Université de Lille

Thèse de doctorat

Information per unit of interaction in stochastic sequential decision making

Quantité d'information par unité d'interaction en apprentissage séquentiel stochastique

Thèse de doctorat de l'Université de Lille
préparée à INRIA Lille Nord-Europe

École doctorale n°631 Mathématiques-Sciences du numérique et de leurs interactions, (MADIS)
Spécialité de doctorat: Informatique

Thèse préparée sous la direction de Odalric-Ambrym Maillard par

FABIEN PESQUEREL

Soutenue le 04 décembre 2023

Composition du Jury :

Bruno Gaujal Directeur de recherche, INRIA (POLARIS)	Président du jury
Alexandra Carpentier Professeur, Université de Potsdam (IFM)	Rapportrice
Alexandre Proutière Professeur, KTH (EES)	Rapporteur
Bruno Gaujal Directeur de recherche, INRIA (POLARIS)	Examinateur
Claire Vernade Chargé de recherche, Université de Tübingen (Lifelong Reinforcement Learning)	Examinatrice
Odalric-Ambrym Maillard Chargé de rercherche, INRIA (SCOOL)	Directeur de thèse

Acknowledgments

I dedicate this manuscript to my loved ones. To my parents, Thierry and Isabelle, whose unwavering support from the very beginning of my life is immeasurable. To my sister, Floriane, whose guidance and support have helped me overcome many hardships. To Cannelle, whose significance in my life is truly unfathomable.

I express my deep gratitude to my advisor, Odalric-Ambrym, for placing trust in me and guiding me through my journey in research during my PhD. Your mentorship has been invaluable, and I am grateful for the opportunities you have provided.

To you, the reader, I extend my appreciation for dedicating your time to read this manuscript. I sincerely hope that the content within proves to be meaningful and insightful.

Fabien Pesquerel

Preface

The process of solving

When pondering about the concepts of *question* and *answer*, one may delve into well established scientific fields as well as discover our own way to *question and answer* those very concepts. When *asking* a dictionary what a *question* is (or more precisely, means) one can read that "a question is an utterance which serves as a request for **information**". Let's emphasize the fact that the dictionary is the mean of **interaction** with the environment through which we gained access to the desired information, a definition. The **environment** is the set of all existing words (is such a set really known?). The definition is the answer to the question *what is the answer?*, and the dictionary is the answer to the question *how did we find the information?*. If we were to answer the question *how long to find the answer?*, we could say that a dictionary sorted in alphabetical order is an efficient mean of information retrieval.* We could then ask how the dictionary was built in the first place. This would take us to the fields of linguistics and philosophy of language and make emerge at least three definitions: a semantic one, a pragmatic one and a syntactic one. While interesting, we will not investigate those concepts from the field of linguistic. Rather, we will adopt a more *symbolic* approach. We will use a language having a coherent and strong **syntax**, being incredibly **pragmatic**, and having a **semantic** power that allows for a large variety of models: **mathematics**. In this thesis, mathematics will be the way to model problems and the way to answer the problems.

In this thesis, we will be interested in mathematically model **the process of solving**. We will not follow the path of logician nor study set and category theories. We will adopt a *physicist mindset* and try to model something that is, hopefully, closer to what is being done as of writing those lines, *trials and errors*: **an empirically motivated model of process solving**. Scientists, such as linguists and physicists, often have **objects** that they want to study. To study those object, scientists have tools, be it abstract or concrete, that allows them to inspect and *interact* with the objects. Scientists also have ways to assess and express the *validity* and scientific interest of their findings. For instance, the new model may be more powerful or the new results may explain something that was not explained until now.

Modeling how the scientific mind solve natural questions about the world is the main motivation of this thesis' author.

The physics of learning

In this thesis, we will emphasize some ways of thinking that are rarely seen in the machine learning community. However, those methods are the bread and butter of the physics community. In particular, we will sometimes write about machine learning and reinforcement learning as if we were describing a natural phenomenon and uncovering the **natural laws of learning**. Since most of the things we study in this thesis stems from mathematical constructions, this idea may seem preposterous to some, but I genuinely think that there is a lot of insight to earn from adopting a physicist lens. In particular, it allows to *phrase* the theorem in a way to makes them easier to grasp with the mind.

For instance, when dealing with the complexity of a sequential learning problem, I will introduce and talk about **information per unit of interaction** or **progress per unit of interaction**. After reading those expressions, a reader should already have an image of what they will refer to, at least at an intuitive level.

When talking about **unit of interaction**, we give a unit, or *dimension*, to the quantity named *interaction*. Indeed, another tool that we will take from the physicist toolbox is the **dimensional analysis**. Usually in mathematics, quantities are dimensionless. However, if we assign a fictitious dimension to some quantities of interest, one can check the soundness of some formulas and even guess what a formula should look like. For instance, it

* when you think about it, we all learned about prefix tree and dichotomy at a very young age.

would be strange to have an *entropy-dimension* equating a *probability-dimension*. With those two previous ideas in mind, we can start thinking about the *derivative of entropy with respect to the number of samples*. While this sentence may not have a clear mathematical meaning, it really helps to build a strong physical sense for the tools and theorems we are manipulating.

A last important tool that we will use is **scaling**. In particular, we will at some point use the concept of **intensive and extensive properties**. When studying a physical system, one can ask how would quantities of interest varies if we were to multiply the *size* of the system. Intensive properties, such as pressure or temperature, will not change. Extensive properties, such as volume, will change proportionally with the size of the system. Obviously, not all quantities are extensive or intensive, but it is nonetheless useful to ask the following question when solving a problem: if we were to multiply the size of the problem by a factor, how should my quantities of interest change? For instance, when solving a sequential learning problem, we strongly advocate that **no explicit reference of the origin of time should be used in algorithms**. While it may not pose any mathematical problem, we do think that it does not make sense from physical standpoint. In particular, we think that doing so can only impair finite time performances of algorithms. Such an intuition is confirmed by experimental results.

Knowing why, knowing what

When solving a sequential stochastic optimization problem such as a bandit problem, there is a difference between **knowing why** and **knowing what**. What we mean by that is that you can *know what to play* without a perfect knowledge of the system, without exactly *knowing why*. In other words, **playing optimally is different from having a good estimation**. While it may sound trivial, I think that it is an important message that should be emphasized. Often, when solving a bandit problem, we estimate the expected returns of different actions, and we use that information to know what to play as the next action. It is almost an automatism, but it should not necessarily be the case. After all, the information needed to know what to play is different from the information needed to estimate the expected returns of the actions. Because those objectives are quite close in most of the studied cases, these two objectives seems highly intertwined, but it does not have to be the case. I will illustrate this fact in this thesis. However, I think that it could be a good thing for the community to develop more tools, mathematical and algorithmic, to directly tackle the problem of *knowing what* rather than using more and more refined *knowing why* tools that are often more concerned about estimating and concentration than sequential problem-solving.

Résumé

Dans cette thèse, nous nous interrogeons sur la vitesse à laquelle on peut résoudre un problème stochastique inconnu. À cette fin, nous introduisons deux domaines de recherche connus sous le nom de Bandit et d'Apprentissage par Renforcement. Dans ces deux champs d'étude, un agent doit séquentiellement prendre des décisions qui affecteront un signal de récompense qu'il reçoit. L'agent ne connaît pas l'environnement avec lequel il interagit, mais pourtant souhaite maximiser sa récompense moyenne à long terme. Plus précisément, on étudie des problèmes de décision stochastique dans lesquels l'agent cherche à maximiser sa récompense moyenne. Dans ces problèmes dit d'apprentissage stochastique, l'agent interagit séquentiellement avec un système dynamique, sans aucune réinitialisation, dans une suite unique, infinie et ininterrompue d'observations, d'actions et de récompenses tout en essayant de maximiser ses récompenses totales accumulées au fil du temps.

Nous commençons par présenter le problème de Bandit, dans lequel l'ensemble des décisions est constant, et définissons ce que l'on entend par *résoudre le problème*. Parmi ces agents, certains sont meilleurs que tous les autres et sont dits optimaux. Nous nous concentrons d'abord sur la manière de tirer le maximum d'information de chaque interaction avec le système en revisitant un algorithme optimal et en réduisant sa complexité numérique. Tout en conservant l'optimalité de la méthode initiale, la méthode proposée réduit la complexité numérique et permet donc d'extraire d'avantage d'information d'un échantillon par unité de temps de calcul. Nous étudions ensuite un problème structuré intéressant dans lequel il est possible d'exploiter la structure sans l'estimer.

Ensuite nous nous consacrons à l'Apprentissage par Renforcement, dans lequel les décisions qu'un agent peut prendre dépendent d'une notion d'état. Chaque fois qu'un agent prend une décision, il reçoit une récompense et l'état change selon une loi de transition sur les états. Sous une certaine hypothèse, dite ergodique, un taux optimal de résolution par unité d'interaction est connu et nous introduisons un algorithme dont nous pouvons prouver qu'il est optimal et nous montrons qu'il est numériquement efficace. Dans un dernier chapitre, nous tentons de mieux comprendre ce qu'implique la supression de l'hypothèse d'ergodicité. Nous considérons le problème *a priori* plus simple où les transitions sont connues. Cependant, même sous cette hypothèse, il n'est pas évident de comprendre correctement la vitesse à laquelle des informations peuvent être acquises sur une solution optimale.

Abstract

In this thesis, we wonder about the rate at which one can solve an unknown stochastic problem. To this purpose we introduce two research fields known as Bandit and Reinforcement Learning. In these two settings, a learner must sequentially makes decision that will affect a reward signal that the learner receive. The learner does not know the environment with which it is interaction, yet wish to maximize its average reward in the long run. More specifically, we are interested in studying some form of stochastic decision problem under the average-reward criterion in which a learning algorithm interacts sequentially with a dynamical system, without any reset, in a single and infinite sequence of observations, actions, and rewards while trying to maximize its total accumulated rewards over time.

We first introduce **Bandit**, in which the set of decision is constant and introduce what is meant by *solving the problem*. Amongst those learners, some are better than all the others, and called optimal. We first focus on how to make the most out of each interaction with the system by revisiting an optimal algorithm, and reduce its numerical complexity. Therefore, the information extracted from each sample, per-time-step, is larger since the optimality remains. Then we study an interesting structured problem in which one can exploit the structure without estimating it.

Afterward we introduce **Reinforcement Learning**, in which the decision a learner can make depend on a notion of state. Each time a learner makes a decision, it receives a reward and the state change according to transition law on the set of states. In some setting, known as ergodic, an optimal rate of solving is known and we introduce a knew algorithm that we can prove to be optimal and show to be numerically efficient. In a final chapter, we make a step in the direction of removing the ergodic assumption by considering the *a priori* simpler problem where the transitions are known. Yet, correctly understanding the rate at which information can be acquired about an optimal solution is already not easy.

Contents

List of Figures

List of Algorithms

List of Tables

The complexity of solving a problem

1

When thinking about the **process of solving**, it is hard not to think about the **complexity of solving**. In this thesis, we will be concerned about **solving an optimization problem while being uncertain about the very problem we want to solve**. There will be a set of decisions that are real random variables. Values taken when sampling a decision is called a reward, and we want to find the decision of maximal expected rewards. The set of decisions might be just that, a set, and the problem is said to be a **bandit problem**. The set of decisions might have some additional structure and depending on it, it will be called a structured bandit or a **reinforcement learning problem**. We will be solving the optimization problem sequentially and our main focus will be to measure a notion of **random complexity**. While related to, this notion which we will call the **regret of a problem**, is not equivalent to the notion of sample complexity. The reason it is not equivalent, is that our focus is to **find a solution** to the problem rather than correctly **estimate the function** from which the problem arises. Similarly to the concept of **algorithmic complexity**, the notion of regret will be related to some important quantities describing the problem, *e.g.* the size of the decision set. When additional structure is considered on the set of decisions, it is important to have an intuitive idea of how the complexity should scale with those parameters.

Before delving into the core of this thesis, let's describe a few simple problems that are all related to bandit and reinforcement learning. I hope those problems will help build some intuition and convince the reader of the scientific interest of the *physics-oriented* approach. In the last section, we will briefly talk about modeling sequential decision problems. In particular, I will mention a sequential problem that is particularly dear to me, **scientific discovery**. I hope that the works done during this thesis will be useful to uncover the veil of **the physics of scientific discovery**.

1.1 Deterministic problems

The maximum of a set of numbers

Here is the simple problem. You are given a finite set of numbers and your task is to find the element of maximal value. We will talk about a simple algorithm to solve the task and consider its complexity. I advocate that there is more to this statement than one may assume and that clarifying it may avoid a lot of confusion when studying the more complex bandit and reinforcement learning problems.

It is implicit that we can **interact** with elements from the set by looking at them. That is to say, there exist a set of pointers or labels $\mathscr{L}$ and a selector function s from $\mathscr{L}$ to the set of values $\mathscr{V}$:

$$s : \mathscr{L} \to \mathscr{V}.$$

Our task is to find *an element*, **not** *a value*. That is to say, we are looking for an object belonging to the pointer set, not the value set. Once we have a correct algorithm that computes the maximal element, we can forget altogether about the value set. When asked about the maximal element, we can confidently answer the stored value and this element can be passed by without reference to its value. This illustrates the difference between *knowing why* and *knowing what*.

It is implicit that computing a pointer or label is easy. That is to say, the set of labels is either explicitly given to us or implicitly as a simple constructive rule. It can be *integers from 1 to n* or *the* n^{th} *first successors of 1*. It will not be something like *the* n^{th} *first prime numbers after* $10^{10^{10}}$. In a nutshell, the complexity of computing a pointer or label is not considered.

It is implicit that we have only access to a limited number of mathematical operators to solve the problem. For instance, we do not assume the existence of a $\max : (\mathcal{L}, s, \mathcal{V}) \to \mathcal{L}$ operator that return the maximal element. Rather, we have to define a set of *elementary operations* and compute the complexity of an algorithm as the number of used operations to compute a solution. All operations are done sequentially. It is implicit that we can only point to one element at a time, allocate some memory to store and access relevant information, and perform **pairwise comparison** of values. We interact with the problem by pointing to elements, and we gain information by comparing elements.

Please remark that in determining the maximal elements, we will only be concerned about the relative order between values and not about the value of the difference between values. It is implicit that the elementary operations do not depend on the difference and therefore, the span of values, nor the gaps between values should appear in any measure of complexity of the problem. A measure of the complexity of the problem should be value-agnostic.

It is implicit that there is no prior information about the values that is contained within the labels and labelling function. Anticipating on the next sections, one can see that this will be the case for unstructured bandit problem but will not necessarily be true for structured bandit and reinforcement learning problems.

To know the maximal element, all values have to be inspected at least once. Hence, the number of elementary operations is larger than the size of the value set, $|\mathcal{V}|$. On the other hand, one can describe a very simple algorithm matching this complexity lower bound. Initialize a couple label-value $m = (\star, s(\star))$ with some arbitrary chosen element $\star \in \mathcal{L}$. Iterate through the label set and, using s, inspect the values. Compare the value of the currently inspected element e with the stored value $s(\star)$ of m. If the value of the currently inspected element $s(e)$ is larger than that of $\star$, then set $m = (e, s(e))$. Otherwise, pass and inspect the next element. Such an algorithm uses a constant number of elementary operations per element and inspect each element only once. Therefore, the complexity of the algorithm is proportional to the size and in the context of algorithmic complexity, we say that it matches the complexity lower bound.

In this sequential procedure, what is the **progress per unit of interaction**? Imagine that the value 0 means that we have no information at all on the

problem and that the problem is 0% solved. Progress of 1 means that we have 100% solved the problem and gathered all the necessary information in a meaningful way. We represent in Figure 1.1 the three measures of progress that we present: crude, probabilistic, and entropic.

A crude way to measure progress would be to say that, since nothing is certain until all values have been inspected, the progress per unit of interaction is always 0, except for the last inspection where the progress is 1. Similarly to a phase transition, all of a sudden, the problem is solved. This approach is interesting because **it gives information about the completeness of the task**. It is represented by the *crude information* curve in Figure 1.1.

Another way of measuring the progress per unit of interaction would be to count the remaining number of elements to inspect before the problem is solved. Normalizing such a quantity, the progress per unit of interaction would be $1/|\mathcal{V}|$ and the total progress would be the sum of the *progresses per step*. One can see that this approach consists in averaging (or amortizing) the costs of the crude approach by the size of the problem. This approach is interesting because **it gives information about the remaining number of interactions** before the task is over. This point view is deeply related to a probabilistic* viewpoint on the problem. When running the sequential algorithm, we can consider the probability p_t that the stored label-value at iteration t is the one corresponding to the maximal element. We have the simple recurrence relation

$$p_{t+1} = p_t + (1 - p_t)\frac{1}{|\mathcal{V}| - t}$$

with the boundary conditions $p_0 = 0$ and $p_t = 1$ whenever $t \geq |\mathcal{V}|$. One could argue that the term $(1 - p_t)\frac{1}{|\mathcal{V}|-t}$ is a good candidate for the denomination of *information per time step*. It indeed has the dimension of a *probability per unit of interaction* since it is a *probability* divided by a *number of iterations to go*. Solving this equation, we get that $p_t = \frac{t}{|\mathcal{V}|}$ and therefore, $(1-p_t)\frac{1}{|\mathcal{V}|-t} = \frac{1}{|\mathcal{V}|}$. We therefore retrieve from a probabilistic and physics-oriented interpretation our aforementioned quantity of progress per step. It is represented by the *probabilistic information* curve in Figure 1.1.

Finally, we present one last interesting way of measuring the complexity of this problem, **entropy**. Entropy can naturally emerge from the aforementioned probabilistic model. However, let's first expose how the entropy term $\log|\mathcal{L}|$ also stems from a theoretical computer science viewpoint. When it comes to the description of the algorithm, there is really one **variable** that depends on the problem, that is the cardinal $|\mathcal{L}|$ of the label set. The length of a program describing how to solve the problem is thus a constant plus a variable length depending on $|\mathcal{L}|$. It is a known fact that whatever the base we choose, the description length of $|\mathcal{L}|$ grows logarithmically with $|\mathcal{L}|$. This approach is interesting because **it gives information about the length of a program (or strategy)** that is necessary to solve the problem. Hence, one could argue that $t \mapsto \log_{|\mathcal{L}|} t$ is a good measure of progress and that $\log_{|\mathcal{L}|} \frac{t+1}{t}$ is a relevant measure of progress per step at step t. One can see that the level of progress is step (or time) dependent. The progress function is concave, meaning that the more advanced the step, the less we gain information and the

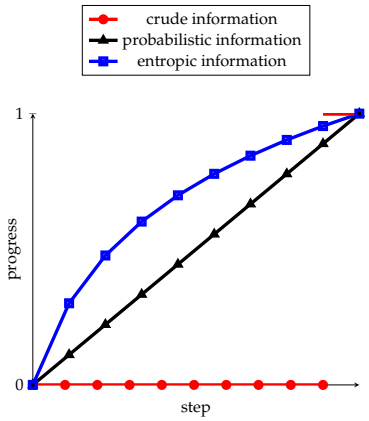

Figure 1.1: Progress functions for the max-element problem as computed by the different complexity measures.

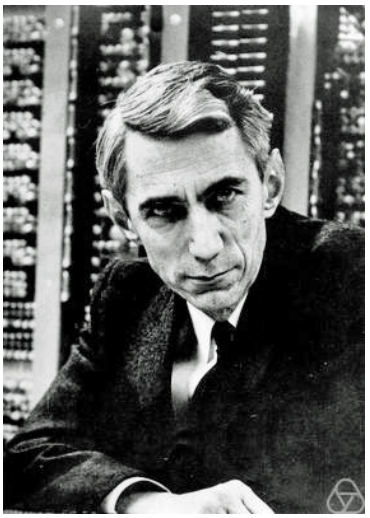

Figure 1.2: Claude Elwood Shannon (1916-2001). American mathematician, electrical engineer, computer scientist and cryptographer known as the one of the father of **information theory**

* assuming uniform permutation of the labels

slower the progress. This is somewhat intuitive that, as the problem is being solved, the higher the probability that we already found the maximal element and the less information we gain per step. Because the information per step is like a derivative, a decreasing information per step means a concave progress function. It is represented by the *entropic information* curve in Figure 1.1. However, we do know that in reality, we have to check all elements at least once to solve our problem. Apart from being the log-likelihood ratio $\log \frac{p_{t+1}}{p_t}$, how does this viewpoint connect with the previous one?

Figure 1.3: Henri-Léon Lebesgue (1875-1941). French mathematician known for his **theory of integration** originally published in his dissertation *Intégrale, longueur, aire* in 1902.

To make appear the $\log |\mathscr{L}|$ term, we have to slightly change our computing paradigm ; from an algebra to analysis. During this thesis, we will encounter a lot of properties that can be understood from either of the two intertwined viewpoints. The paradigm of interest is the one of **measure theory**. In this context, our elementary operations will consist in applying a well-chosen measure to a measurable set of a σ-field defined over the label space $\mathscr{L}$. We gain information on the problem by *measuring*. A good candidate for the complexity of the problem is the **number of measures** before one can solve the problem. We consider the power set $2^{\mathscr{L}}$ as our σ-field over the label space, and we show that we can solve the problem using *dichotomy*. Using this method, we show that the number of measures necessary to solve the problem is $\log |\mathscr{L}|$. The algorithm proceeds as follows. We consider the measure $\mathfrak{m} : 2^{\mathscr{L}} \to \{0, 1\}$ that is such that $\mathfrak{m}(A) = 1$ is the maximal element is in A and $\mathfrak{m}(A) = 0$ otherwise. We partition $\mathscr{L} = \mathscr{L}_1 \sqcup \mathscr{L}_2$ into two disjoint sets of the same size (plus or minus one element). Using $\mathfrak{m}$, we measure $\mathscr{L}_1$ and gain the knowledge of the subset in which the maximum is lying. Recursively picking the subset in which the maximal element is, we halve the size of the subset in which we are looking for the maximal element until both the considered subset are of size one and perform one ultimate measure. The number of measures is therefore $\log |\mathscr{L}|$. Not so coincidentally, this is also the entropy of the uniform distribution on a set of size $|\mathscr{L}|$. This is also equal to $-\log(1 - p_t)\frac{1}{|\mathscr{V}|-t}$, the log of our previous constant probability progress per step.

Figure 1.4: Félix Édouard Justin Émile Borel (1871-1956). French mathematician known for his founding work in **measure theory and probability**.

Let's wrap this introductory example by answering one last question: how does this measure-related logarithmic complexity relates more deeply to the previous algebraic-related complexity? To connect the two, we need to consider the algebraic complexity of performing a measure. From an algorithmic viewpoint, the following reasoning is a bit absurd, but it serves the purpose of illustrating how algebra and measure works together when determining the complexity of a problem. To algebraically perform the measure and determine whether the maximal element is in the first or second subset of the partition, we compute the maximal element of both partition, compare them and based on that, we can assign the measure. Such a procedure can be done recursively. Of course, we should be able to determine the maximal element just after the first split, but this is a thought experiment for the sake of connecting different thinking methods. What is the number of elementary operations? For the first split, we inspect $|\mathscr{L}|$ elements for the measure; for the second split, we inspect $|\mathscr{L}|/2$ elements; for the third split, we inspect $|\mathscr{L}|/2^2$ elements... The number of elementary operations $|\mathscr{L}|\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{\log|\mathscr{L}|}}\right)$ is therefore upper bounded by $2|\mathscr{L}|$ and lower bounded by $|\mathscr{L}|$. When accounting for the **algebraic cost of measuring** and using the measure-oriented algorithm,

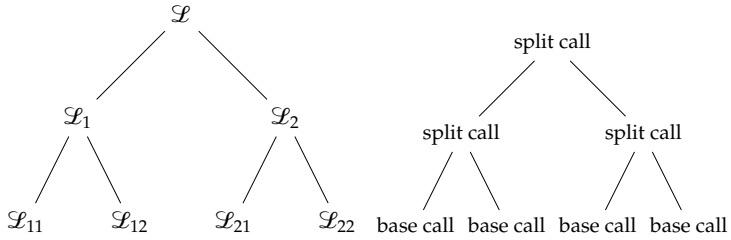

Figure 1.5: Recursive splitting of the label set $\mathscr{L}$ and associated call tree. The order of the call is assumed to correspond to a *preorder traversal* of the splitting tree. Successive call are stored on a stack as depicted in Figure 1.6.

we retrieve the algebraic complexity of solving the problem.

Of course, one can perform a deeper merging of those two ways of thinking by considering a recursive algorithm. The base case occurs when the size is of size one in which case it returns the only element of the set. Otherwise, the label set is split in two roughly equal subsets and the algorithm is called recursively on each of the subset until the base case is reached. When returning from a split, the algorithm return the maximal element of the two elements returned by the two children call.

After splitting, instead of directly computing and comparing the maximal values, we call our procedure on the two constructed subsets and solve the same problem of computing the maximal value. From the knowledge of the maximal elements on the two subproblems, we can compute the solution to our original problem. Thus, the problem of finding the maximal element of a set have the subproblem substructure and can be solved using an algorithmic method that is close to, if not only named differently, **dynamic programming**.

base call on $\mathscr{L}_{11}$
split call on $\mathscr{L}_1$
split call on $\mathscr{L}$

Figure 1.6: Execution stack after right before the first base call return. The base call on $\mathscr{L}_{11}$ will return its value to its parent *split call on* $\mathscr{L}_1$ which will then execute a base call on $\mathscr{L}_{12}$.

How does this relate to the works presented in this thesis? However simple the maximal element problem appears to be, one can see how instructive it can be. **Bandits**, that we introduce in the next section, are connected to this problem. The main feature of bandit is that instead of dealing with the tuple $(\mathscr{L}, s, \mathscr{V})$ where $\mathscr{V}$ is a set of numbers, we will now consider the case where $\mathscr{V}$ is a functional space. Elements of $\mathscr{V}$ will be real random variable satisfying some properties. The maximal element problem will be equivalent to the best element identification problem. Interestingly, the introduced stochasticity brings out new questions that could not be asked in the deterministic setting, *e.g.* what is the expected number of time that we sampled the best element[†] after playing for T steps. For those reasons, the title of Section 3.1 introduces bandits as *a zeroth order model of decision-making*. We will have to determine a relevant measure of the complexity and craft a relevant measure of progress. Understanding a broader notion of complexity and building intuition in simple yet informative examples will hopefully help the more difficult problems presented in this thesis.

Path of maximal reward

A graph is a pair G = (V, E), where V is a set whose elements are called vertices, and E is a set of paired vertices, whose elements are called edges.

[†] *i.e.* a random variable with the largest expected value

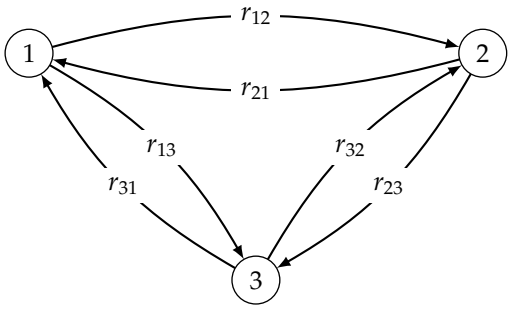

Figure 1.7: A graph in which edges are argument of a reward function.

The shortest & longest path problems

In graph theory, the shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized.

In graph theory and theoretical computer science, the longest path problem is the problem of finding a simple path of maximum length in a given graph. A path is called simple if it does not have any repeated vertices; the length of a path may either be measured by its number of edges, or (in weighted graphs) by the sum of the weights of its edges. In contrast to the shortest path problem, which can be solved in polynomial time in graphs without negative-weight cycles, the longest path problem is NP-hard and the decision version of the problem, which asks whether a path exists of at least some given length, is NP-complete. This means that the decision problem cannot be solved in polynomial time for arbitrary graphs unless P = NP. Stronger hardness results are also known showing that it is difficult to approximate. However, it has a linear time solution for directed acyclic graphs, which has important applications in finding the critical path in scheduling problems.

1.2 Stochastic problems

The maximum expected value of a set of random variable

Here we add some stochasticity to the simple *maximum of a set of numbers* problem. You are given a finite set of real valued random variables and your task is to find the element of maximal expected value. To the layman reader, this sentence might seem so close to the previous one[1] presented in Section 1.1 that once the former problem is solved, the latter should be close enough to be solved. However, adding stochasticity to the problem makes it a research problem called **Bandit**[2]. We will surely delve into the topic latter on but right now, let's focus on the problem statement and clarify some technical words as well as implicit assumption.

1: You are given a finite set of numbers and your task is to find the element of maximal value.

2: More precisely, this is the problem of **best arm identification**.

It is implicit that we can **interact** with elements from the set by looking at them. Compared to the previous situation, the interaction is two-fold. First, there exist a set of pointers or labels $\mathscr{L}$ and a selector function s from $\mathscr{L}$ to the set of random variables $\mathscr{B}$:

$$s : \mathscr{L} \to \mathscr{B}.$$

Second, given a selected random variable $s(a)$ for $a \in \mathscr{L}$, it is implicit that we can sample from it. That is to say, we gain information about elements from the set by selecting them and sampling them. One cannot access information about a selected random variable such as moments for example. Therefore, it is implicit that there is no prior information about the expected values that is contained within the labels and labelling function[3]. As previously, we can only point to one element at a time, allocate some memory to store and access relevant information, and perform **pairwise comparison** of values. Please refer to Section 1.1 for other implicit assumptions, particularly about label computation.

3: Of course, one can always add some structure to the problem later on.

It is somewhat implicit that the expected values of the random variables are all finite. We discuss this assumption in depth in Section 1.3. The main and most important semantic vagueness appearing in the statement lies in the definition of the word *find*. Because of the newly introduced stochastic nature of the problem, uncertainty about our knowledge of the expected values hinder our solving process and ability to *know for certain* the maximal element. For the problem to be interesting, it should be the case that some information is learned about elements as we select and sample random variables. As we gain information, our uncertainty about the maximal element, should reduce in a *measurable* way. Thus, a meaning can be given to the words **solving**, as a *reduction in uncertainty*, and **to find**, as *achieving a low enough uncertainty*. To quantify the information gain, some assumptions on the considered random variables will be necessary and are still a topic discussed in research papers.

Again, we emphasize that our task is to find *an element*, **not** *an expected value*. In other words, knowing the maximal element is different from knowing the maximal expected values. In particular, the uncertainty about which element is maximal can (and often) decrease far more quickly than the uncertainty about the expected value of that said maximal element. While knowledge about the expected values gives information about the maximal element problem we wish to solve, *solving the problem* should not be confused with *estimating the problem*. We now illustrate this fact with a simple instance of the problem.

The problem is as follows. We make the assumption that all the random variables have a support of finite length at most one. We want to solve the problem of finding the element with maximal expected value. Because labels do not give any information about the random variables and no dependencies is assumed between the random variables, we have to start by selecting and sampling each element once. Imagine that doing so, we get $|\mathscr{L}|$ samples such that for all $a, b \in \mathscr{L}$ with $a \neq b$, the respective samples x_a and x_b are separated by more than 2, $|x_a - x_b| \geq 2$. Then, we can already output a maximal element, $\star \in \operatorname{argmax}_{a \in \mathscr{L}} x_a$, the element of whose first sample is maximal. The reason is that because the supports of the random variables are of length at most one, we know from the distances between samples that, in fact, all the supports are disjoints. This situation is illustrated in Figure 1.8 where we can visualize why distances larger than two between samples induce non-overlapping support. In this case, one can see that despite a large uncertainty about the expected values of all samples we have complete certainty about a solution to the problem. For each random variable $s(a)$, $a \in \mathscr{L}$, one can only affirm that its expected value is located in the interval $[x_a - 1, x_a + 1]$, which we can agree, is not a lot of information and a lot of uncertainty. Nonetheless, the

In fact, it is easy to prove that only the largest of samples need to be 2-separated from all other samples to conclude that we found a maximal element.

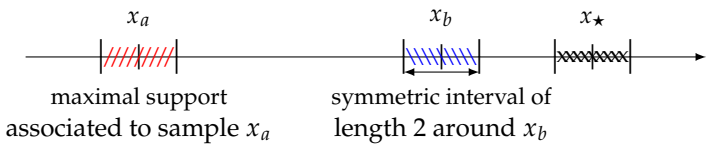

Figure 1.8: The first samples induce a structure of non-overlapping supports

Knowing the ordering permutation is indeed too much information, and we can see why only the 2-sepration of the largest sample is necessary.

relative order of all expected values is known with certainty thanks to the 2-separation of all samples. Starting to connect this setting with more complicated one that are to come, we can say that, for label $a \in \mathcal{L}$ such that its first sample is such that $x_a + 1 < x_\star - 1$, *its likelihood of optimality given information is null*.

In this thesis, we will talk about the **likelihood of optimality** as a mean to emphasize the difference between estimating the stochastic problem and solving the stochastic problem.

If we were to describe the progress per unit of interaction, one would get something very similar to the deterministic case. Of course, even when supports are separated, the first samples do not need to contain this information by being 2-separated. It could very well be that the supports are in fact of lengths much smaller than one (the extreme case being Dirac distributions) thus making useless the strategy tracking the spread of samples to eventually separate supports (which would work if supports are of length exactly one and all strictly separated). It could also be that the supports are not disjoints as illustrated in Figure 1.9 where the true supports of three distributions are depicted along with the position of their true expected values. In this scenario, discriminating between label $\star$ and label b may require advanced techniques presented thereafter. In

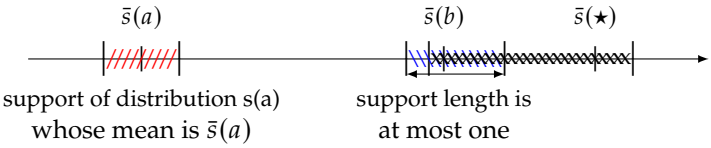

Figure 1.9: Overlapping supports

those cases, sophisticated methods to compute a *likelihood of optimality* for each of the elements are required.

Please remark how the complexity of a problem is sometimes dependent on the gap between the optimal value and other values while it was not the case in the deterministic setting. That complexity somehow has to depend on a function of the random variables and their expected values can be understood as maximally utilizing the hypothesis at hand. In the deterministic setting, we only used the fact that there exist a total order on elements in $\mathcal{V}$ and a way to access the comparison operator. As those are the only properties used to define the problem, any good notion of complexity cannot make any other assumption and will only depend on the number of elements and number of comparisons. In this stochastic setting, the set of possible values is $\mathbb{R}$, endowed with its usual structures, in particular, its $\mathbb{R}$-vector space structure, its canonical topology, Borel σ-algebra and Lebesgue measure. Altogether, this allows to give a precise meaning to the concept real random variables and expected values. If we were to define another criterion of maximality, the problem statement might look completely different. For instance, we could be looking for

the random variable of maximal possible value, knowing that all random variables take their values in a totally ordered set without additional structure. The fact that $\mathbb{R}$ is endowed with its topology and the nature of the problem posed as a maximum over expected values indicates that the *likelihood of optimality* of elements is likely to be computed using analytical properties of $\mathbb{R}$, in particular continuous functions. This is because the topology defines a notion of neighborhood which is the set-approach to define a notion of proximity or vicinity between values. In fact, this goes one step further with $\mathbb{R}$ since its topology can be recovered from its usual metric. Using all the structural assumptions at hand, it is likely that any satisfying notion of complexity of a problem will depend analytically on the random variables, for instance through the distances between the optimal expected value and other values.

This simple example can also teach something about the notion of *complexity to solve a problem*. Our intuition is that an interesting generalization of our previous deterministic complexity should be related to the **number of interaction before solving**. The main difficulty is that because of the stochasticity, there is *a priori* no unique definition of *solved*. Rather, there is a continuum of uncertainty about the solution to the problem, from fully uncertain to completely solved. Therefore, our future definition of complexity should be close to the mathematical transcription of the **number of interaction before a level of uncertainty is reached**. Then, the **progress per unit of interaction** would be related to the decrease in uncertainty per sample. The precise dimension (or unit) of this progress will depend on the method used to measure uncertainty and information. If uncertainty is measured using interval of fixed confidence, then the speed of progress is measured in length per sample, where the length of a confidence interval is a function of the number of samples.

Recall that we are studying the problem of finding the element with maximal expected value in a set of random variables having a support of finite length at most one. From the previous analysis and our definition of complexity, some instances of the problem are *easy* to solve, *e.g.* when the supports are 2-separated, while others are *hard* to solve, *e.g.* when all supports are the same and non-degenerate. Thus emerge two notions of complexity: **instance dependent** and **worst case**. The **instance dependent** complexity will refer to the complexity of solving the specific instance at hand. The **worst case** complexity refers to the maximal instance complexity that can be measured on a class of problems. For a given notion of solving, let $\mathscr{C}(s, \mathscr{L}, \mathscr{B})$ represents the complexity of solving the specific instance specified by the sets $\mathscr{L}, \mathscr{B}$ and the map $s : \mathscr{L} \to \mathscr{B}$. Let

$$\mathscr{P}_k = \{(s, \mathscr{L}, \mathscr{B}) \,||\mathscr{L}| = k, \lambda(cvx(supp(X))) \leq 1 \forall X \in \mathscr{B}\}$$

denotes the set of all instances with k elements, where $\lambda(cvx(supp(X)))$ is read as the Lebesgue measure of the convex envelope of the support of the random variable X. Then a worst case complexity could be $\max_{i \in \mathscr{P}_k} \mathscr{C}(i)$, the maximal complexity on all instances of a specific class of problem. For this notion of complexity to be meaningful and useful, we see that the class of problem should be carefully chosen, not too small nor too large.

Obviously, without structural assumption, $\mathscr{C}(s, \mathscr{L}, \mathscr{B}) = \mathscr{C}(s', \mathscr{L}', \mathscr{B})$ for all $\mathscr{L}, \mathscr{L}'$ and bijection, s, s'. That is, a good notion of complexity should be independent of the label set and map function, $\mathscr{C}(s, \mathscr{L}, \mathscr{B}) = c(\mathscr{B})$.

Interestingly, the introduced stochasticity brings out new questions that

could not really be asked in the deterministic setting. Of particular interest is the number of interaction with each element to get a certain level of certainty about the solution to a specific instance. For a given level of certainty, what is the distribution of those number of interactions? For a given level of certainty, what is the minimal level of interaction? For a given number of interaction, what is the strategy that interacts with the maximal element the largest number of time?

Bandits, that we introduce in the next section of this thesis, is a framework that is used to study those concepts and questions. We foresaw the notion of sample complexity of elements and how it can be helpful to characterize the complexity of a problem. In this stochastic setting we intuited that a good notion of information per unit of interaction should say something about the level of uncertainty about the solution to the problem at hand. Anticipating a probability viewpoint on the space of solutions, the *solution distribution* should "converge" to that of a Dirac on the true solution to the problem. A notion of distance to that Dirac could be a measure of progress and the rate of convergence, as a function of the number of interactions, could be a measure of progress per interaction. Those themes will be further developed in this thesis.

1.3 Problematic problems

Finding the right model

Once a (good) theoretical model is defined, it becomes interesting on its own, questions about the model can be asked, researched, debated and answered. Often, models are built on other models. Sometimes, models interact with other models and results about one model can shed light or ask new question about another one. It can therefore be useful to take the time to think about the model we study, to ponder over the relevance of the model, the axioms and assumptions that the model is based on and the other used models that constrain what can be *said* about and on the model. As we saw, an algebraic point of view on the complexity cannot express the same facts as an information-theory point of view on the complexity. Therefore, when set on studying a concept or topic such as **decision**, finding the right model is an important problem. Finding the right pieces of mathematics on which we base our model is equally important. Fortunately, a physics oriented mindset can help to give intuition about the right tools to choose (or build!).

I would argue that even the most abstract mathematics are models of things that a human wish to better understand. For instance, one could wish to understand mathematically the concept of *regularity* and end up building a pretty abstract model. What does it mean for something to be regular? What are the things that can be regular? It is easy to envision many research directions and models from this simple question. Even with multiple models of regularity, one could ask how those models are related to each other and specify what we mean by *related*. In this thesis, we are interested in better understanding the concept of **decision**. A question one may ask at some point is: should a concept of *good decision* be based on some *regularities* in the available *information*? Terms that are

italic are ought to be mathematically defined at some point if one want to give an answer to this question.

While mathematics is the preferred methods of this thesis, there are many fields from which in which the concept of decision is studied. There fundamental and subtle differences in the definitions of *decision* in those field and all them would be a good starting point for a mathematical formulation. However, an overlap in the models is bound to happen due the abstraction power of mathematics and the fact that the definitions of decisions between those fields are all related in some ways. To name a few, here are some fields from which one can borrow a first intuition about the concept of decision: neurosciences, history, philosophy, biology, legal sciences, sport sciences... In this thesis, we are interested in modelling a form of *rational decision* that is closely related to the concept of *decision* that can be found in the field of *economy*. This can particularly be seen through the *numerical reward* assumption that can be closely related to the notion of *utility* in economics.

Given a particular field, what kind of model naturally emerge in your mind? Is it different from Bandit? From reinforcement learning?

Game theory is one approach to the problem of modeling rational decisions. While game theory can be defined as the study of mathematical models of strategic interactions among rational agents, I do think that a similar definition can hold for competitive multi-agents reinforcement learning. However, game theory differs from Bandits theory and reinforcement learning in multiple ways. Reinforcement Learning differs from game theory because it emphasizes the notions of *sequentiality*, *game against nature*, *stochasticity* and most of the work of the community is concerned about the *learnability*. Also, it is far less focused on economics and political concepts than game theory. Surely there are bridges, in particular through the notion of worst-case lower bound that is similar to a minimax approach to the concept of complexity in the sense that we are looking for the worst case possible expected utility (reward) of a strategy among all possible environments.

Figure 1.10: John von Neumann (1903-1957). Hungarian-American mathematician, physicist, computer scientist, engineer and polymath. Known partly for his founding work in **economics and game theory**.

It is sometimes a problematic problem when multiple discipline tackle similar questions without speaking the same language, sometimes hindering the amount of interaction between communities having similar interests. On the other hand, this diversity of models and tools to tackle the problem (while daunting when we must choose) is beneficial to inspire new ideas and foster creativity. In this thesis and other works from the community, some hypotheses are therefore driven by applications. Others are driven by the need to handle the mathematical complexity of the model and our owns ability to derive results. Finding the right model that allows us say something about the application that we have in mind while being general enough to be useful as an abstract model but not so much that it cannot be theoretically handled anymore is a complicated task.

In this thesis, we want to study the concept of *decision* and more specifically, in the concept of *sequential decision-making*. We are also interested in studying those decisions that allows us to solve some problems, and we are interested in understanding a notion of *information per unit of interaction*. In this thesis, we will see that Bandits and reinforcement learning are one way to mathematically tackle such concepts.

Measuring: alter or ignore but remain uncertain

Imagine that you want to know which of three persons is the angriest. You decide that the person with which you want to interact the most should be the less angry. In this example, *angry* is a numerical cost that you want to minimize across the courses of your interaction. This example is somewhat similar to that of best arm identification. Can you imagine asking *are you angry?* repeatedly to the three persons without changing their internal sate? Without actually making them angry?

In this section, I would like to preemptively write about one of the less intellectually satisfying assumption that is make about the systems that are studied in this thesis, *stationary condition*. In order to derive non-trivial notions of *learnability*, useful lower bounds, and usable concepts, it is often necessary to make a stationary condition assumption. Because we are modelling the process of sequentially solving a problem in an environment, a stationary condition is unlikely to be reasonable in most cases. Obviously, researcher are well aware of this sometimes unsatisfactory assumption but still use it because it is also quite hard to model a relevant process of non-stationary condition. The main question being, what the non-stationary conditions should look like? The second one being, how are the learnability results affected by various and different assumptions? We surely do not want the environment to be so non-stationary that nothing can be learned from it. In this section, I would like to discuss one argument in favor of this assumption as well as an approach to model non-stationary condition that I have not encountered during my PhD.

4: The timescale of an experiment is called the time **horizon**.

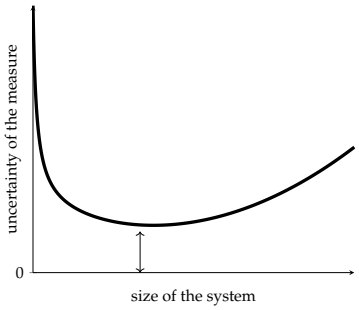

Figure 1.11: Uncertainty of the measure as a function of the *size* of the system when the measure instrument and the method of measurement are fixed

Thinking about physical processes, stationary assumption is often a matter of timescale. For instance, the laws of physics surely can be learned from experiments and deemed stationary on a timescale starting way before we started researching them and way after Earth would disappear. When baking a cake in an oven, the system cake can be considered roughly stationary for a few seconds. If we were to run experiments order of magnitudes faster than the second, then it is safe to assume a stationary assumption. If we want to study a lightning, then the timescale to consider the system close to stationary is likely to be the nanosecond since a lightning is made of short strokes of roughly 65 microseconds. Therefore, the stationary assumption made by Bandits and reinforcement learning is a satisfactory one whenever the timescale of the experiments[4] is small compared to the time constant of the studied system.

However, is it enough? To gain information with the system, an interaction is necessary. Bandits and reinforcement learning model this interaction through the notion of sampling. When sampling, an agent measures a numerical signal. This measure can have at least two effects: making the measure inaccurate and making the system non-stationary. Apart from imperfection coming from the instrument used to measure and the potential inherent stochasticity of the system, another reason for the inaccuracy of the measure is the following. To sample is to measure. To measure is to interact. To interact is to modify. To modify is to make the result of a measurement inaccurate. Therefore, measuring is a source of inaccuracy in the measure.[5] However, there are situations in which the system can safely be assumed to be stable relative to a measurement

5: Thought experiment. Imagine measuring the temperature of water that is within a glass. Using a physical thermometer would surely modify the temperature of the water and therefore, the resulting measure does not correspond to the initial temperature of water. If we were to use a laser to measure the temperature, this laser would surely heat the water, at least locally. It would make the measurement inaccurate too. Furthermore, because the instrument is *more local* than the *system* and that the modification of the property that we want to measure is global, can we still talk about *the* temperature of the system without waiting for a new equilibrium? However, we don't know if the equilibrium is reached without measuring. The process of measuring the temperature of a glass of water is inherently inaccurate. . .

process. That is the case when interaction with the system do not modify the system because of a difference in *scale* or, because the system restore quickly to its previous state (or equilibrium) after the interaction. For the latter case, this restoration can be envisioned through a mechanism of dissipation of the energy injected in the system through the measurement. This amortization can also be interpreted as the fact that the system is under the influence of other factors that are much larger in influence than our sampling system. Whatever the interpretation, the time constant of damping should be order of magnitudes smaller than the time interval between two samples. For the former case, if we were to measure the position of a planet, surely the instruments that we positioned and the photons we captured to perform the measurement would modify the system. However, this influence is negligible to the point that we can consider the interaction to not modify the system.

In the aforementioned cases, one can see that the time constant of the system dissipation effect of the measure should be much smaller than the time interval between two measures to safely assume the non-stationary condition. Otherwise, the sampling process can be a source of non-stationarity. Previously, we saw that the sampling process can be source of noise and uncertainty in the measure. Now we focus on the fact that the sampling process can induce a lasting and measurable dynamic. One can measure with our measurement system the effect of measure. For a Bandit problem, that would mean that at any given time, the random variables that we are studying are functions of the number of times those have been sampled. The rationale is very similar to the previous one. To sample is to measure. To measure is to interact. To interact is to modify. Therefore, to sample is to add dynamics to the system. Modelling the sampling process and incorporating into various results of *learnability* is an interesting question for the Bandits and sequential learning community. A natural question would be: what can be learned about a system from which the observation operator has such and such properties? Therefore, we see that **interaction through measurement** is a source of **non-stationarity** in the system that an agent is interacting with.

In a sense, one can give an original interpretation of reinforcement interpreting the transition matrix as a measuring operator. After measuring the numerical reward associated to an available action, the system, represented by its state is modified as a consequence of the measure. This modification will impact the next measures and the modification is understood through the model of the transition matrix that specify the effect of a given interaction on the state of the system. This interpretation is somewhat limited in scope because the transition is rather interpreted as an effect of *playing* an action rather than *sampling* and action. It can be seen from the fact that the most common starting point to study reinforcement learning is the Markov control model, that originates from control theory. Rather, the thing I want to discuss here is that *observing* the result of an action is an *interaction*. Knowing the state of the system is due to a usually non-modelled interaction with the system.[6]

This why I advocate that a good model of non-stationary condition is one that consider **sampling** as an **operator** on the problem. This operator can be the result of an integration of the effect of sampling on the time interval between two interaction with the system. It is a useful model if we consider that the system has a damping mechanism.[7] Differences in

6: I think that this viewpoint could prove to be interesting to study a new kind of partially observable Markov decision process (PO-MDP).

7: Thought experiment. Imagine a bandit problem in which we wish to choose between three building architectures based on the response of the building to one kind of earthquake of a given magnitude. You build three small models and run your favorite bandit algorithm to choose between the three. The sampling process is to simulate an earthquake on the chosen model. If you don't wait enough between two samples, then the previous interaction will affect the next sample you get through a new simulated earthquake.

systems and operators will lead to different *learnability* results, hopefully general enough to be useful. I think that, in the future of reinforcement learning, exploring the concept of *measure* could lead to interesting mathematical results with great applications. On could start by deriving a *no free lunch* theorem of measurability.

Causation & correlation

correlation $\not\Rightarrow$ causation
causation $\not\Rightarrow$ correlation

It is widely known that correlation does not guarantee causality. Less it is that the converse, the existence of causality without correlation, is possible. Intuitively, the perfect chaotic system, *e.g.* modelling the perfect storm, would exhibit such a strange property. This extremely non-linear model pushes the limit of our definition and intuition of *measurability* and *learnability*.

For a dynamical system $\mathcal{D}$, and an initial condition $x, t \mapsto \mathcal{D}(x, t)$ defines the trajectory within the domain of the system.

As an example, imagine a bounded continuous dynamical system $\mathcal{D}$ in $[0,1]^2$. Denoting P_0 a set of initial condition (time 0), we denote by $P_T = \mathcal{D}(P_0, T)$ the set of positions at time T after letting the system evolve according its dynamic. Calling λ_2 the Lebesgue measure in $\mathbb{R}^2$, imagine that the system is such that for all $\epsilon > 0$ and all convex set $P_0 \in [0,1]^2$ such that $\lambda_2(P_0) = \epsilon$, we have $\lambda_2(P_T) \simeq 1$. That is to say, whatever the size and location of the starting conditions, the whole space is almost filled at time T[8]. If we are unable to sample positions before time T, then, despite the causality, we cannot *measure* meaningful correlation and therefore cannot *learn* about the system. Usually, this phenomenon is illustrated by attractors such as the Ueda attractor that is depicted in Figure 1.12.

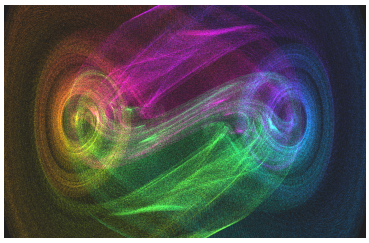

Figure 1.12: Ueda attractor (phase space)

Theoretically speaking, we can imagine going as far as envisaging that the set from which we want to learn are non-measurable. Anticipating on what follow, suppose that we seek to solve a control problem, *i.e.* find an optimal strategy (or control) mapping situation to action given a numeric notion of optimality. If the set of measurable control is empty, then one cannot *learn* an optimal control in the usual sense of learning. Also, if the conditions to be met in order to achieve a specific outcome define a non-measurable set, then given the current framework of *statistical learning*, it is impossible to *learn* useful correlation in this system.

The Bandit and reinforcement learning communities often discuss the various assumptions of the model such as bounded rewards, light tail distributions, finite state space, ergodic dynamic... The reward hypothesis is also a topic of discussion and several papers and research group are focusing on reward free formulations of the kind of problems that are addressed in this thesis. However, the measurability and learnability assumptions are almost always implicit. It is my intuition that the research community would benefit from more projects that study *learning* in mathematical universes using different structural assumptions.

Summary of contributions 2

Bandits

Bandits with groups of similar arms

NeurIPS **2021**

In the paper ***Stochastic bandits with groups of similar arms*** and published at NeurIPS 2021 with Hassan Saber, and Odalric-Ambrym Maillard, we consider a variant of the stochastic multi-armed bandit problem where arms are known to be organized into different groups having the same mean. The groups are unknown but a lower bound q on their size is known. This situation typically appears when each arm can be described with a list of categorical attributes, and the (unknown) mean reward function only depends on a subset of them, the others being redundant. In this case, q is linked naturally to the number of attributes considered redundant, and the number of categories of each attribute. For this structured problem of practical relevance, we first derive the asymptotic regret lower bound and corresponding constrained optimization problem. They reveal the achievable regret can be substantially reduced when compared to the unstructured setup, possibly by a factor q. However, solving exactly the exact constrained optimization problem involves a combinatorial problem. We introduce a lower-bound inspired strategy involving a computationally efficient relaxation that is based on a sorting mechanism. We further prove it achieves a lower bound close to the optimal one up to a controlled factor, and achieves an asymptotic regret q times smaller than the unstructured one. We believe this shows it is a valuable strategy for the practitioner. Last, we illustrate the performance of the considered strategy on numerical experiments involving a large number of arms.

Approximation of the unlikelihood of optimality

NeurIPS **2023**

In the paper ***Fast Asymptotically Optimal Algorithms for Non Parametric Stochastic Bandits*** and published at NeurIPS 2023 with Dorian Baudry, Rémy Degenne and Odalric-Ambrym Maillard, we consider the problem of regret minimization in non-parametric stochastic bandits. When the rewards are known to be bounded from above, there exists asymptotically optimal algorithms, with asymptotic regret depending on an infimum of Kullback-Leibler divergences (KL). These algorithms are computationally expensive and require storing all past rewards, thus simpler but non-optimal algorithms are often used instead. We introduce several methods to approximate the infimum KL which reduce drastically the computational and memory costs of existing optimal algorithms, while keeping their regret guaranties. We apply our findings to design new variants of the MED and IMED algorithms, and demonstrate their interest with extensive numerical simulations.

Reinforcement Learning

Learning in ergodic Markov decision processes

NeurIPS **2022**

In the paper ***IMED-RL: Regret optimal learning of ergodic Markov decision processes*** and published at NeurIPS 2022 with Odalric-Ambrym Maillard, We consider reinforcement learning in a discrete, undiscounted, infinite-horizon Markov Decision Problem (MDP) under the average reward criterion, and focus on the minimization of the regret with respect to an optimal policy, when the learner does not know the rewards nor the transitions of the MDP. In light of their success at regret minimization in multi-armed bandits, popular bandit strategies, such as the optimistic UCB, KL-UCB or the Bayesian Thompson sampling strategy, have been extended to the MDP setup. Despite some key successes, existing strategies for solving this problem either fail to be provably asymptotically optimal, or suffer from prohibitive burn-in phase and computational complexity when implemented in practice. In this work, we shed a novel light on regret minimization strategies, by extending to reinforcement learning the computationally appealing Indexed Minimum Empirical Divergence (IMED) bandit algorithm. Traditional asymptotic problem-dependent lower bounds on the regret are known under the assumption that the MDP is *ergodic*. Under this assumption, we introduce IMED-RL and prove that its regret upper bound asymptotically matches the regret lower bound. We discuss both the case when the supports of transitions are unknown, and the more informative but a priori harder-to-exploit-optimally case when they are known. Rewards are assumed light-tailed, semi-bounded from above. Last, we provide numerical illustrations on classical tabular MDPs, *ergodic* and *communicating* only, showing the competitiveness of IMED-RL in finite-time against state-of-the-art algorithms. IMED-RL also benefits from a light complexity.

Regret in communicating MDPs with known dynamics

ACML **2023**

In the paper ***Logarithmic regret in communicating MDPs: Leveraging known dynamics with bandits*** and published at ACML 2023 with Hassan Saber, Mohammad Sadegh Talebi and Odalric-Ambrym Maillard, we study regret minimization in an average-reward and communicating Markov Decision Process (MDP) with known dynamics, but unknown reward function. Although learning in such MDPs is a priori easier than in fully unknown ones, they are still largely challenging as they include as special cases large classes of problems such as combinatorial semi-bandits. Leveraging the knowledge on transition function in regret minimization, in a statistically efficient way, appears largely unexplored. As it is conjectured that achieving exact optimality in generic MDPs is NP-hard, even with known transitions, we focus on a computationally efficient relaxation, at the cost of achieving order-optimal logarithmic regret instead of exact optimality. We contribute to filling this gap by introducing a novel algorithm based on the popular Indexed Minimum Empirical Divergence strategy for bandits. A key component of the

proposed algorithm is a carefully designed stopping criterion leveraging the recurrent classes induced by stationary policies. We derive a non-asymptotic, problem-dependent, and logarithmic regret bound for this algorithm, which relies on a novel regret decomposition leveraging the structure. We further provide an efficient implementation and experiments illustrating its promising empirical performance.

Benchmarking Bandit algorithms: Forban

In the literature, almost all algorithmic ideas must be evaluated through numerical experimentation. Algorithms introduced in this thesis are no exception and will be benchmarked. For Reinforcement Learning, we mostly used the average-reward Reinforcement Learning package that was developed by my advisor, Odalric-Ambrym Maillard. For experimenting with Bandits, I developed Forban.[1] This library allows to easily create new Bandit algorithms thanks to a class 'SequentiAlg', instantiate a Bandit problem thanks to the functions imported from 'forban.bandits' and then benchmark those algorithms on the Bandit instance using the 'Experiment' class. Each class in Forban has multiple methods whose syntax are similar to those of the popular Scipy library. The '.fit(horizon)' method can be used on a 'SequentiAlg' instance to run a single experiment. Using '.run(nbr_exp, horizon)' on an 'Experiment' instance allows computing various prescribed statistics about a Bandit experiment. For instance, mean regret, median regret, quantiles, standard deviation, and other statistics can be computed automatically thanks to the 'Experiment' class. Such a class can be easily modified to add statistics that are relevant for the practitioner. At the end of an experiment, one is often interested in visualizing the results in some form. The '.plot()' precisely do that. It plots the various needed statistics. Having implemented such a library is light contribution but it was useful for other members and me. Some of the plotting functions were used in the RLberry library that is developed by some members of the SCOOL laboratory. In the page below, we show an example of how 'Forban' can be used to implement the IMED-kl strategy, lines 15-30, that we present in Chapter 4. The shown example consider a simple Bernoulli Bandit problem with 4 arms, lines 10-11. The usage of '.fit()', line 37, and 'Experiment', lines 43-51, is also presented.

1: For non-French speakers: **Forban** is a French word for pirate which is a sort of bandit that **explores and exploits its environment**! It can be found on github.

How to use Forban?

Minimal working example of how to use 'Forban' to implement the IMED-kl strategy, and test it on a simple Bernoulli Bandit problem.

```
import numpy as np
from bandits import BernoulliBandit
from sequentialg import SequentiAlg
from utils import Experiment, klBernoulli

# Define a Bernoulli bandit problem
# All standard deviation are set to one here
# See bandits.py for more models

means = [0.2, 0.5, 0., 0.3]
bandit = BernoulliBandit(means)

# Create the IMED class that inherits from SequentiAlg

class IMED(SequentiAlg):
    def __init__(self, bandit,
                 name="IMED",
                 params={'init': -np.inf, 'kl':klBernoulli}):
        SequentiAlg.__init__(self, bandit, name=name, params=params)
        self.kl = params['kl']

    def compute_indices(self):
        max_mean = np.max(self.means)
        if self.all_selected:
            self.indices = self.nbr_pulls*self.kl(self.means, max_mean)
                           + np.log(self.nbr_pulls)
        else:
            for arm in np.where(self.nbr_pulls != 0)[0]:
                self.indices[arm] = self.nbr_pulls[arm]*self.kl(self.means[arm], max_mean)
                                    + np.log(self.nbr_pulls[arm])

# IMED instance

imed = IMED(bandit)

# Run it using the fit method...

horizon = 500
imed.fit(horizon)

# ... or run multiple experiments

experiment = Experiment([imed], bandit,
                        statistics={'mean':True, 'std':True,'quantile':True, 'pulls':False},
                        complexity=True)
nbr_exp = 200
experiment.run(nbr_exp, horizon)

# and plot the results

experiment.plot()
```

Bandit

A set of choices: Bandit 3

3.1 A zeroth order model of decision-making

We model a **zeroth order model** of **sequential decision-making**. This model is called a **Bandit**. In our model, an **agent**, will have to **sequentially** make **decision**. Often, the agent is an **algorithm**. Mathematically, the **sequential** aspect of the decision-making process is modeled by the fact that the agent **must** make a decision sequentially at **abstract time steps** that belong to an **ordered set**. In our model, the set that is used to represent the sequential nature of the model and to index various elements of interest is arbitrary. Therefore, the ordered set by which we index the sequence of decision contains **no information** and a specific time step element is **not an information**. While it is natural to model the time and index elements using the set of natural numbers, $\mathbb{N}$, I think that it may be harmful since it makes it easier to mix **counting** with **time elapsing**. The former is a **measure of information**, the latter does not correspond to anything. This point will be make clearer once we start presenting Bandit algorithms. Mathematically, **making a decision** is modeled thanks to a **set** whose elements are interpreted as **decision** or **action**. The consequence of making a decision from this set is made known to the agent thanks to a **numerical signal** belonging to the set of real numbers, $\mathbb{R}$. Interestingly, we are interested to model only those situations in which the agent can observe the consequences of its decisions and only those. A **Bandit problem** emerges when the agent has an objective that is a function of the Bandit model. The agent will have to solve its objective using gathered information about the model through interaction with it. Of course, its interaction is modeled through the sequence of decisions and associated numerical rewards. Therefore, we can see that some additional structure might be necessary on the decision-making model to model how the agent **measure the consequences of its decision with respect to its internal objective**. **In the zeroth order model, the numerical signal associated to a decision is *constant***. However, we can add some flavor to the meaning of the word **constant**. In the first case that we present, the numerical signal is indeed represent as a constant function of the decision. This is the **Bandit control model**. In the second case, the numerical signal is a constant random variable from which we sample from. This is the **Bandit learning model**, or simply **Bandit**.

Remark. In science, order of approximation is way to referring to the accuracy of an approximation. It can be informal, as it is here, or formal, particularly when referring to the Taylor expansion of a function.

In the Bandit control model, a decision is a pointer to a real number while in the Bandit learning model, a decision is a pointer to a real random variable. In the Bandit control model, the consequence of a decision is the real number pointed by the decision while in the Bandit learning model, it is a sample from the real random variable pointed by the decision.

Bandit Control Model

In a Bandit model the agent must be able to **measure** the consequences of its decision. By measuring, we mean that it must be able to evaluate

The name ***Bandit*** comes the fact that *slot machines* in casino are also known as *"one-armed bandits"*.

the consequences of its decision. This motivates the following definition of a **Bandit control model**.

Definition 3.1.1 (Bandit Control Model) *A **Bandit control model** is a couple*

$$(A, \mu)$$

consisting of

1. *a **Borel space** A, called the control or **action set** i.e. A is a couple $(A, \mathcal{A})$ where $\mathcal{A}$ is a σ-algebra on A;*
2. *a **measurable function** $\mu : A \to \mathbb{R}$ called the **reward** (or reward per stage) function. The set $\mathbb{R}$ is endowed with its usual σ-algebra, i.e. its Borel set derived from its standard topology.*

Depending on the context (optimization objective), μ may instead be called the one stage *cost* function and denoted by the letter c.

As a matter of fact, one could define a Bandit control model as a couple (A, β) where $\beta : A \to \mathcal{P}(\mathbb{R})$ is a measurable function the set of real probability distributions $\mathcal{P}(\mathbb{R})$. In the control model, we assume that β is known and, furthermore, that the expected value $\mu : a \mapsto \mu(a) = \mathbb{E}_{X\sim\beta(a)(X)}$ can be computed. The sufficient and necessary information to define a control model in the following is the access to the function μ. This is why we denote the control model by (A, μ) which differentiate it from the learning model that is later introduced and denoted (A, β).

While it is not a problem by any mean, it is worth noting that our very first definition of a **reward** relies on a usual model of the set of real numbers. Because of the applications we have in mind, it is not problem, but one can still keep in mind that other notions of learning could emerge from other assumptions. The intuition behind this definition stems from the kind of thing we want to say abut the strategy of an agent. We may want the strategy of an agent to be random. We may also want the strategy of an agent to depend on its past actions. Depending on the objective of the agent, we may therefore need to measure the actions and its consequences. In other word, we may want to attribute a probability to a part of the action space and give it later some interpretation. To measure the results of taken actions, one must be able to **push forward** the measure of a strategy in the action space to a measure of the results of a strategy in the value space. Therefore, it only makes sense to assume that the reward function is measurable. Given a measure of decisions, we want to measure consequences; hence the definition.

In a control model, the reward function is assumed to be known to the learner. The task of an agent will be to compute a strategy that suits its objective using this knowledge. One reason for this assumption is that, often, the reward function is a function from a (small) finite set of action and hence deemed easy to learn since it is just an array. This is the situation we describe in Section 1.1. Another reason is that the reward function is deemed to hard or costly to learn and should therefore be modelled through other means and be given as an input to the agent, along with its objective. Therefore, the quality of a strategy would be dependent on both the agent and the quality of the reward model.

In a Bandit model, the agent takes decisions **sequentially**. It is allowed to remember past information. In a Bandit control model, the reward function will be assumed to be known to the agent. Thus, all the informa-

tion about the past is contained in the sequence of taken actions. This motivates the following definition of **history**.

Definition 3.1.2 (Bandit Control History) *Given a Bandit control model* (A, μ) *and its set* A *of feasible controls, for all* $t \in \mathbb{N}$*, one can define the set of* ***admissible histories*** *up to time* t

$$H_0 = \emptyset$$
$$H_t = A \times H_{t-1} = A \times A^{t-1} \qquad \forall t \in \mathbb{N}_*$$

An element $h_t \in H_t$ *is called an admissible t-history or a* ***t-history*** *and is a vector of the form*

$$h_t = (a_0, \cdots, a_{t-1})$$

with $a_k \in A$ *for* $k \in [0, t-1]$.

For all t*, the space* H_t *is equipped with a canonical* σ*-field inherited from the action set* $(A, \mathcal{A})$ *i.e.,* H_t *is a measurable space with* σ*-field* $\mathcal{A}^{\otimes t}$.

The definition of **history** allows modelling past information after a given number of interaction. I would to like to insist on the fact that, we should make a difference between the set $\mathbb{N}$ of natural numbers that is used to index the sequence of actions and the set $\mathbb{N}$ of natural number when its is viewed as the domain of a statistics, where statistics is understood as a function of the history. For instance, $t \in \mathbb{N}$ can represent the **number of interactions** with the system up to the t^{th} index of the sequence that is also written $t \in \mathbb{N}$. However, I will insist that the index should not be considered as information. Only a function from the history should be considered as such. For instance, the starting index of the sequence of decision could be set at 10^80 without changing the number of interactions. For this reason, I think that most of the time, we should not say *"at time t"* but rather say *"after t interactions"*. The differences might sound absurd, but I think that this is a source of confusion and that the former phrasing does not provide the correct intuition about the process of learning which can hinder our algorithmic thinking.

In a Bandit model, the agent must **take decisions**. Therefore, one must model how an agent take decision. This model is important because it will specify the strategies that are available to our learner and, in learning models, what the agent can and cannot learn. Stemming from our economic understanding of the concept, a decision is a sort of mapping from situations to actions. In particular, we need to model two type of questions an agent may have. First, given a situation, what is the probability of such and such action. In other word, given a situation, what is a measure of the next action an agent will take. This model the fact that an agent should take decisions based on situations. Second, given a decision and a probability measure over situations, what are the situations that lead to the probability that the given decision is larger than a threshold. In other word, we want the ability to push forward any measure on the space of situation to a measure on the space of probability modeling the decision-making strategy. This motivates the following definition of **control policy**, *i.e.* strategy, that relies on the mathematical concept of **stochastic kernel**.

Definition 3.1.3 (Stochastic Kernel) *Given two Borel spaces* $(X, \mathcal{X})$ *and* $(Y, \mathcal{Y})$*, a* ***stochastic kernel*** *on* X *given* Y *is a function* $\mathbb{P}(\cdot|\cdot) : X \times Y \to \mathbb{R}$ *such that*

1. $\mathbb{P}(\cdot|y)$ *is a probability measure on* X *for each fixed* $y \in Y$*;*
2. $\mathbb{P}(B|\cdot)$ *is a measurable function on* Y *for each fixed* $B \in \mathcal{X}$*.*

The set of all stochastic kernels on X *given* Y *is denoted by* $\mathcal{P}(X|Y)$

Replacing the space X by the set of decisions and the space Y by the set of situations, *i.e.* histories, the notion of stochastic kernel seems to be the perfect tool to model the way an agent takes decision. Hence, the following definition of **Bandit control policy**.

Definition 3.1.4 (Bandit Control Policy) *A* ***randomized control policy****, or* ***control policy****, is a sequence* $\pi = (\pi_t)_{t\in\mathbb{N}}$ *of stochastic kernels* $\pi_t \in \mathcal{P}(A|H_t)$ *on the control set* A *given histories* H_t*.*

The set of all policies is denoted by Π*.*

The word **policy** is akin to the concept of **strategy** that we mostly used until now. We remark that, in this model, a decision is made based only in the past history and that the index of time is not part of the information.

The Definition 3.1.4 is general, and it is easy to imagine situation in which strategies are much simpler in the sense that only a part of the history is used, or the randomness is removed. However, we do need a general definition because without knowing the specific objective of an agent, one cannot say anything about the complexity of the needed strategy. Rather, this very general notion of **randomized control policy** is useful because it seems general enough to capture a lot of potential decision-making models and specific enough that we can start thinking about precise mathematical statements about it. One can hope that for some objective, an *"optimal"* strategy should belong to a much simpler subclass of the space of control policy. The importance is twofold. First, a narrower definition of control and stronger assumptions would surely allow deriving more precise mathematical results and lead to a more in depth understanding of the model. Second, keeping application in mind, a smaller search space would surely mean a more efficient search algorithm. The purpose being to be able to compute a strategy corresponding to an objective, we somehow need to make sure that our assumptions lead to the existence of tractable algorithms. Thinking backward, one may be interested in simpler strategies and wonder about the space of objectives that allow an *"optimal"* policy to belong to the restricted set of strategies. Either ways, a **typology of policies** is needed.

We distinguish policies by their level of randomness and dependence on the past. A **deterministic control policy** is a sequence of kernels that are similar to deterministic functions of the past, that is to say, mappings from the space of histories to the space of actions. Since we adopted a measure theoretic approach, we must use the Dirac distribution in order to model deterministic choices. While it may seem cumbersome, it is the way of unifying all types of policies that we present. A **randomized history-agnostic policy** is a sequence of kernels that are independent of history, that is to say, a sequence of probability distribution on the

action space. A **deterministic history-agnostic policy** is a sequence of kernels that are independent of history and Dirac distribution, *i.e.* akin to a deterministic sequence of actions. A **randomized stationary policy** is a sequence of kernels that are all the same kernel. Like the history-agnostic policies, there is no dependency on the history, and additionally there is no dependency on the time set that indexes all sequences. A **deterministic stationary policy** is a sequence of kernels that are all the same Dirac distribution, *i.e.* akin to a sequence of the same action. A deterministic stationary policy is, in a sense, the simplest form of policy, always playing the same action.

Definition 3.1.5 (Typology of Bandit Control Policies) *A **control policy** $\pi = (\pi_t)_t \in \Pi$ is said to be a*

1. ***deterministic control policy*** *if there exists a sequence $(f_t)_t$ of measurable functions $f_t : H_t \to A$ such that for all $t \in \mathbb{N}$,*

$$\forall h_t \in H_t, \pi_t(\cdot|h_t) = \delta_{f_t(h_t)}(\cdot)$$

That is to say, $\pi_t(\cdot|h_t)$ is a Dirac distribution at $f_t(h_t)$. The set of deterministic control policies is denoted Π_{DC}.

Deterministic cases corresponds to randomized special cases in which the distribution of interest is a Dirac distributions.

Note the inclusion $\Pi_{DC} \subseteq \Pi$.

2. ***randomized history-agnostic policy*** *if there exists a sequence $(\phi_t)_t$ of probability distributions $\phi_t \in \mathcal{P}(A)$ such that for all $t \in \mathbb{N}$,*

$$\forall h_t \in H_t, \pi_t(\cdot|h_t) = \phi_t(\cdot)$$

The set of randomized history-agnostic policies is denoted Π_{RA}.

Note the inclusion $\Pi_{RA} \subseteq \Pi$.

3. ***deterministic history-agnostic policy*** *if there exists a sequence $(a_t)_t$ of actions $a_t \in A$ such that for all $t \in \mathbb{N}$,*

$$\forall h_t \in H_t, \pi_t(\cdot|h_t) = \delta_{a_t}(\cdot)$$

The set of deterministic history-agnostic policies is denoted Π_{DA}.

Note the inclusion $\Pi_{DA} \subseteq \Pi_{RA}$.

4. ***randomized stationary policy*** *if there exists a distribution $\phi \in \mathcal{P}(A)$ such that for all $t \in \mathbb{N}$,*

$$\forall h_t \in H_t, \pi_t(\cdot|h_t) = \phi(\cdot)$$

The set of randomized stationary policies is denoted Π_{RS}.

Note the inclusion $\Pi_{RS} \subseteq \Pi_{RA}$.

5. ***deterministic stationary policy*** *if there exists an action $a \in A$ such that for all $t \in \mathbb{N}$,*

$$\forall h_t \in H_t, \pi_t(\cdot|h_t) = \delta_a(\cdot)$$

The set of deterministic stationary policies is denoted Π_{DS}.

Note the inclusion $\Pi_{DS} \subseteq \Pi_{RS}$.

This typology is important because it is easy to understand in the Bandit model, and it will be used again in the more complicated Markov model that we present in subsequent sections. Another reason this typology is important is to understand the concept of **lower bound** that we will introduce soon. When a standard objective, called **regret**, will be defined, we will be facing the problem of comparing our strategy to other strategies. In particular, the notion of *best strategy* depends *a priori* on the space in which we are looking for a strategy. Understanding the different types of policies therefore is essential to understand the definition of the

performances metric that are used by the Bandit community.

In a Bandit model, the agent must have an **objective** which drives its decision-making process. The objective of the agent must depend on its only available information. In the case of the Bandit control model, the information that is known to the agent prior to the beginning of the interaction is the set of action that is assumed to contain no relevant information and the reward function. Because of our measure theoretic approach, it makes sense to assume that the objective will be a *"measurable problem"* of the reward function. Most of the practical objectives that we can think of would need the following finiteness assumption.

Assumption 3.1.1 (Finite Rewards) *The Bandit control model* (A, μ) *satisfies the* ***finite reward*** *assumption if for all actions* $a \in A$ *we have* $-\infty < \mu(a) < +\infty$.

This hypothesis is reasonable when interpreting the reward function as representing some physical measurement or the result of a computation based on physical phenomenon. It is safe to say that, in nature, there is nothing that can be measured or computed to be infinite and therefore, the finite reward Assumption 3.1.1 is a light one.

As it has already be envisioned in Section 1.1, the **objective** of an agent on a Bandit model is to find a strategy that maximizes a function of the sequence of observed rewards. More precisely, we will be interested in one objective that can be written as a **numerical optimization problem** of the reward function. This objective will be called the **average reward criterion.** Before asking an agent to compute or find an optimal strategy, *i.e.* compute the optimal argument of the objective written as optimization problem, we ensure that such a strategy will exist in the first place. This backward thought process explain the following assumption.

Assumption 3.1.2 (Optimal Action Feasibility) *The Bandit control model* (A, μ) *satisfies the* ***optimal action feasibility*** *if it satisfies the finite reward Assumption 3.1.1 the supremum exists,* $-\infty < \sup_{a \in A} \mu(a) < +\infty$*, and it is feasible,* $\sup_{a \in A} \mu(a) = \max_{a \in A} \mu(a)$.

The Assumption[1] 3.1.2 is necessary to ensure the **feasibility** of the Bandit objective. The **average reward criterion** is the usual objective used in the Bandit community while it is almost never presented as in the next Definition 3.1.6. However, the next definition makes the connection with Reinforcement learning much smoother and connect the criterion that is used here in Bandit and the similar average reward criterion that is used in Reinforcement learning. The average reward criterion in Reinforcement learning can be a source of confusion and the discounted reward criterion is often preferred. Presenting the usual Bandit objective as the average reward criterion will hopefully make the Reinforcement learning criterion easier to understand. Without further ado, we define the **cumulated reward** and **average reward**, concepts on which the **average reward criterion** is based.

Definition 3.1.6 (Bandit Cumulated Reward) *Let* $\nu = (A, \mu)$ *be a Bandit control model satisfying the finite reward 3.1.1 and optimal feasibility 3.1.2*

1: A more general **assumption** would be that given an objective, an "optimal" solution can be found in the **closure** of the space on which the problem is defined. Here, this space is simply the action space, but it could be some Cartesian product of multiple action spaces. Hence, we see that this assumption is a **topological** assumption.

Both the sup and max functions are measurable.

*Assumptions, and let $\pi \in \Pi$ be a control policy. The **n-stage cumulated reward** of the policy π on the Bandit ν is defined as*

$$\mathcal{G}_\nu(n;\pi) = \mathbb{E}_\pi\left(\sum_{t=1}^{n} \mu(a_t)\right). \tag{3.1}$$

*The **average reward** of the policy π on the Bandit ν is defined as*

$$\mathcal{G}_\nu(\pi) = \liminf_{n\to\infty} \frac{\mathcal{G}_\nu(n;\pi)}{n}. \tag{3.2}$$

A few remarks. The **n-stage cumulated reward** is always defined thanks to the finite assumption. It is an **expectation** over all possible histories generated by the stochastic process induced by the policy. This expectation also depends on the bandit problem and the chosen time horizon. Once the policy and problem are fixed, the n-stage cumulated reward is a function of the number of interactions. If we want to define an objective that is only based on the policy and the problem, then one should find a way to remove this dependence on the number of interactions. It means taking a limit of some sort. If it is defined, an interesting quantity would be the limit of the cumulated reward difference between two successor stages,

$$\liminf_{n\to\infty} \mathcal{G}_\nu(n+1;\pi) - \mathcal{G}_\nu(n;\pi) = \liminf_{n\to\infty} \frac{\mathcal{G}_\nu(n+1;\pi) - \mathcal{G}_\nu(n;\pi)}{(n+1)-n}.$$

A good criterion would be to try to maximize this quantity. However, the Bandit community prefers to use another criterion based on the **average reward** that is defined in Equation 3.2 of Definition 3.1.6. While these two quantities are closely related and quite often equal in cases of interest, this is not always the case.[2] The latter definition, the average reward, is a better measure of the (average) **reward per unit of interaction**. The lim inf is necessary for mathematical reason because the limit is nor guaranteed to exist *a priori*. The choice of infimum rather than supremum is based on the fact that the objective, that we define now, is to maximize the average reward.

2: Imagine that the agent collect the following sequence of rewards, 1, -1, 1, -1, 1, -1,. . . Then the average reward is equal to 0 while the other "derivative" criterion is equal to -1 (because of the lim inf).

Definition 3.1.7 (Bandit control objective) *Let $\nu = (A, \mu)$ be a Bandit control model satisfying the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions. The **Bandit control objective** is to find a control policy $\pi^\star$ such that*

$$\mathcal{G}_\nu(\pi^\star) = \sup_{\pi\in\Pi} \mathcal{G}_\nu(\pi). \tag{3.3}$$

*The quantity $\sup_{\pi\in\Pi} \mathcal{G}_\nu(\pi)$ is called the **optimal value** of the average reward criterion, and we denote it $g^\star$ and a policy $\pi^\star$ satisfying Equation 3.3 is called an average reward **optimal policy**.*

The Bandit control objective is also called the **average reward criterion**.

While the optimal value is unique, there may be several optimal policies.

In this thesis, we will make another assumption on the type of bandit problem that we study, that the cardinal of the action set is finite. Of course, one could also assume other assumption on the reward function, *e.g.* it could be a concave function of the action and that the action set is a closed convex set of $\mathbb{R}^d$. If the action set is finite and no structure is assumed, then one cannot gain an information about the result of an action by playing another action. However, we will see in the Bandit

Another popular framework that is not presented in this thesis is called *linear Bandits*.

learning setting, that, when playing an action some information is gained about actions that are not played, even in the completely unstructured finite Bandit setting.

Assumption 3.1.3 (Finite Bandit) *The Bandit control model (A, μ) is said to be **finite** if A is a discrete set whose cardinal is finite, $|A| < +\infty$. The considered σ-algebra $\mathcal{A}$ is the power set 2^A.*

Amongst the types of policies presented in the typology 3.1.5, the deterministic stationary class is the smallest one with respect to the inclusion order. For those policies, the average reward is easy to compute and is a true limit in the sense that the lim inf is equal to the lim sup in the Definition of the average reward 6.1.6. In the Bandit setting, for this type of policy, it is possible to abuse notation and confuse the average reward of the policy playing a deterministic action constantly and the reward associated to that action. In this thesis, we try to distinguish between the two thanks to the following notational convention.

Notation 3.1.1 *Let $\nu = (A, \mu)$ satisfying the finite reward assumption, then the average reward of a deterministic stationary policy $\pi_t = \delta_a$ exists, is equal to $\mu(a)$ and is denoted by μ_a.*

The fact that one can confuse the function computing the average reward of a deterministic policy with the reward function of the Bandit problem is the source of a lot of simplification. Formally, a lot of simplification of the Bandit model stems from the following equation,

$$\mathbb{E}_\pi\left(\sum_{t=1}^{n} \mu\left(a_t\right)\right) = \mathbb{E}_\pi\left(\sum_{t=1}^{n} \mu_{a_t}\right).$$

Conceptually, mixing actions and mixing policies are two different things but in the Bandit problem, actions and policies are deeply intertwined, especially at the notation level. However, by keeping this fact in mind, the transition to the more dynamic and complicated Markov control model will be smoother.

It is also useful to have a notation for the maximal value of the reward function associated to a Bandit model.

Notation 3.1.2 *Let $\nu = (A, \mu)$ satisfying the finite reward and optimal feasibility assumptions. Then, we denote $\mu^\star$ the maximal reward, i.e., $\mu^\star = \max_{a \in A} \mu(a)$.*

This notation is useful to state the next Theorem 3.1.1 since it is not hard to guess what an optimal control policy would be in the case of the Bandit control model. It is also useful to define the concept of **regret** which is the usual way to present the objective criterion of Bandit agents. However, such an objective only makes sense once the concept of cumulated reward is correctly defined and there are potentially several valid notions of (problem dependent) regret. Furthermore, such a notation really is useful once it is understood that the maximal reward will correspond to the maximal average reward possible. We are now ready to state

the following theorem about the existence of an optimal policy and its associated average reward.

Theorem 3.1.1 (Existence of an optimal policy) *Let $\nu = (A, \mu)$ be a finite Bandit control model satisfying the finite reward and optimal feasibility assumptions. Then the optimal value satisfies the equation*

$$g^\star = \max_{a \in A} \mu_a = \max_{a \in A} \mu(a) = \mu^\star, \tag{3.4}$$

i.e., $\max_{a \in A} \mu(a) = \sup_{\pi \in \Pi} \mathcal{G}_\nu(\pi)$. *Thanks to the optimal feasibility assumption, there exists a* ***deterministic stationary policy*** $\pi^\star$ *that is* ***optimal****,*

$$\pi^\star(\cdot|h_t) = \delta_\star(\cdot), \tag{3.5}$$

where $\star \in \operatorname{argmax}_{a \in A} \mu(a)$ *is called an* ***optimal action****.*

Note that $\mu^\star$ refers to the optimal value of the problem while $\mu_\star$ refers to average reward of an optimal deterministic stationary policy with action $\star \in \operatorname{argmax}_{a \in A} \mu(a)$. Of course, $\mu_\star = \mu^\star$ for any $\star \in \operatorname{argmax}_{a \in A} \mu(a)$.

The connoisseur would have surely spotted that the Equation 3.4,

$$g^\star = \max_{a \in A} \mu(a),$$

is an instance of the **optimal Bellman equation**, an equation that we will discuss in the next sections about Reinforcement learning. On the same note, the simpler equation

$$\mu_a = \mu(a),$$

is the Bellman equation associated to a deterministic stationary policy of action a. While by definition, optimal policy is an asymptotic notion, in the case of bandit, an optimal policy is anytime optimal, *i.e.*, $\pi^\star$ is a maximizer of the n-stage cumulated reward for all time horizon n.[3]

3: Note that an optimal policy is also optimal for the γ-discounted criterion $\mathbb{E}(\sum_n \mu(a_n))$ for all discount factor γ. Hence, Bandit do indeed enjoy a lot of nice properties.

The Theorem 3.1.1 proves that, as numerical values, the optimal average reward and maximal reward are equal. Intuitively, this is not a very surprising result because there is no influence of the past decision of the agent on the present decision, *i.e.*, there is no dynamic. However, the detailed path we took to present all the results set the stage for the more complicated Reinforcement learning model. It should also be reminded that while $\mu^\star$, the maximal reward, has the dimension of a reward, $g^\star$, the optimal value, has the dimension of a reward per unit of interaction. The former quantity refers to the reward function while the latter refers to a value computed from the interaction of a policy with the Bandit model.

Usually, in the Bandit literature, one is more concerned about **minimizing the regret** than **maximizing the average reward**. While the two objectives lead to the same set of optimal policies, the cumulated reward objective is more suited for generalization to Reinforcement learning and does not bear any ambiguity contrary to the notion of regret. While this notion will become clearer in the learning section, the **regret** aims to capture a **cost** of making suboptimal decisions. This way of phrasing the notion of regret emphasize on the individual decisions of the sequence and already shape a mathematical definition. Another way of phrasing this concept is that the regret aims to capture the cost of using a suboptimal strategy. In this sentence, the accent is put on the more *macro* concept of strategy and

less on the more *micro* concept of decision. Formally, one can define an n-stage control regret in one of the three equivalent following ways.

Definition 3.1.8 (n-stage Control Regret) *Let $\nu = (A, \mu)$ be a Bandit control model satisfying the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions, and let $\pi \in \Pi$ be a control policy. The **n-stage regret** of the policy π on the Bandit ν is defined as*

$$\mathcal{R}_\nu(n;\pi) = \max_{d\in\Pi}\left(\mathbb{E}_d\left(\sum_{t=1}^{n}\mu(a_t)\right)\right) - \mathbb{E}_\pi\left(\sum_{t=1}^{n}\mu(a_t)\right) \tag{3.6}$$

$$= ng^\star - \mathbb{E}_\pi\left(\sum_{t=1}^{n}\mu(a_t)\right) \tag{3.7}$$

$$= \mathbb{E}_\pi\left(\sum_{t=1}^{n}\mu^\star - \mu(a_t)\right) \tag{3.8}$$

Finite time macroscopic viewpoint. The regret is defined at the policy level.

Asymptotic macroscopic viewpoint. The regret is defined at the policy level using the asymptotic notion of gain, *i.e.* the average reward per unit of interaction.

Microscopic viewpoint. The regret is defined at the decision level.

While all the definitions are formally equal expressions, they convey a different meaning. In more complicated cases, it is possible that those definitions will not be equal. The Bandit model is convenient because those quantities are equal, yet we can start to understand the differences in those definitions within this simple framework. The Equation 3.6 compare the differences between the best possible expected cumulated reward a policy could have achieved up to the prescribed stage to the expected cumulated reward of the strategy we consider. *A priori*, the best finite time expected cumulated reward does depend on the prescribed stage. However, it is not the case in the Bandit model. This possibility makes another definition appealing, the one represented by Equation 3.7. In this case, we consider the integration of the optimal average reward per unit of interaction, $g^\star$, over the prescribed horizon, n, that is $ng^\star$. This asymptotic quantity is based on the best asymptotic growth rate of reward, computed as $g^\star$. It can be compared to the finite time quantity represented by the n-stage cumulated reward of the considered policy.[4] While interesting and probably easier to handle mathematically, we can anticipate that some information is lost compared to the previous definition. Finally, instead of comparing the cumulated rewards of strategies, one can compare local rewards directly. This viewpoint, adopted by Equation 3.8, is made possible and easy in the Bandit model because decision are completely local and $\mu^\star = g^\star = \mu_\star$.

4: While $ng^\star = n\mu^\star$, I do not think that the right-hand side of the equation should be used as such because it does not convey the right meaning. On the other hand, $n\mu_\star$ would be a better choice as it represents the n times the gain of an optimal deterministic stationary policy $\star$.

Associated to the concept of gain is the concept of asymptotic regret.

Definition 3.1.9 (Control Regret) *Let $\nu = (A, \mu)$ be a Bandit control model satisfying the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions, and let $\pi \in \Pi$ be a control policy. Let $\mathcal{R}_\nu(n;\pi)$ be the **n-stage regret** of the policy π on the Bandit ν as in Definition 3.1.8. The **average regret** of the policy π on the Bandit ν is defined as*

$$\mathcal{R}_\nu(\pi) = \limsup_{n\to\infty}\frac{\mathcal{R}_\nu(n;\pi)}{n}. \tag{3.9}$$

This quantity can be interpreted as the asymptotic cost per unit of interaction of using a given strategy over an optimal one. While, like the average reward criterion, the regret of a policy depends on both the

Bandit setting and the strategy, the optimal value is problem dependent in the case of the average reward but always zero for the regret formulation of the objective. This make the following objective for most researcher in the field.

Definition 3.1.10 (Control Regret objective) *Let $\nu = (A, \mu)$ be a Bandit control model satisfying the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions. The **regret control objective** is to find a control policy $\pi^\star$ such that*

$$\mathscr{R}_\nu\left(\pi^\star\right) = \inf_{\pi\in\Pi} \mathscr{R}_\nu\left(\pi\right). \tag{3.10}$$

The regret control objective is also called the **average regret criterion**.

From a policy standpoint, this objective is equivalent to the average reward criterion and the optimal set of policies is the same.

We now briefly present some of the core ideas that will be used in Bandit learning in the context of Bandit control. It is interesting to do so because the Bandit control model can be interpreted as a Bandit learning model in which all the reward distributions are Dirac. The agent is assumed to know that all reward distributions are Dirac. Contrary to what we did until now, we do not assume anymore that the agent has the complete knowledge of the reward function. Instead, we only assume that the agent knows the finite set of actions and can evaluate the reward function. The agent must find a strategy to suffer the minimal amount of regret possible. Because of its simplicity, we call this the **Bandit control solving problem**. One could even argue that **solving is the simplest form of learning**.

Bandit Control Solving Problem

This problem is presented extensively in the Section 1.1. The results we present here illustrate the kind of result that the community is interested in. This subsection also illustrate the rationale that is used to derive the result we are interested in.

To say something meaningful, one must confine the class of policy (or agent) that we want to derive results about. In the finite Bandit control problem, we are interested in solving the problem. By solving, it is meant that the agent must, at some point, always decide to select a maximal action. That is to say, that at some time, the regret per unit of interaction, $\mu^\star - \mu(a_t)$, of a solving agent is always null and the (asymptotic) regret is therefore equal to zero. Formally, we define the class of Bandit solving algorithm.

Definition 3.1.11 (Bandit solving algorithm) *An agent is said to be a solving agent if, whatever the finite Bandit problem, its expected time before its sequence of decisions is a sequence of optimal decisions is finite. That is to say, the strategy $\pi = (\pi_1, \pi_2, \ldots, \pi_t, \ldots)$ is such that, there exists a finite number of interaction $t^\star_{\pi,\nu} < \infty$,*

$$t^\star_{\pi,\nu} = \mathbb{E}_{\pi,\nu}\left(\min_{t\in\mathbb{N}}\left\{|H_t| \mid supp\left(\pi_t\right) \subseteq \operatorname*{argmax}_{a\in A} \mu\right\}\right).$$

This definition of a solving algorithm seems to be quite reasonable in the sense that we are only considering those algorithms that do indeed solve the problem. On the other hand, it does not seem to be too restrictive in the sense that we try to not impose unnecessary conditions or restrictions on the class we consider. Nonetheless, it is necessary to make a mathematical choice to describe what is meant by **solving**. The main restriction comes from the fact that the agent cannot be a *specialist* and always decide on the same action without taking into account the specific Bandit problem at hand. To us, **solving** implies a form of **convergence of the strategy** to an optimal policy. Characterizing what is meant by the convergence of a strategy, *i.e.* convergence of a sequence of kernels, is obviously left to the researcher's discretion. Studying if different notions of convergences implies different notions of learning is an interesting topic of discussion and research. To keep things simple, we choose to define a solving algorithm using an expected hitting time criterion, presented in Definition 3.1.11.

Now that we defined the class of learning algorithms, a question emerges. What is the minimal number of mistakes a solving agent must make in its history of interaction. To put it differently, what is the minimal amount of regret it must suffer from its **solving phase**. The concept of **problem dependent lower bound** answer this question. We talk about **problem dependent** because the regret lower bound we derive does depend on the specific Bandit instance. This is to distinguish from **worst case** lower bound that characterize the complexity of a **class** of Bandit problem. We give an example at the end of this section.

Theorem 3.1.2 (Regret Control Lower Bound) *Let $\nu = (A, \mu)$ be a finite Bandit control model satisfying the finite reward and optimal feasibility assumptions. Let n be an integer such that $n > |A|$. The n-stage regret of any* ***solving algorithm*** *π is lower bounded by*

$$\mathscr{R}_{\nu}(n;\pi) \geq \sum_{a \in \mathcal{A}} \mu^{\star} - \mu(a), \tag{3.11}$$

and the regret of any solving algorithm is lower bounded by zero, i.e.

$$\mathscr{R}(\pi) \geq 0. \tag{3.12}$$

The n-stage lower bound is easily computed from the fact that all actions must be selected at least once. Further the lower bound is said to be **tight** because there exist an algorithm whose regret upper bound matches this lower bound. This algorithm is presented in Section 1.1 and formally written as Algorithm 1 under the name **action iteration**, a direct reference to the policy iteration algorithm that we present later on. The n-stage regret lower bound is finite and a constant that is independent of the number $n \geq |A|$ of interaction. Thus, the asymptotic regret is lower bounded by zero. Crafting an optimal algorithm, *i.e.* a strategy matching a lower bound, is often complicated. However, the Bandit control problem provide a simple case. From the lower bound, we can see that an optimal algorithm can gain all the necessary information in $|A|$ interactions since the regret is a constant as soon as n is larger than the cardinal of the set A of actions. To see an algorithm emerges from the lower bound, we first remark that each time the agent decide on action, a, a cost of $\mu^{\star} - \mu(a)$ is paid in the regret. Therefore, we can informally write the n-stage lower

bound as

$$\mathscr{R}_{\nu}(n;\pi) \geq \sum_{a \in \mathscr{A}} \left(\mu^{\star} - \mu(a)\right) T_a(n),$$

where $T(a)$ represents the expected number of time a "lower bound performant" strategy would have picked the action a after n interactions. In the case of a Bandit control problem, we can see that for all action $a \in A$ and all large enough number of interactions $n \in \mathbb{N}$, $T_a(n) = 1$. The lower bound therefore seems to point to an algorithm that is such that the expected number of time an action is picked is only one. It is remarkable that this number is common to all action, therefore independent of the reward, and is independent of time, meaning that all useful information can be gathered in a finite number of interaction, here one. This is perfectly aligned with our knowledge of the Dirac distribution. Once we know that we deal with a Dirac distribution, everything there is to know can be learned after one interaction with the distribution, *i.e.*, one sample.

Therefore, the following algorithm.

Algorithm 1: action iteration: optimal bandit control algorithm

Input: A finite Bandit control model (A, μ) satisfying the finite reward and optimal feasibility assumptions;
A stopping horizon $T \in \mathbb{N}$ such that $T \geq |A|$;

1 Initialize the best current reward to $m = -\infty$;
2 Initialize the best current arm pointer $\star$;
3 Initialize step counter as $t = 0$;
4 **forall** $a \in A$ **do**
5 Play action a and collect reward $\mu(a)$;
6 **if** $\mu(a) > m$ **then**
7 $m \leftarrow \mu(a)$;
8 $\star \leftarrow a$;
9 $t \leftarrow t + 1$;
10 **while** $t < T$ **do**
11 Play action $\star$;
12 $t \leftarrow t + 1$;

This algorithm is **optimal**. The optimal nature of an agent is usually characterized by a regret upper bound that, in some sense, has to match the regret lower bound of the problem. Here, the n-stage lower bound is a constant function of n and therefore, we will deem an algorithm to be optimal if its n-stage regret is equal to the regret lower bound. However, it is often the case that optimality is stated using a regret upper bound of the considered algorithm. The reason is that the n-stage regret is a non-decreasing function of n, and we decide that an algorithm if the first order term of the regret lower bound matches the first order term of a regret upper bound of the considered algorithm. In our case, the regret lower bound is constant and therefore, an optimal algorithm must exactly match that constant.

All the regret of the action iteration strategy is incurred between the lines 4-9 of Algorithm 1. Thus, the following regret upper bound result.

Theorem 3.1.3 (Regret Upper Bound of action iteration 1) *The n-stage regret of the action iteration Algorithm 1 is equal to, and therefore upper bounded by the right-hand side of the following expression,*

$$\mathscr{R}_{\nu}(n;\pi) \overset{=}{\leq} \sum_{a \in \mathcal{A}} \mu^{\star} - \mu(a). \tag{3.13}$$

Therefore, ***action iteration*** *is an* ***optimal algorithm****.*

With this example, we just illustrated that using Dirac in some problems build a bridge between the computer science approach of the theory of complexity and the learning approach of the theory of complexity. With this built bridge, we now advance to the **Bandit learning model**.

Bandit Learning Model

While instructive, the finite Bandit control problem is a bit too easy to be really interesting from a research point of view. However, if we were to add structure to the action set or another type of constraints on the reward function, the problem may become incredibly harder and more interesting. This would be one way to proceed, but it is not the way of this thesis in which we are interested in understanding, first in simple scenario, what does it mean to sequentially learn and **model the process of solving**. Our goal is to clarify the concept of progress and information per unit of interaction. We saw in Section 1.2 that uncertainty is inherent to the concept of measure and interaction. **Learning and solving are intertwined** in a complex dynamic that is sometimes described as the **exploration-exploitation trade off**. In the Bandit learning model, the problem an agent wish to solve is mostly unknown. Therefore, one must learn the (parameters of the) problem in order to solve it. Simultaneously, we want to interact with what is considered to be the solution of the problem which can be done only if the problem has been solved. A tension will appear between this necessity of sufficient knowledge and necessity of solving with good or optimal algorithm solving this tension by being able to collect the sufficient and necessary amount of information to sufficiently enough solve the problem.

We follow the same presentation as previously to introduce the **Bandit learning model**. The main difference with the Bandit control problem is that the reward functions are now unknown real random variables from which one can no longer assume being able to compute the expected values. One can only gain information about the distributions by sampling the distributions, which model the interaction with the problem. The difference between control and learning will therefore be more on the objective and possible performances rather than the definition of Bandit.

Definition 3.1.12 (Bandit learning model) *A* ***Bandit learning model*** *is a couple*

$$(A, \beta)$$

consisting of

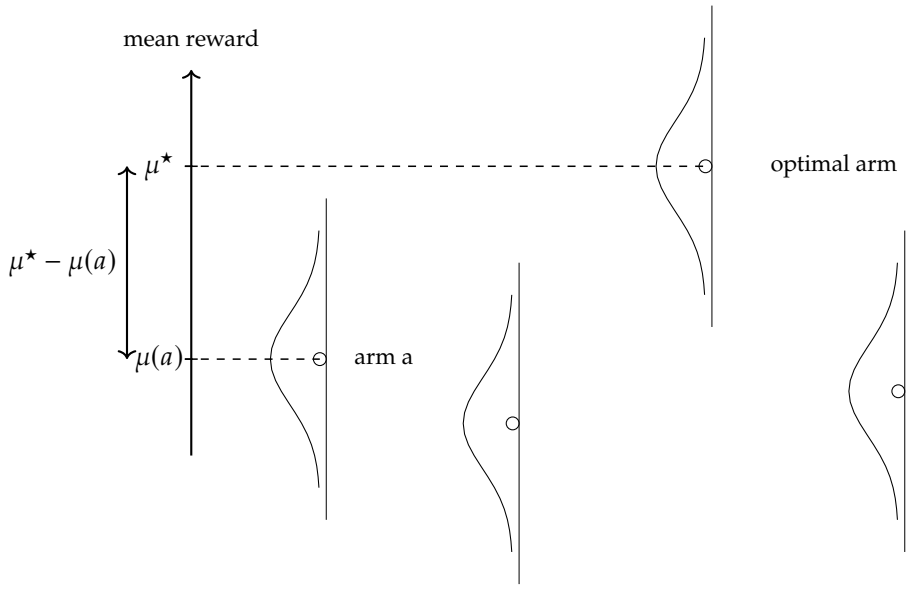

Figure 3.1: Graphical representation of a 4-arms Bandit problem with Gaussian like reward distributions

1. *a* ***Borel space*** A*, called the control or* ***action set*** *i.e.* A *is a couple* $(A, \mathcal{A})$ *where* $\mathcal{A}$ *is a* σ*-algebra on* A*;*
2. *a* ***stochastic kernel*** β *on* $(\mathbb{R}, \mathcal{B}(R))$ *given* $(A, \mathcal{A})$*.* $\beta(\cdot|a)$ *is called the* ***reward distribution*** *of arm* a*.*

The reward distribution of an action a *will be denoted as* $\beta(a)$*. The measurable function* $\mu : A \longrightarrow \mathbb{R}$ *that computes the expected value of a reward distribution*

$$\mu(a) = \mathbb{E}_{X\sim\beta(a)}(X) \tag{3.14}$$

is called the ***reward*** *(or reward per stage) function and* $\mu(a)$ *is the reward of action* a*.*

Notation 3.1.3 (Reward distribution) *Given a Bandit tuple* (A, β) *as in Definition 3.1.12, we will denote by* $\mathcal{F}$ *the set that the random variables* $(\beta(\cdot|a))_{a\in A}$ *belong to.*

We will abuse notation a bit and denote $\beta(a)$ *the random variable associated with arm* a*.*

Sometimes, the distribution or law of $\beta(a)$ *will be denoted* F_a*. Corresponding empirical quantities will be denoted using a hat, such as* $\hat{F}_a$*.*

In the Bandit community, the action set is also called the **set of arms** and an action is called an **arm**.[5] A canonical Bandit control model $(A, \mu(\cdot \mid \beta))$ can be derived from a Bandit learning model (A, β), where the reward function of the control model is such that $\mu(a|\beta) = \mathbb{E}_{X\sim\beta(a)}(X)$. For each arm, the reward of the control model is the expected reward of the arm in the original problem.

5: In the Bandit literature, the term **arm** is preferred over the term **action**. This terminology is linked to the very term Bandit that, we recall, originates from the term **one-armed bandit**, a synonym of **slot machine**.

In the Bandit learning model, the random variables of interest are assumed to have a **finite expected value**. This is the first **necessary condition** to define our problem of average reward maximization. Therefore, no random variable is assumed to have a Cauchy distribution. In a sense, it is somewhat convenient since we should be mostly interested in those

things that can be learned from statistical methods and with a somehow reasonable sample complexity. Cauchy distribution and the likes are more suited to model extreme event that cannot, almost by definition, often repeat and should therefore fall under the umbrella of other learning methods.

A learning agent will use prior information about the problem, usually the class $\mathcal{F}$ of distribution, and will past information, from a sequence of interactions, to make present decision. This model the sequential aspect of the decision-making model. The past information that is available to the agent is the **history**.

Definition 3.1.13 (History) *Given a Bandit learning model (A, β), for all $t \in \mathbb{N}$, one can define the set of **admissible histories** up to the t^{th} interaction,*

$$H_0 = \emptyset$$
$$H_t = A \times \mathbb{R} \times H_{t-1} = A \times \mathbb{R} \times A^{t-1} \qquad \forall t \in \mathbb{N}_*$$

An element $h_t \in H_t$ is called an admissible t-history or a t-history and is a vector of the form

$$h_t = (a_0, x_0, \cdots, a_{t-1}, x_{t-1})$$

with $a_k \in A$ and $x_k \in \mathbb{R}$ for $k \in [0, t-1]$.

The **history** therefore models the sequence of decision and observation that are available to the agent. Compared to the full control model, rewards are part of the history because, compared to the control model in which the learner has the knowledge of the reward function, the observed numerical signal potentially contain new information about the Bandit problem that can be used by the agent. In the Bandit learning model, while the set $\mathcal{F}$ is assumed to be known, precise information about the expected value of an arm can only be gained through sampling. In the bridging example between the Bandit control section and this Bandit learning section, the set $\mathcal{F}$ was the set of all real valued Dirac distributions. While $\mathcal{F}$ gives prior information about the arms, it is not enough to compute the arm with the largest expected value. To do so, interaction with all the arms is necessary to gather information. Thanks to the hypothesis on the set $\mathcal{F}$, only the first reward associated to a given action contains new information about that action. The problem is therefore solved after interaction with each arm once.

The history grows linearly with the number of interactions. From a theoretical viewpoint, this will not be an issue nor a matter of particular importance. From a practical question, it is interesting to start pondering over whether storing all the history is necessary, and what are the situations in which its may not be necessary[6]. As already illustrated with Dirac distribution, it may not be necessary to store all the history. Two question arise. What are the sets $\mathcal{F}$ that allows for a lossless compression of the history? If distributions in $\mathcal{F}$ are parametric and known to possess a sufficient statistic of low dimension, *e.g.* exponential family, then it is certainly possible. The second question is to quantify the loss of progress per unit samples that we discard. That is to say, if we decide to compress the history, what are the expected loss of information regarding the problem we are trying to solve. We will see that some strategies are suboptimal because of history compression but still manage to have

6: When the word necessary appears in a mathematical essay, the word sufficient (statistics) is not too far away.

competitive performances compared to optimal strategies. In this thesis, we try to address some of those issues in Chapter 4 that is based on the paper *Fast Asymptotically Optimal Algorithms for Non-Parametric Stochastic Bandits*.

Definition 3.1.14 (Bandit Learning Policy) *A **bandit learning policy**, or **randomized bandit learning policy** or **policy**, is a sequence $\pi = (\pi_t)_{t\in\mathbb{N}}$ of stochastic kernels $\pi_t \in \mathcal{P}(A|H_t)$ on the control set A given histories H_t.*

The set of all policies is denoted by Π.

This definition is exactly the same as Definition 3.1.4 and what makes those set of policies different is the sequence of histories. The reason for this similarity is that our definition of a policy as a sequence of kernels based on past information is general enough to capture the essence of the concept of **past-information based decision-making**. The typology of policies that is made in Definition 3.1.5 is the same for the Bandit learning model. We again insist that in this model, a decision is made based only in the past history and that the index of the sequence, also time index, is not part of the information. In popular algorithms, when the time index is used, one should read it as the total number of interactions with the Bandit problem. It is a quantity that can be computed from the history H. By the Definition 3.1.13, the total number of interaction is half the length of the history.

Recall that a kernel allows us two important things. First to attribute probabilities over decisions based on past information. Second, given a present decision, to push forward a probability over the history to the space of probability distributions (of choosing that action).

Bandit objectives In a Bandit model, the agent must have an **objective** which drives its decision-making process. In the Bandit learning community, two objectives are popular.

Best arm identification The first, that we do not study in this thesis, is called the **best arm identification** problem. As the name suggest, an agent with such an objective aims to find an optimal arm within the set of available actions using the gathered information through the course of its interactions. Of course, some constraints are necessary for the problem to be well-defined. A first option is to restrict the number of interactions with the Bandit model. Given this maximal number of interactions, the agent must then maximize its probability of outputting an optimal arm by the end of the allowed interaction. Such a probability obviously depends on assumptions that are made about the probability distributions, $\mathcal{F}$. A second option is to restrict the uncertainty about the answer of the agent by the end of the interaction. In this case, the agent is free to interact with the model for as long as it deems necessary for its perceived uncertainty about its knowledge of a best arm to go below a threshold prescribed by the practitioner.

This objective is useful to model situations in which making decisions that are not optimal, *i.e.* not associated to a maximal expected reward, is not seen as harmful during the testing phase. For instance, this can be the case if the practitioner can access a **perfect simulator** that model the reward distributions associated to a set of decisions. There are cases were sampling from a distribution is possible but knowing the distributions statistics is impossible or prohibitive, hence the success of Monte Carlo simulations. In this case, the practitioner may want to *find* the optimal

decision in the simulator before actually implementing it in real life. The cost of making suboptimal decisions is therefore completely virtual. When the practitioner must take a decision within a given relatively short timeframe[7] compared to the number of interactions that would be necessary to achieve the preferred level of uncertainty, the framework that constrain the number of interactions is better. In this case, the total time is equal to the number of interactions multiplied by the time it takes to sample a reward and process that sample. When the practitioner is not constrained by the time but rather by its uncertainty level about its decision, the framework that precisely constrains this is the preferred methodology.

7: Here *short time* is to be compared to the number of interaction that would be necessary to achieve a good enough precision for the practitioner. That is to say, the number of interactions is *short* compared to the sample complexity required to achieve a level of uncertainty that would be preferable to the practitioner.

Regret minimization The second, that we study in this thesis, is called the **cumulated reward maximization** or **regret minimization** problem. As the name suggest, an agent with such an objective aims to get, through the course of its interactions, a sequence of rewards that, when added up, is maximal. As such, the problem is still ill-defined because of the inherent randomness of the model. What will matter is the expected cumulated reward of a policy. Certainly, assumptions about the real random variable, *i.e.* assumption on $\mathcal{F}$, are necessary. An assumption about a rate of convergence of the empirical mean to the mean or an assumption about the dispersion of samples around the true mean will be necessary to characterize the behavior of an agent. Concentration results and large deviation theory, as presented in the eponymous Dembo and Zeitouni's book [1] provides useful tool to model such situations. At the heart, the main question an agent ask about an arm is, how *typical* is the sequence of random rewards that has been sampled? Thus, the importance of the Sanov's Theorem. This objective is useful to model situations in which one must repeatedly make decision with uncertain outcome while the expected outcome of the agent's decision has consequences that cannot be ignored. In that case, an objective would be to try as hard as possible to minimize the number of decision that are suboptimal, where suboptimal means that the expected reward of sampled decision is less than the maximal possible expected reward in the set of available decisions.

[1]: Dembo et al. (2010), *Large Deviations Techniques and Applications*

Depending on the context, working with expected value without constraints on the standard deviation, quantiles, or some other measure of deviation to the expected reward might not be the right model to consider. In other words, some applications requires a measure of the risk to be taken into account. However, the fact that risk is not explicitly included in the performance objective of a strategy allows for some really greedy strategies to be used. In Section 5 that is devoted to the paper *stochastic bandits with groups of similar arms*, we illustrate how the possibility of using a large risk taking strategy can work to our advantage.

Expected Cumulated Reward The n**-stage cumulated reward** measure the expected performance of an agent on a given Bandit problem after n interactions. It is used to define the asymptotic measure of performance, the expected **average reward**. The performance of an algorithm will be measured by comparing its expected cumulated reward to the best possible cumulated reward achievable by a given class of algorithm. This comparison is often phrased using the notion of **expected regret**, which

we introduce after defining the cumulated reward. For our problem to make sense, we will assume from now on that the considered Bandit learning problem are such that their associated Bandit control problem satisfies the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions, and that it is finite, *i.e.* the cardinal of the set of arm is finite.

Definition 3.1.15 (Bandit Expected Cumulated Reward) *Let $\nu = (A, \beta)$ be a Bandit learning model such that its associated control model $(A, \mu(\cdot|\beta))$ satisfies the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions, and let $\pi \in \Pi$ be a learning policy. The n**-stage expected cumulated reward** of the policy π on the Bandit ν is defined as*

$$\mathcal{G}_\nu(n;\pi) = \mathbb{E}_{\pi,\nu}\left(\sum_{t=1}^{n} x_t\right). \tag{3.15}$$

*The **expected average reward** of the policy π on the Bandit ν is defined as*

$$\mathcal{G}_\nu(\pi) = \liminf_{n\to\infty} \frac{\mathcal{G}_\nu(n;\pi)}{n}. \tag{3.16}$$

Remark While the n-stage reward is defined using a sum over the time index, $\mathcal{G}_\nu(n;\pi) = \mathbb{E}_{\pi,\nu}\left(\sum_{t=1}^{n} x_t\right)$, we should really see it as a sum of the rewards elements H_{reward} of the history H which is a random sequence depending on the problem ν and policy π,

$$\mathcal{G}_\nu(n;\pi) = \mathbb{E}_{\pi,\nu}\left(\sum_{x\in H_{\text{reward}}} x\right). \tag{3.17}$$

The implicit constraint is that the number of interactions is n, *i.e.* the size of the history is $2n$.

The n**-stage cumulated reward** is always defined thanks to all the assumptions. It is an **expectation** over all possible histories generated by the stochastic process induced by the **policy** on the **Bandit problem**. The **expectation** depends on the Bandit problem because the sequence of rewards is generated from samples of the reward distributions. By conditioning each reward x_t on a_t, *i.e.* on past information that is available since choosing an action precede the reward sample, and using the law of total expectation[8], one can derive that Equation 3.15 can be written in another form that depends only on the decisions and their associated expected rewards,

$$\mathcal{G}_\nu(n;\pi) = \mathbb{E}_{\pi,\nu}\left(\sum_{t=1}^{n} \mu(a_t|\beta)\right), \tag{3.18}$$

where the expectation is of course still subject to the Bandit problem ν because choices made by the policy depends on the history which itself depends on the problem ν. This equation can be written as a function of H_{action}, the extracted sequence of action from the history,

$$\mathcal{G}_\nu(n;\pi) = \mathbb{E}_{\pi,\nu}\left(\sum_{a\in H_{\text{reward}}} \mu(a|\beta)\right). \tag{3.19}$$

8: This proposition states that one can iterate expectations in the sense that $\mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X \mid Y))$ where the expectations and random variables are assumed to be correctly defined.

Number of samples In our example ending the section devoted to Bandit control, we foresaw that the number of times an arm has been selected up to a given stage n could prove relevant. In particular, we envision how useful it could be in the design of lower bound based algorithms. Therefore, we introduce the following notation.

Recall that the name Bandit comes from a casino term to designate a slot machine. Therefore, the verb **to pull** is used by the Bandit research community to mean **to sample**. The verb **to play** is also used as a synonym. For instance, one could say that an agent pulled or played arm $a \in A$ to mean that it sampled the distribution $\beta(a)$ associated to the action a.

Notation 3.1.4 (Number of pulls) *Given a Bandit problem $\nu = (A, \beta)$ and a policy π, the **random variable** $N_a(n)$ counts the number of time action a has been pulled by the policy π after n interactions,*

$$N_a(n) = \sum_{t=1}^{n} \mathbb{1}\{a_t = a\}. \tag{3.20}$$

While this random variable depends on ν and π, no ambiguity will occur from removing the explicit dependency from $N_a(n)$. Denoting it $N_{\nu,\pi,a}(n)$ would be too cumbersome.

Again, N_a is a statistic, *i.e.* measurable function, of the history H. We will therefore denote it $N_a(H)$ to emphasize that the main random object from which information come is the history. The associated overloaded definition is

$$N_a(H) = \sum_{d \in H_{\text{action}}} \mathbb{1}\{d = a\}\ , \tag{3.21}$$

where H_{action} is the extracted sequence of action from the history.

Using the law of total expectation to write Equation 3.15 in its most used and usable form. The idea is that we can regroup actions and count them together when they are the same. Formally, we derive that

$$\mathcal{G}_\nu(n;\pi) = \sum_{a \in A} \mu(a) \mathbb{E}_{\pi,\nu}(N_a(n)). \tag{3.22}$$

History dependent form The Definition 3.1.15 makes the cumulated reward a function of n, the number of interaction with the Bandit problem. This is convenient since we will want to study the behavior of this function $\mathcal{G} : \mathbb{N} \to \mathbb{R}$. However, I would prefer to define the cumulated reward as a random variable explicitly function of the history H,

$$G(H) = \sum_{x \in H_{\text{reward}}} x\ ,$$

and the expected cumulated reward as $\mathbb{E}_{H\sim(\pi,\nu)}(G(H))$. In my opinion, this way of writing thing gives better intuitions and help consider different research questions compared to the usual time-related notation. The history related notations help to avoid using the time index n where it is not needed and may help the thought process of solving the Bandit learning problem.

Average Reward Criterion One can now define the **Bandit learning objective**. With this objective in mind, we will define the **class of learning policy**, similarly to what we did when defining the class of solving polices for the Bandit control problem, at the end of the previous section. The Bandit learning objective on a Bandit model (A, β) is defined with respect to the Bandit control model in the sense that the learning objective is

to craft a learning policy that "*converges*" to an optimal control policy on the associated control problem $(A, \mu(\cdot|\beta))$. Deterministic policies are included in the set Π of all policies defined in 3.1.14.

Definition 3.1.16 (Bandit learning weak objective) *The **Bandit learning weak objective** is to **find a policy** $\pi^\star$ such that, for all finite Bandit learning model $\nu = (A, \beta)$ satisfying the finite reward and optimal feasibility assumptions,*

$$\mathcal{G}_\nu\left(\pi^\star\right) = \sup_{\pi\in\Pi} \mathcal{G}_\nu(\pi)\,. \tag{3.23}$$

To achieve this objective, the policy $\pi^\star$ must sequentially extract information about the expected reward function through its interactions with the distributions of the problem.

This criterion 3.1.16 is also called the (weak) **average reward criterion**, in particular in the context of Reinforcement Learning. It is interesting because it only says something about the asymptotic sampling rate of suboptimal actions. By Definition 3.1.15 of the expected cumulated reward, the optimality criterion implies that the sampling rate $\frac{\mathbb{E}N_a(n)}{n}$ of a suboptimal arm a should converge to zero, as the number of interactions n grows, for any policy $\pi^\star$ satisfying the Equation 3.23 of the Bandit learning criterion 3.1.16, *i.e.*

$$\frac{\mathbb{E}_{\pi^\star,\nu}\left(N_a(n)\right)}{n} \underset{n\to\infty}{\longrightarrow} 0$$

for a suboptimal arm $a \in A$. Asymptotically, the learning of a good enough policy satisfying the Equation 3.23 is not visible in the sense that suboptimal arm are pulled a sublinear number of the interactions. This definition weak objective is used to define the set of policies that are said to be **consistent** in some early work, see [2–4].

[2]: Robbins (1952), 'Some aspects of the sequential design of experiments'
[3]: Fox et al. (1973), 'Adaptive Policies for Markov Renewal Programs'
[4]: Lai et al. (1985), 'Asymptotically efficient adaptive allocation rules'

However, this criterion is *weak* in the sense that it does not discriminate between good enough policies that are such that $N_a(n) \propto \sqrt{n}$ and $N_a(n) = \log n$ while the two are order of magnitudes different. Ideally, one would want the *fastest* possible convergence, *i.e.* smallest sampling rate of suboptimal arm. As we saw in the Dirac example, a sampling rate of zero is possible because information for solving the problem is available after a finite number of samples. On the other hand, for Cauchy distribution, no certain convergence to a good enough policy is possible because the problem is ill-defined. It is likely that properties of the distributions $\mathcal{F}$ will be key to understand the optimal sampling rate. We prefer to talk about sampling rate because we will mainly be interested in first order term in a Taylor expansion of $\mathbb{E}_{\pi,\nu}\left(N_a(n)\right)$ that asymptotically matters.

Uniformly fast converging policy However, to derive more interesting guarantees and a more interested theory, one must restrict the class of what we consider to be **learning** policy. We did a similar thing in the section devoted to the Bandit control model, where we defined the **solving algorithms** to be those algorithms that interact with each suboptimal arm a **finite** number of time, see Definition 3.1.11. A policy π

that is consistent, *i.e.* satisfy

$$\frac{\mathbb{E}_{\pi,\nu}\left(N_a(n)\right)}{n} \underset{n\to\infty}{\longrightarrow} 0\ ,$$

is considered to be *weakly* learning the Bandit problem, while a **uniformly fast converging** learner π satisfy the stronger property that

$$\frac{\mathbb{E}_{\pi,\nu}\left(N_a(n)\right)}{n^\alpha} \underset{n\to\infty}{\longrightarrow} 0$$

for all $\alpha > 0$. This basically say that a policy is considered *learning* is its expected number of suboptimal samples grows at most logarithmically. This mechanically reduces the set of Bandit problem we can consider to be *learnable*. One can even see the two kind of restrictions to be *dual*. In one case, one restrict the class of problem we wish to solve, *e.g.* the class of Bandit problem with light tail distributions. On this class of problem, there exists consistent policies with the smallest possible sampling rate of suboptimal arm, logarithmic. In the other case, one can restrict the class of algorithm that we consider, *e.g.* uniformly fast converging. This restricts the class of problem we may consider since we want those algorithms to be consistent, *i.e.* solve the problem asymptotically. Because of the restriction on their sampling rate of suboptimal arms, algorithms is the considered class must gather information quickly enough, which is possible only if the problem allows to do it.

Towards an optimality criterion Given a set Ω of policies, some are considered to be better than others. Some policies are considered so much better than they are deemed the epithet **optimal**. To compare policies, one must create an **order** between them. For this purpose, a function $\mathcal{O} : \Omega \to \mathbb{R}$, or to any ordered set would be sufficient. Interestingly, to compute a notion of best or optimal policy, we do not need the codomain to be totally ordered. Instead, we only need $\mathcal{O}(\Omega)$ to be partially ordered, have at least one maximal element and such that all maximal elements are equivalent in some sense. More on that when writing about policy iteration in Reinforcement Learning. This order is created by the **cumulated reward** or **regret** functions. Since we did not yet introduce the regret function, we focus on the cumulated reward. For the order to be useful, it must somehow uniquely determine the set of **optimal** policies. In this thesis, we will focus on what is called **problem dependent** optimality as opposed to min-max or **worst case** optimality. We fix a class ζ of Bandit problems on which policies in Ω can be consistent. A worst case study would be interested in characterizing the maximal gain that is achievable by policy its worst corresponding Bandit problem, $\max_{\pi\in\Omega}\min_{\nu\in\zeta}\mathcal{G}_\nu(n;\pi)$. In the problem dependent setting, one is more interested in a notion of *best achievable gain in a specific instance* $\nu \in \zeta$ for a generic adaptive policy $\pi \in \Omega$ that is not fine-tuned on ν but rather the same for all problem ν. Roughly speaking, an algorithm π will be better than an algorithm π' if its gain on the instance ν is larger than the one of π',

$$\mathcal{G}_\nu\left(n|\pi\right) \geq \mathcal{G}_\nu\left(n|\pi'\right)\ .$$

Ideally, one would π to satisfy this equation for all number n of interaction and problem ν in the considered class ζ. In general, it is not possible to find such a policy π that maximizes the gain uniformly on ζ for any fixed

n. Rather, one would use an asymptotic notion of optimality, closer to the original weak criterion which is about asymptotic sampling rate.

Uniformly maximum convergence rate The term **fast convergence** referred to the highly sublinear sampling rate of suboptimal arms of a **consistent** policy, *i.e.* a $o(n^\alpha)$ for all $\alpha > 0$. The **uniform** nature of **uniform fast convergent** algorithm resides in the independence to the specific Bandit problem ν in the considered class ζ, meaning that the fast convergence, or sublinear sampling rate, must be achieved on all problem within the class of problems. A policy will be optimal if it is uniformly fast convergent with **maximal convergence rate**. Furthermore, this maximal convergence rate must be achieved **uniformly** on the class ζ of problems. Formally, a policy π has a **uniformly maximum convergence rate** if it is such that for all policy π' in Ω,

$$\liminf_{n\to\infty} \frac{\mathcal{G}_\nu(n;\pi)}{\mathcal{G}_\nu(n;\pi')} \geq 1 \ ,$$

for all Bandit problem $\nu \in \zeta$. The term **maximum** is formalized by the *greater than one* and the **convergence rate** is conveniently written using the **limit** of the **ratio** of the gain. The term **uniform** again comes from the fact that this inequality must be satisfied for all Bandit problem in the considered set ζ. Usually, Ω will simply be the set of **uniformly fast convergent** policies in the forthcoming theorems. However, in some cases, it may be a smaller set due to some structural assumption or additional constraint to our problem. For instance, one may be interested in those policies that are linear in some parameters, or those policies that do not store all the history of information but rather compress information with some loss, but not the point that the policy is no longer uniformly fast convergent.

Remark Somehow, I always felt that this notion of **uniformly fast convergent** policy was a bit overlooked in most Bandit presentations and lectures. Even in papers, this assumption always seemed to be a bit *ad hoc* and with the sole justification of *it makes things work*. However, since we do not need this assumption to craft algorithms with uniformly fast convergent rate, *e.g.* UCB [5], it somehow justifies the interest the community to this specific class of algorithm. However, we do originally need assumption on the class of Bandit problem we consider to make said algorithm be uniformly fast, *e.g.* with element of $\mathcal{F}$ being sub-Gaussian of known proxy-variance. Seeing assumptions on Bandit problems and the class of algorithms that solve them as somehow dual from each other[9] help to better make sense of this notion. I think that this notion of **uniformly fast convergence** put a lot of weight on the *practical viewpoint* by specifying what we consider to be practical Bandit learner. Mechanically, it also restrains the class of problem that we can learn and perhaps, restrict our thoughts on other interesting learning frameworks where learning that fast is not possible.

[5]: Auer et al. (2002), 'Finite-time analysis of the multiarmed bandit problem'

9: Do we add constraint on the type of problem we solve or on the practical algorithms that we consider solvers of problem.

As a final remark, we point that the term **uniformly fast convergent** is sometimes called **uniformly efficient** policy. We also note that the main reference book used in Bandit [6] uses the term **consistent** policy, Definition 16.1 page 207 of the September 2023 version, which is not what

[6]: Lattimore et al. (2020), *Bandit algorithms*

was done in original papers, *e.g.* [7, 8]. The term **strongly consistent** is preferred in some work, *e.g.* [9]. This diversity of terms and absence of consensus on an expression is, in my eye, an indicator that this classic hypothesis is not discussed that often in the community.

[7]: Burnetas et al. (1996), 'Optimal adaptive policies for sequential allocation problems'
[8]: Lai et al. (1985), 'Asymptotically efficient adaptive allocation rules'
[9]: Kaufmann et al. (2012), 'Thompson Sampling: An Asymptotically Optimal Finite-Time Analysis'

Bandit learning objective: minimizing the regret Usually, in the Bandit literature, one is more concerned about **minimizing the regret** than **maximizing the average reward**. While the two objectives lead to the same set of optimal policies, the cumulated reward objective is more suited for immediate generalization to Reinforcement learning. Furthermore, the average reward criterion is somewhat overlooked in the Reinforcement Learning community where the discounted reward and even finite horizon settings are favored. However, this criterion is a more direct generalization of the traditional Bandit learning setting. An in depth understanding of the connection between the two framework, stemming from the former really help better grasping more difficult concept from the latter. For reference, the average reward criterion is presented in chapter 10 of the reference book [10].

[10]: Sutton et al. (2018), *Reinforcement learning: An introduction*

Where gain measure the raw cumulated rewards without any reference to a standard value, the **regret** aims to capture a **cost** of making suboptimal decisions which necessitate a comparison to a reference that is considered or known to be optimal. While the comparison is quite unambiguous in the Bandit learning setting, a bit of definition-based ambiguity can emerge in the Reinforcement Learning setting. This is due to non-locality issue that appears due to the added dynamics in the Reinforcement Learning setting. However, the ambiguity is kind of raised if we understand the **regret** as an analytical tool to measure, compute and study the **rate of convergence of a learning strategy on** (A, β) **to an optimal control strategy on** $(A, \mu(\cdot|\beta))$. The rate of convergence of a strategy π is measured by the rate of convergence of the sequence of average cumulated reward $\frac{\mathcal{G}_\nu(n;\pi)}{n}$ to $g^\star = \max_a \mu(a)$. We are interested in those strategies that are such that $\left|\frac{\mathcal{G}_\nu(n;\pi)}{n} - g^\star\right|$ converges to zero when the number of interactions n grows and such that the convergence to zero is as fast as possible. This quantity is called the **regret**, and it admits several useful representations each having an interesting interpretation.

Formally, one can define an n-stage control regret in one of the three equivalent following ways.

Definition 3.1.17 (n-stage Learning Regret) *Let $\nu = (A, \beta)$ be a Bandit Learning model satisfying the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions, and let $\pi \in \Pi$ be a learning policy. Let $(A, \mu(\cdot|\beta)) = (A, \mu)$ be the associated Bandit control problem and denote by $\mathfrak{D}$ the set of deterministic*

*control policy. The n-**stage regret** of the policy π on the Bandit ν is*

$$\mathcal{R}_\nu(n;\pi) = \max_{d\in\mathcal{D}} \left(\mathbb{E}_d \left(\sum_{t=1}^{n} \mu(a_t) \right) \right) - \mathbb{E}_{\pi,\nu} \left(\sum_{t=1}^{n} \mu(a_t) \right) \tag{3.24}$$

$$= n g^\star - \mathbb{E}_{\pi,\nu} \left(\sum_{t=1}^{n} \mu(a_t) \right) \tag{3.25}$$

$$= \mathbb{E}_{\pi,\nu} \left(\sum_{t=1}^{n} \mu^\star - \mu(a_t) \right) \tag{3.26}$$

Finite time macroscopic viewpoint. The regret is defined at the policy level.

Asymptotic macroscopic viewpoint. The regret is defined at the policy level using the asymptotic notion of gain, *i.e.* the average reward per unit of interaction.

Microscopic viewpoint. The regret is defined at the decision level.

The Equation 3.24 and the Equation 3.25 are valid definitions of the analytical object that is the regret, *i.e.* by definition these equations measure the difference of gain to that of the used strategy. In Reinforcement Learning, where dynamic is added, these two quantity are not equal which is why we take the time to distinguish between the finite time and asymptotic viewpoint. When interested in asymptotic convergence rate, the difference between the two will vanish, and we will find that the generalization of Equation 3.25 to be more practical, especially as it is already written in a partially asymptotic fashion. On the other hand, the Equation 3.26 is a consequence of one of the above two definitions (but it is easier to start with the second Equation 3.25). Indeed, this equation can be derived from the fact that all decisions are locals, *i.e.* the action set and β function are independent of the history, and the fact that $\mu^\star = g^\star = \mu_\star$. This equation allows us to understand the regret at a per-step scale, more micro, rather than at the original policy scale, more macro. The macro behavior can be understood as the integration of a lot of suboptimality micro costs.

Notation 3.1.5 (Suboptimality gap) *The quantity $\mu^\star - \mu(a)$, sometimes denoted Δ_a without explicit reference to the original bandit problem, is called the **suboptimality gap** of arm a.*

Equation 3.26 then explain the regret as an integration of all suboptimality gaps suffered by the learner along its learning trajectory, *i.e.* its history.

In the section devoted to the Bandit control model, we explained that its is convenient that those quantities are equal but that we should not forget that they convey different meanings and interpretations. We recall a few key points. The Equation 3.24 compare the differences between the best possible expected cumulated reward a deterministic control policy could have achieved up to the prescribed stage to the expected cumulated reward of the strategy we consider. The reason we consider the set $\mathcal{D}$ of deterministic control policy is that we know from the previous part that there exist an optimal control policy within that set. A similar fact will be used when defining the regret in Reinforcement Learning. *A priori*, the best finite time expected cumulated reward does depend on the prescribed stage. However, it is not the case in the Bandit model. This possibility makes another definition appealing, the one represented by Equation 3.25. In this case, we consider the integration of the optimal average reward per unit of interaction, $g^\star$, over the prescribed horizon, n, that is $ng^\star$. This asymptotic quantity is based on the best asymptotic growth rate of reward, computed as $g^\star$. It can be compared to the finite time quantity represented by the n-stage cumulated reward of the

considered policy.[10] This asymptotic viewpoint lead us to the definition of asymptotic regret and **consistent policy**.

10: While $ng^\star = n\mu^\star$, I do not think that the right-hand side of the equation should be used as such because it does not convey the right meaning. On the other hand, $n\mu_\star$ would be a better choice as it represents the n times the gain of an optimal deterministic stationary policy $\star$.

Definition 3.1.18 (Learning Regret) *Let $\nu = (A, \beta)$ be a Bandit Learning model satisfying the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions, and let $\pi \in \Pi$ be a control policy. Let $\mathcal{R}_\nu(n;\pi)$ be the n-**stage regret** of the policy π on the Bandit ν as in Definition 3.1.17. The **average regret** of the policy π on the Bandit ν is defined as*

$$\mathcal{R}_\nu(\pi) = \limsup_{n\to\infty} \frac{\mathcal{R}_\nu(n;\pi)}{n}. \tag{3.27}$$

Consistent policies This concept is useful in the sense because it allows to define the set of consistent polices, *i.e.* those policies that converge, regret-wise, to an optimal control policy. However, in the opinion of many theoreticians and practitioners alike, it is a weak criterion and another must be later introduced to define what is deemed to be call and optimal set of policies.

Definition 3.1.19 (Consistent policy) *A policy π is consistent on a set ζ of Bandit Learning problem if its average regret is equal to zero for all $\nu \in \zeta$, i.e.*

$$\mathcal{R}_\nu(\pi) = 0\ .$$

*This characterizes the desired notion of convergence in the **policy space***

Note that the convergence, because it is interested in the regret, does not mean that the policy ends up being stationary or deterministic. One can only say that the asymptotic support of decision is within the subset of maximal arms. Interestingly, there are popular strategies that are studied, such as Explore-Then-Commit (ETC), that are known to be non-consistent policies, see [11]. In the Reinforcement Learning literature, the ϵ-greedy strategy also is non-consistent (in the sense that it does not converge to an optimal strategy), yet popular, see [10]. One of the main reason is that asymptotic convergence is little to nothing to the practitioner if one cannot see in the reasonable experimental time the convergence regime. Non-consistent strategies with smaller finite time regret are therefore sometimes deemed better. This is reminiscent of other phenomenon that sometimes appear in the field of computational statistics, such as MCMC methods, where prohibitive *burn-in* phase of optimal algorithm makes suboptimal algorithm more suitable to practical usages.

[11]: Garivier et al. (2016), 'On Explore-Then-Commit strategies'

[10]: Sutton et al. (2018), *Reinforcement learning: An introduction*

Amongst consistent policies, one may be interested in a subset of those policies. The subset of polices that we are interested in and that will restrict the set of problem that can be solved is the subset of uniformly fast convergent polices.

Definition 3.1.20 (Uniformly fast convergent policies) *Given a set ζ of Bandit problems, a policy π is uniformly fast convergent on ζ if for all Bandit problem $\nu \in \zeta$, the regret $n \mapsto \mathcal{R}_\nu(n;\pi)$ is negligible compared to n^α for all $\alpha > 0$, i.e.*

$$\forall \nu \in \zeta,\ \mathcal{R}_\nu(n;\pi) = o\left(n^\alpha\right)\ .$$

Again, we insist that the word uniform is attached to the set ζ and the word fast to the highly sublinear growth rate of the regret function $o(n^\alpha)$ for all positive α.

Amongst the subset of policies of interest, here the subset of uniformly fast convergent polices, some may converge faster than others on some problem. Some may converge faster than any other policies on all problem in the set ζ. Those policies have the maximal convergence rate and achieve this maximality criterion uniformly on ζ.

Definition 3.1.21 (Uniformly maximal convergence rate policies) *Given a set ζ of Bandit problems, a policy $\pi^\star$ is said to have uniformly maximal convergent rate on ζ if for all Bandit problem $\nu \in \zeta$, and all uniformly fast convergent policy π the regret $n \mapsto \mathcal{R}_\nu(n;\pi^\star)$ is asymptotically smaller than the regret $n \mapsto \mathcal{R}_\nu(n;\pi)$, i.e.*

$$\forall \nu \in \zeta, \ \limsup_{n\to\infty} \frac{\mathcal{R}_\nu(n;\pi^\star)}{\mathcal{R}_\nu(n;\pi)} \leq 1 \ .$$

We consider the lim sup of the ratio because the limit may not exist and the lim sup is more restrictive than a lim inf and better suited to define a sound notion of optimality. Note how this definition corresponds to the one we introduce as a lim inf on the gain. It can be read as the fact that we want to control the worst adherent point of the sequence of regret ratio.

Existence a policy with uniformly maximal convergence rate While the non-emptiness of the set of policies with uniform maximal convergence rate is unknown at this point, one can still define it. The proof of existence of such a policy with maximal convergence rate is hopefully possible and even better, it is a **constructive proof**.[11] Upon deriving a lower bound on the regret characterizing the minimal regret that any uniformly fast convergent policy must incur, hence also uniformly maximal convergent rate policies, one would dispose of a practical analytical tool to check if a policy is within the set of uniformly maximal convergent policies. A lower bound has the generic form

$$\liminf_{n\to\infty} \frac{\mathcal{R}_\nu(n;\pi)}{\log(n)} \geq \mathcal{C}(\nu;\zeta) \ ,$$

where $\mathcal{C}(\nu;\zeta)$ is a term that depend on the problem ν and the class ζ that the problem is assumed to belong to. Because it depends on ν, it is problem dependent. It also depends on ζ, the class of problem ν belong to. By constructing a policy matching this lower bound, *i.e.* such that the regret upper bound of such a policy is equal to the computed lower bound, one would prove that this interesting set of optimal policy is not empty. The proof of existence of optimal policies is therefore constructive.

11: While I do consider myself a *constructivist* I do find it more satisfying to have a constructive proof whenever possible. This may be why my PhD ended up being about mathematical algorithm, constructive method by definition.

Remark The class ζ represent prior knowledge that we have on the problems that a learner may solve and the properties of this class condition the difficulty of a given problem. With this notion of *prior*

knowledge, the difficulty comes from the *discrimination of specific problem-dependent information* of the Bandit problem ν within the class ζ. In the following, the problem-dependent information that we want to discriminate is the index $\star$ of an optimal arm. If ζ has few "statistically easy to discriminate" problems, then the constant will be smaller than if ζ has a lot of "statistically hard to discriminate" problems. The comparison of the regret to a logarithmic term is not too surprising given the fact that we study uniformly fast converging algorithm which mechanically enforce a logarithmic (at most) growth rate of the regret function.

It is linked to the problem of the "*number of questions* to ask in order to discriminate between distributions", hence linked to the classical theory of information as in Shannon information theory. Concepts such as typical sequences or typical set are therefore commonly used in the Bandit literature even if the community does not use this term. We recall that a typical set $\{x_i\}_{1\leq i\leq n}$ of a random variable X is such that

$$nH(X) - \epsilon \leq -\log \mathbb{P}\left(\{x_i\}_{1\leq i\leq n}\right) \leq nH(X) + \epsilon \ ,$$

where $H(X)$ is the entropy of random variable X. In the context of this thesis, one should think of H as the measure of the difficulty to discriminate X from the uniform distribution in the set of random variable on a finite alphabet. In this thesis, we will be interested in the event discriminating an optimal arm $\star$ from all suboptimal ones $a \in A$, $\{\hat{\mu}_\star > \hat{\mu}_a\}$. The log probability of this event will be controlled by $N_a \mathcal{I}_\zeta(a, \mu_\star; \nu)$, where $\mathcal{I}_\zeta(a, \mu_\star)$ is a measure of the difficulty to discriminate the arm a from an optimal one in the problem ν given the prior knowledge of the class ζ of distributions. For the knowledgeable reader, is reminiscent of certain events used in various regret proof of optimal algorithms. More on that topic later in the manuscript.

Remark Even considering a set of Bandit problems ζ on which the set of fast converging policies is not empty, I do think that it is not *a priori* trivial that the set of uniformly maximal convergent policies is not empty. If the set of considered problem ζ is too large, or has a strange structure (*e.g.* some kind of "holes" or several connected components), then I think that the set of optimal policies in the sense of *uniformly* maximal convergence rate may be empty. It may be interesting to identify such sets ζ and study the property of (a notion similar to) "maximal elements" in the set of uniformly fast convergent policies.

Bandit Learning Problem

Given a class ζ of Bandit problems, theoretician and practitioner interested in solving the Bandit learning problem generally want to at least craft a **uniformly fast convergent algorithm** as in Definition 3.1.20. Ideally, they want to find a **uniformly maximal convergent algorithm** as in Definition 3.1.21, *i.e.* a uniformly fast convergent policy with the minimal regret growth rate.[12] Sometimes, there are arguments that justify choosing a uniformly fast over a uniformly maximal policy, *e.g.* numerical complexity.

12: We mention that, as a sum of positive term, the regret is a non-decreasing function. This justifies the usage of the term **growth rate**.

Optimal policies In this thesis, we will define a policy to be **optimal** if it is **uniformly maximal convergent**. This is a biased choice that is a bit different from the definition that is used in the Bandit community, *e.g.* as in [6]. Usually, the definition of optimality that is used is what we will call a **criterion** or **characterization** of optimality, which can be deduced from the following Definition 3.1.22

[6]: Lattimore et al. (2020), *Bandit algorithms*

Definition 3.1.22 (Optimal Bandit policy) *Given a set ζ of Bandit problems, a policy $\pi^\star$ is said to be **optimal** if it is **uniformly fast convergent** and if it has a **uniformly maximal convergent rate** on ζ.*

That is to say, for all Bandit problem $\nu \in \zeta$, and all uniformly fast convergent policy π the regret $n \mapsto \mathcal{R}_\nu(n;\pi^\star)$ is asymptotically smaller than the regret $n \mapsto \mathcal{R}_\nu(n;\pi)$, i.e.

$$\forall \nu \in \zeta, \ \limsup_{n\to\infty} \frac{\mathcal{R}_\nu(n;\pi^\star)}{\mathcal{R}_\nu(n;\pi)} \leq 1 \ .$$

Since a policy with uniformly maximal convergence rate is by definition also uniformly fast, there is a bit of redundancy in the above definition which hopefully, help to better grasp the properties of what we call an **optimal policy**.

Optimality criterion Upon defining a notion of **optimal strategy**, one may wonder whether the class of optimal algorithm is non-empty, and one may wonder how to **assess optimality** of a given strategy. For the purpose of assessing the optimality of a Bandit strategy, we will state a powerful theorem that characterize optimal policies. This characterization is often used as a definition of optimality.

From Definition 3.1.17, the n-stage regret is

$$\mathcal{R}_\nu(n;\pi) = \mathbb{E}_{\pi,\nu}\left(\sum_{t=1}^{n} \mu^\star - \mu(a_t)\right) \ ,$$

which can be written as a sum over the action space A using counting random variables N_a and suboptimality gaps $\Delta_a = \mu^\star - \mu(a)$ (implicitly depending on $\mu^\star$).

Lemma 3.1.4 (Action decomposition of n-stage regret) *The n-stage regret $\mathcal{R}_\nu(n;\pi)$ of a policy π on a Bandit problem ν, as defined in Definition 3.1.17, can be decomposed as a sum along the action space A of problem $\nu = (A, \beta)$,*

$$\mathcal{R}_\nu(n;\pi) = \sum_{a\in A} \Delta_a \mathbb{E}_{\pi,\nu}(N_a(n)) \ . \tag{3.28}$$

From this decomposition 3.28, and the definition of a uniformly fast convergent policy, one can see that for all suboptimal action a, *i.e.* such that $\Delta_a > 0$,

$$\mathbb{E}_{\pi,\nu}(N_a(n)) \leq \frac{1}{\Delta_a}\mathcal{R}_\nu(n;\pi) = o(n^\alpha) \tag{3.29}$$

for all positive α. Thus, as a direct consequence of both the regret and fast convergence definitions, the expected number of pulls of a suboptimal arm is $o(n^\alpha)$, *i.e.* $\mathbb{E}_{\pi,\nu}(N_a(n)) = o(n^\alpha)$ for all $\alpha > 0$. Together, these

definitions imply that the number of pulls of suboptimal arms has at most a logarithmic growth rate.[13] However, the growth rate can be highly sub-logarithmic, and even equal to zero, as we already saw when considering Bandit problem with Dirac distributions and more generally, with disjoints distribution supports.

13: I think that it is important to note that this is more a consequence of the uniformly fast convergence hypothesis we make on the space of policies that we are interested in rather than a consequence of the forthcoming criterion "teaching us that a reasonable strategy can expect to pull the suboptimal arms a logarithmic number of times" as it can sometimes be read. . .

The action decomposition 3.28 and independence of cross-information between arms, *i.e.* the absence of structure in the Bandit problem, suggest that we may find a criterion that is based on the number of pulls of suboptimal arms. This criterion will be related to the asymptotic growth rate of the n-stage regret since we do not assume anything related to the finite time performance of the considered algorithms. When designing optimal algorithms, one would also be concerned about finite time performances. Sometimes, one can prove interesting finite time upper bound on the regret of our strategies, and the algorithmic design can be of great help to better understand the finite time behavior of the studied algorithm. The optimality criterion will answer the question about the minimal growth rate of mistakes an efficient learning agent must make in its history of interaction which is related to the minimal growth rate of regret it must suffer due to its **uncertainty** about the problem. The criterion takes the form of already introduced **problem dependent lower bound**.

Regret lower bound: the Bandit learning speed

A regret lower bound has the generic form

$$\liminf_{n\to\infty} \frac{\mathcal{R}_\nu(n;\pi)}{\log(n)} \geq \mathcal{C}(\nu;\zeta) \ ,$$

where $\mathcal{C}(\nu;\zeta)$ is a term that depend on the problem ν and the class ζ that the problem is assumed to belong to. Because it depends on ν, it is problem dependent. Because it depends on ζ, it is class dependent. The studied ratio, $\frac{\mathcal{R}_\nu(n;\pi)}{\log(n)}$, comparing the regret to a logarithm, comes from the uniformly fast convergence hypothesis. The direction of the inequality, lower bounding the regret, comes from the fact that we are interested in characterizing the uniformly maximal convergent polices, *i.e.* the minimal regret growth rate that any policy must suffer from the learning, *i.e.* from the uncertainty. We recall that we consider unstructured Bandit problems and that their associated control model at least satisfy the finite reward 3.1.1 and optimal feasibility 3.1.2 Assumptions. Because we assume an absence of structure, a class ζ of problem can be represented as a product of set of independent distributions, one set per arm. Given a set A of arm, we will write a class ζ as $\zeta := \bigotimes_{a\in A} \mathcal{F}_a$ where $\mathcal{F}_a$ is the set of distribution the reward $\beta(a)$ can belong to, independently of the distribution of all other arms. The sets $\mathcal{F}_a$ can be the same for all arms, *e.g.* the set of (m, M)-bounded distributions.

The unlikelihood of optimality viewpoint

There are two types of lower bound that are related to the design of two types of optimal algorithms. Both of the lower bounds and algorithms can be seen as dual in the sense that constraints and convex function to

minimize are exchanged. In a sense, this is an application of the classic optimality result about the primal and dual problems. However, there are a lot of technicalities in both the proofs that make the problem of deriving a lower bound far more than applying the primal-dual method and make the duality gap vanish.

The unlikelihood of optimality The first lower bound is based on a quantity that, in this thesis, we call the **unlikelihood of optimality**. We give it such a name for two reason. First it is a quantity that will be encountered time and time again in this manuscript and being able to name it is useful. Second, we think that this name conveys a useful meaning of the mathematical quantity and offer an intuition of how it should mathematically be used based on how it is used when we wish to explain algorithmic ideas.

Definition 3.1.23 (Unlikelihood of optimality) *Let A be a finite set of arms, $\zeta = \bigotimes_{a\in A} \mathcal{F}_a$ be a class of Bandit problems and $\nu = (A, \beta) \in \zeta$ be a Bandit problem within that class. We denote $\mathcal{E}_{\mathcal{F}_a}(\beta(a), \mu^\star)$ and call **unlikelihood** of optimality the quantity defined as*

$$\mathcal{E}_{\mathcal{F}_a}\left(\beta(a), \mu^\star\right) = \inf_F \left\{ KL\left(\beta(a), F\right) \mid F \in \mathcal{F}_a, \mathbb{E}_{X\sim F}(X) > \mu^\star \right\} \ , \quad (3.30)$$

where KL denotes the Kullback-Leibler divergence between two distributions. When the set of constraints is empty, the quantity is defined as $+\infty$.

Figure 3.2: Solomon Kullback (1907-1994)

The unlikelihood of optimality is written as a constrained optimization problem. Sometimes, we will call the unlikelihood of optimality of arm a, the **exponential rate of discrimination** of arm a from optimality. Indeed, the probability of arm a to be empirically optimal, $\mathbb{P}_\beta\left(\hat{\mu}_a > \hat{\mu}_\star\right)$ *i.e.* the probability of arm a to be the solution of the Bandit control problem will be linked and almost proportional to

$$\exp\left(-\mathcal{E}_{\mathcal{F}_a}\left(\beta(a), \mu^\star\right)\right)^{N_a(n)} \ .$$

This remark alone almost describe the MED [12] algorithm that we later introduce.

[12]: Honda et al. (2011), 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'

We read $KL\left(\beta(a)|F\right)$ as, the unlikelihood of *believing* samples comes from distribution F when the true distribution samples are coming from is $\beta(a)$. The unlikelihood of optimality is defined using two constraints that aims at computing how likely it is that reward distribution of arm a may be seen as optimal. First, the optimization search space is constrained to those distributions that the model ζ allows us to consider, *i.e.* the set $\mathcal{F}_a$ for arm a. Second, amongst all those possible distributions for arm a, we consider only those that would make the arm a optimal within the considered Bandit problem, *i.e.* whose expected value is larger than the optimal reward, $\mathbb{E}_{X\sim F}(X) > \mu^\star$. Intuitively, the larger the unlikelihood of optimality, the less likely it is that the arm a can be confused with an optimal distribution within ζ for the problem μ. Samples from arm a will quickly be discriminated as coming from a distribution that is highly unlikely to be optimal. On the contrary, the smaller the unlikelihood of optimality, the *KL-closer* is the reward distribution of arm a to an optimal distribution. Collections of samples coming from arm a have a higher

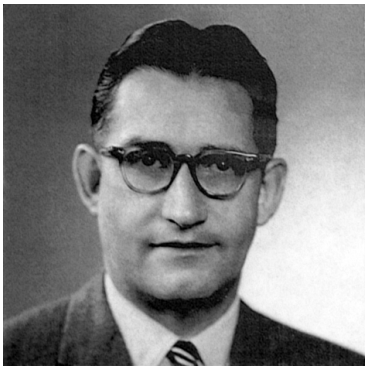

Figure 3.3: Richard Leibler (1914-2003)

probability of being confused with a sample collection coming from a distribution in $\mathcal{F}_a$ and that is optimal for the problem ν being solved by an agent. When the set of constraints is empty, the unlikelihood of optimality is infinite because the reward distribution of arm a cannot be modified into an optimal reward distribution using assumption $\mathcal{F}_a$ of class ζ. Even when we do not mention it, it goes without saying that optimality or unlikelihood of optimality is not a property of an arm. It is a property of an arm with respect to the class ζ of problem and with respect to the specific problem that is considered. An arm is not the best *per se*, it is maximal compared to other arms.

We introduce a useful notation to represent the set of optimal arms in a Bandit problem.

Notation 3.1.6 (Optimal set of arms) *Given a Bandit problem $\nu = (A, \beta)$, we denote by $A^\star(\beta)$ or $A^\star(\nu)$ or simply $A^\star$ when there is no possible confusion, the set of arm with maximal expected reward, i.e.*

$$A^\star(\beta) = \operatorname*{argmax}_{a \in A} \mu(a|\beta) \ . \tag{3.31}$$

Interestingly, this set is equal to the set of arms with a zero valued unlikelihood of optimality,

$$A^\star(\beta) = \left\{ a \in A \mid \mathcal{E}_{\mathcal{F}_a}\left(\beta(a), \mu^\star\right) = 0 \right\} \ . \tag{3.32}$$

Even more interestingly, one can write this same set as the set of minimizers of the unlikelihood of optimality,

$$A^\star(\beta) = \operatorname*{argmin}_{a \in A} \mathcal{E}_{\mathcal{F}_a}\left(\beta(a), \mu^\star\right) \ . \tag{3.33}$$

Equation 3.33 is interesting because it relates the unlikelihood of optimality to the set of optimal arms, Equation 3.31. Already can we see a form of interesting duality between these two quantities, maximal expected reward and minimal unlikelihood of estimation. Intuitively, the function $\mu \to \mathcal{E}_{\mathcal{F}_a}(\beta(a), \mu)$ is non-decreasing, making the connection even more fruitful as we will later see when introducing optimal algorithms.

Regret lower bound: the unlikelihood of optimality viewpoint With all those intuitions in mind, one can state the first regret lower bound. This form is handy because of Equation 3.35 that allows to easily individually separate the sample rate of each arm and to isolate their contribution to the regret logarithmic growth rate, Equation 3.34.

Theorem 3.1.5 (**Regret lower bound**: unlikelihood of optimality) *Let A be a finite set of arms, $\zeta = \bigotimes_{a \in A} \mathcal{F}_a$ be a class of Bandit problems and $\nu = (A, \beta) \in \zeta$ be a Bandit problem within that class. Let's denote by $A_{\log}$ (or $A_{\log}(\beta)$ when necessary) the set of arm with a finite unlikelihood of optimality, i.e.*

$$A_{\log} = \left\{ a \in A \mid \mathcal{E}_{\mathcal{F}_a}\left(\beta(a), \mu^\star\right) < +\infty \right\} \setminus A^\star$$

where $A_{\log}$ implicitly depend on ν and ζ.

Then, for all uniformly fast convergent policy π, and therefore uniformly maximal converging rate policy, the growth rate of the regret $\mathcal{R}_\nu(n;\pi)$ is lower bounded,

$$\liminf_{n\to\infty} \frac{\mathcal{R}_\nu(n;\pi)}{\log n} \geq \sum_{a\in A_{\log}} \frac{\mu^\star - \mu(a|\beta)}{\mathcal{E}_{\mathcal{F}_a}(\beta(a),\mu^\star)} \,. \tag{3.34}$$

Furthermore, the growth rate of the expected number of samples of any suboptimal arm $a \in A_{\log}$ is lower bounded,

$$\liminf_{n\to\infty} \frac{\mathbb{E}_{\pi,\nu} N_a(n)}{\log n} \geq \frac{1}{\mathcal{E}_{\mathcal{F}_a}(\beta(a),\mu^\star)} \,. \tag{3.35}$$

For arms $a \in A \setminus (A_{\log} \cup A^\star)$, i.e. suboptimal with an infinite unlikelihood of optimality, the expected number of pulls grows sub-logarithmically, $\mathbb{E}_{\pi,\nu} N_a(n) = o(\log n)$. Such arms do not contribute to the logarithmic growth rate of the regret.

Theorem 3.1.5 is very instructive. It can be found in the original work of [13] and has been rederived using different methods multiple times in the literature, in an effort to better understand this fundamental result, *e.g.* in [14]. Quite intuitively, it maximally uses the **uniformly fast convergence** hypothesis in the sense that it compares the regret function $n \mapsto \mathcal{R}_\nu(n;\pi)$ to the "*smallest*" of all function of the form n^α with positive α, *i.e.* the largest of $o(n^\alpha)$ function, the logarithm $n \mapsto \log n$. With this fast convergence hypothesis alone, and without adding assumption about the Bandit set ζ, this theorem allows us to compute the asymptotic sampling rate, Equation 3.35, only of those arms that are pulled exactly at a logarithmic rate. Maximally using the available hypothesis, the theorem cannot inform us about sub-logarithmic convergence rate. However, the theorem does provide an interesting insight about those arms that could have sub-logarithmic sampling rate, they are the arms with an infinite unlikelihood of optimality. *A priori*, those arms could be discriminated as suboptimal at a much faster rate than the arms that are not infinitely unlikely.

[13]: Burnetas et al. (1996), 'Optimal adaptive policies for sequential allocation problems'

[14]: Garivier et al. (2016), 'Explore first, exploit next: The true shape of regret in bandit problems'

The regret lower bound is thus written as a sum over a subset of arms,

$$A_{\log} = \left\{ a \in A \mid 0 < \mathcal{E}_{\mathcal{F}_a}\left(\beta(a),\mu^\star\right) < +\infty \right\} \setminus A^\star \,,$$

made of those arms that not optimal, *i.e.* arms that do not contribute to the regret, and those arms that have a finite unlikelihood of optimality, *i.e.* those arms that must be sampled at a logarithmic rate according to Theorem 3.1.5.

The Equation 3.34 is about the logarithmic rate of the regret any uniformly fast convergent and therefore uniformly maximally convergent policy must at least suffer from. This regret comes from the uncertainty about the very problem the agent is solving and arms that must be pulled at a logarithmic rate are precisely those arms with a finite non-zero unlikelihood of optimality. Interestingly, in the case of unstructured Bandit, one can isolate the sampling rate of arms in $A_{\log}$ as shown in Equation 3.35. This equation shows that logarithmic sampling rate of any arm in $A_{\log}$ is lower bounded by the invert ratio of its unlikelihood of optimality. The larger the unlikelihood of optimality, the more likely it is that the arm is confused for an optimal one, the larger its logarithmic sampling

rate. Retrospectively, one can read the Equation 3.34 by separating the individual regret contribution of each arm in $A_{\log}$ as the ratio of the suboptimality gap and unlikelihood ratio,

$$\frac{\mu^\star - \mu(a|\beta)}{\mathcal{E}_{\mathcal{F}_a}(\beta(a), \mu^\star)} .$$

Remark: Arms close to be optimal may greatly contribute to the regret The closer the expected value of a suboptimal arm is to be optimal, the smaller the suboptimality gap. One may be tempted to say that its contribution to the regret is small or smaller than those arms with larger suboptimality gap. However, it may not be the case. Indeed, usually, the smaller the suboptimality gap, the more likely the arm is to be confused for an optimal one and the smaller its unlikelihood of optimality. Therefore, when an arm is "close" to be optimal, its per-sample contribution to the regret is small because the suboptimality gap is small, but it often contributes to the regret due its higher sampling rate because its unlikelihood of optimality is smaller. Therefore, there are cases (and more often than not) when the contribution to the regret of arms that are close to be optimal is larger than the contribution of arms that are highly suboptimal. For instance, considering a Bandit problem with Gaussian distributions with known identical variance for all arms, the suboptimality gap of a suboptimal arm a is $\mu^\star - \mu(a)$ and the value of $\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)$ is proportional to $(\mu^\star - \mu(a))^2$. Thus, the contribution of arm a to the logarithmic regret growth rate is proportional to

$$\frac{\mu^\star - \mu(a)}{(\mu^\star - \mu(a))^2} = \frac{1}{\mu^\star - \mu(a)} ,$$

an increasing function of $\mu(a) < \mu^\star$.

n-stage lower bounds & leading orders This Theorem 3.1.5 allows us to write several inequalities using small-o and Ω notations to express, perhaps in a more readable way, what can be said about the n-stage regret function, and the number of pulls of the different type of arms. We distinguish three types of arms, optimal ones in $A^\star$, non-optimal finitely unlikely arms in $A_{\log}$, and the remaining arms in $A \setminus \left(A^\star \cup A_{\log}\right)$ that are suboptimal and have an infinite unlikelihood of optimality.

Equation 3.34 can be used to derive that the n-stage regret is such that

$$\mathcal{R}_\nu(n; \pi) \geq \sum_{a \in A_{\log}} \frac{\mu^\star - \mu(a|\beta)}{\mathcal{E}_{\mathcal{F}_a}(\beta(a), \mu^\star)} \log n + o(\log n) .$$

This notation also is an asymptotic one but may be easier to read than the original equation. The expected number of pulls of an arm $a \in A_{\log}$ is such that,

$$\mathbb{E}_{\pi,\nu} N_a(n) \geq \frac{\log n}{\mathcal{E}_{\mathcal{F}_a}(\beta(a), \mu^\star)} + o(\log n) \tag{3.36}$$

and the expected number of pulls of an arm $a \in A \setminus \left(A_{\log} \cup A^\star\right)$ is lower bounded by a sub-logarithmic term,

$$\mathbb{E}_{\pi,\nu} N_a(n) \geq o(\log n) .$$

An optimal algorithm $\pi^\star$ will be such that those inequalities are equalities. In practice, we will prove an upper bound on the algorithmic regret, upper bound that should match the right-hand side of those inequalities. In particular, for unstructured Bandit problem, we will be interested in proving an upper bound on the individual number of pulls, reversing inequalities 3.36. Assuming reversing the inequalities, the fact that $\sum_{a\in A} N_a(n) = n$, thus $\sum_{a\in A} \mathbb{E}_{\pi,\nu} N_a(n) = n$, implies that a strategy matching the lower bound will pull optimal arms a linear amount of the interactions,

$$\sum_{a\in A^\star} \mathbb{E}_{\pi,\nu} N_a(n) = \Omega(n) \,,$$

thus efficiently learning a solving policy.

Usual case in the literature In practice, Theorem 3.1.5 will be used for unstructured Bandit where all the distribution spaces $\{\mathcal{F}_a\}_a$ will be identical to a space $\mathcal{F}$, *i.e.* $\mathcal{F}_a = \mathcal{F}$ for all $a \in A$. Therefore, usually the space $A_{\log}$ is one of the following two: $A_{\log} = A \setminus A^\star$, *e.g.* if $\mathcal{F}$ is the space of Bernoulli distributions with expected values in $[0,1)$, or $A_{\log} = A \setminus A^\star = \emptyset$, *e.g.* if $\mathcal{F}$ is the space of bounded Dirac distributions in (m, M). However, if the space $\mathcal{F}$ is more complex, then there might be some cases where $A_{\log}$ is none of the above. For instance, if $\mathcal{F}$ is the space of distributions with all their mass in $(0, 0.3)$ or in $(0.6, 1)$, then if $\mu^\star \in (0.6, 1)$, for any distribution with all its mass in the smaller interval $(0, 0.3)$, its unlikelihood of optimality is infinite and is not in $A_{\log}$. This is because one cannot transform a distribution having all its mass in an interval into another of disjoint support without paying an infinite Kullback-Leibler cost. If $\mu^\star \in (0, 0.3)$, then this same considered arm would have a finite unlikelihood of optimality and would belong to $A_{\log}$. Therefore, even for unstructured Bandit, there are problems where a distribution has a finite unlikelihood of optimality and other where it has an infinite one, for the same class of Bandit problem.

Algorithmic applications Once a lower bound on a Bandit class ζ has been computed, the next step usually is to find an algorithm that is optimal on that class, *i.e.* an algorithm with uniformly maximal convergent rate, with said rate computed in the lower bound of Theorem 3.1.5. The lower bound presented in this theorem, and in particular the lower bound on the sampling rate of suboptimal arms Equation 3.35 is directly connected to a class of algorithms that we call MED-like algorithms. The MED acronym will stand for Minimal Empirical Divergence. In the next section, we will see how this lower bound can be used to directly derive three algorithms, MED [12], DMED [15], and IMED [16].

We now present another type of regret lower bound, analytically equal to one the presented in Theorem 3.1.5 (a comforting fact), that is connected to another family of algorithm that we call UCB-like algorithms. The UCB acronym will stand for Upper Confidence Bound. This view point, we call it the **allocation-constrained optimism**, and we see it as dual of the **unlikelihood of optimality** viewpoint.

[12]: Honda et al. (2011), 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'

[15]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models'

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

The allocation-constrained optimism viewpoint

When are two problems indistinguishable? Are two Bandit problems $\beta \in \zeta$ and $\beta' \in \zeta$ indistinguishable when $KL(\beta(a), \beta'(a)) = 0$ for all arms? This is a strong sense of indistinguishability, that we may want to call equality, because all reward distributions must be the same. Or are they indistinguishable when $A^\star(\beta) = A^\star(\beta')$? This second viewpoint is weaker than the first proposed and relies only on the comparison of the set of optimal solutions. In the space ζ, one could say that two problems $\beta \in \zeta$ and $\beta' \in \zeta$ are *indistinguishable* if they have the same set of optimal arms, *i.e.* the same set of solutions to the Bandit control problem. From the *solution-viewpoint*, two such problems are not fundamentally different. However, two problems are fundamentally different if they do not share the same set of solutions and can be distinguished from there set of solutions. In this pint of view, two problems can only be distinguished at the level of control policies and not at the level of the problem. **Finding an optimal policy is all we are interested in.** As we stated in the introduction, we are interested in finding solutions to uncertain problem and *a priori* not interested in knowing the uncertain problem. Gaining knowledge about the uncertain problem will be a mean to achieve the true goal of the Bandit learner, which is playing a solution of the control problem as often as possible in the long run. By long run, we mean *infinite*, hence we are more interested in asymptotic rates rather than exact number of interactions. It therefore makes sense to build an object that is more concerned about the set of solutions than the specific values of the problem. Of course, solutions of the Bandit control problem still depend on the values of the problem as well as the class ζ problems belong to.

However, this second notion of indistinguishability is too weak. From an available information viewpoint, an optimal policy will play in the set of optimal actions, observe rewards and can extract information from this set whereas it cannot remain optimal and extract information from suboptimal arms. Therefore, a better notion of indistinguishability should take the possibility of collecting and processing information from the set of optimal arms $A^\star(\beta)$ of a Bandit problem $\beta \in \zeta$. Thus, our final notion of indistinguishability. If two bandit problems β and β' (with same set of arms) cannot be distinguished by any optimal policy, then they are said to be indistinguishable.

Definition 3.1.24 (Instinguishable set) *Let ζ be a set of Bandit problems and $\beta \in \zeta$ be a Bandit problem. We denote $\mathscr{S}(\beta; \zeta)$ the set of Bandit problem in ζ that are* ***indistinguishable*** *from Bandit problem β,*

$$\mathscr{S}(\beta;\zeta) = \left\{ \beta' \in \zeta \mid \begin{array}{l} A^\star(\beta) = A^\star(\beta') \\ \forall a \in A^\star(\beta)\,,\ KL(\beta(a), \beta'(a)) = 0 \end{array} \right\}. \tag{3.37}$$

Therefore, two problems are indistinguishable if they share the same set of optimal arms and distributions of optimal arms are unchanged. This definition is coherent with both our requirements to compare problem from an optimal control policy viewpoint and the possibility for such policies to collect and process information that allows to distinguish distributional changes of an optimal arm.

What can optimal policies distinguish? In light of the previous paragraph, the answer to that question is quite clear. The set of optimal policies on a Bandit problem $\beta \in zeta$ can only distinguish those problems β' that are such that there exist an arm $a \in A^\star(\beta)$ with $KL\left(\beta(a), \beta'(a)\right) > 0$ because an optimal control on β is only allowed to play those arms that are in $A^\star(\beta)$. Hence, the set of distinguishable problems $\mathscr{P}\left(\beta\right)$ is

$$\mathscr{P}\left(\beta\right) = \left\{\beta' \in \zeta \mid \exists a \in A^\star(\beta),\ KL\left(\beta(a), \beta'(a)\right) > 0\right\} \ .$$

On the contrary, only allowing ourself to play within the set of optimal arms in β, $A^\star(\beta)$, one cannot distinguish from β the complement in ζ of $\mathscr{P}\left(\beta\right)$,

$$\overline{\mathscr{P}\left(\beta\right)} = \left\{\beta' \in \zeta \mid \forall a \in A^\star(\beta),\ KL\left(\beta(a), \beta'(a)\right) = 0\right\} \ .$$

Problems that would make a set of policies suboptimal Amongst those Bandit problem in $\overline{\mathscr{P}\left(\beta\right)}$ that the set $A^\star(\beta)$ of optimal control cannot distinguish from the original problem β, there are some that would make all policies in $A^\star(\beta)$ suboptimal. Those are the problems that would make a control suffer linear regret. Those are the problems that a learner cannot distinguish from playing only actions in its currently computed set of optimal controls but must discriminate if it does not want to suffer a linear regret. If the agent cannot distinguish its current problem β from another one β' where at least one optimal action in β' is on the set of optimal arms for β, then an algorithm playing in the set $A^\star(\beta)$ on the problem β' can gain enough information to discriminate that existing common optimal action. On the other hand, if $A^\star(\beta) \cap A^\star(\beta') = \emptyset$, then an optimal control on β, *i.e.* a greedy learner, cannot distinguish its problem from the set of alternative where its optimal set is made of suboptimal controls only. We call this set the **control-indistinguishable optimal alternative** and denote it $\zeta\left(\beta\right)$.

Definition 3.1.25 (**control-indistinguishable optimal alternative**) *Let ζ be a set of Bandit problems and $\beta \in \zeta$ be a Bandit problem. We denote $\zeta\left(\beta\right)$ the set of Bandit problem in ζ that are **indistinguishable** from Bandit problem β and whose set of optimal control has empty intersection with the set $A^\star(\beta)$ of optimal arms in the original problem β,*

$$\zeta\left(\beta\right) = \left\{\beta' \in \zeta \mid \begin{array}{l} A^\star\left(\beta\right) \cap A^\star\left(\beta'\right) = \emptyset, \\ \forall a \in A^\star\left(\beta\right),\ KL\left(\beta(a), \beta'(a)\right) = 0 \end{array}\right\} \ . \tag{3.38}$$

From a learning perspective, Bandit problems in this set $\zeta(\beta)$ cannot be distinguished from β by only playing action in $A^\star(\beta)$, *i.e.* by being greedy on the computed information, while problems in this set can make the learner suffer a linear regret. From a learner perspective who is uncertain about the problem, there is a necessity to gain information on that set of problems, which means, sampling actions that are optimal for at least one problem in $\zeta(\beta)$. The regret lower bound answer the question of the frequency this necessity to sample arms that are *a priori* suboptimals. This necessity to deviate from greedy decision is the **cost of uncertainty** and is often worded as the **exploitation-exploration trade off**.

Cost of discriminating A learner that only greedily exploits its current knowledge of optimality, there is a probability that a lack of information about suboptimal arms wrongs it into computing a false set of optimal arms. Playing arms in $A^\star(\beta)$ can only do that much, that is bringing information about actions in $A^\star(\beta)$. To gain discriminative power, one must sample outside this set of optimal controls. One must deviate only to avoid those arms that could make the learner suffer linear regret. The question that is answered by the lower bound is, how often should we sample arms that are suboptimal? From an asymptotic viewpoint, due to the uniformly fast convergence hypothesis, arms that are in $A^\star(\beta)$ are assumed to be perfectly known while the uncertainty is on those arms that are pulled in $o(n^\alpha)$ for all $\alpha > 0$. As before, it means that we should be considering the logarithmic rate of sampling of suboptimal arms. The discriminative power of a policy π such with the allocation scheme $m_a(n)$ after n time steps, *i.e.* $m_a(n)$ samples were collected from arm a, is measured by

$$KL(m(n) \otimes \beta, m(n) \otimes \beta') = \sum_{a \in A} m_a(n) KL(\beta, \beta') \ ,$$

where $m(n) \otimes \beta$ is the distribution of sampling arm a with normalized probability $m_a(n)/n$. Since we are only interested in discriminating those arms that not optimal in β because an optimal control policy will not sample them, we prefer to measure the discriminative power of an allocation scheme only on those arms that are not in $A^\star$. Pushing it one step further, we are better interested in the logarithmic discriminative rate where we replace $m_a(n)$ by $m_a(n)/\log n$ and call η_a the superior limit as n tends to infinity. Therefore, it only makes sense to look for those policies with logarithmic sampling rate of suboptimal policies that satisfy a discriminating constraint of the form

$$\sum_{a \in A \setminus A^\star(\beta)} \eta_a KL(\beta(a), \beta'(a)) \geq \gamma \ . \tag{3.39}$$

Of course, such a constraint should be satisfied on all the problem $\beta \in \zeta(\beta)$, the control-indistinguishable optimal alternative set. Thus, the final form of the constraint,

$$\inf_{\beta' \in \zeta(\beta)} \sum_{a \in A \setminus A^\star(\beta)} \eta_a KL(\beta(a), \beta'(a)) \geq \gamma \ . \tag{3.40}$$

It is interesting to note that when sampling a suboptimal arm a, we gain information about many problems in $zeta(\beta)$ because from gaining information about a, we gain information about all those problems in which a is optimal. Note that if there is no distribution in $\mathcal{F}_a$ making a optimal without modifying the set of optimal distribution, then there is no $\beta' \in \zeta(\beta)$ where a is an optimal arm and therefore, its logarithmic sampling rate η_a surely will be equal to zero in the allocation objective because such an arm does not have any logarithmic discriminative power and should be played sub-logarithmically.

Cost of uncertainty To have enough discriminating power, we now understand that a uniformly fast convergent policy must satisfy logarithmic

rate allocation constraint of the form,

$$\inf_{\beta' \in \zeta(\beta)} \sum_{a \in A \setminus A^\star(\beta)} \eta_a KL\left(\beta(a), \beta'(a)\right) \geq \gamma \ ,$$

with $\eta_a \geq 0$ since it is an allocation rate and $\zeta(\beta)$ the control-indistinguishable optimal alternative set. The logarithmic growth rate of regret due to a logarithmic allocation rate $(\eta_a)_a$ is

$$\sum_{a \in A \setminus A^\star} \eta_a \left(\mu^\star - \mu(a|\beta)\right) \ .$$

A lower bound on the regret growth rate therefore is written as the infimum over all the allocation rates that satisfy the discrimination constraint,

$$\inf_{(\eta_a)_a} \sum_{a \in A \setminus A^\star} \eta_a \left(\mu^\star - \mu(a|\beta)\right) \ .$$

Link with the unlikelihood of optimality Before finally giving the complete lower bound, we emphasize a link between the allocation constraint and the unlikelihood of optimality. Because of the unstructured assumption on the Bandit problem, arms can be "modified" independently of each other, and we can write the allocation-exploration constraint as $|A \setminus A^\star|$ constraints of the form,

$$\inf_{\beta'(a) \in \mathcal{F}_a(\mu^\star)} \eta_a KL\left(\beta(a), \beta'(a)\right) \geq \gamma \ ,$$

where

$$\mathcal{F}_a\left(\mu^\star\right) = \left\{F \in \mathcal{F}_a \mid \mathbb{E}_{X \sim F}\left(X\right) > \mu^\star\right\} \ .$$

Therefore, we retrieve the unlikelihood of optimality from this analysis.

Constraints on the unlikelihood of optimality The unlikelihood of optimality of an arm a is defined as the infimum of the Kullback-Leibler divergence, $F \longrightarrow KL\left(F_a, F\right)$, where the distribution F is constrained to belong to set, which we denote $\mathcal{F}_a\left(\mu^\star\right)$ or $\mathcal{F}_a\left(\beta\right)$, with

$$\mathcal{F}_a\left(\mu^\star\right) = \left\{F \in \mathcal{F}_a \mid \mathbb{E}_{X \sim F}\left(X\right) > \mu^\star\right\} \ ,$$

where $\mu^\star = \max_{a \in A} \mu\left(a|\beta\right)$ is the maximal reward on the Bandit problem β. For the computation of the unlikelihood of optimality, this set is of uttermost important because the Kullback-Leibler divergence is quantity that is only concerned about the *distributions* of random variables, and not about specific domain values.[14] The way unlikelihood of optimality integrates the expected reward is through the constraints, which link a *distribution* to an associated *random variable*. In $\mathcal{F}_a$, distributions are reward distributions of associated reward random variables. However, the sets $\mathcal{F}_a\left(\beta\right)$ can be of independent interest because together, those sets allow to **construct the set of all Bandit problem with a different set of solutions**.

14: For discrete random variables,

$$KL(p, q) = \sum_i p_i \log \frac{p_i}{q_i},$$

and there is no reference to the specif value attached to an index i. Any good notion of unlikelihood of optimality **must** depend on the domain to take the expected reward and optimal expected reward $\mu^\star$ into account.

Regret lower bound: the allocation-constrained optimality viewpoint

Using all the knowledge from the previous analysis, we are ready to state the allocation-constrained based lower bound on the regret. The only unknown that was left and is derived mathematically is the threshold γ. However, the mentioned link with the unlikelihood of optimality points out that $\gamma = 1$ is a good candidate. We hope that the previous analysis helps to read *in plain English* the following lower bound, which, at first, appears like an intricate optimization problem.

Theorem 3.1.6 (**Regret lower bound**: allocation-constrained optimality viewpoint) *Let A be a finite set of arms, $\zeta = \bigotimes_{a\in A} \mathcal{F}_a$ be a class of Bandit problems and $\nu = (A, \beta) \in \zeta$ be a Bandit problem within that class.*

Then, for all uniformly fast convergent policy π, and therefore uniformly maximal converging rate policy, the growth rate of the regret $\mathcal{R}_\nu(n;\pi)$ is lower bounded,

$$\begin{aligned} \liminf_{n\to\infty} \frac{\mathcal{R}_\nu(n;\pi)}{\log n} \geq \inf_{(\eta_a)_{a\in A}} & \sum_{a\in A} \eta_a \left(\mu^\star - \mu(a|\beta)\right) \\ & s.t.\ \eta_a \geq 0 \ \forall a \in A\ , \\ & \inf_{\beta'\in\zeta(\beta)} \sum_{a\in A} \eta_a KL(\beta(a), \beta'(a)) \geq 1 \end{aligned} \quad , \tag{3.41}$$

where the set $\zeta(\beta)$ is defined using $A^\star(\beta) = \operatorname{argmax}_{a\in A} \mu(a|\beta)$, the set of maximal arms in a Bandit problem (A, β), as

$$\zeta(\beta) = \left\{ \beta' \in \bigotimes_{a\in A} \mathcal{F}_a \mid \begin{array}{l} \mathcal{A}^\star(\beta) \cap \mathcal{A}^\star(\beta') = \emptyset, \\ \forall a \in \mathcal{A}^\star(\beta)\ ,\ KL(\beta(a), \beta'(a)) = 0 \end{array} \right\} . \tag{3.42}$$

$\zeta(\beta)$ is the set of Bandit problem $\beta' \in \zeta$ with different optimal set and such that the distribution of optimal arm in β is unchanged in β', i.e. only the distribution of suboptimal arms is allowed to be changed and at least one must be transformed into an optimal arm within β'.

[17]: Graves et al. (1997), 'Asymptotically efficient adaptive choice of control laws incontrolled markov chains'

This form is first found in the seminal work of [17]. The analysis preceding the theorem statement gave a lot of intuitions. Still, one can *a posteriori* see that Theorem 3.1.6 is very instructive, and we can make a small analysis that is similar to the one we did for Theorem 3.1.5. Quite intuitively, it maximally uses the **uniformly fast convergence** hypothesis in the sense that it compares the regret function $n \mapsto \mathcal{R}_\nu(n;\pi)$ to the "*smallest*" of all function of the form n^α with positive α, *i.e.* the largest of $o(n^\alpha)$ function, the logarithm $n \mapsto \log n$. With this fast convergence hypothesis alone, and without adding assumption about the Bandit set ζ, this theorem allows us to compute the asymptotic sampling rate, constraints on $(\eta_a)_a$ Equation 3.41, only of those arms that are pulled exactly at a logarithmic rate. Maximally using the available hypothesis, the theorem cannot inform us about sub-logarithmic convergence rate. Indeed, those arms are not in the set $\zeta(\beta)$ and therefore, their logarithmic sampling rate saturate the constraint by being equal to zero.

As in Theorem 3.1.5, the regret lower bound is written as a sum over a

subset of arms,

$$A_{\log} = \left\{ a \in A \mid 0 < \mathcal{E}_{\mathcal{F}_a}\left(\beta(a), \mu^\star\right) < +\infty \right\} \setminus A^\star \ ,$$

made of those arms that are not optimal, optimal arms that do not contribute to the regret, and those arms that have a finite unlikelihood of optimality, *i.e.* those arms that must be sampled at a logarithmic rate. Indeed, under the unstructured assumption, those arms can be made optimal under the ζ hypothesis and corresponds to those arms with non-trivial (equal to zero) logarithmic sampling rate constraint.

While the Theorem 3.1.6 is more general than the Theorem 3.1.5 in the sense that it can be used as such for structured Bandit problems (with a general ζ instead of decomposable $\zeta = \bigotimes_a \mathcal{F}_a$), it does not make it immediate to the eye that one can isolate the sampling rate of arms in $A_{\log}$ as shown in Equation 3.35.

The space $\zeta(\beta)$ can be written as a union of Bandit problems where we constrain that at least one suboptimal arm is, if possible, made optimal,

$$\zeta(\beta) = \bigcup_{a \notin A^\star(\beta)} \left\{ \beta' \in \bigotimes_{a' \in A} \mathcal{F}_{a'} \mid \mathbb{E}_{F'_a}(\beta'(a)) > \max_{a \in A} \mu(a|\beta) \right\} , \quad (3.43)$$

where some sets might be empty due to the fact that one cannot find in $\mathcal{F}_a$ a reward distribution $\beta'(a) \sim F'_a$ such that arm a has an expected reward larger than $\mu^\star = \max_{a \in A} \mu(a|\beta)$. That is to say, there are some suboptimal arms that cannot be made optimal without changing the distribution of optimal arms in the original problem. The suboptimal arms that can be transformed into optimal ones without changing the distribution of optimal ones are the arms that are within the $A_{\log}$ set defined in the unlikelihood of optimality viewpoint. In fact, fully exploiting the unstructured assumption, one could replace the set $\zeta(\beta)$ by the set

$$\bigcup_{a \notin A^\star(\beta)} \left\{ \beta' \in \bigotimes_{a' \in A} \mathcal{F}_{a'} \mid \begin{array}{l} E_{F'_a}(\beta'(a)) > \mu^\star, \\ \forall a' \neq a,\ KL(\beta(a')|\beta'(a')) = 0 \end{array} \right\} ,$$

which amounts to consider all the arm-constraints independently and computing the logarithmic sampling rate using the sets $\mathcal{F}_a(\mu^\star)$.

Usual case in the literature Similarly to Theorem 3.1.5, Theorem 3.1.6 will often be used for unstructured Bandit where all the distribution spaces $\{\mathcal{F}_a\}_a$ will be identical to a space $\mathcal{F}$, *i.e.* $\mathcal{F}_a = \mathcal{F}$ for all $a \in A$ and therefore $\bigotimes_{a \in A} \mathcal{F}_a = \mathcal{F}^{\otimes |A|}$. However, because it does not distinguish between arms *a priori* in the set $\zeta(\beta)$ and constrain all the logarithmic sampling rates in a single combined inequality, it is more suited to the analysis of structured Bandits. This constrained optimization formulation provide an easier to take structural assumption into account in the regret lower bound.

Algorithmic applications Similarly to what we said after Theorem 3.1.5, once a lower bound on a Bandit class ζ has been computed, the next step usually is to find an algorithm that is optimal on that class, *i.e.* an

algorithm with uniformly maximal convergent rate, with said rate computed in the lower bound of Theorem 3.1.5. The allocation-constrained optimality lower bound presented in Theorem 3.1.6 is directly connected to a class of algorithms that we call UCB-like algorithms. The UCB acronym will stand for Upper Confidence Bound. In the next section, we will see how this lower bound can be used to directly derive several algorithms, KL-UCB [18–22] and UCB [5]. We mention that this lower bound inspired other UCB-like algorithm such as the Bayesian-UCB [23]. We also connect this allocation-constrained optimality to the class of TS-like algorithms where TS stands for Thompson Sampling [24, 25]. In particular, it is related to the recent NPTS [26] algorithm.

From a policy and algorithmic viewpoint, the allocation-constrained Equation 3.41 of Theorem 3.1.6 offers an interesting perspective because a set of **logarithmic rates of allocation on suboptimal arms** can readily be seen as a **sampling policy** or **algorithm**. Therefore, the infimum over the rates $(\eta_a)_a$, $\inf_{(\eta_a)_a}$ in Equation 3.41 gain from being seen as implicitly an infimum over policies, where $\eta_a(\pi)$ is the logarithmic sampling rate of a policy π. This indicate that η_a is a good candidate for a logarithmic sampling rate of a suboptimal arm a.

[18]: Lai (1987), 'Adaptive treatment allocation and the multi-armed bandit problem'

[19]: Agrawal et al. (1989), 'Asymptotically efficient adaptive allocation schemes for controlled iid processes: Finite parameter space'

[20]: Agrawal (1995), 'Sample mean based index policies by O (log n) regret for the multi-armed bandit problem'

[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'

[22]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation.'

[5]: Auer et al. (2002), 'Finite-time analysis of the multiarmed bandit problem'

[23]: Kaufmann et al. (2012), 'On Bayesian Upper Confidence Bounds for Bandit Problems'

[24]: Thompson (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'

[25]: Thompson (1935), 'On a criterion for the rejection of observations and the distribution of the ratio of deviation to sample standard deviation'

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

Wrap it up: towards optimal algorithms

Dual lower bounds & dual algorithms The two regret lower bounds presented in this section can be seen as dual and are the roots of two families of algorithms, MED-like algorithms and UCB-like algorithms. For MED-like algorithms, inspired by the unlikelihood of optimality lower bound, algorithms are using an empirical unlikelihood of optimality to assess the frequency at which suboptimal arms must be sampled. The more unlikely the optimality, the less the strategy should sample. The less unlikely the optimality, the more the strategy should sample. In short, we will sample an arm if its empirical number of samples does not match with its unlikelihood of optimality. For UCB-like algorithms, inspired by the allocation-constrained optimality, algorithms are using the empirical allocation rate to compute a distribution $\beta'(a)$ (more precisely, its expected value $\mu(a|\beta')$) in $\zeta(\beta)$ that satisfy the allocation constraints. This distribution must satisfy the allocation constraint but, at the same time, it must respect the information that have been collected on arm a. Therefore, one would be looking for an arm in $\mathcal{F}_a(\mu^\star)$ that will saturate the constraint. Amongst those possible distribution, a group is more appealing, the group of distributions with the largest possible expected values. Indeed, those are the distributions, that, while respecting the allocation constraints, would make the algorithm suffer the largest possible regret if this expected value were true. This point of view is sometimes phrased as **optimism in face of uncertainty**. This wording is quite understandable, but I do not think that it reflects the **policy viewpoint** that seems to be emerging from this lower bound. Rather, I prefer the term **upper confidence bound** because it does not carry the meaning of *optimism*. There is no optimism from the policy viewpoint, rather, the policy computes the largest possible regret it could incur from each of the empirical suboptimal arms and decide to sample the one from which this likely regret is the largest (if positive). While it is indeed an optimism of expected reward from an arm distribution viewpoint,

it is not optimism but rather an upper confidence bound on the *largest likely regret* that the policy can suffer based on current information.

The frequency at which an optimal algorithm will sample arms, we will link it to the quantity we call **regret per unit of interaction**. The adaptiveness of the considered policies, *i.e.* the **speed at which it learns**, is asymptotically controlled by a minimal sampling rate given by each of the two lower bounds. The frequency at which suboptimal actions are sampled is controlled by the **information per unit of interaction**. The logarithmic rate implies that there is a diminishing return on the value of information, while it is still necessary to sample suboptimal actions. We recall all those terms here to foster a physical process viewpoint on the algorithms we are going to present.

Tight lower bound and optimal algorithms

Given a class ζ of Bandit problems, a regret lower bound informs us of the minimal logarithmic growth rate that any uniformly fast convergent policy must suffer from the uncertainty of the problem it is learning to solve. By inclusion, this inequality applies to the class of policy with a uniformly maximal convergence rate. Good candidates for uniformly maximal convergence rate are precisely those policies that may satisfy the lower bound. More precisely, polices that satisfy the lower bound uniformly on class ζ are uniformly fast convergent policies and called optimal. A lower bound will be said to be tight if there exist a policy matching its regret logarithmic growth rate. This will be the case for the problem of bounded Bandits. A Bandit problem (A, β) is said to be bounded if for all arm $a \in A$, the support $\text{supp}\,(\beta(a))$ of the reward distribution associated to arm a is lower bounded by a known finite number $m \in \mathbb{R}$ and upper bounded by a known finite number $M \in \mathbb{R}$, with the constraint $m < M$. The space ζ is therefore parameterized by the two quantities m and M. In this case, optimal and numerically efficient algorithms are known.

When presenting the two types of generic regret lower bounds we alluded to how those are linked to two types of algorithmic designs. In the next sections, we will more deeply explain how the two types of lower bounds intimately relate to MED-like and UCB-like algorithms. In the previous section devoted to the Bandit control problem, we presented a simple Dirac problem showing how one can make use of the regret lower bound to design an optimal learning policy. The next sections will be devoted to addressing the more complicated case of the generic lower bound presented in this section.

Finite time From an asymptotic optimality standpoint, finite time regret or n-stage regret does not really matter as long as, in the long run, the sampling rate corresponds to the optimal ones. However, finite time performances are quite important for the practitioner. While more difficult and tricky to tackle than asymptotic properties, finite time behavior of adaptive sampling strategies is an interesting and important topic for the theoretician. Because of the theoretical difficulty of deriving fine finite time results, it is interesting to deeply think about the algorithmic design, the empirical quantities that the learner can access, and how to

relate them. This is what we do in the next section, before presenting Bandit algorithms. Another important point is the one of **numerical complexity**. Mathematically, there is no notion of elapsed time between two successive choices made by the algorithm. However, in the tangible world, the learner may be constrained by how much time or memory resources it can use to compute the next action it decides to sample.

Wrap it up A Bandit problem (A, β) is specified by a finite set of arms A and a function β that maps an arm $a \in A$ to real-valued random variable $\beta(a)$ of probability distribution $F_a \in \mathcal{F}_a$ where $\mathcal{F}_a$ is known to the learner. The random variables $\beta(a)$ are called reward distributions and the expected reward of an arm a is denoted $\mu(a|\beta)$ or $\mu(a)$ and corresponds to the expected value of the random variable $\beta(a)$, $\mu(a) = \mathbb{E}_{F_a}(\beta(a))$. The interaction of an agent or policy with a Bandit problem proceeds as follows. At each interaction $t \in N$, the learner chooses an arm $a_t \in A$ based on the past observations and decisions, then receives and observes a sample X_t (called the reward), conditionally independent, sampled from $\beta(a_t)$. The goal of a learner is to maximize the cumulative reward received over time, *i.e.* over the sequence of interactions. The mean of each arm is unknown, which makes the problem non-trivial, hence the learner should adjust its sampling strategy based on past information obtained from drawing different arms in order to maximize the expected sum of rewards. The maximal expected value of a finite Bandit problem is denoted by $\mu^\star$, defined as $\mu^\star = \max_{a \in A} \mu(a)$. The performance of the strategy used by the agent is measured by the *regret*, that compares the expected sum of rewards obtained by an oracle that would constantly pull an optimal arm and the ones obtained by the learner, up to some time horizon n, that we assume is unknown to the learner.

3.2 Solving the problem

A Bandit problem with a well-defined optimization problem and objective is given to us. The regret lower bound indicates that, within the considered class of algorithm, it is impossible to beat a certain level of sampling performance due to the uncertain nature of the very problem we are solving, the one of interacting with an optimal action. The existence and construction of such an algorithm therefore remains. Can we find a finite sequence of rigorous instructions, that can be used to solve a class of Bandit problems where distributions are specified to belong to a subset $\mathcal{F}$? What are the specifications for performing calculations and data processing of the collected information? At any given interaction moment, the main question the algorithm must answer is the one concerning the choice of the next action to interact with. To do so, it is allowed to process past information however we may prescribe it to do, but we insist that the purpose is to **compute the next action**. This purpose should not be confused with intermediate goals such as estimation. . . .

Generic Bandit Algorithms

Similarly to what was done in the section dedicated to the Bandit control model, in our computation framework, we are bound to perform comparisons. Therefore, a general bandit algorithm is always based on the computation of **numerical quantities**, numbers, one for each arm. Comparisons are made between those numbers and the next arm to sample is chosen based on the result of those comparisons. Formally, those numerical quantities are the results of a finite function $\mathcal{I} : A \to \mathbb{R}$ therefore used to compute the next action to play. While not written for the moment, the function $\mathcal{I}$ depends on $\mathcal{F}$, the history of information H and potentially on a random number generator. Usually, an argmax or argmin of this function is used as the next action to play. That is to say, the next action s to be played is such that $s \in \operatorname{argmax}_{a \in A} \mathcal{I}(a)$ or $s \in \operatorname{argmin}_{a \in A} \mathcal{I}(a)$. This is almost a consequence of the comparison-based computation framework that we are using. The physical interpretation of learning will be emphasized through the special attention given to the intensive and extensive properties. I think that when designing algorithms, the following question is worth pondering upon: what are the algorithmic quantities that should scale with the problem?

One could replace $\mathcal{I}$ by $\widetilde{\mathcal{I}} = \mathcal{I} - \text{rule}_a \mathcal{I}$ if the rule used to pick the next action, *argrule*, is not that of an argmax or argmin. rule_a is used to compute the value of $\mathcal{I}$ for the chosen arm.

We formalize three algorithmic designs that are used in the Bandit literature, the **index based** in Algorithm 2, the **distribution based** in Algorithm 4, and the **set based** in Algorithm 6. The set based algorithmic design might also be called **round based** design. As we will see, set based design can be built from index based and distribution based design, but we nonetheless identify this design as a useful category.

Index Based Algorithms

Definition 3.2.1 (Arm Index Function) *Let (A, β) be a Bandit problem with $\beta : A \to \mathcal{F}$ a known subset of distributions on $\mathbb{R}$. Let H be an admissible history on the Bandit problem (A, β). Then, an **index** of arm $a \in A$ is a*

measurable function

$$\mathcal{I}_{\mathcal{F}}(\cdot; a) : H \mapsto \mathcal{I}_{\mathcal{F}}(\cdot; a) \in \mathbb{R} .$$

In practice, an arm index function will be used to compute a notion of *score* for an arm a given a history H and a hypothesis set $\mathcal{F}$. An **index based Bandit policy** would then compute the $\operatorname{argmin}_{a \in A} \mathcal{I}_{\mathcal{F}}(H; a)$ or $\operatorname{argmax}_{a \in A} \mathcal{I}_{\mathcal{F}}(H; a)$ to select the next arm to sample. Afterward, the algorithm would update the history, compute the new indices and repeat this process. We remark that our definition of an index slightly differs from the conventional one used in the Bandit literature, see [6], in which an index only depends on the history of samples collected from the distribution $\beta(a)$ associated to arm $a \in A$. What we defined is called a generalized index, see [12]. However, I would rather advocate that what is called a generalized index would better be called an index and that what is called an index would rather be called restricted index. I cannot find a good reason to exclude information from the history H when defining the index object. Sure, if we think about an index as related to the confidence interval of an arm, then it would only depend on the history associated to that arm. However, as we already emphasized, estimating the problem is a different objective than solving the problem. While the solving part of the design might be in the argmax or argmin, I do think that it is better if it is present in the index construction. However, we will see that when considering distributions instead of index, things might be different. When using an index to compute the next action to sample, one is more interested in computing a quantity related to the likelihood of optimality of this action. Rather, we will see that some strategies, that we call MED-like strategies, *e.g.* [16], are interested in computing the **empirical unlikelihood of optimality** of each arm. Other strategies, *e.g.* [21] and [5], are more interested in computing the **maximal likely reward** of an arm. In this section, I will also explain how to improve the original KL-UCB and enhanced KL-UCB+ algorithm by considering the Definition 3.2.1 and by thinking about extensive and intensive quantities. In Algorithm 2,

[6]: Lattimore et al. (2020), *Bandit algorithms*

[12]: Honda et al. (2011), 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'

[5]: Auer et al. (2002), 'Finite-time analysis of the multiarmed bandit problem'

Algorithm 2: Generic generalized index bandit policy

Input: A bandit tuple (A, β) as in Definition 3.1.12;
An index function $\mathcal{I}_{\mathcal{F}}$ as in Definition 3.2.1;

The history H is a *multiset*. The data structure used to represent H numerically is a *dictionary* or *hash table*.

```
Initialize history H as H = ∅;
for t ∈ ℕ do
    forall a ∈ A do
        Compute index I_a = 𝓘(H; a);
    Compute a ∈ argmax_{e∈A} I_e;
    Sample a reward r ~ β(a) from reward distribution β(a);
    Update history, H ← H ∪ {(a, r)};
```

we illustrate how a generic index based bandit policy work. While the index is computed using the whole history H, the function $\mathcal{I}_{\mathcal{F}}$ is in fact more a sequence of function defined on size increasing t-histories where t is the number of interactions. Of course, depending on the specific index, it is possible to write the index function of a specific algorithm as depending on the previous index value and the current accessed information, or depending only on the history of a specific arm. However,

those are related to practical implementation points that are left for the discussion about the specific algorithms that we present in this thesis.

In some cases, the index might not be a map to $\mathbb{R}$ but to a real valued distribution.

Definition 3.2.2 (Randomized Arm Index Function) *Let (A, β) be a Bandit problem with $\beta : A \to \mathcal{F}$ a known subset of distributions on $\mathbb{R}$. Let H be an admissible history on the Bandit problem (A, β). Then, a **random index** of arm $a \in A$ is a measurable function*

$$\mathcal{I}_{\mathcal{F}}(\cdot; a) : H \mapsto \mathcal{I}_{\mathcal{F}}(\cdot; a) \in \mathcal{P}(\mathbb{R}) \ .,$$

where $\mathcal{P}(\mathbb{R})$ is the space of probability distributions on $\mathbb{R}$.

In such a case, a sample is collected from each of the random index in order to compute the next arm to play. This is illustrated in the Algorithm 3. Compared to Algorithm 2 the original *compute index* of

Algorithm 3: Generic generalized random index bandit policy

Input: A bandit tuple (A, β) as in Definition 3.1.12;
An index function $\mathcal{I}_{\mathcal{F}}$ as in Definition 3.2.1;

```
1 Initialize history H as H = ∅;
2 for t ∈ ℕ do
3     forall a ∈ A do
4         Sample the computed random index I_a ~ 𝓘(H; a);
5     Compute a ∈ argmax_{e∈𝒜} I_e;
6     Sample a reward r ~ β(a) from reward distribution β(a);
7     Update history, H ← H ∪ {(a, r)};
```

The history H is a *multiset*. The data structure used to represent H numerically is a *dictionary* or *hash table*.

line 4 has been replaced by *sample the computed random index*. Interestingly, in the seminal work of Thompson [24] that pioneer the research field on Bandit, such an algorithmic design is present. It is based on the construction of a posterior distribution $\mathcal{I}(H; a)$ associated to arm a that we can sample from. Samples are then used to create a t-ranking of the arms, *i.e.* a ranking of the arms after the t^{th} interaction, from which the maximal element is selected as the next arm to be sampled. This interesting idea has been developed further in papers from Agrawal et al. [27, 28], Kaufmann et al. [29]. A survey of such algorithmic design can be found in [30]. More recently, Riou and colleagues introduced the NPTS (Non-Parametric Thompson Sampling) algorithm [26] which has the desirable property of being non-parametric.

[24]: Thompson (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'

[27]: Agrawal et al. (2012), 'Analysis of Thompson Sampling for the multi-armed bandit problem'

[28]: Agrawal et al. (2013), 'Further Optimal Regret Bounds for Thompson Sampling'

[29]: Kaufmann et al. (2012), 'Thompson sampling: An asymptotically optimal finite-time analysis'

[30]: Russo et al. (2018), 'A Tutorial on Thompson Sampling'

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

Distribution Based Algorithms

A **distribution based Bandit policy** would compute a probability distribution $\mathcal{I}_{\mathcal{F}}(H) \in \mathcal{P}(A)$ on the set of arm. It is formalized in Definition 3.2.3.

Definition 3.2.3 (Arm Distribution Function) *Let (A, β) be a Bandit problem with $\beta : A \to \mathcal{F}$ a known subset of distributions on $\mathbb{R}$. Let H be*

an admissible history on the Bandit problem (A, β)*. Then, a* ***distribution function*** *is a measurable function*

$$\mathcal{I}_{\mathcal{F}} : H \mapsto \mathcal{I}_{\mathcal{F}}(H) \in \mathcal{P}(A) \ .$$

In practice, in such a distribution, the mass $\mathcal{I}_{\mathcal{F}}(H)(a)$ will be the probability of selecting the arm a. Likely, it will be related to the *probability of optimality* of the arm a given a history H and a hypothesis set $\mathcal{F}$. After sampling an arm $a \in A$ according to the law of $\mathcal{I}_{\mathcal{F}}(H) \in \mathcal{P}(A)$, the algorithm samples a reward from the distribution $\beta(a)$ and update the history. This process is then repeated in loop starting at line 2 of Algorithm 4.

Algorithm 4: Generic generalized random bandit policy

Input: A bandit tuple (A, β) as in Definition 3.1.12;
An index function $\mathcal{I}_{\mathcal{F}}$ as in Definition 3.2.1;

The history H is a *multiset*. The data structure used to represent H numerically is a *dictionary* or *hash table*.

1 Initialize history H as $H = \emptyset$;
2 **for** $t \in \mathbb{N}$ **do**
3 Compute distribution $\mathcal{I}_{\mathcal{F}}(H) \in \mathcal{P}(A)$;
4 Sample $a \in \operatorname{argmax}_{e \in \mathcal{A}} I_e$;
5 Sample a reward $r \sim \beta(a)$ from reward distribution $\beta(a)$;
6 Update history, $H \leftarrow H \cup \{(a, r)\}$;

In Algorithm 4, we illustrate how a generic distribution based bandit policy work. It should be noted that while theoretically, it is useful to distinguish the computation of $\mathcal{I}_{\mathcal{F}}(H) \in \mathcal{P}(A)$ line 2 from the sampling $a \sim \mathcal{I}_{\mathcal{F}}(H)$ line 3, it is sometimes not necessary and even detrimental to do so from a practical standpoint. Again, this is all about the complexity of what we **really aim** to do. **Simulating** or **sampling** from a probability distribution does not have the same complexity as **computing** the probability distribution. This simple yet important remark led to the creation of Markov chain Monte Carlo (MCMC) methods, that are a class of algorithms for sampling from a probability distribution. This class of algorithms is still an active research field. [15] Thus, while the distribution is computed using the whole history H, the distribution $\mathcal{I}_{\mathcal{F}}$ is in fact more a sequence of function defined on size increasing t-histories where t is the number of interactions. Depending on the specific distribution, it is possible to write the distribution function of a specific algorithm as depending on the previous distribution and the current accessed information, or depending only on the history of a specific arm. However, those are related to practical implementation points that are left for the discussion about the specific algorithms that we present in this thesis.

15: It can also be connected to a research field that has regained popularity due to its connection with the neural networks literature, **generative model**.

This algorithmic design is embodied by the MED algorithm [12]. In the MED algorithm, unnormalized weights $P(H; a)$ are computed and the arm to be played is sampled from the normalized multinomial distribution $Mult\left(p(H; a)\right)_a$ of parameter $\left(p(H; a)\right)_a \in \triangle_{|A|}$ the probability simplex of dimension $|A|$. The elements of the vectorized parameter are such that $p(H; a) = \frac{P(H;a)}{\sum_a P(H;a)}$. One advantage of such a stochastic design is that it can be helpful to *implicitly* add some constraints or stability property to the sampling method. By *implicit*, we mean that some interesting properties, that would be difficult to write in a deterministic form, might be satisfied on average or with a high enough probability. Such properties are satisfied

[12]: Honda et al. (2011), 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'

implicitly without the need of writing algorithmic rules to ensure the algorithm will satisfy them. Sometimes, it makes it easier to write ideas down. Sometimes, one can phrase a good algorithmic idea by replacing a deterministic statement about Bandit control into an *expected statement* about Bandit learning. For instance, in the bandit **control** problem we **follow the leader** while in the bandit **learning** problem we **expect to follow the leader**. Instead of looking for a deterministic rule that would allow us to get this desired behavior, one can instead think of a distribution over the space of arm that would, given collected information, behave this intended way. The idea of a stochastic follow the leader quickly emerge. While the specific implementation and mathematical way to do this is not trivial, I do think that a significant part of the thought process has been done once we settle on this idea of a **stochastic follow the leader**. This shows that there is more connection between the index based approach and the distribution based approach. The link between the two is the **randomized index approach** defined in Definition 3.[16]

16: More often than not, one should give a try to the idea of replacing a stochastic quantify by a Dirac distribution and see if this trigger some ideas.

Link with *Randomized Arm Index Function* As already recalled, sampling from a distribution is a different task than computing it. In the follow the leader approach, one pick the best computed expected value $\text{argmax}_a \mu_a$ as the arm to play. This is possible in the Bandit control problem after observing all arms once due to the perfect knowledge of the expected returns. Here, we want to sample from a relevant "argmax_a distribution". The MED algorithm is capable of computing such a distribution. Other algorithmic design may not and while computing such a distribution might be difficult, one can sometimes find ways to sample from it. Indeed, one can write the Algorithm 3 that consider *randomized arm index functions* as an instance of Algorithm 4 that consider an *arms distribution function*. The lines 3-5 from randomized index Algorithm 3 can be written as sampling from a distribution in $\mathcal{P}(A)$. The probability mass $\mathcal{I}_{\text{Distr.}}(H;a)$ is such that

$$\mathcal{I}_{\text{Distr.}}(H;a) = \mathbb{P}\left(\mathcal{I}(H;a) = \max_{a' \in A} \mathcal{I}(H;a')\right)$$

where $\mathcal{I}(H;a')$ are the original $|A|$ random variable that defined the randomized index (see Definition 3.2.2) in Algorithm 3. While computing those probability mass given the distributions $\mathcal{I}(H;\cdot)$ might be difficult, one can see that sampling from such a distribution is much easier. One only have to sample an index for each arm, which amounts to sampling a random permutation, and compute the argmax. The Algorithm 5 details how to sample an arm from the distribution induced by a randomized index function. Remark that this method can be applied with deterministic

Algorithm 5: Sampling from probability distribution on A corresponding to a randomized index

Input: An randomized index function $\mathcal{I}_{\mathcal{F}}$ as in Definition 3.2.2;

1 **for** $a \in A$ **do**

2 Sample the computed random index $I_a \sim \mathcal{I}(H;a)$;

3 Compute and return $a \in \text{argmax}_{e \in \mathcal{A}} I_e$;

indexes that are considered as Dirac distributions.

In the previous section devoted to index based strategies, we mentioned that some strategies, that we call MED-like strategies, *e.g.* [16], are interested in computing the **empirical unlikelihood of optimality** of each arm. As the name suggest, this is the case of the MED algorithm [12] whose probability distribution on the set of arms is based on the unlikelihood of optimality. [17] We saw that other strategies, such as KL-UCB [21], are more interested in computing the **maximal likely reward** of an arm. The corresponding algorithms in the distribution based setting are those related to **Thompson sampling** (TS), see [24]. In TS-like strategies, we sample an *expected value* for each arm and use those sampled values to compute the next arm to play by picking an argmax_a of those values. TS-like strategies are using randomized indexes but can be seen as distribution based thanks to the procedure explained in Algorithm 5 which show how to sample an arm from a distribution based on randomized indexes. When analyzing the algorithm, one can see that what we are really doing in this case is sampling a random permutation of the arms but the only interesting quantity for us is the argmax. The work that is the closer to correspond to the KL-UCB algorithm in this distribution based one is NPTS, recently introduced by Riou and colleagues in [26] which has the desirable property of being non-parametric.

In both the index based and distribution based setting there exists algorithms derived from the **unlikelihood of optimality** and **maximal likely reward** viewpoints.

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

[12]: Honda et al. (2011), 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'

17: We will see that it is akin to a Boltzmann distribution with the number of samples similar to the invert of a temperature and the energy of an action linked to an information-based distance between a suboptimal arm and the optimal one. The optimal arm is grounded, *i.e.* it is the arm with minimal level of energy (which is always defined up to an additive constant). Raising the temperature means to remove samples while increasing it means adding samples, and we can understand intuitively how this should affect the repartition of *particles or samples* in the different energy levels.

[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'

[24]: Thompson (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

Set Based Algorithms

We now discuss a final algorithmic design, set based, that is encountered in the bandit literature. From an intuition point of view, I do think that it is one of the harder to *feel* and, in my opinion, it is an interesting idea that may have useful additional properties that set this design apart. In particular, it may be that such a design help with finite time properties and stabilize the sampling process in the early interactions, when information is scarce and the problem *highly uncertain*. The reason is that a **set based Bandit policy** sequentially computes *sets of arm* to be played. This differs from the previous design where policies compute a sequence of arm or *singleton*. Formally, we define an **arm set function** in Definition 3.2.4.

Definition 3.2.4 (Arm Set Function) *Let (A, β) be a Bandit problem with $\beta : A \to \mathcal{F}$ a known subset of distributions on $\mathbb{R}$. Let H be an admissible history on the Bandit problem (A, β). Then, an **arm set function** is a measurable function*

$$\mathcal{I}_{\mathcal{F}} : H \mapsto \mathcal{I}_{\mathcal{F}}(H) \subseteq A \ .$$

In Algorithm 6, we illustrate how a generic set based bandit policy work. Contrary to the index and distribution based algorithms, the index $t \in \mathbb{N}$ line 2 does not count the total number of interactions with the Bandit problem. Indeed, within an iteration of the loop line 2, one can sample several arms and update the history several times. This can be seen from the loop starting line 4 which iterates through all the arms that belong to

the set of arms $\mathcal{A}$ to be sampled, computed line 3 using the prescribed arm set function. In the context of set based Bandit policy, the index t is

Algorithm 6: Generic generalized set-based bandit policy

Input: A bandit tuple (A, β) as in Definition 3.1.12;
An arm set function $\mathcal{I}_{\mathcal{F}}$ as in Definition 3.2.4;

```
1 Initialize history H as H = ∅;
2 for t ∈ ℕ do
3     Compute the subset 𝒜 := 𝓘_ℱ(H) ⊆ A;
4     forall a ∈ 𝒜 do
5         Sample a reward r ~ β(a) from reward distribution β(a);
6         Update history, H ← H ∪ {(a, r)};
```

The history H is a *multiset*. The data structure used to represent H numerically is a *dictionary* or *hash table*.

usually referred to as a **round**. The number of round is therefore smaller, often strictly, than the number of interactions with the problem, which can be computed from the history.

Set based Bandit policies uses indexes as in Definition 3.2.1 or distributions as in Definition 3.2.2 in order to compute the sequence of sets. For instance, given an index function $\mathcal{I}^{\text{index}}_{\mathcal{F}}$, a set could be all the arms such that their index is above a certain threshold $\gamma(H)$, *i.e.* $\mathcal{I}_{\mathcal{F}}(H) = \left\{a \in A | \mathcal{I}^{\text{index}}_{\mathcal{F}}(H; a) \geq \gamma(H)\right\}$. This is for instance what is done in the DMED [15] algorithm. Duel based algorithm, as in [31] are also set based. In such a design, a reference arm, called leader, is chosen, and its index is compared to the indexes of other arms, called challengers. The duel is won by a challenger if its index is larger than that of the leader. The computed set is then made of all those arms that won their duel or the singleton leader arm if no challenger won. The most simple duel-like structure that one can think of is one induced by an index policy, Algorithm 2. Instead of computing $a \in \text{argmax}_{a \in A} I_a$ in line 5 of this Algorithm 2, one can simply define the set $\mathcal{A} := \mathcal{I}_{\mathcal{F}}(H)$ of line 3 in Algorithm 6 to be equal to that argmax, *i.e.* $\mathcal{I}_{\mathcal{F}}(H) = \text{argmax}_{a \in A} I_a$. In index based strategies, ties are usually broken using the number of samples of arm, choosing the less sampled arm, and eventually according to an arbitrary rule, *e.g.* uniformly at random. Here, the idea would be to not decide on any tiebreaker but to sample all the arms within the argmax at the considered round. I am not aware of any algorithm that exist in the literature implementing this simple modification. [18]

[15]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models'

[31]: Chan (2020), 'The multi-armed bandit problem: An efficient nonparametric solution'

18: I hope to present such a modification of the IMED algorithm at my oral presentation.

In popular set-based algorithms, the computed set will often be the singleton made of the empirical maximal arm (hopefully) or a singleton made of a suboptimal arm that would have been chosen by the index or distribution based policy used to build the set. However, this phenomenon occurs mostly asymptotically. When the information is scarce and the number of interactions is still small, the computed set might contain a large portion of the arms. This design may prevent from being greedy too early by mitigating the effect of error inducing first samples.

Index to set based: the symmetry viewpoint

Modification of an index based policy Above, we described a strategy to transform an index based strategy into a set based one. What would

intuitively justify such an approach? Can such an intuition support the last claim we made about mitigating short horizon regret? I would argue that, yes we can! The idea is that our algorithmic design should be **invariant to similar conditions**. A difference in the number of samples from two arms should **only** come from the problem and not from the algorithm.

Breaking symmetries More precisely, imagine the following scenario. After computing all indexes, there are two arms a and a' having the same index, $I_a = I_{a'}$, and this value is assumed to be maximal. From an information viewpoint, what distinguish those two arms? In my opinion, nothing. However, one could still argue that one of the arms has a smaller number of associated samples, therefore a less reliable index and that we should break the tie by sampling it. This is a fair point. However, there are situations in which both the indexes and number of pulls are equal. Imagine that this is the case, what do we do? An index based algorithm would arbitrarily break the ties, usually uniformly at random. Therefore, on average, two such arms are indiscernible. However, the index based policy would have to compute which of the two arms to choose, and the *sampling trajectories* would then depend on an event that is independent of the problem. Furthermore, it discriminates between arm that, according to the used statistics (index and number of samples), are indiscernible. On average, the *symmetries* of the problem are observed, *i.e.* on average two indistinguishable arms are processed equally by the algorithm, but on a specific random run, two indistinguishable arms might not be processed equally by the algorithm and *symmetries* are not observed.

Restoring symmetries The transformation procedure of an index based policy into a set based one that we described allows to observe the *symmetries* of observations. Instead of choosing an arm a in the argmax_a of indexes, $a \in \operatorname{argmax}_a \mathcal{I}_{\mathcal{F}}^{\text{index}}(H; a)$, one can compute the set of arm $\mathcal{A} := \mathcal{I}_{\mathcal{F}}(H)$ to be sampled at considered round as

$$\mathcal{I}_{\mathcal{F}}(H) = \operatorname*{argmax}_{a \in A} \mathcal{I}_{\mathcal{F}}^{\text{index}}(H; a) \ .$$

As stated, the number of samples may be an interesting quantity to break ties between arms of equal indexes, and we could therefore consider the following set construction,

$$\mathcal{I}_{\mathcal{F}}(H) = \operatorname*{argmin}_{a \in \operatorname{argmax}_{a \in A} \mathcal{I}_{\mathcal{F}}^{\text{index}}(H;a)} N_a(H) \ .$$

In this set based setting, if $\mathcal{I}_{\mathcal{F}}(H) = \{a, a'\}$, then both the arms are sampled and any subsequent decisions would be based on the collected samples and not on the arbitrary choices of an arm at some point. **The symmetry breaks when the problem breaks it, not when the algorithm breaks it.** This restoration of the symmetry may help in the early stages, when several arms may be in $\mathcal{I}_{\mathcal{F}}$. In particular, because it may slightly increase exploration in the short horizon, it may help reduce the variance of the regret in larger horizon, when the sets are almost always singleton.

Reducing the regret variance Think about a three arms Bernoulli Bandit problem with means 0.4, 0.5, 0.6. If all the three arms have an initial sample equal to zero, which is possible with non-negligible probability, then an index based policy would pick the next arm randomly. It may be the arm of mean 0.5, and it may obtain a reward of 1 this time. This would increase the likelihood of sampling this same arm, which is suboptimal. There a $2/3$ probability that such an event occur if all initial samples are 0s. Afterward, the index of arm with mean 0.4 and optimal arm with mean 0.6 are the same. Therefore, an index based policy would have $1/2$ probability of sampling the suboptimal one. If it gets a 1, then it would again increase the likelihood of sampling one of the two suboptimal arms. Therefore, one could see that, even if on average, the algorithm will behave correctly and have a logarithmic regret, on a specific instance, it may suffer for quite a long time from initial mistakes and lack of short horizon exploration. The described problem is solved when considering set based algorithmic design. While bad event can still occur, its probability is reduced to the minimum as such an event is now only due to the problem and not due to the random choices of indistinguishable arms. In the initial state described, a set based transformation of the same index policy would add all arms to the set and will only discriminate between them when samples collections would start to differ. While this may add a bit of exploration in the early stages, it certainly removes some in later stages. Therefore, I do think that such an algorithmic design help to reduce the standard deviation of the regret function, a random variable.

Tu sum it up. In index based strategies, exogenous source of randomness is used in the form of a way to break ties between indistinguishable arms. By choosing a set based strategy and deciding not to break the ties, we remove such an external dependency where *external* means that some choices are made based on measures that are *independent* of the problem. Therefore, by removing a source of arbitrary or randomness, the variance of the algorithmic regret mechanically diminish. The external source of randomness, or arbitrary decision is to distinguish from the internal source of randomness used by distribution based strategies. In my opinion, as far as possible, **algorithms should be invariant to symmetries of information by design**. This fact may explain why the MED algorithm seems to empirically have a lower regret variance than its deterministic counterpart, IMED. Also, This idea might also help understand what algorithms are worst-case (or min-max) optimal. Indeed, if the algorithm is not using any source of exogenous randomness/information, then there is less to exploit by an adversary.

Towards the DMED algorithmic design In the previous paragraphs, we saw that transforming an index based strategy into a set based one could be beneficial. However, some strategies such as the DMED [15] algorithm uses another transformation of an initial index based policy.[19] Given an index function $\mathcal{I}_{\mathcal{F}}^{\text{index}}$, a set is built as the arms having their indexes above a certain threshold $\gamma(H)$, *i.e.* $\mathcal{I}_{\mathcal{F}}(H) = \left\{a \in A | \mathcal{I}_{\mathcal{F}}^{\text{index}}(H; a) \geq \gamma(H)\right\}$. This extends the previous transformation as $\gamma(H)$ could be chosen as the maximum of index, $\gamma(H) = \max_a \mathcal{I}_{\mathcal{F}}^{\text{index}}(H; a)$ thus making $\mathcal{I}_{\mathcal{F}}(H)$ equal to the argmax_a. There are at least two related reasons as to why this transformation makes sense.

[15]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models'

19: In this section we present set based policy as transformation of index based one, but one could obviously define a set based policy independently of a reference to an index one. We do so to better connect algorithmic design together.

The first is about **uncertainty**. Given the history and a desired level of precision $\delta(H)$ that depends on the history, one could say that the size of a confidence interval of precision $\delta(H)$ for the value of $\max_a \mathcal{I}^{\text{index}}_{\mathcal{F}}(H;a)$ is $\epsilon(H)$. That is to say, with probability more than $\delta(H)$, the true (*i.e.* expected) value of $\max_a \mathcal{I}^{\text{index}}_{\mathcal{F}}(H;a)$ is located within an interval of length $2\epsilon(H)$ centered in the empirical value $\max_a \mathcal{I}^{\text{index}}_{\mathcal{F}}(H;a)$. In this scenario, does it make sense to discriminate between a maximal arm in $\text{argmax}_a \mathcal{I}^{\text{index}}_{\mathcal{F}}(H;a)$ and another arm a' that is such that $\mathcal{I}^{\text{index}}_{\mathcal{F}}(H;a) - \mathcal{I}^{\text{index}}_{\mathcal{F}}(H;a') < \epsilon(H)$. If the available information allows us to discriminate between arms up to $\epsilon(H)$ given the wanted precision $\delta(H)$, then one probably should not discriminate action choices based on index that are less than $\epsilon(H)$-separated. In short, we should wonder whether it makes sense to discriminate between action that not statistically different and if our algorithm will process them too differently. Asymptotically, this may not change anything but for short horizon, this may experimentally boost performances and reduce the regret variance.

The second is about **threshold**. Instead of comparing indexes to each others, one may imagine comparing the indexes to a threshold $\gamma(H)$ that depends on the history. For instance, if the index of an arm is related to its likelihood of optimality, then it may make sense to decide to sample an arm as long as its likelihood of optimality is above a certain threshold. The threshold, depending on the history, *a priori* depends on the empirically optimal arm, but it may not be necessary if the index already take that information into consideration. In the event there is *jump* in the optimal arm belief, *i.e.* the algorithm computed it was one and regression to the mean prove the algorithm wrong, then a lot of computed quantity changes and the problem the algorithm was solving is transformed into the one corresponding to the newly identified best arm. In that case, it may make sense to consider a threshold and set based strategy to ensure a sufficient amount of new exploration given the new conditions. In the long run, the sets should not be too different from singleton but in short horizon, this may make a difference.

Why does algorithmic design matter? The reasons we insisted on those algorithmic design are the following. First, we wanted to introduce some physical way of thinking that we hope to deliver in this thesis. The main example of this section is the one of symmetry preserving design. The second is that finite time theoretical guarantees are hard to get. The study of the variance of the regret is also a hard problem compared to the expected value which is the topic of most research paper in the bandit community. However, we show that some algorithmic design, such as set based, by removing an exogenous source of randomness mechanically reduces the variance of the history and therefore regret distribution. By simply considering the algorithmic design, we could not conclude about the variance of a specific algorithm but could nonetheless say something about the variance of a transformed design relative to an initial algorithmic one. In my opinion, better understanding of short horizon behavior is important for theoretician and practitioner alike while difficult to obtain. Algorithmic design may help to better understand sampling algorithm from both a theoretical and experimental viewpoints.

3.3 Algorithms in the literature

In order to minimize the regret, a learner faces the classical exploration/-exploitation trade-off: it needs to balance *exploration*, that is gaining information about the expected values of the arms by sampling them, and *exploitation*, that is playing the most promising arm sufficiently often. This Many algorithms have been proposed to solve the multi-armed bandits problem (see [32] for a survey). In this section, we present and review some of the Bandit algorithms that can be found in the Bandit literature.

Complexity

For each type of algorithm (index, distribution, or set based), we present how its significant feature is computed. In this thesis, we attach great importance to the time complexity and, to a lesser degree, space complexity of the crafted algorithms. We summarize in Table 3.1 the complexity required to compute the index (or distribution for MED and NPTS) of the Bandit algorithms that are the most mentioned and used in this thesis.

The time complexity is written as a function of the number of collected samples from a suboptimal arm and the per-interaction cost of solving can be deduced from it. The space complexity relates to the required memory needed as a function of the number n of collected samples from a suboptimal arm. To compute those costs, we computed the number

Table 3.1: Time complexity and space complexity needed to compute the index of a suboptimal arm a after n samples have been collected from its associated distribution $\beta(a)$.

Algorithm	Time complexity	Space complexity	Constant	Optimality
KL-UCB [21]	$\mathcal{O}(n\log(n)^2)$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
kl-UCB [21]	$\mathcal{O}(\log(n))$	$\mathcal{O}(1)$	$\frac{1}{kl(p(a),p^\star)}$	Sub-opt.
UCB [5]	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\frac{1}{2(\mu^\star-\mu(a))^2}$	Sub-opt.
NPTS [26]	$\mathcal{O}(n)$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
MED [33]	$\mathcal{O}(n\log(n))$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
IMED [16]	$\mathcal{O}(n\log(n))$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.

of "elementary" $\mathcal{O}(1)$ operations computed at each interaction. Amongst elementary operations, we included sequential updates of means, of number of pulls. The time complexity takes into account a precision of $\frac{1}{n}$ on the computation of the indexes. Indeed, due to limited resources, one must compute numerical quantities up to a given precision, which we set at $\frac{1}{n}$. The constant column refers to the asymptotic logarithmic sampling rate of a suboptimal arm $a \in \mathcal{A}$ in the unstructured Bandit. In the kl-UCB algorithm, kl refers to the KL-divergence of the Bernoulli distribution.

Unstructured Bandits

The study of the lower bounds had a crucial impact on the development of provably asymptotically optimal strategies. In the case of *unstructured* bandit $\mathcal{B} = \mathcal{F}^A$, this includes strategies that are categorized in the literature as based on *Optimism in Face of Uncertainty*. As we mentioned

in the previous section, this category of algorithms is based on the allocation-constrained viewpoint, lower bound of Theorem 3.1.6. Instead of speaking about optimism in face of uncertainty, we prefer to talk, from the agent viewpoint, of worst case scenario given information and level of uncertainty. The KL-UCB [18, 34, 35] Algorithm 7 is an asymptotically optimal algorithm whose design reflects the lower bound.

[18]: Lai (1987), 'Adaptive treatment allocation and the multi-armed bandit problem'
[34]: Cappé et al. (2013), 'Kullback–Leibler upper confidence bounds for optimal sequential allocation'
[35]: Maillard (2018), 'Boundary Crossing Probabilities for General Exponential Families'

Algorithm 7: Index of the KL-UCB policy

Input: Arm $a \in \mathcal{A}$ of a bandit tuple $(\mathcal{A}, s, \mathcal{V})$;
A family $\mathcal{F}$ of distributions whose $\mathcal{V}$ is a subset;
History H of a sequential sampling of the bandit tuple;
A confidence function $\mathcal{C} : H \to \mathbb{R}_+$;

if *arm a has been sampled* **then**
  Extract history of samples H_a of arm a from H;
  return $\sup_{\mu \in \Xi(\mathcal{F})} \{\mathcal{E}_{\mathcal{F}}(H_a, \mu) \leq \mathcal{C}(H)\}$;
else
  return $+\infty$;

$\Xi(\mathcal{F})$ is the set (usually an interval) of all expected values of real random variables having a law in the set $\mathcal{F}$. Formally, $\Xi(\mathcal{F}) = \{\mathbb{E}(X) | X \sim \mathcal{F}\}$.

The confidence function that is used depends on the choice of the specific family $\mathcal{F}$. For distributions with bounded rewards, the original algorithm has $\mathcal{C}$ defined as a function of the total number t of interactions, and is such that $\mathcal{C}(t) = \log t + c \log\log t$ with $c > 2$. As we said, the presence of the total number of interactions since the beginning of interactions, while not bad *per se*, is not what we want to call a good algorithmic design. Rather, one would prefer to use $\mathcal{C}(H) = \max_a N_a(H)$, where $N_a(H)$ is the number of times arm a has been sampled in the history H of interactions. Indeed, $\max_a N_a(H)$ represent, in terms of the sample complexity, the maximal amount of information one could have had on any given arm, according to the history of samples. From a small number of samples regime viewpoint, it does not make much sense to consider t. Assume the situation where we solve a problem with 10 arms. After sampling all 10 arms exactly once, the constraint $\log t$ is equal to $\log 10$. Now assume that instead of 10 arms, we have a 1000 arms bandit problem. After sampling all 1000 arms exactly once, *i.e.* gathering the mandatory initial information, the constraint $\log t$ is equal to $\log 1000$, three times $\log 10$. However, the number of samples per arm is the same in the two problems. This is why we argue that the total number of interaction should not appear in the definitions of indexes and that we should rather consider quantities computed from the history H and number of samples. Of course, from an asymptotic viewpoint, since $\max_a N_a(H) \geq \frac{t}{|\mathcal{A}|}$, this does not change anything. From a finite time standpoint, this may change everything. In the KL-UCB algorithm, one must find the maximal expected regret one can occur from an arm give the allocation constraints computed from the history H, a form that is similar to that in Theorem 3.1.6. The computation of an empirical $\mathcal{E}_{\mathcal{F}}$ must be done $\log N_a(H)$ times for an arm a when the desired precision on the value of the maximal reward is $\frac{1}{N_a(H)}$. Since computing the $\mathcal{E}_{\mathcal{F}}$ is akin to solving a convex optimization problem as shown in [16], and has a complexity of $\mathcal{O}(n \log(n))$, the computational cost of KL-UCB can become quite heavy. In the case of bounded reward, it is possible to consider kl-UCB, where the parameterization of distributions by the

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

expected value makes the computation of index faster, albeit at the cost of optimality. The computation of the Bernoulli kl is $\mathcal{O}(1)$, but must be

Algorithm 8: Index of the kl-UCB policy

Input: Arm $a \in \mathcal{A}$ of a bandit tuple $(\mathcal{A}, s, \mathcal{V})$;
A family $\mathcal{F}$ of distributions whose $\mathcal{V}$ is a subset;
History H of a sequential sampling of the bandit tuple;
A confidence function $\mathcal{C} : H \to \mathbb{R}_+$;

```
if arm a has been sampled then
    Extract history of samples H_a of arm a from H;
    return sup_{μ∈Ξ(F)} {kl(μ(H_a), μ) ≤ C(H)};
else
    return +∞;
```

$\Xi(\mathcal{F})$ is the set (usually an interval) of all expected values of real random variables having a law in the set $\mathcal{F}$. Formally, $\Xi(\mathcal{F}) = \{\mathbb{E}(X) | X \sim \mathcal{F}\}$.

performed $\mathcal{O}(\log n)$ times at each time step. Yet, such a cost is reasonable since the number n of number of samples from suboptimal arm should be logarithmic in the number of interactions. Interestingly, the optimality of KL-UCB was only proved recently [36], as the seminal work of [21, 37] only proved it for Multinomial distributions.

[36]: Agrawal et al. (2021), 'Regret Minimization in Heavy-Tailed Bandits'
[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'
[37]: Burnetas et al. (1996), 'Optimal adaptive policies for sequential allocation problems'

Finally, the most celebrated UCB-type of algorithm in the literature is the eponymous Upper Confidence Bound (UCB) algorithms [5, 38]), for

Algorithm 9: Index of the UCB policy

Input: Arm $a \in \mathcal{A}$ of a bandit tuple $(\mathcal{A}, s, \mathcal{V})$;
History H of a sequential sampling of the bandit tuple;
A confidence (or bonus) function $\mathcal{C} : H \to \mathbb{R}_+$;

```
if arm a has been sampled then
    Extract history of samples H_a of arm a from H;
    return μ(H_a) + C(H);
else
    return +∞;
```

Please recall that $\mu : \mathcal{H} \to \mathbb{R}$ is the function that computes an empirical mean from a history of samples. We also write $\mu(H_a) = \mu_a$.

which the computation of the KL-UCB index can be expressed in a closed form solution, using a quadratic (Gaussian) KL, under the σ-subGaussian hypothesis (a bounded distribution is σ-subGaussian).

Exploiting the unlikelihood of optimality viewpoint of Theorem 3.1.5, we have the family of MED-like algorithms: MED, DMED and IMED [12, 16, 33], that are proven asymptotically optimal for various families $\mathcal{F}$ (e.g. bounded support, semi-bounded support with log-Laplace function defined in a neighborhood of zero), and directly exploit the lower bound of Theorem 3.34 in their structure. The lower bound state that the sampling rate $\frac{N_a(H)}{t}$ of suboptimal arms should be larger than the normalized probability $\exp(-N_a(H)\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star))$, the probability of action a to be optimal in the problem, given all the current information. Such a point of view naturally lead to the MED Algorithm 10. Another way of viewing this lower bound is to say that an algorithm should never sample an arm less than its unlikelihood of optimality measure. That is to say, one should always have that $\exp(-N_a(H)\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)) \leq \alpha(H)\frac{N_a(H)}{t}$, where

[12]: Honda et al. (2011), 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'
[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'
[33]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models.'

Algorithm 10: Distribution of the MED policy

Input: Arm $a \in \mathcal{A}$ of a bandit tuple $(\mathcal{A}, s, \mathcal{V})$;
A family $\mathcal{F}$ of distributions whose $\mathcal{V}$ is a subset;
History H of a sequential sampling of the bandit tuple;

if *arm a has been sampled* **then**
  Compute maximal empirical expected reward, $\mu^\star(H)$;
  Compute number of samples of arm a, $N_a(H)$;
  Extract history of samples H_a of arm a from H;
  return $-N_a(H)\mathcal{E}_{\mathcal{F}}(H_a, \mu^\star(H)) - \log N_a(H)$;
else
  return $+\infty$;

$\alpha(H) = \sum_a \exp\left(-N_a(H)\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)\right)$ is a normalization factor. Taking the logarithm, this gives two algorithms, DMED and IMED.

DMED 11 is round based and sample at each round all the arms that violate the constraint that we just expressed.

Algorithm 11: Computed set by the DMED policy

Input: Arm $a \in \mathcal{A}$ of a bandit tuple $(\mathcal{A}, s, \mathcal{V})$;
A family $\mathcal{F}$ of distributions whose $\mathcal{V}$ is a subset;
History H of a sequential sampling of the bandit tuple;

Compute number n of total interactions from history H;
return $\{a \in \mathcal{A} \mid N_a(H)\mathcal{E}_{\mathcal{F}}(H_a, \mu^\star(H)) \log N_a(H) \geq \log T\}$

IMED 12 is indexed based and sample the arm that is the closest to violate the constraint or violate the constraint the most. As we can see,

Algorithm 12: Index of the IMED policy

Input: Arm $a \in \mathcal{A}$ of a bandit tuple $(\mathcal{A}, s, \mathcal{V})$;
A family $\mathcal{F}$ of distributions whose $\mathcal{V}$ is a subset;
History H of a sequential sampling of the bandit tuple;

if *arm a has been sampled* **then**
  Compute maximal empirical expected reward, $\mu^\star(H)$;
  Compute number of samples of arm a, $N_a(H)$;
  Extract history of samples H_a of arm a from H;
  return $-N_a(H)\mathcal{E}_{\mathcal{F}}(H_a, \mu^\star(H)) - \log N_a(H)$;
else
  return $+\infty$;

both the MED and IMED algorithm do not explicitly refer to the number of interactions, which in my opinion, explain their superior numerical performances. Furthermore, the MED-like algorithms only require to compute the empirical $\mathcal{E}_{\mathcal{F}}$, making them faster than KL-UCB. Following our intuition, while $\mathcal{C}(H) = \log t$ in the original design of DMED, we think that it should be replaced by $\mathcal{C}(H) = \log \max_a N_a(H)$.

Alternative asymptotically optimal strategies include the Thompson Sampling (TS) [39, 40], which uses a Bayesian posterior distribution given

[39]: Thompson (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'
[40]: Agrawal et al. (2012), 'Analysis of Thompson Sampling for the Multi-armed Bandit Problem'

a specific prior, whose optimality was shown in [41]. See also [42] for other randomized algorithms and [31, 43, 44] for recent non-parametric extensions using re-sampling methods. In a recent paper [26], the NPTS algorithm was introduced which stands for Non-Parametric Thompson Sampling.

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

Algorithm 13: Distribution of the NPTS policy

Input: Arm $a \in \mathcal{A}$ of a bandit tuple $(\mathcal{A}, s, \mathcal{V})$;
A family $\mathcal{F}$ of distributions whose $\mathcal{V}$ is a subset;
An upper bound M on the supports of reward distributions;
History H of a sequential sampling of the bandit tuple;

if *arm a has been sampled* **then**
  Compute number of samples of arm a, $N_a(H)$;
  Extract history of samples H_a of arm a from H;
  Sample $N_a(H) + 1$ weights $(w_i)_i$ from Dirichlet distribution of order $N_a(H) + 1$ and parameters all equals to 1, where $i \in [1, N_a(H)]$ also index the elements in $H_a \cup \{M\}$;
  return $\sum_i w_i x_i$;
else
  return $+\infty$;

The main cost comes from the sampling of $n+1$ weights from a Dirichlet distribution of parameter $(1, \dots, 1)$ (n ones). This sampling is linear in n, since it can be done by sampling $n+1$ i.i.d. samples $R_1, \dots, R_{n+1}$ from the exponential distribution $\mathcal{E}(1)$, and then defining $w_i = \frac{R_i}{\sum_{j=1}^{n+1} R_j}$ for any $i \in [n+1]$. This justifies the cost in $\mathcal{O}(n)$ in the table.

Structured Bandits

In this thesis, we will be interested in a structured Bandit problem, Chapter 5, and a Reinforcement Learning problem that can be seen as a highly structured Bandit problem, Chapter 8. By structure, it is meant that there is an assumption that the expected reward function satisfy some sort of constraint, restricting the size of the set a given Bandit instance can belong too, compared to the unstructured case. This usually translate into a smaller lower bound on the logarithmic growth rate, *i.e.* there is information to be exploited that can improve the sample efficiency. However, deriving lower bound is not a trivial task in general and crafting algorithms that can exploit the structural knowledge an even harder task. Yet, several instances of structured bandits received considerable attention in the last few years. This is the case for instance of linear bandits, see [43, 45–47] and [48], Lipschitz bandits [49–51], unimodal bandits [52–54], or combinatorial bandits [55, 56], and more recently [57]. A generic asymptotically optimal algorithm, called OSSB (Optimal Structured Stochastic Bandit), has been introduced in the work of [58], and proven to be asymptotically optimal for all structures satisfying some weak properties that include all the aforementioned structures. Although being asymptotically optimal this algorithm often suffers from a long burn-in phase that may hinder its finite practical performance. It further comes with high computational price as it requires solving, at each step, an empirical version of the optimization problem defined by

[48]: Lattimore et al. (2017), 'The End of Optimism? An Asymptotic Analysis of Finite-Armed Linear Bandits'
[49]: Magureanu et al. (2014), 'Lipschitz Bandits: Regret Lower Bounds and Optimal Algorithms'
[50]: Wang et al. (2020), 'Towards Practical Lipschitz Bandits'
[51]: Lu et al. (2019), 'Optimal Algorithms for Lipschitz Bandits with Heavy-tailed Rewards'
[58]: Combes et al. (2017), 'Minimal exploration in structured stochastic bandits'

the lower bound on the logarithmic growth rate of regret. This motivates the quest for alternative strategies, perhaps less generic but better suited to a specific structure. Inspired by combinatorial structures for which computing $\mathfrak{C}_{\mathcal{D}}(\nu)$ is simply not feasible, a relaxation of the generic constrained optimization problem was recently proposed in [59]. The authors show that this comes at the price of trading-off regret optimality for computational efficiency. Indeed, in some structure, combinatorial properties are at stake and asymptotically optimal algorithms may require solving combinatorial optimization problems (see [59]) related to the optimization problem that define the regret lower bound. In order to exploit the combinatorial structures in a numerically efficient way, research has been made in how to relax these combinatorial optimization problems while preserving theoretical properties on the regret of the relaxed algorithms (see [57, 59]). In Chapter 5 of this thesis, we will work on such a structured Bandit problem and craft an algorithm that is based on a relaxation of the associated combinatorial Bandit problem, yet has a controlled regret. Similarly, in Chapter 8, we deal with a structured Bandit problem on the combinatorial space of deterministic stationary policies and derive assumptions and guarantees that are sufficient to craft an interesting problem and efficient algorithm.

[57]: Cuvelier et al. (2021), 'Statistically Efficient, Polynomial-Time Algorithms for Combinatorial Semi-Bandits'
[59]: Cuvelier et al. (2021), 'Asymptotically optimal strategies for combinatorial semi-bandits in polynomial time'

3.4 Summary of contributions

NeurIPS **2021**

Bandits with groups of similar arms

In the paper ***Stochastic bandits with groups of similar arms*** and published at NeurIPS 2021 with Hassan Saber, and Odalric-Ambrym Maillard, we consider a variant of the stochastic multi-armed bandit problem where arms are known to be organized into different groups having the same mean. The groups are unknown but a lower bound q on their size is known. This situation typically appears when each arm can be described with a list of categorical attributes, and the (unknown) mean reward function only depends on a subset of them, the others being redundant. In this case, q is linked naturally to the number of attributes considered redundant, and the number of categories of each attribute. For this structured problem of practical relevance, we first derive the asymptotic regret lower bound and corresponding constrained optimization problem. They reveal the achievable regret can be substantially reduced when compared to the unstructured setup, possibly by a factor q. However, solving exactly the exact constrained optimization problem involves a combinatorial problem. We introduce a lower-bound inspired strategy involving a computationally efficient relaxation that is based on a sorting mechanism. We further prove it achieves a lower bound close to the optimal one up to a controlled factor, and achieves an asymptotic regret q times smaller than the unstructured one. We believe this shows it is a valuable strategy for the practitioner. Last, we illustrate the performance of the considered strategy on numerical experiments involving a large number of arms.

NeurIPS **2023**

Approximation of the unlikelihood of optimality

In the paper ***Fast Asymptotically Optimal Algorithms for Non-Parametric Stochastic Bandits*** and published at NeurIPS 2023 with Dorian Baudry, Rémy Degenne and Odalric-Ambrym Maillard, we consider the problem of regret minimization in non-parametric stochastic bandits. When the rewards are known to be bounded from above, there exists asymptotically optimal algorithms, with asymptotic regret depending on an infimum of Kullback-Leibler divergences (KL). These algorithms are computationally expensive and require storing all past rewards, thus simpler but non-optimal algorithms are often used instead. We introduce several methods to approximate the infimum KL which reduce drastically the computational and memory costs of existing optimal algorithms, while keeping their regret guaranties. We apply our findings to design new variants of the MED and IMED algorithms, and demonstrate their interest with extensive numerical simulations.

Towards an optimal information usage

4

In practice, the choice of a bandit algorithm may be motivated by its theoretical guarantees, but also by its computation and memory costs. This section try to answer what can be said about this topic.

We set the scene. Consider the problem of regret minimization in non-parametric stochastic bandits. When the rewards are known to be bounded from above, there exists asymptotically optimal algorithms, with asymptotic regret depending on an infimum of Kullback-Leibler divergences. These algorithms are computationally expensive and require storing all past rewards, thus simpler but non-optimal algorithms are often used instead. In this section we wonder whether one can efficiently approximate the infimum KL and reduce the computational and memory burden of existing algorithms while maintaining this desirable optimality property, *i.e.* the optimal regret upper bound.

This part is based on a paper, *Fast Asymptotically Optimal Algorithms for Non-Parametric Stochastic Bandits*, that was written in 2023 with Dorian Baudry, Rémy Degenne, and Odalric-Ambrym Maillard. It should be mentioned that I had the chance to briefly expose my initial ideas to Junya Honda when he came to visit the laboratory, and who was kind enough to listen to me. When I briefed him about my experimental results and my lack of theoretical ones, he told me that it was likely that my policy was too greedy and would probably lack theoretical guarantees. While it might not be true, it helped me reconsider my original idea and, in the end, led me to talk about this Odalric-Ambrym who kindly encourages me to start a collaboration with Dorian, and then Rémy. This whole project could not have been what it is without Rémy's idea to use portfolio's algorithms. I hope that the questions that are raised in this section concerning portfolio's algorithm will be researched and answered by the community as it seems a very important topic to me.

4.1 Space, time, & sample complexities

At the dawn of this project is me pondering over the following fact. To compute the next action to play using the IMED algorithm, it is necessary to solve $|A|$ optimization problems using the samples collected for each arm. As the number of samples for each arm grows, so is the numerical complexity of picking the next arm to play. Therefore, the more information, *i.e.* the more samples, we have, the more we must compute to pick the next arm. While we know that the strategy will almost surely always play an action of maximal empirical mean after a large enough number of interaction, the computational complexity to choose that same arm again and again is increasing. The complexity increases each time we pick an empirically suboptimal arm. The same can be said about the KL-UCB algorithm. On the other hand, a strategy like UCB does not have this behavior. The computational complexity to pick the next action to play is a constant function of the number of interaction. At the

extreme, a strategy like ETC, has a constant complexity until a threshold of interactions is reached and then has virtually no computational cost, albeit some memory lookup is always necessary. However, UCB is not an **optimal** strategy. ETC even less. Thus, the question, is the computational cost of known optimal strategy a **necessary cost** to pay in order to be optimal? Can we do better? **Is it possible to be more sample efficient in the sense that less computation is done with each sample while optimally is still guaranteed?**

Of course, **time complexity**, or computational complexity, is not the only complexity that matters in the design of sequential learning algorithms. **Space complexity** also is important. The core question being, how much should an algorithm memorize from its past interactions? Is it necessary to store the history of all rewards to remain optimal? Surely this space complexity question is linked to the concept of sufficient statistics, *i.e.* how much can we compress without loss the information that is contained within a sequence of samples. While it is linked, the two topic are different because the Bandit objective is not to learn the problem, *i.e.* the distributions, it is to solve the problem. Nonetheless, it is true that if we assume some constraints on the set of considered distributions such that a sufficient of low dimension statistics exist and is easily updatable within a scheme akin to a Markov Chain Monte Carlo method, one can have optimal Bandit algorithm with a small space and time complexity.

We recall that an optimal algorithm achieves a logarithmic regret with a constant of $\sum_{a:\Delta_a>0} \frac{\mu^\star - \mu(a)}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$.

State of affairs Optimal algorithms are quite costly: IMED/MED [16, 33] and KL-UCB [21] need to compute an empirical infimum, $\mathcal{E}_{\mathcal{F}}$, that takes the form of an optimization problem. This computation needs to be done at each interaction.

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'
[33]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models.'
[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'

For some parametric families (e.g. Gaussian, Bernoulli), $\mathcal{E}_{\mathcal{F}}$ has a convenient closed-form expression. One can leverage this knowledge to greatly reduce the time and space complexity required to compute the sequence of actions. However, for more general non-parametric families of distributions this may not be the case. In this section we consider the most famous example of such non-parametric model, where the learner only knows that the distributions are bounded, *i.e.* the support of any distribution is bounded in $[m, M]$. All the used algorithm in this section will use the knowledge of the value of the upper bound M. However, the knowledge of the value of the lower bound m on the support will not be used by all algorithms. In this section, the non-parametric family $\mathcal{F}_{[m,M]}$ that the Bandit distribution belongs to is defined as

$$\mathcal{F}_{[m,M]} = \{F \in \mathcal{P}(\mathbb{R}) : \ \mathrm{supp}(F) \subseteq [m, M] \subset \mathbb{R}; m < M\}, \tag{4.1}$$

and we will denote it simply $\mathcal{F}$ without explicit mention to the support. Usually, the lower bound will be $m = 0$ and the upper bound $M = 1$. Doing so prove to simplify the notations, especially in the kl-UCB algorithm because it allows to confuse the (empirical) expected value with a Bernoulli parameter. On the other hand, it somehow adds a bit of confusion because one should not use values with KL divergences that are really about distributions. Instead, we prefer to denote $p(\mu)$, or $p(F)$ the "Bernoulli projection" of $F \in \mathcal{F}$, with $p(\mu) = \frac{\mu - m}{M - m}$. It should be noted that all those families are, from a bijection point of view, all the

same since one can map $\mathcal{F}_{[m,M]}$ to $\mathcal{F}_{[m',M']}$ using a simple full rank linear transformation.

For this family $\mathcal{F}$, computing the empirical infimum $\mathcal{E}_{\mathcal{F}}$ costs $\mathcal{O}(n \log(n))$ when n sampled have been collected, and when solved with precision $1/n$. Among those optimal algorithms, some are better off than others when it comes to the numerical complexity. The KL-UCB algorithm suffer a larger complexity than IMED because for one arm, several $\mathcal{E}_{\mathcal{F}}$ must be computed at each round. On the other hand, the cost of computing an index in the NPTS [26] algorithm is a linear function of the number of samples once the weights used by the algorithm have been sampled from a Dirichlet distribution. Sampling the weights using an acceptance-rejection algorithm is also a linear function of the number of samples. In all cases, the complexity costs, space and time, increase with the number of interaction. This is one of the reason that cheaper suboptimal alternatives are often enough considered in place of optimal algorithms.[1] For instance, UCB [5] requires a memory that is independent of the number of interaction and a time complexity that is constant per time step, *i.e.* independent of the number of interaction. It is however suboptimal in the sense that it achieves logarithmic regret but with a multiplicative constant of $\mathcal{O}\left(\sum_{a:\mu^\star-\mu(a)>0} \left(\mu^\star - \mu(a)\right)^{-1}\right)$. More generally, all the algorithms designed for $1/4$-sub-gaussian distributions can be used on $\mathcal{F}$ if the rewards are rescaled in $[0,1]$. A finer approximation consists in using the KL divergence of Bernoulli distributions, denoted kl (lowercase), that lower bounds $\mathcal{E}_{\mathcal{F}}$ [21]. We note that some of these approximations are sensitive to the value (and knowledge) of the lower bound of the support m, contrarily to asymptotically optimal algorithms. Recall that IMED/MED algorithms only assume an upper bound on the support of the distributions which explains this absence of sensitivity when assuming a bounded support. It is interesting that optimal algorithm does not exploit the knowledge of the lower bound when other concentration assumptions are made or deduced about distributions. We recall that by analyzing the dual problem $\mathcal{E}_{\mathcal{F}}$ for upper bounded distributions, Honda and Takemura [33] obtained that

$$\forall F \in \mathcal{F},\ \mu \leq M \ :\quad \mathcal{E}_{\mathcal{F}}(F,\mu) = \max_{\lambda \in \left[0, \frac{1}{M-\mu}\right]} \mathbb{E}_{X\sim F}\left[\log\left(1 - \nu\left(X-\mu\right)\right)\right] \ ,$$

and we will denote $D\left(\nu; X, \mu\right) = \mathbb{E}_{X\sim F}\left[\log\left(1-\nu\left(X-\mu\right)\right)\right]$ for X of law $F \in \mathcal{F}$.

Before presenting the work done in the paper this section is based on, we briefly introduce the idea on which this project was initially based on.

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

1: Another reason is the (apparent) complexity of implementation of some optimal algorithms compared to suboptimal one. Another is that *numerical* optimality (in some benchmark of interest for the practitioner) is sometimes different from *theoretical* optimality, especially in finite time.

[5]: Auer et al. (2002), 'Finite-time analysis of the multiarmed bandit problem'

[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'

[33]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models.'

First idea

The purpose was to find a way to compute the $\mathcal{E}_{\mathcal{F}}$ that is used in the IMED algorithm using an iterative scheme, akin to gradient descent. The benefits would be a constant memory need per arm and a time complexity that is independent of the number of interaction. Because the $\mathcal{E}_{\mathcal{F}}$ is basically, the maximization of a concave function that is written as an expectation, a MCMC algorithmic scheme seemed like the perfect mathematical tool. The trick is that the expected value is computed using

a random variable but also depends on another random variable. Those two random variables might not even be estimated at the same speed.

Specifically, one is interested in maximizing in ν, and under some constraints, the quantity

$$D(\nu; X, \mu) = \mathbb{E}_X(\log(1 - (X - \mu)\nu)).$$

We know that μ corresponds to an expected value so that we can in fact write the function to study as

$$D(\nu; X, X^\star) = \mathbb{E}_X\left(\log\left(1 - \left(X - \mathbb{E}_{X^\star}\left(X^\star\right)\right)\nu\right)\right). \tag{4.2}$$

In practice, both X and $X^\star$, thus μ, will be empirically estimated quantities, but not with the same precision, *i.e.* number of samples. $X^\star$ should correspond to a distribution that has been sampled a linear amount of the interactions, while X might have been sampled as little as a logarithmic fraction of interactions. At least at a high level, it is safe to assume that the number of samples associated to $X^\star$ is always larger than the number of samples associated to X. If it is not, one could always envision a way of forcing it, either by sampling the empirical best arm upon detection that it is not the most sampled, or by considering the concept of *leader arm*, the arm that has been sampled the most, instead of empirical best arm. As a matter of fact, such a remark inspired the final design of the algorithms of the paper, but more about that later.

The nested expected values formulation of Equation 4.2 emphasizes the fact that the outer expectation depends on another expectation. However, the inner expectation should not hinder the quality of the outer expectation, especially in the asymptotic regime, and even in the finite time regime if we consider the outer random variable to always have fewer samples than the inner random variable. This should be the case by construction rule of the algorithm. However, it is always the case that, at a lower level, things are not so obvious in the finite time regime. Nonetheless, knowing that one must maximize the outer expectation given the inner, this formulation call for a *double online optimization problem formulation*. First, the inner expectation is estimated. Second, one use a sequential Optimality algorithm to solve the outer problem. This somehow indicates that the outer online scheme should have a larger time step than the inner. We should probably do so in order to give the outer product to adapt to fluctuation of the estimated inner expected value. If the inner value converges faster than the outer expectation, then the problem is almost like a classic MCMC problem. We later call this situation a *quasi-static* optimization problem,[2] that is when the inner problem is almost constant compared to the outer problem, where constant is measured with respect to the size of the learning rates.

2: We borrow this term to physics because it conveys the intuition of the inner and outer dynamics. In physics, a quasi-static process is also known as a quasi-equilibrium process.

A double online optimization problem Clarifying our goal, recall that, at the t^{th} interaction, the IMED algorithm computes for each arm $a \in A$,

$$\mathcal{E}_{\mathcal{F}}\left(\hat{X}_a(t), \hat{\mu}^\star(t)\right) = \sup_{0 \leq \nu \leq \frac{1}{1-\hat{\mu}^\star(t)}} \mathbb{E}_{\hat{X}_a(t)}\left(\log\left(1 - \left(\hat{X}_a - \hat{\mu}^\star(t)\right)\nu\right)\right),$$

that is to say,

$$\mathcal{E}_{\mathcal{F}}\left(\hat{X}_a(t), \hat{X}^\star(t)\right) = \sup_{0 \leq \nu \leq \frac{1}{1-\mathbb{E}\left(\hat{X}^\star(t)\right)}} \mathbb{E}_{\hat{X}_a(t)}\left(\log\left(1 - \left(\hat{X}_a - \mathbb{E}\left(\hat{X}^\star(t)\right)\right)\nu\right)\right),$$

where $\hat{X}_a$ denotes the empirical distribution of arm a and $\hat{X}^\star$ is the empirical distribution of an arm with maximal empirical mean. In the IMED algorithm, those quantities are computed for all arms at every interaction. However, once the number of interaction is large, $\hat{\mu}^\star(t)$ should not change much because a best arm should have been sampled a linear amount of time and its random fluctuation would be controlled by a concentration hypothesis on the considered set of distributions. As written above, it would make sense to introduce the concept of *leader arm*, the arm that has been sampled the most. On one hand, one can try to prove that the empirical best arm has been sampled a linear fraction of the interaction and therefore prove that concentration inequalities can be applied. On the other hand, one can try to prove that the leader arm[3], on which we know that we can apply concentration inequalities, is indeed a best arm. Similarly to what is done in the paper of Honda and Takemura [16], we choose $\hat{\mu}^\star$ to be the empirical best mean.

3: The leader arm has been sampled more than $\frac{n}{|A|}$ times after n^{th} interactions, *i.e.* a linear function of the number of interactions.

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

The algorithmic idea is the following. The inner expected value is simply the empirical expected value of the optimal random variable and, for a suboptimal arm, does not depend on the value of the outer expected value. Whenever the empirical best arm is sampled, the empirical best mean will be updated using a stochastic gradient descent, using the usual learning rate of $\frac{1}{n}$ where n is the number of samples.[4] Whenever a suboptimal arm is sampled, the supremum that we want to compute will be updated using a stochastic gradient ascent scheme. The sampled gradient will be using the latest estimate of the maximal empirical mean. Because the estimated gradient depend on an inner random variable, the empirical best mean, that is independent of the outer random variable, used for the stochastic update of the $\mathcal{E}_{\mathcal{F}}$, we have to mitigate the fluctuation of the outer gradient with respect to the noise induced by the inner expected value. To do so, we choose the learning rate of the outer problem to be larger than that of the inner problem. Namely, the leaning rate will be $\frac{\log n}{n}$, where n is the number of samples. The intuition is that the outer and inner learning rates should be set so that the dependence of the outer problem on the fluctuation of the inner problem is almost removed. That is to say, the inner problem should be *quasi-static* with respect to the outer problem. This way, even at a small number of interactions, when the number of samples are of the same order of magnitude, the learning rate of the outer scheme is always larger than the learning rate of the inner scheme (but we should make sure that the empirical best is the most sampled). In other words, if the magnitude of the gradient is controlled, the fluctuations of the inner random variable should be small (first order Taylor approximation) compared to the random fluctuation induced by the outer random variable and its larger learning rate. In the long run, this effect would be magnified by the fact that the number of samples of the inner and outer random variables should be exponentially different.

4: The usual online update of the empirical mean can be seen as a stochastic gradient descent of the function $\mu \mapsto \mathbb{E}_X (X - \mu)^2$, where the used leaning step should satisfy Robins-Monro conditions.

The algorithm proceeded as follows. At each interaction t, an arm $a_t \in \arg\min_a I_a(t)$ is sampled, where I_a is the IMED-like index of arm

a. Then, statistical quantities of the sampled arm, and only this arm, are updated using the information extracted from the interaction. If the sampled arm a_t is suboptimal, then the number of samples N_{a_t}, the invert learning rate[5] α_{a_t}, the empirical mean $\hat{\mu}_{a_t}$, the empirical optimal parameter of the outer problem ν_{a_t}, the empirical unlikelihood of optimality $\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)$, K_{a_t}, and the index I_{a_t} are updated according the following rules:

5: I think that its helps to see the invert learning rate as a form of pseudo-count.

$$N_{a_t}(t+1) = N_{a_t}(t) + 1 \tag{4.3}$$

$$\alpha_{a_t}(t+1) = \frac{\log(2) N_{a_t}(t+1)}{\log(N_{a_t}(t+1)+1)}, \tag{4.4}$$

$$\hat{\mu}_{a_t}(t+1) = \hat{\mu}_{a_t}(t) + \frac{1}{N_{a_t}}\left(X_t - \hat{\mu}_{a_t}\right), \tag{4.5}$$

$$\nu_{a_t}(t+1) = \nu_{a_t}(t) + \frac{1}{\alpha_{a_t}} \frac{\hat{\mu}^\star(t) - X_t}{1 - (X_t - \hat{\mu}^\star(t))\,\nu_{a_t}(t)}, \tag{4.6}$$

$$K_{a_t}(t+1) = K_{a_t}(t) + \frac{1}{\alpha_{a_t}}\left(\log\left(1 - \left(X_t - \hat{\mu}^\star(t)\right)\nu_{a_t}\right) - K_{a_t}(t)\right), \tag{4.7}$$

$$I_{a_t}(t+1) = \alpha_{a_t} K_{a_t} + \log\left(N_{a_t}\right). \tag{4.8}$$

For the empirical computation of ν, a projection step to ensure the constraints $0 \leq \nu \leq \frac{1}{1-\mu^\star}$ is necessary. In the future, it could be good idea to investigate other numerical methods such as the **Frank–Wolfe algorithm** which has the advantage of computing a sequence of parameters that stays in the feasible set by construction.

If the sampled arm a_t is the empirically optimal one, then, one only need to update its number of samples, and more importantly, update its empirical mean using the usual learning rate of $\frac{1}{N_\star(t)}$, *i.e.* the relevant quantities are

$$N_{a_t}(t+1) = N_{a_t}(t) + 1 \tag{4.9}$$

$$\alpha_{a_t}(t+1) = \frac{\log(2) N_{a_t}(t+1)}{\log(N_{a_t}(t+1)+1)}, \tag{4.10}$$

$$\hat{\mu}_{a_t}(t+1) = \hat{\mu}_{a_t}(t) + \frac{1}{N_{a_t}}\left(X_t - \hat{\mu}_{a_t}\right), \tag{4.11}$$

$$I_{a_t}(t+1) = \log\left(N_{a_t}\right). \tag{4.12}$$

Interestingly, based on the intuition of the different terms in the IMED index, I am convinced that different (pseudo)-counts should be used for the term in front of the $\mathcal{E}_{\mathcal{F}}$ approximation and within the log. The count within the log accounts for the true frequency at which the agent interacted with the arm in the history. The IMED index allows to compare this true frequency of play with a term akin to a likelihood of optimality, $N_a \mathcal{E}_{\mathcal{F}}\left(\hat{X}_a, \hat{\mu}^\star\right)$. Intuitively, the frequency of play $\frac{N_a(t)}{t}$ should such that it is roughly equal to $\exp\left(-N_a \mathcal{E}_{\mathcal{F}}\left(\hat{X}_a, \hat{\mu}^\star\right)\right)$ normalized so that all those exponential terms sum to one across all arms. In the original IMED (and MED) algorithm, this likelihood of optimality depends on the amount of interactions. In our case, more uncertainty about the empirical $\mathcal{E}_{\mathcal{F}}$ is due to the stochastic gradient scheme. A good proxy for the total number of interactions is therefore the pseudo-count α_a. Furthermore, one can even think about it in terms of homogeneity. The update rule of K_a, the

empirical $\mathcal{E}_{\mathcal{F}}$, makes appear the learning rate $\frac{1}{\alpha_a}$, and it is homogenous to a likelihood per pseudo-interaction. To recover an expression with the good unit, a likelihood, one should use the pseudo-count as a multiplicative factor. Because $N_a(t) < \alpha_a(t)$, it means that compared to the original IMED index with the same approximation of the $\mathcal{E}_{\mathcal{F}}$, a suboptimal arm is considered closer to be under-sampled. The presented scheme therefore favor a bit more exploration than the original IMED algorithm when approximations of $\mathcal{E}_{\mathcal{F}}$ are the same. This is perfectly coherent and aligned with the additional noise induced by the sequential gradient ascent scheme.

What can be said about such an algorithm?

Complexities of the first intuition The above algorithm, if proven correct, solve the problems that were mentioned at the beginning of this section. The space complexity is reduced to a $\mathcal{O}(1)$, *i.e.* it only needs a constant memory for each arm and that memory space does not depend on the number of interactions. The time complexity is reduced to a $\mathcal{O}(1)$ per interaction, *i.e.* it only needs a constant time per interaction to update the statistical quantities. Since the gradient of the outer function has been computed by hand, the updates are really a finite amount of evaluations of functions. Several experiments were run with this algorithm and its performances where always good. Its regret was always sublinear and when plotting it with logarithmic scale, does seem logarithmic in the number of interactions. Compared to IMED, it was sometimes better and sometimes a bit worse depending on the setting. The initial slope always seemed a bit higher which is easily explained by the enforced slightly over exploration that is induced by the pseudo-count scheme and potentially, the stochastic noise induced by the scheme. Overall, the algorithm works well. From a theoretical standpoint, one can hope that the algorithm is indeed optimal. If we take the pseudo-count in the frequency of sample that we chose, and assume that K_a is a good approximation of $\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)$, then we have that

$$\frac{N_a(t)}{t} \propto \exp\left(-\frac{N_a(t)\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)}{\log N_a(t)}\right), \tag{4.13}$$

were the right-hand side should be compared to the original IMED one, $\exp\left(-N_a(t)\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)\right)$. It is not immediately obvious that this frequency of play will achieve the right constant that we target for asymptotic optimality. Looking carefully, we see that the Equation 4.13 means that

$$N_a(t) \simeq \exp\left(W_{-1}\left(-\frac{\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)}{\log t}\right)\right),$$

where W_{-1} is the real branch of the Lambert W function with negative argument. A first order expansion then shows that

$$N_a(t) \simeq \frac{\log t}{\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)}$$

which is the target quantity.

Unfortunately, despite a promising first approach and apparently theoretically tuned learning rates, when attempting to prove a regret upper bound in this algorithm, several attempts failed. I would be happy

is someone takes the torch of this line of work and manage to prove theoretical guarantees. During my PhD, I could not.

It is time to try another approach, surely less greedy in the desired results.

Cost/performance trade-off To start with, we lay down in Table 4.1 the state of play. We consider optimal and common suboptimal algorithms (*Algorithm* column) for the unstructured bandit problem (A, β). The general assumptions would be those assumed in the paper [16] of Honda and Takemura, with upper bounded support of rewards and existence of the log-Laplace transform in a neighborhood of zero. However, some suboptimal algorithms need stronger assumptions, in particular, bounded reward support, *i.e.* bounded from above but also from below. As stated above, this is our chosen assumption for this section. For each of the considered algorithm π, a regret upper bound can be written as

$$\mathscr{R}_{(A,\beta)}(T;\pi) \leq \sum_{\substack{a\in A \\ \mu^\star-\mu(a)>0}} \frac{\mu^\star-\mu(a)}{\mathscr{C}_{(A,\beta)}(a;\pi)} \log T + o(\log T),$$

where $\mathscr{C}_{(A,\beta)}(a;\pi)^{-1}$ is interpreted as the necessary logarithmic sample complexity rate of arm a in Bandit (A, β) to separate it from an optimal arm when using the policy π.

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

Figure 4.1: Pierre-Simon Marquis de Laplace (1749-1827)

The larger this quantity, the larger the number of pulls of suboptimal action a. The smaller, the easier it is for policy π to separate this action from the optimal set and thus, the smaller the number of associated pulls of suboptimal arm a. We highlight the fact that the regret upper bound, but also the regret lower bound, is written as a sum over all suboptimal actions and that the logarithmic sample rates are independent of each others because of the unstructured assumption on the Bandit problem. Therefore, in the following table, we talk about the sample rate of a suboptimal arm in a given Bandit problem, without mentioning its absent dependency to other suboptimal arms. The optimal sample rate is given by the infimum Kullback-Leibler, $\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$. This rate corresponds to the *Constant* column in Table 4.1. In the *Optimality* column, we report the optimal nature of the algorithm, that can be seen from its suboptimal sampling rates. All the presented algorithm compute, at each interaction and for each suboptimal arm, one quantity. All the computed quantities are then used to compute the next action. The cost of choosing the next action once the quantities have been computed is $\mathcal{O}(|A|)$, *e.g.* computing an argmin for IMED, an argmax for NPTS or KL-UCB, and sampling according a discrete distribution over $|A|$ categories for MED. To compare algorithm, we thus only report the time complexity of computing the quantity of interest of a suboptimal arm when the number of sample associated to that arm is n, *Time complexity* column. We also report in the *Space complexity* column, the required memory to store the necessary information of a suboptimal arm after n interactions with that arm. Please note that, in the following Table 4.1, n corresponds to the number of samples of a suboptimal arm of algorithms having a logarithmic regret upper bound. Thus, as an order of magnitude, one can always think that $n \sim \log T$, where T is the total number of interactions.

Table 4.1: Time complexity and space complexity needed to compute the index of a suboptimal arm a after n samples have been collected from its associated distribution $\beta(a)$.

Algorithm	Time complexity	Space complexity	Constant	Optimality
KL-UCB [21]	$\mathcal{O}(n\log(n)^2)$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
kl-UCB [21]	$\mathcal{O}(\log(n))$	$\mathcal{O}(1)$	$\frac{1}{kl(p(a),p^\star)}$	Sub-opt.
UCB [5]	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\frac{1}{2(\mu^\star-\mu(a))^2}$	Sub-opt.
NPTS [26]	$\mathcal{O}(n)$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
MED [33]	$\mathcal{O}(n\log(n))$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
IMED [16]	$\mathcal{O}(n\log(n))$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.

The algorithm kl-UCB need a bounded reward assumption while the UCB algorithm need a sub-Gaussian reward assumption. The sub-Gaussian property of a distribution is guaranteed if it is bounded. Optimal algorithms work with the stronger bounded assumption reward but do not make use of the lower bound on the support of reward. To compare algorithms, we need to assume the same set of hypothesis on the set of considered distributions, $\mathcal{F}$. In the following comparison, we assume that the set $\mathcal{F}$ is the set of bounded distributions, whose support are in $[m, M]$, and whose expected value are in $[m, M[$, where $-\infty < m < M < \infty$. In the kl-UCB algorithm, the Kullback-Leibler divergence of Bernoulli distribution is used. We recall that $p(a) = \frac{\mu(a)-m}{M-m}$, the barycenter parameter that allows to write $\mu(a)$ as a convex combination, *i.e.* barycenter, of the two extremities, $\mu(a) = (1-p(a))\,m + p(a)M$. Furthermore, $p^\star = max_{a\in A}p(a)$. Often, rewards are assumed to be bounded in $[0, 1]$, and we have the numerical identity $p(a) = \mu(a)$ and the sampling rate is written using $kl\left(\mu(a), \mu^\star\right)$ which is equal numerically to $kl\left(p(a), p^\star\right)$. While this is not wrong, I think that it provides the wrong intuition since the Kullback-Leibler divergence is only concerned about the distribution and not the specific support of the distribution. Taking into account the support into consideration in the computation of the infimum Kullback-Leibler problem is precisely what makes the quantity $\mathcal{E}_{\mathcal{F}}$ support dependent. To avoid confusion and wrong intuition, a value of the support should not be used as the parameter of a Kullback-Leibler divergence. Writing the Bernoulli kl like this makes it already obvious that the suboptimal kl-UCB is in fact part of a family of algorithms where the bounded support is discretized into $\sigma \geq 2$ points, two of which are necessarily located on the boundary of the support. We elaborate on this discretization procedure later in the section. First we briefly compare the logarithmic sampling rate of the algorithms in Table 4.1.

For simplicity, we assume that $m = 0$ and $M = 1$. All results hold for the general range $[m, M]$ up to a rescaling of the rewards,

$$\begin{aligned} p : [m, M] &\longrightarrow [0, 1] \\ x &\longmapsto \frac{x-m}{M-m} \end{aligned}$$

which really can be seen as a way of mapping the support space to the space of Bernoulli distributions. That is to say, a random variable in $\mathcal{F}_{[m,M]}$ with law F and expected value μ can be mapped to Bernoulli distribution of parameter $p\left(\mu\right)$. This is just projecting F to a binarized random variable with probability $p(\mu)$ associated to M and $1-p(\mu)$

associated to m. Except if F already is in this form, this transformation induce a loss of information. Anticipating a bit on what follows, one can already imagine what are the case in which the "loss of information" is the largest. It is when the original distribution has a density with respect to the Lebesgue measure and a bell shape-like curve centered at the middle of the support. The smaller the variance, the closer to a Dirac, the larger the loss of information. Real random variables whose support is included in $[0,1]$ are $\frac{1}{4}$ sub-Gaussian, which is the used proxy standard deviation that is used in the following.

Sub-optimality in Table 4.1 Consider $F \in \mathcal{F}$ of mean $\mu \in [0,1[$, representing the distribution and expected value of a suboptimal arm and $\mu^\star \in [\mu, 1[$ representing the expected value of an optimal arm. In this normalized case, the Bernoulli parameters are equal to the expected values, $p(\mu) = \mu$ and $p(\mu^\star) = \mu^\star$. The following result, known from [21],

[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'

$$\mathcal{E}_{\mathcal{F}}\left(F, \mu^\star\right) \geq kl\left(p(\mu), p\left(\mu^\star\right)\right) \geq 2\left(\mu^\star - \mu(a)\right)^2. \tag{4.14}$$

This inequality 4.14 prove that the asymptotic performances of algorithms in Table 4.1 can be compared. The UCB algorithm has the worst sample rate of suboptimal actions, followed by kl-UCB which itself is worse than optimal algorithms. In fact, those two inequalities could be the time to recall what a good logarithmic sample rate $\mathcal{C}_{\mathcal{F}}(F, \mu^\star)^{-1}$ is. For a distribution $F \in \mathcal{F}$ whose mean μ is smaller than $\mu^\star$, the function of interest should be such that $\mathcal{E}_{\mathcal{F}}(F, \mu^\star) \geq \mathcal{C}_{\mathcal{F}}(F, \mu^\star) > 0$. The first inequality with $\mathcal{E}_{\mathcal{F}}$ guarantees that a strategy with the $\mathcal{C}_{\mathcal{F}}^{-1}$ sampling rate is over-sampling suboptimal arms. It is therefore never greedier than the greedier a policy can be. On the other hand, its over-exploration cannot be infinite (inequality with zero) since we would suffer a linear regret in that case. Another condition is that this quantity should be equal to zero for an optimal distribution. An easy way to craft a function like this is to transform the distribution $F \in \mathcal{F}$ into another easier to handle distribution $T(F) \in \mathcal{D}$ and compute the infimum Kullback-Leibler divergence on the space $\mathcal{D}$ which is always smaller than the original one because of the **data processing inequality**. When the processing function T is lossy, the lost information makes it harder to distinguish between distributions and force the user of this information process to over explore compare to an agent willing to pay the cost of computing $\mathcal{E}_{\mathcal{F}}$ rather than $\mathcal{E}_{\mathcal{D}}$.

A natural question is to determine the magnitude of the gap in the inequalities presented in Equation 4.14. First, we illustrate this gap with an example that consider a problem of crop-management optimization in agriculture. The SCOOL team, the INRIA team I belonged to, owe quite a lot of this example to Odalric-Ambrym who made a praiseworthy effort to push members into using this crop-management problem and other environment related distributions. In Figure 4.2, we represent seven distributions of crop yields generated from the DSSAT simulator [60], corresponding to different crop-management policies. Those seven distributions constitute what we call the DSSAT problem in this thesis. The abscissa is indexed by the yield (a reward) of an agro-policy in a given agricultural environment. A sample of an agro-policy (an arm) is generated after simulating the yield of such a policy on randomly generated environmental conditions, *e.g.* temperature and humidity.

[60]: Hoogenboom et al. (2019), 'The DSSAT crop modeling ecosystem'

DSSAT: **D**ecision **S**upport **S**ystem for **A**grotechnology **T**ransfer.

The distributions are naturally bounded and non-parametric, due to the physical constraints of the problem. This plot visually confirms the

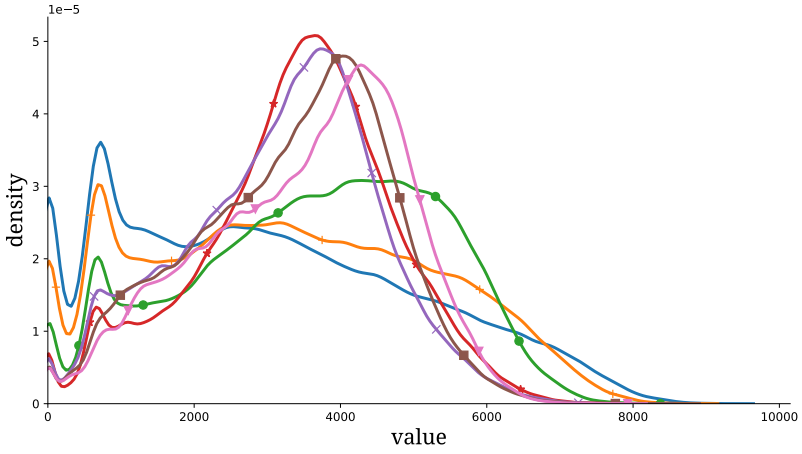

Figure 4.2: Distributions of the DSSAT Bandit problem

multimodality of all distributions and the difficulty to fit this problem into a parametric framework.

In Table 4.2 we compare the evaluation of $\mathcal{E}_{\mathcal{F}}$ to kl and $(\mu - \mu^\star)^2$ for each arm, except the optimal one because all those functions evaluate to zero for the optimal arm. This Table 4.2 empirically confirm the

$\frac{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}{\mathrm{kl}(p(\mu(a)),p(\mu^\star))}$	4.39	5.22	12.04	10.19	11.73	11.04
$\frac{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}{2(\mu(a)-\mu^\star)^2}$	4.72	5.55	12.73	10.87	12.44	11.63

Table 4.2: List of all ratios for the DSSAT bandit problem

theoretical importance of using the true $\mathcal{E}_{\mathcal{F}}$ quantities over their usual relaxed versions. One can indeed see that the asymptotic over sampling of suboptimal is significant since suboptimal arms can be sampled as much as 12 times more than they should due to the loss of information in using another KL in the infimum problem. The inequalities 4.14 is verified since all the ratios are larger than one and, column by column, the lower line ($2(\mu - \mu^\star)^2$) elements are larger than the upper line (kl). One can even compute how much larger are the regret upper bound constant[6] corresponding to kl and $2(\mu - \mu^\star)^2$. For kl, the regret constant is 8.95 times larger than the optimal regret constant. For $2(\mu - \mu^\star)^2$, the regret is 9.48 times larger than the optimal regret constant. We include this problem in the experimental Section 4.5 to check the practical consequences of this computed asymptotic property.

6: multiplicative constant in front of the log.

We now turn to a more theoretical result proving our aforementioned intuition about the loss of information induced by the Bernoulli-projection, that a centered bell curve with very small variance, or simply, a Dirac, is the worst case scenario. The Lemma 4.1.1 demonstrates that the ratio between $\mathcal{E}_{\mathcal{F}}$ and kl can be arbitrarily large if the distribution $F \in \mathcal{F}$ is *far* from being Bernoulli. But first, let's focus on the inequality relating the Bernoulli kl and quadratic $2(\mu - \mu^\star)^2$ expressions of Equation 4.14. In the following we assume that Δ is small to perform some first-order approximations. To simplify the small analysis, which only serve the purpose of building and proving intuition, we consider a distribution $F \in \mathcal{F}$ of expected value μ and a number $\mu^\star$, akin to a maximal reward,

such that $p(\mu) = p(\mu^\star) - \Delta$ with $0 < Delta << 1$ very small. Then, a first order Taylor expansion shows that $kl\left(p(\mu), p(\mu^\star)\right) \approx \frac{\Delta^2}{p(\mu)(1-p(\mu))}$. Compare to the quadratic $2\Delta^2$, the kl better exploit the bounded domain knowledge and adapts to the true variance of the Bernoulli distribution, thus exploiting better the geometry of $\mathcal{F}$ (here, the bounded assumption). The sub-Gaussian approximation (of the Bernoulli kl) is tight only if $p(\mu)$ is close to $1/2$ but not near the boundaries of the support.

We now investigate the gap between $\mathcal{E}_\mathcal{F}$ and kl, and recall that the two quantities match for Bernoulli distributions.

Lemma 4.1.1 (Comparison between kl and $\mathcal{E}_\mathcal{F}$) *For $\mu \in (0,1)$ and $F \in \mathcal{F}_{[0,1]}$ of mean $\mu - \Delta$, the scaling of $\frac{\mathcal{E}_\mathcal{F}(F,\mu)}{\mathrm{kl}(\mu_F,\mu)}$ can be as large as $\frac{\mu}{\Delta}$, with the maximum achieved when F is the Dirac distribution with support $\mu - \Delta$.*

Proof. First, the Jensen inequality and the dual representation of $\mathcal{E}_\mathcal{F}$ provide that

$$\mathcal{E}_\mathcal{F}(F,\mu) \leq \log\left(1 + \frac{\Delta}{1-\mu}\right) = \mathcal{E}_\mathcal{F}(\delta_{\mu-\Delta},\mu) \ ,$$

where δ_x is the Dirac distribution whose unitary mass is located at x.

Dividing both sides by $\mathrm{kl}(\mu-\Delta,\mu)$ and using a first order approximation of the right-hand side fraction, we get that for small values of Δ,

$$\frac{\mathcal{E}_\mathcal{F}(\delta_{\mu-\Delta},\mu)}{\mathrm{kl}(\mu-\Delta,\mu)} \approx \left(\frac{\Delta}{1-\mu}\right) \times \left(\frac{\mu(1-\mu)}{\Delta^2}\right) = \frac{\mu}{\Delta} \underset{\Delta\to 0}{\to} +\infty \ .$$

□

We could argue that Dirac (or highly concentrated) distributions may be unlikely in most applications, however the use-case provided by the DSSAT distribution (Figure 4.2 and Table 4.2) shows that *large* ratios[7] $\mathcal{E}_\mathcal{F}/\mathrm{kl}$ may already be observed with more natural examples.

7: Suffering 10 times the regret we would if using an optimal strategy may be considered large, in our opinion.

Cost/performance trade-off: from Bernoulli to Multinomial We saw that mapping a distribution in $\mathcal{F}$ to a Bernoulli distribution was way to reduce the computational cost of choosing an action. Both the time and space complexity were positively affected by such a mapping. However, this simplification came at the cost of a loss of information and therefore a proven suboptimality. Bernoulli distributions are the simplest case of *discretization* or *multinomialization* of distributions in $\mathcal{F}$. I prefer to talk about *multinomialization* because this term really shows that we are acting at the level of the distribution and at the level of the support. We map a distribution from $\mathcal{F}$ to another distribution within a set $\mathcal{D}$ in order to ease the computation of the infimum Kullback-Leibler divergence problem, from $\mathcal{E}_\mathcal{F}$ to $\mathcal{E}_\mathcal{D}$. A multinomialization procedure was described in a recent work [26] inspired by a binarization procedure described in [40]. We detail this discretization procedure and explore the suboptimality induced by the processing of the random variables. In particular, we ponder over the dependency of the suboptimality with respect to the size of the discretization. Intuitively, if we multinomialize a distribution

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

[40]: Agrawal et al. (2012), 'Analysis of Thompson Sampling for the Multi-armed Bandit Problem'

on $p \geq 2$ ticks evenly separated in $[mnM]$, then the precision of the approximation of the $\mathcal{E}_{\mathcal{F}}$ should be related to $\frac{M-m}{p-1}$. Please note that $p-1$ is the number of intervals so that $\frac{M-m}{p-1}$ is the size of a sub-interval of the subdivision. Also, please remark that $p-1$ is the number of independent parameters, or degree of freedom in the multinomial model. The algorithm we describe now to multinomialize a distribution in $\mathcal{F}$ involve a transformation of the reward random variable but recall that we are mostly interested in creating a random variable with an easier to handle law and that the way the support and values intervene in the description is more of a convenient way to write the transformation.

The process described in [26] consists in *discretizing* the rewards: a reward random variable $X \in \mathcal{F}_{[m,M]}$ is transformed into a multinomial distribution $Y \in \mathcal{D} = \mathcal{M}_p\left(\{x_1 = m, \dots, x_p = M\}\right)$ on some finite subdivision $\sigma = \left\{x_1, \dots, x_p\right\}$ of the original support with Y satisfying the minimal constraint that $\mathbb{E}(Y|X, \sigma) = \mathbb{E}(X)$. The subdivision can be arbitrary, as long as the boundaries m and M belong to it, but we will more naturally use the regular subdivision with sub-interval of equal length. The constraint is always feasible because the original expected value belong to the interval, and therefore can be written as a convex combination of the points of the subdivision. In fact, as it is shown by the Bernoulli case, only the extremal points are necessary. Hence, with this transformation, the expectations of the arms are unchanged while the memory to store the past information is reduced to $\mathcal{O}(p)$ per arm and the computation time of approximation $\mathcal{E}_{\mathcal{M}_p}$ of the infimum KL $\mathcal{E}_{\mathcal{F}}$ for discretized distributions is proportional to M. The expected value constraint is not enough to uniquely determine the multinomialization scheme, except when $p = 2$. Whatever the subdivision of size p, the multiplicative constant, *i.e.* the logarithmic sampling rate, of the $\log(T)$ term in the regret upper bound can be arbitrarily large compared with the optimal one.

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

We consider an arbitrarily non-degenerate subdivision of size $p \in \mathbb{N}$ with the constraint $p \geq 2$, $\sigma = \{x_1 = m, \dots, x_p = M\}$. By non-degenerate, we mean that $m = x_1 < x_2 < \cdots < x_{p-1} < x_p = M$ where the inequalities are strict and boundary conditions satisfied. Let $X \in \mathcal{F}$ be reward distribution. The multinomialization process transforming $X \in \mathcal{F}$ into $Y \in \mathcal{M}_p(\sigma)$ work by generalizing the Bernoulli process. Each point $x \in [m, M]$ is seen as the (unique) barycenter of its two closest elements in the subdivision. The weight associated to the element $k \in \sigma$ corresponds to a weight of the probability density that is transferred from x to k. The way it is done in the work [26] is slightly different in the sense that the transformation on the collected reward is not deterministic (assigning a fraction of the weight to the closest elements of the subdivision) but stochastic. Given a computed barycenter weight, a Bernoulli distribution is sampled to determine on which of the two elements we should assign all the weight. This method computes the same random variable as the deterministic method when considering the true random variable X. When samples are collected sequentially, the equality of the parameters of the two constructed multinomial distributions are only equal in expectation. Mathematically, each element of the subdivision x_k are associated with a number $\hat{P}_k$ that count the number of time we observed the reward x_k after the transformation. Those counts are used to compute the empirical multinomial model where the probability associated to x_k is $\frac{\hat{P}_k}{\sum \hat{P}_k}$. In a sense, this is a justification to the stochastic way of doing things

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

because this method is akin to say that we only *observe* the rewards *after transformation* and that the observed reward should belong to σ. Upon observing the real reward and splitting the weights, our algorithm would have access to information it should not have. Formally, upon receiving a sample reward x such that $x_k < x < x_{k+1}$, we compute the barycenter weight $b = \frac{x-x_k}{x_{k+1}-x_k}$. Then, we sample $R = x_k \mathbb{1}\{Z = 0\} + x_{k+1}\mathbb{1}\{Z = 1\}$ where $Z \sim Ber(b)$ and increment the count of sampled $R \in \{x_k, x_{k+1}\}$ by one. If we $x \in \sigma$ then, this is a special case, and we increment the count of element x by one. If this procedure is used with all rewards collected, one can use a bandit algorithm calibrated for multinomial distributions in $\mathcal{M}_p(\sigma)$.

The true random variable Y can be simulated from simulations of X as follows,

1. Sample $x \sim X$,
2. If $x \in \sigma$, output x,
3. Else, compute barycenter parameter $b = \frac{x-x_k}{x_{k+1}-x_k}$ where x satisfies the two inequalities $x_k < x < x_{k+1}$,
4. Then sample $z \sim Ber(b)$,
5. Output $zx_k + (1-z)x_{k+1}$.

The expected value constraint is respected $\mathbb{E}(Y) = \mathbb{E}(X)$, meaning that the order of the arms do not change with this discretization. Furthermore, the transformed bandit problem becomes equivalent to a problem where the distributions would be $(Y_a)_{a\in A}$ where, for each arm $a \in A$, Y_a is the distribution of the discretized rewards with implicit subdivision σ.

The above analysis shows that we can use an asymptotically optimal bandit algorithm for Bandit problem with multinomial distributions family $\mathcal{M}_p(\sigma)$ on the support σ on the modified multinomialized Bandit problem and obtain a logarithmic regret with corresponding multiplicative multinomial constant, $\sum_{\substack{a\in A \\ \mu^star > \mu(a)}} \frac{\mu^\star - \mu(a)}{\mathcal{E}_{\mathcal{M}_p(\sigma)}(Y_a, \mu^\star)}$. The constant factor is optimal despite the transformation if the distributions are already multinomial distributions with the prescribed subdivision. We remark that, *a priori*, a different subdivision could be used for each arm. However, without additional information prior to starting the interaction, there is no reason to pick different subdivisions.

We close this section by discussing the performance that can be achieved with such discretization. Assume that the chosen division σ is regular and of size p, *i.e.* that the length of a sub-interval is constant, $x_{k+1} - x_k = \frac{M-m}{p-1}$ for all defined indexes. Using the data processing inequality and Lemma 4 of [21] we obtain that

[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'

$$\mathcal{E}_{\mathcal{F}}\left(\beta(a), \mu^\star\right) \geq \mathcal{E}_{\mathcal{M}_p(\sigma)}\left(Y_a, \mu^\star\right) \geq \mathcal{E}_{\mathcal{F}}\left(\beta(a), \mu^\star\right) - \frac{1}{p-1}\frac{M-m}{M-\mu^\star}$$

Hence, the multinomial approximation with regular subdivision can be good for large p. This makes perfect sense from an approximation theory point of view since $p-1$ is the number of free parameters to be fitted. We remark that this approximation can be especially good but quite as costly as the original problem if, for a chosen horizon T, p is chosen too close to $\frac{\log T}{\mathcal{E}_{\mathcal{F}}(\min \beta(a), \mu^\star)}$, a quantity that is unknown prior to the interaction (and only known asymptotically in the full experimental setting). However, because of the logarithmic scaling of the number of samples of suboptimal

arms, one can see that, in finite time, such an approximation with a reasonably large parameter p can be quite performant However, from an asymptotic optimality viewpoint, the story is of course quite different. If $\mathcal{E}_{\mathcal{F}}(F_k, \mu^\star) \leq \frac{1}{M-1} \times \frac{B-b}{B-\mu^\star}$ (which is possible since p is fixed without knowing the distributions) then the lower bound is vacuous. As a matter of fact, Lemma 4.1.1 can be extended to any non-degenerate discretization process. If a small, compared to $\frac{M-m}{p-1}$, suboptimality gap $\mu^\star - \mu(a)$ is assumed, one can make the ratio $\mathcal{E}_{\mathcal{F}}(\beta(a), \mu^\star)/\mathcal{E}_{\mathcal{M}_p^\sigma}(Y_a, \mu^\star)$ arbitrarily large. Therefore, while this approximation technique has some convenient properties, it is not entirely satisfying. This justifies our quest for alternatives that would preserve the asymptotic optimality of the algorithms.

Before moving to the next section, we complete the comparison table with the results from the above discretization analysis. The used notations are the same that we used until now, and we attribute this scheme to the work made in the paper introducing the NPTS algorithm [26]. We abuse the notation a bit and denote $\sigma(\beta(a))$ the multinomialization of $\beta(a)$ along the subdivision σ. The notation $\mathcal{E}_{\mathcal{M}_p(\sigma)}$ MED/IMED indicate the MED/IMED algorithm in which the $\mathcal{E}_{\mathcal{F}}$ has been replaced by the $\mathcal{E}_{\mathcal{M}_p(\sigma)}$ on the transformed distribution. We adopt the same convention for $\mathcal{E}_{\mathcal{M}_p(\sigma)}$-UCB. The term σ-NPTS refers to the NPTS algorithm where the rewards are being transformed along the subdivision σ.

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

Table 4.3: Time complexity and space complexity needed to compute the index of a suboptimal arm a after n samples have been collected from its associated distribution $\beta(a)$. Subdivision σ is non-degenerate of size p.

Algorithm	Time complexity	Space complexity	Constant	Optimality
KL-UCB [21]	$\mathcal{O}(n\log(n)^2)$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
kl-UCB [21]	$\mathcal{O}(\log(n)^2)$	$\mathcal{O}(1)$	$\frac{1}{kl(p(a),p^\star)}$	Sub-opt.
$\mathcal{E}_{\mathcal{M}_p(\sigma)}$-UCB [26]	$\mathcal{O}(p\log(n)^2)$	$\mathcal{O}(p)$	$\frac{1}{\mathcal{E}_{\mathcal{M}_p(\sigma)}(\sigma(\beta(a)),\mu^\star)}$	Sub-opt.
UCB [5]	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\frac{1}{2(\mu^\star-\mu(a))^2}$	Sub-opt.
NPTS [26]	$\mathcal{O}(n)$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
σ-NPTS [26]	$\mathcal{O}(p)$	$\mathcal{O}(p)$	$\frac{1}{\mathcal{E}_{\mathcal{M}_p(\sigma)}(\sigma(\beta(a)),\mu^\star)}$	Sub-opt.
$\mathcal{E}_{\mathcal{M}_p(\sigma)}$ MED/IMED [26]	$\mathcal{O}(p\log(n))$	$\mathcal{O}(p)$	$\frac{1}{\mathcal{E}_{\mathcal{M}_p(\sigma)}(\sigma(\beta(a)),\mu^\star)}$	Sub-opt.
MED [12]	$\mathcal{O}(n\log(n))$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.
IMED [16]	$\mathcal{O}(n\log(n))$	n	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$	Opt.

Outline and contribution The results presented in previous paragraphs motivate the search for novel computationally competitive and asymptotically optimal non-parametric bandit algorithms. We build on MED to propose two novel approaches that achieve this goal. By MED, it is meant the original MED algorithm presented in [12] as well as the DMED [33] and IMED [16] approaches that are all based on the computation of the $\mathcal{E}_{\mathcal{F}}$. In the work presented in this section we will be interested in computing numerically competitive approximations of the empirical $\mathcal{E}_{\mathcal{F}}$ that is used by the MED algorithms that would not hinder the optimal nature of the algorithm using this approximation. We first propose FMED (resp. FIMED) as a fast variant of MED (resp. IMED), that computes the empirical $\mathcal{E}_{\mathcal{F}}$ only for the arm that is pulled while a first-order Taylor expansion is used to compute an approximation of the empirical $\mathcal{E}_{\mathcal{F}}$ for

[12]: Honda et al. (2011), 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'

[33]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models.'

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

the other arms. This simple change translates to a considerable speed-up of the algorithms, as for large enough horizons the best empirical arm (for which $\mathcal{E}_{\mathcal{F}}$ is 0) is pulled most of the time, and preserves the theoretical guarantees of the two original algorithms. However, FMED and FIMED still require to store all rewards and to fully compute $\mathcal{E}_{\mathcal{F}}$, albeit such a computation occurs at an exponentially decreasing frequency. Armed with the willingness to tackle this issue, we study another approach, in which estimates of $\mathcal{E}_{\mathcal{F}}$ are computed using an *online portfolio selection* algorithm. We highlight a property that would guarantee that such an algorithm also keeps the guarantees of MED and IMED, while having much faster computation time and having a competitive space complexity of $\mathcal{O}(|A|^2)$, possibly lower.

The base intuition for this section is that, when working with a good enough approximation, **its is highly probable that approximation does not matter in the asymptotically long term**. In other words, when good, *i.e.* with some kind of convergence, approximations do not matter in the long term. In the context of this thesis, this work was motivated by the quest of finding a good way to maximize the rate of information acquisition by an optimal algorithm, *i.e.* reducing the computational complexity while preserving the amount of information acquired (as measured by Optimality criterion), effectively increasing the rate of information acquisition. Hopefully, the reader will find it comforting that it is not compulsory for an optimal algorithm to suffer an increasing numerical complexity per time step while the uncertainty about the solution of the problem is decreasing.

Without further ado, let's present the algorithms, their theoretical guarantees, and numerical performances.

4.2 From intuition to algorithms: Fast MED algorithms

We introduce, recall and clarify some necessary notation used in the remainder of this chapter. In the Bandit problem (A, β), A is the set of arms and $\beta :\rightarrow \mathcal{F}$ maps an arm index $a \in \mathcal{F}$, the space of real random variable with bounded support distribution in a prescribed interval. By a slight abuse of notation, $\beta(a)$ sometimes refers to the reward random variable of arm a as well as the law of that random variable. The empirical distribution of arm a after t interactions with the Bandit problem will be denoted $F_a(t) \in \mathcal{F}$. The maximal empirical reward computed after t interactions with the Bandit problem will be denoted $\mu^\star(t)$. The expected reward of a policy that always play arm a is denoted μ_a and the empirical average reward of such a policy is denoted $\mu_a(t)$. Coincidentally, it also corresponds numerically to the empirical mean of arm a $\mu(t; a)$ while the true expected reward of arm a is $\mu(a)$. Therefore, the maximal empirical reward $\mu^\star(t)$ also corresponds to the maximal of average reward a (stationary) policy can get on the empirical Bandit control problem. In the following, we propose fast variants of MED and IMED, that avoid computing $\mathcal{E}_{\mathcal{F}}(F_k(t), \mu^\star(t))$ for each arm and at each interaction.

On-access full update, off-access convex update: FMED & FIMED

Here is the idea. While an arm $a \in A$ is not sampled, the associated empirical distribution $F_a(t)$ is constant. This is the case because there are no structural assumptions so that one cannot transfer information from one arm to another. Therefore, the function $\mu \mapsto \mathcal{E}_{\mathcal{F}}(F_a(t), \mu)$ is unchanged between two interactions with the arm a. On the other hand, the value $\mu^\star(t)$ is likely to change between two interactions with arm a so that $t \mapsto \mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$ is not a constant function despite its first argument being constant. However, under good behavior of the algorithms, the fluctuations of $\mu^\star(t)$ between two interactions with a suboptimal arm a should be small enough. Therefore, the variations of the unlikelihood function of arm a, $t \mapsto N_a(t)\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$, should be relatively small and controlled between two interactions with arm a. The idea is then to use a first order Taylor expansion of $\mu \mapsto \mathcal{E}_{\mathcal{F}}(F_a(t), \mu)$ to compute a reliable enough approximation of $t \mapsto N_a(t)\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$ between two interactions with arm a. Upon such an interaction, one can update the empirical distribution, and compute the exact solution to the empirical $\mathcal{E}_{\mathcal{F}}$ problem.

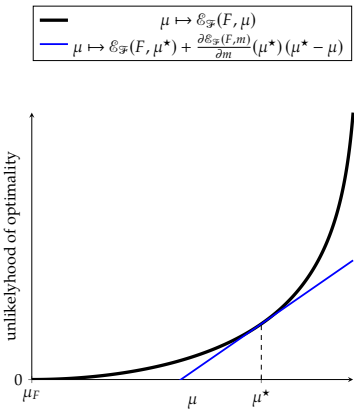

Figure 4.3: The function $\mathcal{E}_{\mathcal{F}}$ is continuous, non-decreasing and convex in its second argument (*i.e.* with fixed input distribution). Hence, the graph is above its tangent at all point. The plot x-axis is one such that $\mu \geq \mu_F$ because for smaller value the $\mathcal{E}_{\mathcal{F}}$ is by definition equal to zero. The plotted curve corresponds to a $\mathcal{E}_{\mathcal{F}}$ of a Bernoulli distribution, but this shape is generic.

The first reason this approximation could be good enough is that the $\mathcal{E}_{\mathcal{F}}$ function is a convex function of its second argument.

Lemma 4.2.1 (Properties of $\mathcal{E}_{\mathcal{F}}$) *Let $\mathcal{F}$ be the space of bounded distribution in $[m, M]$ and F a distribution in $\mathcal{F}$. Then, the function*

$$\mu \mapsto \mathcal{E}_{\mathcal{F}}(F, \mu)$$

is continuous, non-decreasing and convex on the whole domain $[m, M]$. It is also differentiable at all point of the open interval $]m, M[$. In other word, $\mathcal{E}_{\mathcal{F}}$ is continuous, differentiable, non-decreasing and convex in its second argument.

This Lemma 4.2.1 illustrated in Figure 4.3 where the $\mathcal{E}_{\mathcal{F}}$ function corresponding to Bernoulli distributions is plotted. It corresponds to the Theorem 7 of [33]. Thus, by using a first order Taylor expansion, one lower bound the true unlikelihood and therefore, the computed index prescribe to oversample arm a compared to what the original index would ask for. The second is that of that the fluctuations of $\mu^\star(t)$ around the mean should be of order $\frac{1}{\sqrt{t}}$. The number of samples of suboptimal arm a should be of order $\log t$. Therefore, the fluctuations of $t \mapsto N_a(t)\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$ should be of order

$$\max_{\mu \in [\mu^\star - \frac{1}{\sqrt{t}}; \mu^\star + \frac{1}{\sqrt{t}}]} \left| \frac{\partial \mathcal{E}_{\mathcal{F}}(F, m)}{\partial m}(\mu) \right| \times \frac{\log t}{\sqrt{t}},$$

a quantity that can be controlled and converges to zero as the number of interactions increases. The main difficulty of such an approximation scheme lies in the control of what happens before converge, *i.e.* when fluctuations of $\mu^\star(t)$ may be significant, for instance because there is a switch of empirical best arm. Also, the maximal value of the gradient might be quite large whenever $\mu^\star$ is close to the upper boundary M of allowed expected values. This is illustrated in Figure 4.3 where one

[33]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models.'

can see the divergence of the derivative of $\mathcal{E}_{\mathcal{F}}$ with respect to its second argument as μ is approaching the boundary of the allowed expected values.

Deriving the algorithm Let $F \in \mathcal{F}$ be a distribution of bounded support in $[m, M]$ and let $\mu, \mu' \in [m, M[$ be two numbers that represent two expected values. Later on, μ would be the maximal empirical reward at the last exact computation of $\mathcal{E}_{\mathcal{F}}$, $\mathcal{E}_{\mathcal{F}}(F, \mu)$, while μ' would be the current value of the maximal empirical reward. From Lemma 4.2.1, $\mathcal{E}_{\mathcal{F}}$ is non-decreasing and convex in its second argument, therefore,

$$\mathcal{E}_{\mathcal{F}}(F, \mu') \geq \mathcal{E}_{\mathcal{F}}(F, \mu) + \frac{\partial \mathcal{E}_{\mathcal{F}}}{\partial \mu}(F, \mu)(\mu' - \mu) \ . \tag{4.15}$$

Assuming that we can access (and store) the partial derivative of $\mathcal{E}_{\mathcal{F}}$ with respect to its second argument evaluated in μ, $\frac{\partial \mathcal{E}_{\mathcal{F}}}{\partial \mu}(F, \mu)$, then 4.15 is lower approximation of the $\mathcal{E}_{\mathcal{F}}$ that is useful when only the second argument is changing. The closer μ' is from μ, the better this approximation. Fortunately, a result from [16] shows that the partial derivative evaluated at μ is equal to the maximizer argument of the dual function of $\mathcal{E}_{\mathcal{F}}$. This quantity can therefore readily be stored at no additional computational cost and only constant memory cost. Using those facts, we propose FMED and FIMED, respectively **fast** variants of MED and IMED algorithms in which, for each arm $a \in A$, the quantity $\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$ is exactly computed right after a new sample from arm a has been collected, *i.e.* when $a_{t-1} = a$ and otherwise a Taylor approximation is used. The Taylor expansion base interaction point is the last time arm a was sampled. We denote by $s_a(t)$,

$$s_a(t) = \max\{s \leq t \mid a_s = a\} \ ,$$

the last time arm a was sampled prior to time t. As already said in this thesis, this way of phrasing is useful, but one should really think of storing a pointer to the last index in the history sequence that is associated with a sample of arm a. The set of index is ordered, so this operation is possible the same way it is done using natural numbers. In this spirit, recall that we read t as the total number of interactions. The derivative $\frac{\partial \mathcal{E}_{\mathcal{F}}}{\partial \mu}(F_a(t), \mu^\star(t))$ is denoted $\lambda_a(t)$ and is computed as the maximizer of the dual problem of the empirical $\mathcal{E}_{\mathcal{F}}$,

$$\lambda_a(t) = \underset{0 \leq \lambda \leq \frac{1}{M - \mu^\star(t)}}{\operatorname{argmax}} \ \mathbb{E}_{X \sim F_a(t)}\left(\log\left(1 - \left(X - \mu^\star(t)\right)\lambda\right)\right) .$$

Recall that, from [33], the infimum KL divergence $\mathcal{E}_{\mathcal{F}}$ can be computed from its dual representation as

$$\mathcal{E}_{\mathcal{F}}\left(F_a(t), \mu^\star(t)\right) = \max_{0 \leq \lambda \leq \frac{1}{M - \mu^\star(t)}} \mathbb{E}_{X \sim F_a(t)}\left(\log\left(1 - \left(X - \mu^\star(t)\right)\lambda\right)\right) . \tag{4.16}$$

The derivative of interest that we should store is $\lambda_a(s_a(t))$, that is to say, the value of the derivative that has been computed the last time arm a has been sampled. Using all of the above, one can say that the approximation that is used for the empirical $\mathcal{E}_{\mathcal{F}}$ of arm a at a time t that does not immediately succeed an interaction with arm a is the maximum

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

[33]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models.'

of zero and $\mathcal{L}_a(t)$, where

$$\mathcal{L}_a(t) = \mathcal{E}_{\mathcal{F}}\left(F_a\left(t\right), \mu^\star\left(s_a(t)\right)\right) + \lambda_a\left(s_a\left(t\right)\right)\left(\mu^\star\left(t\right) - \mu^\star\left(s_a\left(t\right)\right)\right). \quad (4.17)$$

This Equation 4.17 is exactly the formalization of the intuition we developed in the previous paragraph. We denote the complete expression of the approximation $\mathcal{K}_a(t)$ with

$$\mathcal{K}_a(t) = \max\left\{0, \mathcal{L}_a\left(t\right)\right\}. \quad (4.18)$$

The reason for this max is that $\mathcal{E}_{\mathcal{F}}$ is a positive quantity by construction. Therefore, one can only increase the quality of the approximation by projecting the approximation into the original set it is within. This constraint is not required for the algorithm to work as a negative approximation of the $\mathcal{E}_{\mathcal{F}}$ means that, "surely", this arm will be pulled and a complete computation of the empirical $\mathcal{E}_{\mathcal{F}}$ will then occur. Similarly, among all arms with zero valued approximation of $\mathcal{E}_{\mathcal{F}}$, the one with the smallest number of pulls is sampled, and "surely", it would be the suboptimal arm with this zero valued approximation. The two situations are algorithmically somewhat similar but using the max is theoretically more satisfying and sound.

Algorithms

We present in Algorithm 14 and Algorithm 15 the two fast update algorithms FIMED and FMED respectively.

Aggregating all of the above, we can express the index that is used by our two FMED and FIMED algorithms, $\mathcal{J}_a(t)$ as

$$\mathcal{J}_a(t) = \begin{cases} 0 & \text{if } N_a(t) = 0 \\ \mathcal{E}_{\mathcal{F}}\left(F_a(t), \mu^\star(t)\right) & \text{if } a_{t-1} = a \text{ (exact computation)} \\ \max\left\{0, \mathcal{L}_a\left(t\right)\right\} & \text{otherwise (Taylor expansion)} \end{cases} \quad (4.19)$$

where the *exact computation* is practically done, either at a fixed precision given the knowledge of the time horizon or with increasing precision proportional to $\frac{1}{n}$ where n is the number of samples collected from arm a. However, in the theoretical algorithm and proof of regret, we consider that those computations can be done exactly. At each interaction t, both FMED and FIMED will compute $\mathcal{J}_a(t)$ for all arm $a \in A$ and decide on the next arm to sample based on those information, the unlikelihood of optimality, and the numbers of samples $N_a(t)$. Please recall that $\mathcal{J}_a(t)$ is a useful compact notation to write $\mathcal{J}_a\left(H\right) = \mathcal{J}\left(a, H\right)$ where H is the whole history of all the t interactions. In the spirit of this thesis, we write the algorithm using the history notation and not the "time" notation as we think that no explicit mention of the "time" or total number of interaction should appear in the algorithm. However, in the forthcoming analysis of the regret, we do use the conventional notations that are used by the bandit community. Still, we hope that the reader will be convinced by the advantages of using the history notation.

Algorithm 14: FIMED: Fast IMED algorithm

Input: A bandit tuple $(\mathcal{A}, s, \mathcal{V})$ as in Definition 3.1.12;

The history H is a *multiset*. Similarly to m and λ, the data structure used to represent H numerically is a *dictionary* or *hash table*.

Initialize history H as $H = \emptyset$;
Initialize $k : A \to \mathbb{R}$ that will store the value of the last exact computation of empirical $\mathcal{E}_{\mathcal{F}}$ at the last interaction with arm a;
Initialize $m : A \to \mathbb{R}$ that will store the value of $\mu^\star$ at the last interaction with arm a;
Initialize $\lambda : A \to \mathbb{R}$ that will store the value of the derivative of $\mathcal{E}_{\mathcal{F}}$ evaluated at $\mu^\star$ at the last interaction with arm a;
for $t \in \mathbb{N}$ **do**
  forall $a \in \mathcal{A}$ **do**
    Compute index $J_a = N_a(H)\mathcal{J}(a, H) + \log N_a(H)$;
  Compute $a \in \operatorname{argmax}_{e \in \mathcal{A}} I_e$;
  Sample a reward $r \sim \beta(a)$ from distribution $\beta(a)$ of arm a;
  Update history, $H \leftarrow H \cup \{(a, r)\}$;
  Update maximal map, $m(a) \leftarrow \mu^\star(H)$;
  Update $\mathcal{E}_{\mathcal{F}}$ map, $k(a)$, computed from the exact empirical $\mathcal{E}_{\mathcal{F}}$ solution;
  Update derivative map, $\lambda(a)$, computed from the exact empirical $\mathcal{E}_{\mathcal{F}}$ solution;

Using notations of Algorithm 14 the fast update of the $\mathcal{E}_{\mathcal{F}}$ is written as

$$\mathcal{J}(a, H) = k(a) + \lambda(a)\left(\mu^\star(H) - m(a)\right),$$

where the fast update is used when an exact computation of the empirical $\mathcal{E}_{\mathcal{F}}$ is not done.

Algorithm 15: FMED: Fast MED algorithm

Input: A bandit tuple $(\mathcal{A}, s, \mathcal{V})$ as in Definition 3.1.12;

The history H is a *multiset*. Similarly to m and λ, the data structure used to represent H numerically is a *dictionary* or *hash table*.

Initialize history H as $H = \emptyset$;
Initialize $k : A \to \mathbb{R}$ that will store the value of the last exact computation of empirical $\mathcal{E}_{\mathcal{F}}$ at the last interaction with arm a;
Initialize $m : A \to \mathbb{R}$ that will store the value of $\mu^\star$ at the last interaction with arm a;
Initialize $\lambda : A \to \mathbb{R}$ that will store the value of the derivative of $\mathcal{E}_{\mathcal{F}}$ evaluated at $\mu^\star$ at the last interaction with arm a;
for $t \in \mathbb{N}$ **do**
  forall $a \in \mathcal{A}$ **do**
    Compute exponential weight $P_a = \exp\left(-N_a(H)\mathcal{J}(a, H)\right)$;
  Compute multinomial parameters $p_a = \frac{P_a}{\sum_{a \in A} P_a}$ and vectorized parameter $p = (p_a)_{a \in A}$;
  Sample $a \sim Mult(p)$, a multinomial distribution of parameter p;
  Sample a reward $r \sim \beta(a)$ from distribution $\beta(a)$ of arm a;
  Update history, $H \leftarrow H \cup \{(a, r)\}$;
  Update maximal map, $m(a) \leftarrow \mu^\star(H)$;
  Update $\mathcal{E}_{\mathcal{F}}$ map, $k(a)$, computed from the exact empirical $\mathcal{E}_{\mathcal{F}}$ solution;
  Update derivative map, $\lambda(a)$, computed from the exact empirical $\mathcal{E}_{\mathcal{F}}$ solution;

Despite its simplicity, this method approximating the empirical $\mathcal{E}_{\mathcal{F}}$ already permits a huge gain of computation time. Indeed, the exact computation of empirical $\mathcal{E}_{\mathcal{F}}$ only happens when a suboptimal arm is sampled, that is to say $\mathcal{O}(\log T)$ times.[8] Furthermore, whenever such a costly operation is performed, the computation is only made for one arm, the suboptimal one that has been sampled. Of course, this small analysis is anticipating in the regret upper bound that will confirm the fact that suboptimal arms are indeed sampled a $\mathcal{O}(\log T)$ of the total number T of interactions. Comparatively, in the IMED and MED algorithms, $|A| - 1$ computations of the empirical $\mathcal{E}_{\mathcal{F}}$ are required at each interaction, each of which costing $\mathcal{O}(\log T \log\log T)$ time complexity for a precision of $\frac{1}{\log T}$ on the value of the empirical $\mathcal{E}_{\mathcal{F}}$. This numerical precision of the numerical scheme used to perform the convex optimization problem is the reason of the $\log\log T$ term and is simply read as the log of the number of samples of the relevant suboptimal arm.

8: The empirical value of $\mathcal{E}_{\mathcal{F}}$ of an optimal arm is zero by construction and its derivative also is zero by construction, see Figure 4.3. "Computing" the $\mathcal{E}_{\mathcal{F}}$ of optimal arm is therefore always $\mathcal{O}(1)$ complexity cost.

However, these two algorithms still require to keep every sample in memory because, from time to time, an exact computation of the empirical $\mathcal{E}_{\mathcal{F}}$ must be performed. This is rather non-satisfactory and motivates us to investigate an alternative approach in the next section.

Personal remark

The required memory for FMED/FIMED after T interactions is $\mathcal{O}(T)$, essentially because storing the samples of the optimal arm is necessary, even is those samples are not really used. The only scenario in which those samples would be used is when the optimal arm changes, a rare event. Therefore, I think that one could only keep $\mathcal{O}\left(\log(T)^2\right)$ samples of the optimal arm. The empirical mean of the optimal arm is still computed using the usual scheme and therefore all the collected samples. The number of samples used in the FIMED index would also be equal to the total number of collected samples as long as this arm is optimal. If there is a switch, *i.e.* the optimal arm suddenly is no longer the optimal one, then we use its effective number of samples, the $\mathcal{O}\left(\log(T)^2\right)$ quantity, in both the FMED and FIMED algorithm. With a number of samples proportional to $\log(T)^2$ instead of $\log T$ (now that it is suboptimal), this arm, previously optimal, would be considered vastly oversampled compared to its likelihood of optimality. Preserving $\log(T)^2$ samples compared to $\log T$ seems better to me because, in case of a switch, at least T large enough, $\log(T)^2 \geq c \log T$, for arbitrary positive constant c. Therefore, whatever the newly computed $\mathcal{E}_{\mathcal{F}}$ of this previously optimal arm, we are certain, for T large enough, that the number of samples is strictly (and vastly) greater than the amount of samples $\log T / \mathcal{E}_{\mathcal{F}}$ that would be required by the algorithm. There are therefore no samples to collect right after the suboptimality is discovered, which would be kind of absurd. One can even say that there are no samples to collect for roughly $\log T$ times the number of interactions that already happened. Instead of $\log(T)^2$, I do think that we could use $\log(T)^{1+\epsilon}$ for arbitrary positive ϵ, but it is hard to predict the effect on ϵ on finite time performances, although I think that large ϵ for $\mathcal{O}(|A|)$ interactions would be good. That is to say, at the beginning of interactions, we store all the samples because the probability of switching is non-negligible and, after a while, we use the rule of storing only $\log(T)^2$ samples for the empirical optimal arm.

Therefore, the rule would be to store all the samples for suboptimal arm and store $\max\left(\log(T)^2, |A|\right)$ for the optimal one.

It would surely make an analysis more complicated, but I think that a better rule would be to use the maximum value of $(\mu^\star(t) - \mu_a(t))^{-2}$, the invert of suboptimality gap instead of $|A|$. This would ensure that the number of samples allows a statistically significant separation of the maximal expected value and the closest to be optimal other expected value. Indeed, when the number of samples is n the fluctuations are of order $\frac{1}{\sqrt{n}}$. If we want to avoid switching due to random fluctuations, *i.e.* when the optimal mean fluctuates by more than the minimum of $\mu^\star(t) - \mu_a(t)$, denoted Δ, then one should try to guarantee that the number n of samples is such that

$$\frac{1}{\sqrt{n}} \leq \Delta \Rightarrow n \geq \frac{1}{\Delta^2} \,.$$

Because $\log(T)^2$ is bound to be larger than Δ^{-1} anyway, let's keep the simple memory constraint of $\log T^2$ for the optimal arm. If we were to modify FIMED and FMED the way we just describe, then the space complexity would drop from $\mathcal{O}(T)$ to a significantly lower $\mathcal{O}\left(\log(T)^2\right)$. This would be a huge improvement and, while writing this thesis, I do not have the time to prove this claim, I do think FIMED and FMED with this modification can be proved to be optimal. Assuming this is the case, not only would the amount of information per computation is increasing compared to IMED and MED, but the required memory to make sure of the optimality would also be so much lower!

The final algorithm, which we called AFMED (Amnesic Fast MED) is described Algorithm 16. A similar algorithm, AFIMED (Amnesic Fast IMED) can similarly be described while paying attention to using the right amount value of the number of samples in the log.

Interestingly, both FMED and FIMED are optimal algorithms. It shows that with fewer computations and without stronger assumptions on the set of distributions $\mathcal{F}$, it is possible to get an optimal algorithm. Thus, the amount of progress per unit of computation is better in FMED and FIMED compared to the original MED and IMED algorithms. I would like to encourage the reader to think about this fact proven by FMED and FIMED: optimality can be obtained with an amortized cost per iteration of $\mathcal{O}(|A|)$. The rate at which information is processed to compute an optimal strategy is therefore close to be minimal since this cost still depends on the total number of arm. In a strategy like Follow The Leader or Explore Then Commit, the cost per iteration is independent of the number of arms, except at the unique step in which a leader is computed.

Algorithm 16: AFMED: Amnesic Fast MED algorithm

Input: A bandit tuple $(\mathcal{A}, s, \mathcal{V})$ as in Definition 3.1.12;

Initialize history H as $H = \emptyset$;
Initialize $k : A \to \mathbb{R}$ that will store the value of the last exact computation of empirical $\mathcal{E}_{\mathcal{F}}$ at the last interaction with arm a;
Initialize $m : A \to \mathbb{R}$ that will store the value of $\mu^\star$ at the last interaction with arm a;
Initialize $\lambda : A \to \mathbb{R}$ that will store the value of the derivative of $\mathcal{E}_{\mathcal{F}}$ evaluated at $\mu^\star$ at the last interaction with arm a;
for $t \in \mathbb{N}$ **do**
  forall $a \in A$ **do**
    Compute exponential weight $P_a = \exp\left(-N_a(H)\mathcal{J}\left(a, H\right)\right)$;
  Compute multinomial parameters $p_a = \frac{P_a}{\sum_{a \in A} P_a}$ and vectorized parameter $p = (p_a)_{a \in A}$;
  Sample $a \sim Mult\,(p)$, a multinomial distribution of parameter p;
  Sample a reward $r \sim \beta(a)$ from distribution $\beta(a)$ of arm a;
  if *arm a is empirically optimal* **then**
    if $N_a(H) \leq (\log T)^2$ **then**
      Update history, $H \leftarrow H \cup \{(a, r)\}$;
  else
    Update history, $H \leftarrow H \cup \{(a, r)\}$;
  Update maximal map, $m(a) \leftarrow \mu^\star(H)$;
  Update $\mathcal{E}_{\mathcal{F}}$ map, $k(a)$, computed from the exact empirical $\mathcal{E}_{\mathcal{F}}$ solution;
  Update derivative map, $\lambda(a)$, computed from the exact empirical $\mathcal{E}_{\mathcal{F}}$ solution;

The history H is a *multiset*. Similarly to m and λ, the data structure used to represent H numerically is a *dictionary* or *hash table*.

Personal remark

The idea is that one should not be computing nor comparing indexes at every interaction, especially knowing that most of the time, the comparison will prescribe to follow the leader. Therefore, one could use a small *active action set* made of actions for which we compute the index and pick the next action from that set. The active set will change after every interaction and the computation of the new active set should not be costly, *i.e.* it should be a constant time cost. Let's denote A_t the active set at interaction t. The empirical optimal action should always belong to the active set A_t. Then, it should be that suboptimal actions belong to the active set with roughly equal probability, every $|A|$ interactions or so. The main idea is that suboptimal actions should belong to the active set at a frequency larger than $\frac{\log T}{T}$. Here we use a frequency $\frac{1}{|A|}$, but we could probably be even more aggressive and have an active set consisting of only the empirical best for longer and longer chunk of uninterrupted follow the leader periods, as long as the frequency of suboptimal arms in the active set is large enough, say, larger than $\frac{\log(T)^2}{T}$. If we choose A_t to always be of cardinal 2, then it should contain the current optimal action as well as a suboptimal one. The suboptimal action is chosen using a deterministic scheme based on a cyclic permutation $\sigma : A \to A$ of actions in the action set, as depicted in Figure 4.4. To be more precise, there should be one permutation $\sigma_a : A \setminus \{a\} \to A \setminus \{a\}$ for each action $a \in A$

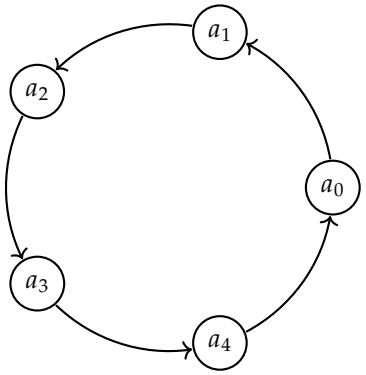

Figure 4.4: Cyclic permutation of a set of 5 actions indexed by $\mathbb{Z}_5$, $\{a_0, a_1, \ldots, a_5\}$.

where the cyclic permutation σ_a is used when action a is empirically optimal. If we denote $\star$ an empirically optimal arm, and assume that it does not change, then, using $\sigma_\star$, one can construct a sequence of active set from which an action is chosen. Starting from $a \neq \star$, the sequence then is

$$\{\star, a\} \to \{\star, \sigma(a)\} \to \{\star, \sigma \circ \sigma_\star(a)\} \to \cdots \to \left\{\star, \sigma_\star^{(T)}(a)\right\}$$

where $\sigma_\star^{(T)}$ denotes T compositions of $\sigma_\star$.

Using this idea, the amount of computation per interaction is therefore independent of the number of arm. We compute the index of an arm only if it in the active set: there is really one index to compute, the one of the suboptimal arm but the amortized cost is $\mathcal{O}(1)$ using the FMED/FIMED strategy. We have only one comparison to perform[9] instead of $|A|$. Therefore, using this strategy with FMED/FIMED, the number of computations per interaction is independent of the number of arms and, thanks to the competitive amortized cost of FMED/FIMED, the number of computations per interaction is reduced to $\mathcal{O}(1)$, which is impressive. If were to combine this scheme with the AFMED or AFIMED algorithms 16, then we would have an algorithm with an optimal number of computations per time step (nothing can be better than $\mathcal{O}(1)$ when comparing quantities with $\mathcal{O}$ notations) and a highly competitive space complexity required after T interactions of $\mathcal{O}\log(T)^2$. It should be noted that checking for the empirically maximal element is not done at a $\mathcal{O}(|A|)$ cost per interaction. Upon sampling the optimal arm, one can just check if the variation of the empirical expected values fluctuates more than the current suboptimality gap, which cost $\mathcal{O}(1)$ and does not happen often (exponentially decreasing probability of happening). If such an unlikely switch occurs, then the new optimal arm is given by the second one and the new position of the previous optimal arm can be found in $\log|A|$ time if the arms are already sorted. An efficient implementation could be to use a max-heap data structure. We call asynchronous the method that we just describe, as all arms are not considered at each interaction, and we call aAFMED (and aAFIMED) the asynchronous Amnesic FMED algorithm that merges the two ideas presented. The pseudocode is presented in Algorithm 17.

9: or only have to sample from a Bernoulli distribution in the case of (F)MED.

In Algorithm 17, indexes are computed line 7 using only the fast update and the exact computation of the empirical $\mathcal{E}_\mathcal{F}$ is performed line 16 whenever a suboptimal arm is sampled. This way, the amortized cost per interaction is $\mathcal{O}(1)$ and the space complexity grows as $\mathcal{O}\left(\log(T)^2\right)$. I believe that such an algorithm can be proven to be optimal with all the tools that are already available. I hope to be able to do the proof of aAFMED before the oral defense of this PhD. We summarize in Table 4.4 the complexities

Table 4.4: Amortized time complexity needed to compute the next suboptimal arm a of distribution $\beta(a)$ to pull and total space complexity required after T interactions.

Algorithm	Time complexity	Space complexity	Constant	Optimality
aAFMED	$\mathcal{O}(1)$	$\mathcal{O}\left(\log(T)^2\right)$	Conjecture $\frac{1}{\mathcal{E}_\mathcal{F}(\beta(a),\mu^\star)}$	Conjecture Opt.

and conjectured performances of the aAFMED algorithm (and implicitly the aAFIMED algorithm).

Algorithm 17: aAFMED: asynchronous Amnesic Fast MED algorithm

Input: A bandit tuple $(\mathcal{A}, s, \mathcal{V})$ as in Definition 3.1.12;

Initialize history H as $H = \emptyset$;
Initialize $k : A \longrightarrow \mathbb{R}$ that will store the value of the last exact computation of empirical $\mathcal{E}_{\mathcal{F}}$ at the last interaction with arm a;
Initialize $m : A \longrightarrow \mathbb{R}$ that will store the value of $\mu^\star$ at the last interaction with arm a;
Initialize $\lambda : A \longrightarrow \mathbb{R}$ that will store the value of the derivative of $\mathcal{E}_{\mathcal{F}}$ evaluated at $\mu^\star$ at the last interaction with arm a;
for $t \in \mathbb{N}$ **do**
  Compute active set A_t using **forall** $a \in A_t$ **do**
    Compute exponential weight $P_a = \exp\left(-N_a(H)\mathcal{J}\left(a, H\right)\right)$;
  Compute multinomial parameters $p_a = \frac{P_a}{\sum_{a \in A_t} P_a}$ and vectorized parameter $p = (p_a)_{a \in A}$;
  Sample $a \sim Bern\left(p\right)$, a Bernoulli distribution of parameter p;
  Sample a reward $r \sim \beta(a)$ from distribution $\beta(a)$ of arm a;
  if *arm a is empirically optimal* **then**
    if $N_a(H) \leq (\log T)^2$ **then**
      Update history, $H \leftarrow H \cup \{(a, r)\}$;
  else
    Update history, $H \leftarrow H \cup \{(a, r)\}$;
    Compute exactly the empirical $\mathcal{E}_{\mathcal{F}}$ using the newly collected sample;
  Update maximal map, $m(a) \leftarrow \mu^\star(H)$;
  Update $\mathcal{E}_{\mathcal{F}}$ map, $k(a)$, computed from the exact empirical $\mathcal{E}_{\mathcal{F}}$ solution;
  Update derivative map, $\lambda(a)$, computed from the exact empirical $\mathcal{E}_{\mathcal{F}}$ solution;

The history H is a *multiset*. Similarly to m and λ, the data structure used to represent H numerically is a *dictionary* or *hash table*.

Regret

The Algorithms FMED 15 and FIMED 14 have, by construction, rather similar theoretical guarantees. This situation is similar to what is known about the MED [33] and IMED [16] algorithms. The proofs techniques that are used in this section and the following aim to emphasize those similarities. Compared to the original articles, we think that the proofs are more general and prioritize general results about common quantities and phenomenon in the MED-like algorithms[10] than any previous work. Similarly to what is done in the more recent work of Charles Riou and Junya Honda [26], we distinguish between two main events respectively called the **pre-convergence** and **post-convergence** terms. Those two expressions captures the intuition that when the number of interaction is *small*, then the information are somehow unreliable and no well-informed optimal decision can be mad; this is pre-convergence. When the number of interactions is *large enough* that some form of knowledge about the optimal decision is certainly obtained, informed optimal decision can be made; this the post-convergence. The smallest amount of time the pre-convergence regime can last in the unstructured Bandit setting is obviously $|A|$ since all arms have to be sampled at least once by any agent. Thus, naming and separating terms this way might be key thought ingredient to better handle finite time analysis of bandit algorithms.

[33]: Honda et al. (2010), 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models.'

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

10: By MED-like algorithm, we mean those algorithms that, just like MED, are based on the minimization (M) of an empirical (E) Kullback-Leibler divergence (D).

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

In Theorem 4.2.2, we state an upper bound on the regret of the FMED Algorithm 15 and FIMED Algorithm 14. The stated result is of course also valid for the original MED and IMED algorithms, which we recall. The optimality of FMED-like algorithms is a direct consequence of this theorem.

Theorem 4.2.2 *Consider (A, β) a Bandit problem where the distribution of $\beta(a)$ is $F_a \in \mathcal{F}$, a set of bounded distributions with known boundaries, $\beta(a) \sim F_a \in \mathcal{F}$ for all $a \in A$. We denote $\mu^\star = \max_{a \in A} \mu(a)$ the maximal expected reward which, we recall, also is the maximal gain possible of the corresponding Bandit control problem.*

Let $T \in \mathbb{N}$ corresponds to a total number of interactions, or time horizon, $a \in A$ be the index of a suboptimal arm, and $\epsilon > 0$ be arbitrary (small) real number. Then, MED, FMED, IMED, and FMED satisfy the following upper bound on the expected number of interactions with arm a,

$$\mathbb{E}\left(N_a(T)\right) \leq \frac{\log(T)}{\mathcal{E}_{\mathcal{F}}(F_a, \mu^\star) - \epsilon} + o_\epsilon(\log(T)) \,, \tag{4.20}$$

where $o_\epsilon(\log(T))$ denotes a term that is asymptotically dominated by $\log(T)$ for any fixed ϵ, but with a polynomial dependency in ϵ^{-1}.

*Furthermore, all the algorithms are **asymptotically optimal**.*

The main result is that FMED and FIMED both preserve the theoretical guarantees of their original algorithm. We apply our analysis to MED and IMED too, in order to exhibit more precisely the new terms induced by the approximation, and their scaling in $\mathcal{O}(\epsilon^{-2})$. Furthermore, we drastically simplify the analysis of IMED compared to [16], although their result is more general because the authors consider the family of upper bounded distributions (no finite lower bound is assumed to be known).

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

To make the proof easier to follow, we first outline a sketch of proof that hopefully will highlight the key points of the reasoning.

Proof sketch. The Definition 4.19 of the approximation $\mathcal{J}_a(t)$ of the empirical $\mathcal{E}_{\mathcal{F}}\left(F_a(t), \mu^\star(t)\right)$ and Equation 4.15 can be read as

$$\mathcal{J}_a\left(t\right) \leq \mathcal{E}_{\mathcal{F}}\left(F_a(t), \mu^\star(t)\right)$$

for all arm $a \in A$ and index of interaction $t \in \mathbb{N}$. The unlikelihood of optimality computed by FMED/FIMED is always smaller than the empirical unlikelihood computed by the original MED/IMED vanilla algorithms. This fact makes FMED and FIMED at least as exploratory as the vanilla algorithms. Their regret is thus smaller in a *pre-convergence* regime, defined by the interactions for which the maximal expected value is ϵ-underestimated, $\mu^\star(t) \leq \mu^\star - \epsilon$. More exploration when things are not well estimated, especially the maximal gain, is better and lead to smaller regret because the algorithm can benefit a *regression to the mean* effect and recover quicker, *i.e.* with less interaction, from statistical errors.

The main challenge is then to prove that using the approximated divergences, $(\mathcal{J}_a(t))_{a \in A}$, do not lead to over-exploration of the suboptimal arms in a *post-convergence* regime, in which the maximal expected value is not ϵ-underestimated, $\mu^\star(t) \geq \mu^\star - \epsilon$. The key is to understand that

over-exploration of suboptimal arm a would have to come from a poor approximation of $\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$ by $\mathcal{J}_a(t)$. Such a scenario happen when $\mu^\star(s_a(t))$, the value of the best empirical mean the last time $(s_a(t))$ arm a has been sampled, is very different from the current value $\mu^\star(t)$ of the empirical best expected value. However, under such a scenario, at the moment suboptimal arm a is sample, the gap between $\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$ and $\mathcal{J}_a(t)$ is nullified, as well as the gap between $\mu^\star(s_a(t))$ and $\mu^\star(t)$. Afterward, in the post-convergence regime, the fluctuation of $\mu^\star(t)$ should be so small that the approximation errors are close to zeros and induce a small number of additional suboptimal interaction. This *small number* is the reason we get an additional $\mathcal{O}(\epsilon^{-2})$ term in the FMED and FIMED regret bound compared to the regret upper bound of the MED and IMED algorithms.

While this sketch of proof is mainly focused on the difference with MED/IMED because we already know the optimality of those algorithms, we recall that our proof will not assume this knowledge and involve proof techniques that allows to rederive the optimality of both the vanilla algorithms while unifying our viewpoint on those MED-like approaches. □

We summarize numerical and theoretical performances of our FMED and FIMED algorithms in Table 4.5 that will be used to complete the original table of comparisons 4.1.

Table 4.5: Amortized time complexity needed to compute the next suboptimal arm a of distribution $\beta(a)$ to pull and total space complexity required after T interactions.

Algorithm	Time complexity	Space complexity	Constant	Optimality
FMED (Alg. 15)	$\mathcal{O}(1)$	$\mathcal{O}(T)$	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$ (Thm. 4.2.2)	Opt.
FIMED (Alg. 14)	$\mathcal{O}(1)$	$\mathcal{O}(T)$	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$ (Thm. 4.2.2)	Opt.

Proof of Theorem 4.2.2

Of empirical divergence, information & optimality

In this section, we provide a proof of Theorem 4.2.2 that unify the proof of FMED, FIMED, MED, and IMED. This proof is important to this thesis because all the four published and presented papers are based on MED-like strategies. The subtitle of this section, **Of empirical divergence, information & optimality**, reflects our viewpoint on the theoretical tools used to derive a proof of MED-like strategies. While attempting to formalize mathematically what it means to *ask questions*, to *get information*, and to *sequentially solve*, we almost surely end up working with **probability theory** and **convex optimization**. The former is powerful mathematical way to express *randomness* and *uncertain information* about problems. The latter only makes sense in some spaces, in particular, those space where sentence like "getting closer to a solution" can be given a mathematical meaning. When a space is endowed with a topology, one can start to talk about the vicinity of some points. When a space is endowed with a metric, one can start to talk about points that are close, getting closer or further. In convex optimization, one can conveniently relate local *informative* values to *desirable* global ones and therefore *get closer* to a given point thanks to *local information*. Things really get interesting when mixing the two topics, probability and metric spaces since we can start say things such as *the probability of two points getting closer*. MED-type strategies make a great work at relating **gathered information** to **empirical unlikelihood of optimality** by mathematically handling **empirical divergence** between estimated distributions and relates this divergence to a probability of optimality.

The tools that are used to prove upper bounds on the logarithmic growth rate of regret on MED-like strategies are versatile and informative. We introduce them now, prior to proving the Theorem 4.2.2. In the following, the set of arms is assumed to be of cardinal K and arms are indexed by $\{1, \dots, K\}$. We assume that arm 1 is optimal which can hold true up to a permutation of indexes.

Toolbox for bounded distributions.

In the proofs we use the following results, that come from several works on bounded distributions in bandits (e.g. [16, 21]).

Lemma 4.2.1 (Useful properties of $\mathcal{E}_{\mathcal{F}}$) *Let $F \in \mathcal{F}$ be a distribution of mean μ_F. We denote by F_n and μ_n respectively the empirical cumulative distribution function (cdf) and empirical mean corresponding to n i.i.d. observations collected from F. It holds that*

1. *$\mathcal{E}_{\mathcal{F}}$ is continuous, non-decreasing and convex in its second argument (Theorem 7 of [33]).*
2. *There exists constants $c > 0$, $\delta_0 > 0$ such that $\forall \mu > \mu_F$, and $\delta \in (0, \delta_0]$ it holds that*

$$\mathbb{P}\left(\mathcal{E}_{\mathcal{F}}(F_n, \mu) \leq \mathcal{E}_{\mathcal{F}}(F, \mu) - \delta\right) \leq e^{-nc\delta^2} .$$

3. *For any $\epsilon > 0$ and $\mu > \mu_n + \epsilon$, it holds that*

$$\mathcal{E}_{\mathcal{F}}(F_n, \mu) - \mathcal{E}_{\mathcal{F}}(F_n, \mu - \epsilon) \geq \delta_\epsilon \coloneqq \frac{2\epsilon^2}{(B-b)^2} .$$

4. *For any $x > 0$, it holds that (Lemma 6 from [21])*

$$\mathbb{P}(\mathcal{E}_{\mathcal{F}}(F_n, \mu_1) > x) \leq e(n+2)e^{-nx}.$$

5. *For any $\epsilon > 0$ (Hoeffding's inequality),*

$$\mathbb{P}(\mu_n \leq \mu_F - \epsilon) \leq e^{-2n\epsilon^2} \quad \text{and} \quad \mathbb{P}(\mu_n \geq \mu_F + \epsilon) \leq e^{-2n\epsilon^2} .$$

Proof. Property 3. can be deduced from Lemma 13 of [33]. Using the convexity of $\mathcal{E}_{\mathcal{F}}$ and Pinsker inequality we obtain that

$$\mathcal{E}_{\mathcal{F}}(F_n, \mu) - \mathcal{E}_{\mathcal{F}}(F_n, \mu - \epsilon) \geq \mathcal{E}_{\mathcal{F}}(F_n, \mu_n + \epsilon) \geq \mathrm{kl}(\mu_n, \mu_n + \epsilon) \geq 2\epsilon^2 \ ,$$

if the range is $[0, 1]$. The factor $\frac{1}{(B-b)^2}$ comes from using this result on rescaled distributions for general ranges.

Then, property 2. can be obtained with a straightforward adaptation of the proof of Lemma 6 of [36] (in their Appendix B.1). Indeed, in this paper the authors consider a non-parametric family of distributions for which $\mathcal{E}_{\mathcal{F}}$ can be expressed in a very analogous way to Equation 4.16. □

Proof of Theorem 4.2.2

We introduce the notation

$$s_k(t) = \inf\{s \leq t : N_k(s) = N_k(t)\} \ ,$$

which is the time step corresponding to the last pull of arm k before time t, and

$$k^\star(t) = \underset{k \in A}{\operatorname{argmax}}\, \mu_k(t) \ ,$$

which is the empirical best arm.

We mention that the regret of all algorithms using the multinomialization trick can be directly deduced from the proof of the general case. Indeed, as the bandit algorithms only see the discretized reward, its regret guarantees are the same as those of a problem where the true distributions are $(F_1^{\mathrm{Mult}}, \dots, F_K^{\mathrm{Mult}})$. Hence, we focus in the following on the guarantees of the standard implementation of the algorithms.

Proof. Equation 4.20 can be proved for all the algorithms with the same general proof scheme. Hence, we propose a proof that tackle them all at once, explicitly mentioning which parts are specific to a given algorithm. We start by proving a first general upper bound.

Lemma 4.2.2 (Generic regret upper bound) *For each suboptimal arm k and any $\epsilon > 0$, FMED and FIMED both satisfy*

$$\mathbb{E}[N_k(T)] \leq u_\epsilon(T) + \underbrace{\mathbb{E}\left[\sum_{t=u}^{T-1} \mathbb{1}(A_{t+1} = k, N_k(t) > u_\epsilon(T), \mathcal{G}_k(t), \mathcal{J}_k(t))\right]}_{\textit{Post-CV}}$$
$$+ \underbrace{\mathbb{E}\left[\sum_{t=u}^{T-1} \mathbb{1}(A_{t+1} = k, N_k(t) > u, \mu^\star(t) \leq \mu_1 - \epsilon_0)\right]}_{\textit{Pre-CV}} + \mathcal{O}_\epsilon(1) \ ,$$

where

$$u_\epsilon(T) = \left\lceil \frac{\log(T)}{\mathcal{E}_{\mathcal{F}}(F_k, \mu_1) - \epsilon} \right\rceil \ ,$$

and using the notation $\epsilon_0 = \frac{B-\mu_1}{4}\epsilon$ and $\epsilon_1 = \epsilon_2 = \frac{\epsilon}{2}$ we define

$$\mathcal{J}_k(t) = \{\mathcal{E}_{\mathcal{F}}(F_k(t), \mu_1 - \epsilon_0) \geq \mathcal{E}_{\mathcal{F}}(F_k, \mu_1) - \epsilon_1\},$$

and

- *For MED and IMED,*

$$\mathcal{G}_k(t) = \left\{\mu^\star(t) \geq \mu_1 - \epsilon_0\right\}$$

- *For FMED and FIMED,*

$$\mathcal{G}_k(t) = \left\{ \begin{cases} \mu^\star(t) \geq \mu_1 - \epsilon_0, \\ \mu^\star(s_k(t)) \geq \mu_1 - \epsilon_0, \\ \dfrac{\mu^\star(t) - \mu^\star(s_k(t))}{B - \mu^\star(s_k(t))} \geq -\epsilon_2 \end{cases} \right\}$$

Proof. We upper bound the number of pulls by distinguishing several cases,

$$\begin{aligned}
\mathbb{E}\left[N_k(T)\right] \leq u_\epsilon(T) &+ \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(A_{t+1} = k, N_k(t) > u_\epsilon(T), \mathcal{G}(t), \mathcal{J}_k(t))\right]}_{\text{Post-CV}} \\
&+ \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(A_{t+1} = k, N_k(t) > u_\epsilon(T), \mu^\star(t) \leq \mu_1 - \epsilon_0)\right]}_{\text{Pre-CV}} \\
&+ \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(A_{t+1} = k, N_k(t) > u_\epsilon(T), \overline{\mathcal{J}}_k(t))\right]}_{\text{CV-Emp}} \\
&+ \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}\left(A_{t+1} = k, \mu^\star(t) \geq \mu_1 - \epsilon_0, \frac{\mu^\star(t) - \mu^\star(s_k(t))}{B - \mu^\star(s_k(t))} \leq -\epsilon_2\right)\right]}_{\text{Var-Best (FMED and FIMED only)}} \\
&+ \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}\left(A_{t+1} = k, \mu^\star(t) \geq \mu_1 - \epsilon_0, \mu^\star(s_k(t)) \leq \mu_1 - \epsilon_0\right)\right]}_{\text{Transition-Best (FMED and FIMED only)}}.
\end{aligned}$$

This upper bound already exhibits $u_\epsilon(T)$, Post-CV and Pre-CV. It remains to prove that the additional terms are upper bounded by constants depending on ϵ. The term CV-Emp is necessary for all algorithms, while the two last terms are specific to FMED and FIMED. Indeed, they tackle the cases where the suboptimal arm k is pulled due to a value of $\mu^\star(s_k(t))$ that deviates from μ_1 and/or $\mu^\star(t)$. We start by upper bounding CV-Emp, writing

$$\begin{aligned}
\text{CV-Emp} &\leq \sum_{t=1}^{T-1} \sum_{n=u_\epsilon(T)}^{T-1} \mathbb{E}\left[\mathbb{1}(A_{t+1} = k, N_k(t) = n, \overline{\mathcal{J}}_{k,n})\right] \\
&\leq \sum_{n=u_\epsilon(T)}^{T-1} \mathbb{P}\left(\mathcal{E}_{\mathcal{F}}(F_{k,n}, \mu_1 - \epsilon_0) \leq \mathcal{E}_{\mathcal{F}}(F_k, \mu_1) - \epsilon_1\right) \\
&\leq \sum_{n=u_\epsilon(T)}^{T-1} \mathbb{P}\left(\mathcal{E}_{\mathcal{F}}(F_{k,n}, \mu_1) \leq \mathcal{E}_{\mathcal{F}}(F_k, \mu_1) + \frac{\epsilon_0}{B - \mu_1} - \epsilon_1\right),
\end{aligned}$$

where the last line comes from Lemma 4 of [21]. We choose $\epsilon_1 = 2\frac{\epsilon_0}{B-\mu_1}$, and obtain that CV-Emp= $\mathcal{O}_{\epsilon_0}(1)$ thanks to property 2. of Lemma 4.2.1. This concludes the proof of Lemma 4.2.2 regarding MED and IMED. We now consider the terms specific to FMED and FIMED.

We start with Var-Best, and analyze more precisely the implications of the event

$$\mathcal{B}_k(t) = \left\{ \mu^\star(t) \geq \mu_1 - \epsilon_0, \frac{\mu^\star(t) - \mu^\star(s_k(t))}{B - \mu^\star(s_k(t))} \leq -\epsilon_2 \right\}.$$

We first prove that the combination of these two events ensure that $\mu^\star(s_k(t)) > \mu_1$ with a proper tuning of ϵ_0 and ϵ_2, since

$$\begin{aligned}
\frac{\mu^\star(t) - \mu^\star(s_k(t))}{B - \mu^\star(s_k(t))} \leq -\epsilon_2 &\Rightarrow \mu^\star(t) - \mu^\star(s_k(t)) \leq -\epsilon_2(B - \mu^\star(s_k(t))) \\
&\Rightarrow \mu^\star(s_k(t)) \geq \frac{\mu^\star(t) + \epsilon_2 B}{1+\epsilon_2} \geq \frac{\mu_1 - \epsilon_0 + \epsilon_2 B}{1+\epsilon_2} \\
&\Rightarrow \mu^\star(s_k(t)) \geq \mu_1 + \frac{\epsilon_2(B-\mu_1) - \epsilon_0}{1+\epsilon_2} .
\end{aligned}$$

Hence, we choose $\epsilon_2 = 2\frac{\epsilon_0}{B-\mu_1}$ to obtain $\mu^\star(s_k(t)) \geq \mu_1 + \frac{\epsilon_0}{3}$, if it holds that $\epsilon_0 \leq B - \mu_1$. Then, we also remark that this event also implies some variation of the best empirical mean between $s_k(t)$ and t. Hence, only two scenarios can make $\mathcal{B}_k(t)$ hold:

- $k^\star(s_k(t))$ is pulled between $s_k(t)$ and t, and $\mu^\star(s_k(t)) \geq \mu_1 + \frac{\epsilon_0}{3}$.
- $k^\star(s_k(t))$ is not pulled between $s_k(t)$ and t, $k^\star(t) = j$ for some $j \neq k^\star(s_k(t))$. In that case, j has been pulled between $s_k(t)$ and t and it holds that $\mu_j(t) = \mu^\star(t) \geq \mu^\star(s_k(t)) \geq \mu_1 + \frac{\epsilon_0}{3}$.

As each of these scenario can cause at most one pull of arm k, we can upper bound Var-Best by simply counting the number of times an arm $j \in [K]$ is pulled (at time t) and either $\mu_j(t) \geq \mu_1 + \epsilon_0$ or $\mu_j(t+1) \geq \mu_1 + \epsilon_0$. We hence obtain that

$$\begin{aligned}
\text{Var-Best} &\leq \sum_{j=1}^{K} \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(A_{t+1} = j, \left\{ \mu_j(t) \geq \mu_1 + \frac{\epsilon_0}{3} \right\} \cup \left\{ \mu_j(t+1) \geq \mu_1 + \frac{\epsilon_0}{3} \right\} \right) \right] \\
&\leq \sum_{j=1}^{K} \sum_{t=1}^{T-1} \sum_{n=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(A_{t+1} = j, N_j(s) = n, \left\{ \mu_{j,n} \geq \mu_1 + \frac{\epsilon_0}{3} \right\} \cup \left\{ \mu_{j,n+1} \geq \mu_1 + \frac{\epsilon_0}{3} \right\} \right) \right] \\
&\leq 2 \sum_{j=1}^{K} \sum_{n=1}^{+\infty} \mathbb{P}\left(\mu_{j,n} \geq \mu_1 + \frac{\epsilon_0}{3} \right) \\
&= \mathcal{O}_{\epsilon_0}(1) .
\end{aligned}$$

We can apply the exact same arguments for transition-best, except that this time we deduce that $k^\star(t)$ was pulled between $s_k(t)$ and t, and that (if $N_{k^\star(t)} = n$) $\mu_{j,n} \leq \mu_1 - \epsilon_0$ **and** $\mu_{j,n+1} \geq \mu_1 + \epsilon_0$. Depending on the identity of $k^\star(t)$ one of these two events has exponentially decreasing probability. Formally, we obtain that

$$\begin{aligned}
\text{Transition-best} &\leq \sum_{j:\mu_j<\mu_1} \sum_{n=1}^{+\infty} \mathbb{P}(\mu_{j,n} \geq \mu_1 - \epsilon_0) + \sum_{j:\mu_j=\mu_1} \sum_{n=1}^{+\infty} \mathbb{P}(\mu_{j,n} \leq \mu_1 - \epsilon_0) \\
&= \mathcal{O}_{\epsilon_0}(1) .
\end{aligned}$$

This concludes the proof of Lemma 4.2.2 by choosing $\epsilon_0 = \frac{B-\mu_1}{4}\epsilon$. □

Building on Lemma 4.2.2, we finish proving Theorem 4.2.2 by upper bounding the post-convergence and pre-convergence terms for all algorithms.

Upper bounding the post-convergence terms

Under $\mathcal{G}_k(t)$ and $\mathcal{J}_k(t)$, it holds thanks to property 1. of Lemma 4.2.1 that $\mathcal{E}_{\mathcal{F}}(F_k(t), \mu^\star(t)) \geq \mathcal{E}_{\mathcal{F}}(F_k, \mu_1) - \epsilon$. Hence, MED satisfies

$$\begin{aligned}
\text{Post-CV}_{MED} &\leq \mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(N_k(t) > u_\epsilon(T), \mathcal{G}_k(t), \mathcal{J}_k(t)) \times p_k(t)\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(N_k(t) > u_\epsilon(T), \mathcal{G}_k(t), \mathcal{J}_k(t)) \exp\left(-N_k(t)\mathcal{K}_k(t)\right)\right] \\
&\leq T \exp\left(-u_\epsilon(T)(\mathcal{E}_{\mathcal{F}}(F_k, \mu_1) - \epsilon)\right) \leq 1 \ ,
\end{aligned}$$

by definition of $u_\epsilon(T)\left\lceil \frac{\log(T)}{\mathcal{E}_{\mathcal{F}}(F_k,\mu_1)-\epsilon} \right\rceil$. Similarly, the design of $\mathcal{G}_k(t)$ for FMED also ensure that $\mathcal{K}_k(t) \geq \mathcal{E}_{\mathcal{F}}(F_k, \mu_1) - \epsilon$ and $\text{Post-CV}_{FMED} \leq 1$ by the same arguments. For IMED, we obtain that

$$\begin{aligned}
\text{Post-CV}_{IMED} &\leq \mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(I_k(t) \leq I_{k^\star(t)}(t), N_k(t) > u_\epsilon(T), \mathcal{G}_k(t), \mathcal{J}_k(t))\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(N_k(t)\mathcal{K}_k(t) \leq \log(N_{k^\star(t)}(t)) < \log(T), N_k(t) > u_\epsilon(T), \mathcal{G}_k(t), \mathcal{J}_k(t))\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(u_\epsilon(T)(\mathcal{E}_{\mathcal{F}}(F_k, \mu_1) - \epsilon) < \log(T))\right] = 0 \ ,
\end{aligned}$$

so arm k is never pulled in the post-convergence regime with IMED. Again, these arguments directly translate to Post-CV_{FIMED}.

Upper bounding pre-convergence term

Interestingly, the design of $\mathcal{K}_k(t)$, satisfying $\mathcal{K}_k(t) \geq \mathcal{E}_{\mathcal{F}}(F_k(t), \mu^\star(t))$, makes FMED and FIMED more exploratory than MED and IMED. Hence, we can unify the proof for the vanilla algorithms and their fast update in this part.

We start with MED. First, we use that for any sampling probability $p_k(t)$ it holds that

$$p_k(t) \leq \frac{p_k(t)}{p_1(t)} p_1(t) \leq \frac{1 - p_1(t)}{p_1(t)} p_1(t) \ ,$$

and obtain that

$$\begin{aligned}
\text{Pre-CV}_{MED} &\coloneqq \mathbb{E}\left[\sum_{t=u}^{T-1} \mathbb{1}(N_k(t) > u, A_{t+1} = k, \mu^\star(t) \leq \mu_1 - \epsilon_0)\right] \\
&= \mathbb{E}\left[\sum_{t=u}^{T-1} \mathbb{1}(N_k(t) > u, \mu^\star(t) \leq \mu_1 - \epsilon_0) p_k(t)\right] \\
&\leq \mathbb{E}\left[\sum_{t=u}^{T-1} \mathbb{1}(\mu_1(t) \leq \mu_1 - \epsilon_0) \left(\frac{1}{p_1(t)} - 1\right) p_1(t)\right] \\
&= \mathbb{E}\left[\sum_{t=u}^{T-1} \mathbb{1}(A_{t+1} = 1, \mu_1(t) \leq \mu_1 - \epsilon_0) \left(\frac{1}{p_1(t)} - 1\right)\right] \ .
\end{aligned}$$

Let us denote by $p_1^{\mathrm{FMED}}(t)$ the sampling probability of arm 1 under FMED. We obtain that

$$p_1^{\mathrm{FMED}}(t) \geq \frac{1}{K}\exp\left(-N_1(t)\mathcal{K}_1(t)\right) \geq \frac{1}{K}\exp\left(-N_1(t)\mathcal{E}_{\mathcal{F}}(F_1(t),\mu^\star(t))\right) \coloneqq p_1^{MED}(t)\ ,$$

where $p_1^{MED}(t)$ is itself a lower bound on the sampling probability of arm 1 under MED (we use this notation with a slight abuse). If follows that Pre-CV$_{FMED}$ admits the same upper bound as Pre-CV$_{MED}$. We first obtain that

$$\begin{aligned}\text{Pre-CV}_{MED} &\leq K\sum_{n=1}^{T-1}\mathbb{E}\left[e^{n\mathcal{E}_{\mathcal{F}}(F_{1,n},\mu_1-\epsilon_0)}-1\right] + (K-1)\sum_{n=1}^{T-1}\mathbb{P}(\mu_{1,n}\leq\mu_1-\epsilon_0)\\ &\leq K\sum_{n=1}^{T-1}\mathbb{E}\left[e^{n\mathcal{E}_{\mathcal{F}}(F_{1,n},\mu_1-\epsilon_0)}-1\right] + \mathcal{O}_{\epsilon_0}(1)\ .\end{aligned}$$

Then using properties 3. and 4. from Lemma 4.2.1 and $\mathcal{E}_{\mathcal{F}}(F_{1,n},\mu_1)\leq\mathcal{K}^+ \coloneqq \log\left(\frac{B-b}{B-\mu_1}\right)$ we conclude that

$$\begin{aligned}\text{Pre-CV}_{MED} &\leq K\sum_{n=1}^{T-1}\int_0^{e^{n\mathcal{K}^+}-1}\mathbb{P}\left(\mathcal{E}_{\mathcal{F}}(F_{1,n},\mu_1)\geq\delta_{\epsilon_0}+\frac{\log(1+x)}{n}\right)\mathrm{d}x\\ &\leq K\sum_{n=1}^{T-1}e(n+2)\int_0^{e^{n\mathcal{K}^+}-1}\frac{e^{-n\delta_{\epsilon_0}}}{1+x}\mathrm{d}x\\ &= Ke\log\left(\frac{B-b}{B-\mu_1}\right)\times\sum_{n=1}^{T-1}n(n+2)e^{-n\delta_{\epsilon_0}}\\ &= \mathcal{O}_{\epsilon_0}(1)\ ,\end{aligned}$$

with $\delta_{\epsilon_0}=\frac{2\epsilon_0^2}{(B-b)^2}$. As a consequence, Pre-CV$_{FMED}$ $=\mathcal{O}_{\epsilon_0}(1)$ too. For FIMED the relationship with the pre-convergence term of IMED is even more straightforward,

$$\begin{aligned}&\text{Pre-CV}_{FIMED} \leq \mathbb{E}\left[\sum_{t=u_\epsilon(T)}^{T-1}\mathbb{1}(N_1(t)\mathcal{K}_1(t)+\log(N_1(t))\geq\log(N_k(t)),N_k(t)>u_\epsilon(T),\mu^\star(t)\leq\mu_1-\epsilon_0)\right]\\ &\leq \mathbb{E}\left[\sum_{t=u_\epsilon(T)}^{T-1}\mathbb{1}(N_1(t)\mathcal{E}_{\mathcal{F}}(F_1(t),\mu^\star(t))+\log(N_1(t))\geq\log(N_k(t)),N_k(t)>u_\epsilon(T),\mu^\star(t)\leq\mu_1-\epsilon_0)\right]\\ &= \text{Pre-CV}_{IMED} = \mathcal{O}_{\epsilon_0}(1)\ .\end{aligned}$$

The last statement can be deduced from the regret analysis presented in [16], but we now propose a simpler proof of independent interest. We use that if $A_{t+1}=k$, then the index of arm k is smaller than the index of arm 1, and that $N_k(t)>u_\epsilon(T)$ to first write that

$$\begin{aligned}\text{Pre-CV}_{IMED} &\leq \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left(A_{t+1}=k,N_k(t)\geq u_\epsilon(T),I_1(t)\geq\log(N_k(t))\right)\right]\\ &\leq \mathbb{E}\left[\sum_{t=u_\epsilon(T)}^{T}\sum_{n_1=1}^{T}\sum_{n_k=u_\epsilon(T)}^{T}\mathbb{1}\left(A_{t+1}=k,N_k(t)=n_k,N_1(t)=n_1,I_1(t)\geq\log(n_k)\right)\right]\\ &\leq \mathbb{E}\left[\sum_{n_1=1}^{T}\sum_{n_k=u_\epsilon(T)}^{T}\mathbb{1}\left(n_1\mathcal{E}_{\mathcal{F}}(F_{1,n},\mu_1-\epsilon_0)+\log(n_1)\geq\log(n_k)\right)\right]\ .\end{aligned}$$

To further upper bound this term, we observe that since $\mathcal{E}_{\mathcal{F}}(F_{1,n}, \mu_1 - \epsilon_0) \leq \log\left(\frac{B-b}{B-\mu_1+\epsilon_0}\right) := \mathcal{K}^+_{\epsilon_0}$, the events considered above cannot happen if $n_1\mathcal{K}^+_{\epsilon_0} + \log(n_1) \leq \log(u_\epsilon(T)) \leq \log(n_k)$. Hence, there exists a function $g : T \mapsto \mathbb{N}$ satisfying $g(T) \to +\infty$ ($g(T)$ would be of order $\log\log(T)$ without the additional $\log(n_1)$) such that

$$\text{Pre-CV}_{IMED} \leq \mathbb{E}\left[\sum_{n_1=g(T)}^{T}\sum_{n_k=u_\epsilon(T)}^{T} \mathbb{1}\left(n_1\mathcal{E}_{\mathcal{F}}(F_{1,n}, \mu_1 - \epsilon_0) + \log(n_1) \geq \log(n_k)\right)\right] .$$

Then, using again properties 3. and 4. of Lemma 4.2.1 we obtain that

$$\begin{aligned}
\text{Pre-CV}_{IMED} &\leq \sum_{n_1=g(u)}^{T}\sum_{n_k=u}^{T} e(n_1+2)e^{-(n_1\delta_{\epsilon_0}+\log(n_k)-\log(n_1))} \\
&\leq e\left(\sum_{n_1=g(T)}^{T} n_1(n_1+2)e^{-n_1\delta_{\epsilon_0}}\right)\left(\sum_{n_k=u}^{T}\frac{1}{n_k}\right) \\
&\leq e\left(\sum_{n_1=g(T)}^{+\infty} n_1(n_1+2)e^{-n_1\delta_{\epsilon_0}}\right)\log\left(\frac{T}{u_\epsilon(T)}\right) \\
&= o_{\epsilon_0}(\log(T)) ,
\end{aligned}$$

where the conclusion comes from the fact that the remaining sum is the remainder of a convergent series, or in other words $\text{Pre-CV}_{IMED} \leq a_T \log(T)$ where $a_T \to 0$. This is sufficient to conclude the proof, by noting that a careful tuning of ϵ_0 (for instance as $\log\log(T)$ allows to obtain that $\limsup \frac{\mathbb{E}[N_k(T)]}{\log(T)} \leq \frac{1}{\mathcal{E}_{\mathcal{F}}(F_k,\mu_1)}$.

□

4.3 Online portfolio optimization: OMED & OIMED

The time complexity of FMED and FIMED which is $\mathcal{O}(T \log T)$ when sampling a suboptimal arm and $\mathcal{O}(1)$ otherwise is somewhat satisfactory. The corresponding amortized time complexity per interaction is $\mathcal{O}(1)$ which is optimal. However, the space complexity is still a linear function $\mathcal{O}(T)$ of the number of interactions T which is burdensome. In this section, we aim to alleviate the space complexity and to craft a fully sequential algorithm with true $\mathcal{O}(1)$ time complexity per interaction for all interactions, not amortized. We build on our intuition of the MCMC scheme introduced at the beginning of this chapter to create a fully sequential update rule of the empirical unlikelihood of optimality, $\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$. Recall that the dual formulation of $\mathcal{E}_{\mathcal{F}}$ is written as a concave linearly constrained maximization problem. The insight is that this dual problem can be reformulated as an **online portfolio selection** problem, a class of online convex optimization problems. More precisely, we express the computation of the dual formulation of $\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$ as a portfolio selection algorithm in dimension 2,

$$\mathcal{E}_{\mathcal{F}}(F_a(t), \mu^\star(t)) = \frac{1}{N_a(t)} \max_{\lambda \in \triangle_2} \sum_{n=1}^{N_a(t)} \log\left(\lambda^T \cdot \left(1, \frac{M - X_{a,n}}{M - \mu^\star(t)}\right)\right) ,$$

where[11] $\triangle_2$ is the probability-simplex of dimension 1 (space of Bernoulli distributions) and $(X_{a,n})_{1 \leq n \leq N_a(t)}$ the $N_a(t)$ reward samples collected from arm a after t interactions with the Bandit problem. This equation simply is a rewriting of the dual formulation of the dual formulation of $\mathcal{E}_{\mathcal{F}}$. We know connect this equation with a portfolio formulation.

11: $x^T \cdot y$ denotes the scalar product of vectors x and y in a real vector space of finite dimension d, here $\mathbb{R}^d$. x^T is the transpose of vector x.

Portfolio formulation

Online portfolio selection is a mathematical framework modeling a sequential interaction between a learner and an adversary. Prior to the sequence of N interactions, the adversary selects a sequence of vectors $x_1, \ldots, x_N$ in $\mathbb{R}^d$ for $d \geq 2$ representing the dimension of the problem. This is equivalent to state that the adversary create a matrix $(x_i)_{i \in [N]}$ where $x_i \in \mathbb{R}^d$ for all $1 \leq i \leq N$. Anticipating on what follows, one can think, in our Bandit setting, that an adversary is picking a random seed of random number generator that generates a sequence of random variable according to a prescribed law. Given a random number generator, **choosing a seed** amounts to **choosing a sequence** of random numbers. By the very nature of a perfect random number generator (*i.e.* with perfect hashing function), an adversary that can only choose a seed is indeed choosing the sequence but not controlling it as it is *impossible* to anticipate the sequence. If the adversary knows the used random generator, it cannot reasonably compute the seed to generate the sequence it would want because of the computational burden it would require (for modern random number generator). While our viewpoint on the adversary would be different, the above explanation shows how an adversary can choose a sequence without really being too *adversary* as it should be the case in the Bandit setting. Of course, some research is done on the topic

of adversarial Bandits but the associated community usually is more concerned about min-max/worst-case type of regret lower bounds.

In our work, the interpretation of $x_i \in \mathbb{R}^d$ is that it corresponds to d numerical signals coming from *d experts* and that we therefore have to weight those signals based on some criteria. In other word, we want to allocate a finite *mass* of resources to the different signals. This interpretation of sequentially allocating a resource (money) to different numerical signals (return of some investment strategy) based on some criteria (risk management, liquidity, politic. . .) explain the original name of this mathematical framework, **online portfolio optimization**.

In our Bandit framework, at interaction $n \in \mathbb{N}$, the learner chooses $\lambda_n \in \triangle_d \subseteq \mathbb{R}^d$ (the simplex of dimension $d-1$) and receives the reward $\log(\lambda_n^\top x_n)$. The reward the learner receive is a measure of performance of the chosen allocation λ_n at interaction n. The optimization criterion, named **regret**, of a learner is eventually based on a function of those per-interaction rewards. The regret of a learner, similarly to the Bandit regret, measure the discrepancy of the cumulated rewards the learner and its sequence of allocations $(\lambda_n)_n$ with the best possible time-uniform allocation λ of resources. The regret of the learner after N iterations is formally defined as

$$R_N(x_1, \dots, x_n) = \max_{\lambda \in \triangle_d} \sum_{n=1}^{N} \log(\lambda^\top x_n) - \sum_{n=1}^{N} \log(\lambda_n^\top x_n)\,. \tag{4.21}$$

The objective of the learner is then to find a sequence of allocations $(\lambda_n)_{n\in\mathbb{N}}$ that minimizes its regret R_N, where λ_n depends only on past information $(\lambda_1, x_1, \dots, \lambda_{n-1}, x_{n-1})$.

As already previewed, the dual of the empirical $\mathscr{E}_{\mathcal{F}}$ problem can be seen as cumulated rewards and its maximization as a portfolio optimization problem. Formally, we express, for any arm $a \in A$, the computation of $\mathscr{E}_{\mathcal{F}}(F_a(t), \mu^\star(t))$ as a portfolio selection algorithm in dimension 2,

$$N_k(t)\mathscr{E}_{\mathcal{F}}(F_k(t), \mu^\star(t)) = \max_{\lambda \in \Delta_2} \sum_{n=1}^{N_k(t)} \log\left(\lambda^\top x_{k,n}(t)\right), \qquad \text{with}$$
$$x_{a,n}(t) = \left(1, \frac{M - X_{a,n}}{M - \mu^\star(t)}\right).$$

Interestingly, the numerical signals of interest $x_{a,n}$ are not constants in the sense that they depend on $\mu^\star(t)$. Rather, those are better seen as real valued function of $\mu^\star$,

$$x_{a,n} : \mu^\star \mapsto \left(1, \frac{M - X_{a,n}}{M - \mu^\star}\right).$$

In reality, $\mu^\star$ won't be fixed but rather a (measurable) function of the number of interactions t,

$$x_{a,n} : t \mapsto \left(1, \frac{M - X_{a,n}}{M - \mu^\star(t)}\right),$$

which is slightly off the original portfolio framework because of its time dependency. This dependence in $\mu^\star(t)$, that is not revealed before time t, of $x_{a,1}(t), \dots, x_{a,n}(t)$ makes this first portfolio formulation a bit

unpractical. Indeed, we emphasize that the allocation λ_n should be based on past information and thus should not use $\mu^\star(t)$.

At the n-th interaction, we can only use the current best empirical mean, observable by arm a, denoted by $\mu^\star_{a,n}$ for simplicity. A portfolio algorithm can only try to minimize a past-information based metric,

$$R_N\left(x_{a,1},\dots,x_{a,N}\right) = \max_{\lambda\in\Delta_d} \sum_{n=1}^{N} \log(\lambda^\top x_n) - \sum_{n=1}^{N} \log(\lambda_n^\top x_n) \quad \text{with} \tag{4.22}$$

$$x_{a,n} = \left(1, \frac{M - X_{a,n}}{M - \mu^\star_{a,n}}\right) , \tag{4.23}$$

which we call **portfolio regret** in the rest of this section. From the definition of $x_{a,n}$, one can see that this regret metric is suitable to evaluate the intrinsic performance of a portfolio algorithm in our framework.

A problem emerges from this formulation is that this measure of regret is now different from our original problem of minimizing the $\mathcal{E}_\mathcal{F}$, *i.e.* maximizing the dual representation. If the present-time dependent rewards $\log\left(\lambda_n^T x_{a,n}(t)\right)$ are too different from the past-time dependent rewards $\log\left(\lambda_n^T x_{a,n}(t)\right)$ that we just introduced in Equation 4.23, then our (already sequential) estimation of $\mathcal{E}_\mathcal{F}$ will suffer. In other words, since the portfolio reward depends on $\mu^\star$, if $\mu^\star_{a,n}$ diverges too often from $\mu^\star(t)$ our estimation of $\mathcal{E}_\mathcal{F}$ will not be accurate. Hence, we also define the **bias** (of an arm a) of the portfolio selection algorithm, caused by its inability to foresee $\mu^\star(t)$,

$$B_{a,N_a(t)} = \max_{\lambda\in\Delta_2} \sum_{n=1}^{N_a(t)} \log(\lambda^\top x_{a,n}) - \max_{\lambda\in\Delta_2} \sum_{n=1}^{N_a(t)} \log\left(\lambda^\top x_{a,n}(t)\right) . \tag{4.24}$$

This term can be studied independently of the portfolio regret and only depends on the variations of the best empirical mean. It is not negligible: we will have to modify the structure of the MED algorithms and assume that the best mean is not too close to the upper boundary M to control its magnitude. Now that we set up what we are aiming to do, minimizing the portfolio regret and controlling the portfolio bias, we ought to take a look at current literature on portfolio algorithms to see if a method suits our needs. Thankfully, one can start by looking at [61] which is a recent (2023) review of portfolio selection algorithms.

[61]: Tsai et al. (2023), 'Online Self-Concordant and Relatively Smooth Minimization, With Applications to Online Portfolio Selection and Learning Quantum States'

Portfolio algorithms Different methods vary based on their computational complexities and their regret guarantees. Some have regret upper bounds as low as $\mathcal{O}(d \log N)$ but are computationally expensive, like UPS [62–64]. Our goal is to use portfolio methods to obtain a computationally efficient $\mathcal{E}_\mathcal{F}$, hence we cannot use those. However, in our case, the constant N will be the number of interaction with a suboptimal arm a after t interactions with the Bandit problem, $N_a(T)$. If our scheme prove to be optimal or at least allows for a logarithmic number of samples of suboptimal arms, it means that $N_a(T)$ will be of order $\log T$. Hence, we can afford a looser regret upper bound since we will use it on supposedly $\mathcal{O}(\log T)$ samples from the suboptimal arms. For instance, the Soft-Bayes [65] algorithm, has a regret of $\mathcal{O}(\sqrt{dN})$, which should translate to $\mathcal{O}(\sqrt{d \log T})$ in our case. Furthermore, it has a $\mathcal{O}(d)$ computational complexity per round, which

[62]: Cover (1991), 'Universal portfolios'
[63]: Cover et al. (1996), 'Universal portfolios with side information'
[64]: Kalai et al. (2002), 'Efficient algorithms for universal portfolios'

[65]: Orseau et al. (2017), 'Soft-bayes: Prod for mixtures of experts with log-loss'

is perfect in our case where $d = 2$. Other algorithms presented in the aforementioned survey but also in [66] achieve intermediate trade-offs.

[66]: Zimmert et al. (2022), 'Pushing the efficiency-regret Pareto frontier for online learning of portfolios and quantum states'

Among the computationally cheap algorithms, we chose Soft-Bayes for its simple implementation and the fact that [65] provide bounds on the regret valid for an adaptive step-size parameter, which we need since we don't know the total number of samples we will see for the arms.

[65]: Orseau et al. (2017), 'Soft-bayes: Prod for mixtures of experts with log-loss'

Apart from FMED and FIMED, the main contribution of the paper this chapter is based on is to introduce the first Bandit algorithm that use a portfolio allocation algorithm to estimate $\mathcal{E}_{\mathcal{F}}$. We call those algorithms **OMED**, Online MED, and **OIMED**, Online IMED. While the link between $\mathcal{E}_{\mathcal{F}}$ and portfolio selection was previously used in Lemma E.1 [36], where the authors use the existence of a portfolio algorithm with logarithmic regret to obtain a concentration inequality on $\mathcal{E}_{\mathcal{F}}$, they only use that observation to obtain a bound for the analysis, and they do not explore any algorithmic use of the portfolio formulation.

[36]: Agrawal et al. (2021), 'Regret Minimization in Heavy-Tailed Bandits'

Algorithm structure of OMED and OIMED The algorithms OMED and OIMED will use a sequential portfolio selection algorithm to compute a sequence of estimations of the $\mathcal{E}_{\mathcal{F}}$. *A priori*, OMED and OIMED are therefore a class of algorithms because one can define one such algorithmic scheme for every suitable portfolio selection algorithm. We therefore first introduce the OMED and OIMED algorithms for a generic portfolio algorithm. As discussed above, using an online estimate of empirical $\mathcal{E}_{\mathcal{F}}$ makes the MED algorithms highly sensitive to variations of the best empirical mean. To mitigate this issue, we apply several structural changes to the algorithms that aim at preventing the portfolio bias to be large. As seen in Figure 4.3, the function $\mu \mapsto \mathcal{E}_{\mathcal{F}}(F, \mu)$ have a derivative that diverges at the upper boundary M of its domain of definition. An unbounded derivative is rarely a good sign of stability for sequential optimization schemes. This is why we will assume the knowledge of an upper bound $\mu^{\max}$ on the maximal expected value $\mu^\star$ of our considered bandit problem with $\mu^{\max} < M$, *i.e.* the known upper bound $\mu^{\max}$ is distinct from the support upper bound M. The gap $M - \mu^{\max}$ allows to "compactify" the domain of $\mathcal{E}_{\mathcal{F}}$ and its continuous derivative with respect to μ and therefore have uniform control on the magnitude of such a derivative. This assumption is not that restrictive but can rather be seen as the introduction of a hyperparameter $\gamma > 0$. Indeed, one way to satisfy the required assumption is to consider the set $\mathcal{F}_\gamma$ of bounded distribution with upper bound on the support now equal to $M + \gamma$. If we consider the Bandit problem with distribution originally in $\mathcal{F}$, then $\mu^\star$ is naturally upper bounded strictly by M which we take as a $\mu^{\max}$ since in the larger class $\mathcal{F}_\gamma$, it is γ-separated from the boundary. That is to say, we consider a Bandit algorithm suitable for $\mathcal{F}_\gamma$ rather than $\mathcal{F}$. In the following, we will consider the knowledge of $\mu^{\max} < M$, smaller than the natural upper bound of class $\mathcal{F}$.

Controlling the fluctuations of $\mu^\star(t)$ should be of uttermost importance in order to tell something meaningful about the bias and, therefore, regret of a portfolio Bandit algorithm. The fluctuation of this quantity is related to concentration assumptions on the distributions (the class $\mathcal{F}$) and the number of samples of the empirical best arm. It would be best if that arm has the largest associated number of samples $\max_a N_a(t)$.

The reason is that, after t interactions, the arm that has been sampled the most has been sampled more than $\frac{t}{|A|}$ times. It means that this number of samples is roughly a linear function of the number of interactions[12] and the uncertainty of the expected value of such an arm can be controlled thanks to concentration hypothesis. Now, it is possible, especially in the first time steps, that an empirical best arm does not correspond to a most sampled arm. In that case, one must close that gap and adopt a greedy strategy by sampling the empirical best arm until it is the most sampled or no longer the best arm. For this purpose, we define a **leader** as $\ell_t \in \text{argmax } N_a(t)$.[13]

12: more precisely, it is bounded by two linear functions, $\frac{t}{|A|} \leq \max_a N_a(t) \leq t$.

13: Ties are broken in favor of the arm with best empirical mean, then at random if several candidates remain.

To better control the bias and convergence of $t \mapsto \mathcal{E}_{\mathcal{F}}(F, \mu^\star(t))$, it will be better if $\mu^\star(t)$ in fact corresponds to $\mu_\star(t)$, *i.e.* if the empirical maximal expected value is *always* the expected value of an arm $\star$. Of course, it is impossible to have this event with probability one. Therefore, we will rather introduce, for a distribution F, $|A|$ estimation of the $\mathcal{E}_{\mathcal{F}}$, one for each arm a in A: $t \mapsto \mathcal{E}_{\mathcal{F}}(F, \mu_a(t))$. Therefore, even if the $\mu^\star(t)$ is not always associated to the same arm, *i.e.* a *jump* in the trajectory occurs, then one can *jump* from one set of $\mathcal{E}_{\mathcal{F}}$ estimation to another one, corresponding to the current empirical maximal arm $\star(t)$. While this is the way we dealt the control of $\mu^\star$, I do think that there is room for improvement and that using $|A|$ estimations of the $\mathcal{E}_{\mathcal{F}}$ might be too much. However, whatever the method, one still have to find a way to make the learner recover fast enough from wrong estimation of $\mathcal{E}_{\mathcal{F}}$ due to using the wrong reward signal. This is because once we switch from one believed-to-be best arm to another one, one may observe that the value of $\mu^\star$ will change a lot in a few iterations (regression to the true mean of the empirical arm) while the value of the $\mathcal{E}_{\mathcal{F}}$ is still highly dependent on the history of previous values of $\mu^\star$.

[31]: Chan (2020), 'The multi-armed bandit problem: An efficient nonparametric solution'

The final structural modification that we add to create OMED and OIMED is to use a **duel-based** algorithm, a design that is inspired by [31]. Duel-based algorithms are based on the concept of duelling a **leader**, that we already introduce, and the other arms, that are called **challengers** in this framework. Challengers compete against the leader in pairwise comparisons called **duels**. The specific arm-specific numerical quantities used for comparison will be defined in a moment, but the reader can already guess some natural candidates from the original design of MED and IMED. In a duel based algorithm, an active set of *arm to be pulled* is maintained. If this set is not empty, then an arm from this set is sampled and removed from the set while no duel is done. If the active set is empty, then duels are performed and the results of the duels are used to add arms to the active set. This set consists of arms that won their duels. This design implies that at the end of the round of comparison, all winning challengers (if any) are pulled. If there are none, the leader is pulled. Hence, several arms can be pulled per round.

To better control the bias and convergence of $t \mapsto \mathcal{E}_{\mathcal{F}}(F, \mu^\star(t))$, we replace $\mu^\star(t)$ by $\mu_{\ell_t}(t)$ for the reference value used by the challengers in their estimations of infimum KLs, and to implement a different estimate of unlikelihood of optimality, *i.e.* $\mathcal{E}_{\mathcal{F}}$ times the number of samples, $L_{a,\ell_t}(t)$ for each possible pair (a, ℓ_t).

Duelling framework We now introduce some notation and terminology to describe the duel between a challenger a and a leader ℓ. We distinguish between several forms of duels. The simpler one, we call it the **greedy duel**. In a greedy duel between a challenger a and leader ℓ_t, the winner is decided based on the maximal value of empirical expected rewards, *i.e.* the winner is argmax $\{a : \mu_a(t),\ \ell_t : \mu_{\ell_t}(t)\}$. In accordance with our previous intuition and analysis, we use greedy duel when $N_{\ell_t}(t)$ is not large enough compared to $N_a(t)$, implying that the estimation of $\mathcal{E}_{\mathcal{F}}$, based on $x_{a,n}\ (\mu_{\ell_t}(t))$, will be hindered by the estimation of $\mu_{\ell_t}(t)$ which is not considered "static enough" compared to the number $N_a(t)$ of samples of the "outer" expectation that we are trying to optimize. In other words, when the leader has not been sampled enough compared to arm a, then the noise in the estimation of the portfolio reward function and $\mathcal{E}_{\mathcal{F}}$ is not dominated by samples collected from arm a and the noise induced by the wrong estimation of the inner parameter $\mu^\star\ \mu_{\ell_t}$ cannot be considered negligible or small enough. In such a case, it is better to use a greedy duel and do not update the estimate of $\mathcal{E}_{\mathcal{F}}$ associated to the couple $(a\ell_t)$ but rather wait for the expected value of the challenger to be considered stable enough compared to the noise induced by the current number of samples (hence learning rate) of arm a. Algorithmically and mathematically, we introduce a binary variable $Z_a(t)$, that indicates if the duel played by arm a at interaction t was greedy ($Z_a(t) = 0$) or not ($Z_a(t) = 1$). The portfolio selection algorithm, like most sequential stochastic optimization algorithm, will use a learning rate that is not fixed but depends on the number of samples collected from a given distribution. In our case, we are minimizing a constrained expected value, $\mathcal{E}_{\mathcal{F}}$ problem, and the collected samples are the portfolio rewards. However, there are two reasons we cannot use the total number $N_a(t)$ of collected samples in our learning rate scheme. First, for an arm a, there are $|A|$ parallel estimations of $\mathcal{E}_{\mathcal{F}}$, one for each possible leader. Therefore, there are $|A|$ parallel learning rate that are based on the number of times arm a has been sampled while the leader was ℓ. We can denote such a quantity by $N_{a,\ell}(t)$, with

$$N_{a,\ell}(t) = \sum_{i=1}^{t-1} \mathbb{1}\left\{a \in \mathcal{A}_{i+1}, \ell_i = \ell\right\}\ ,$$

and we have the relation $N_a(t) = \sum_{\ell \in A} N_{a,\ell}(t)$. The second reason is that $N_{a,\ell}$ does not even correspond to the total number of updates of the $\mathcal{E}_{\mathcal{F}}$ associated to the couple (a, ℓ). As presented, we compute a portfolio reward and then update our estimate of $\mathcal{E}_{\mathcal{F}}$ only when the number of samples of the leader is large enough. Therefore, the number $N_{a,\ell}(t)$ (and *a fortiori* $N_a(t)$) of samples collected from arm a when challenger is ℓ does not correspond to the number of updates of the $\mathcal{E}_{\mathcal{F}}$ associated to the couple (a, ℓ_t). This pseudo-count of samples that corresponds to the used samples of arm a in the update of $\mathcal{E}_{\mathcal{F}}$ associated to leader ℓ is denoted $\widetilde{N}_{a,\ell}(t)$ and is formally equal to

$$\widetilde{N}_{a,\ell}(t) = \sum_{i=1}^{t-1} \mathbb{1}\left\{a \in \mathcal{A}_{i+1}, \ell_i = \ell, Z_a(s) = 1\right\}\ .$$

It is the number of observations of a challenger a collected *against the leader* ℓ after a *non-greedy duel*.

Before addressing the definition of a **non-greedy duel** we formally answer

the question that we already partly answered: what is the criterion to decide whether a duel is greedy? Let $f : \mathbb{N} \to \mathbb{R}_+$ be a function that any linear function is asymptotically negligible compared to f, *i.e.* $n = o(f(n))$. A duel between a challenger a and a leader ℓ is made greedy if the precision of the empirical expected value of the leader ℓ_t is too small compared to the pseudo-count of number of times arm a has been sampled with corresponding leader. Formally, a duel is greedy if $N_\ell(t) \leq f\left(\widetilde{N}_{a,\ell}\right)(t)$. Another reason to make a duel greedy, is when we know for sure that μ_ℓ is wrongly estimated. This is the case when $\mu_\ell(t) > \mu^{\max}$. Furthermore, this also corresponds to the region of the support domain where the gradient that will be used in the portfolio selection algorithm is considered uncontrolled, *i.e.* μ_ℓ is not in the reference compact. Therefore, a duel also is greedy when $\mu_\ell(t) > \mu^{\max}$. Finally, one can say that the two conditions for a duel to be greedy are

$$\mu_\ell(t) > \mu^{\max} ,$$

in which case all duels are greedy because the expected value of the leader is for sure wrongly estimated, and

$$N_\ell(t) \leq f\left(\widetilde{N}_{a,\ell}(t)\right)$$

in which case the leader has not been sampled enough to consider that the stochastic optimization scheme will be using a correct enough estimation of the gradient.

When a duel is deemed to be **non-greedy**, we compare quantities that originates from the original MED and IMED algorithms. For **OIMED**, a non-greedy duel is based on the comparison of IMED indexes of the leader and challenger. The challenger a wins its duel against the leader ℓ if

$$L_{a,\ell}(t) + \log(\widetilde{N}_{a,\ell}(t)) \leq \log(N_\ell(t)) , \tag{4.25}$$

where we recall that $L_{a,\ell}(t)$ is the estimated unlikelihood of optimality $N_a(t)\mathscr{E}_{\mathcal{F}}(F_a, \mu^\star)$ (more precisely of $N_{a,\ell}\mathscr{E}_{\mathcal{F}}(F_a, \mu_\ell)$) of arm a computed by the portfolio selection algorithm. The update rule for $L_{a,\ell}$ is described shortly. For **OMED**, a non-greedy duel is based on the "comparison" of samples from Bernoulli distributions, whose parameters are estimations of $\exp(-N_a\mathscr{E}_{\mathcal{F}}(F_a, \mu^\star))$. This corresponds to the unormalized exponential weights that are used when performing the multinomial sampling in the original MED algorithm. By definition, for the leader arm, $\mathscr{E}_{\mathcal{F}}(F_\ell, \mu_\ell) = 0$ which is the used value for the leader arm. Therefore, the leader arm has a Bernoulli parameter of $e^0 = 1$ and its Bernoulli sample is necessarily one. The only comparison that is therefore made to decide whether a challenger wins a duel or not, is if its sample collected from a Bernoulli of parameter $\exp(-L_{a,\ell}(t))$ is equal to one. The challenger a wins its duel against the leader ℓ if

$$W_a(t) = 1 \text{ where } W_a(t) \sim \text{Ber}(\exp(-L_{a,\ell}(t))) . \tag{4.26}$$

Portfolio update of $\mathscr{E}_{\mathcal{F}}$ The non-greedy duels are based on the estimated unlikelihood of optimality $L_{a,\ell}$ that are computed and updated by the portfolio selection algorithm after each such non-greedy duel. We emphasize that $L_{a,\ell}$ is updated **only** when the duel was **non-greedy**

and won by the challenger which is algorithmically and Mathematically translated by $Z_a(t) = 1$. The update of $L_{a,\ell}(t)$ then is based on the current estimated empirical reward $\mu_\ell(t)$ of the leader and the $N_a(t)^{th}$ collected $X_{a,N_a(t)}$ reward from challenger arm a. This update scheme that combines greedy and non-greedy duel to prescribe whether an update of the estimated unlikelihood of optimality shall happen is the main algorithmic structure that underpins the theoretical and numerical control of the portfolio bias and Consequently, portfolio and Bandit learner regrets. This method is independent of the specific choice if portfolio selection algorithm that used to perform the updates.

We provide in Algorithm 18 an implementation of the OMED algorithm using a generic portfolio selection algorithm in the update of estimated value of unlikelihood of optimality (line 27). In the OMED Algorithm 18, the lines 1-5 corresponds to the initialization of several useful quantities. It should be noted that, while those quantities are initially defined as depending on the whole history, *e.g.* the counting function $N_a(H) = N_a(t)$ or the expected reward $\mu_a(H) = \mu_a(t)$, can be updated sequentially without using the whole history which is therefore not part of the initialization of the OMED algorithm. The loop that begins line 8 is a loop over **rounds**, not number of interactions. Indeed, several arms can be sampled per round as it can be seen in the loop starting line 19 that is on a set $\mathcal{A}$ updated lines 12 and 17. In the OMED Algorithm 18, the lines 7-8 corresponds to the computation of the leader for round $t \in \mathbb{N}$. The loop staring line 9 contains all the duels. First, line 10-13 are the greedy duels and then lines 15-18 are the non-greedy duels. For each type of duels, we update the set $\mathcal{A}$ of arms of winner and the values of Z_a indicating whether the duel was greedy or not. The block starting line 19 check whether one duel was won by a challenger, *i.e.* if the set $\mathcal{A}$ is empty or not. If this set is empty, then the leader will be the only played arm at that round. Of course, by definition $L_{\ell,\ell} = 0$ for all $\ell \in A$ so the boolean Z_ℓ indicating whether one should use the portfolio selection algorithm is set to 0 line 21. Line 22 starts the loop that iterates over all the arms that must be sampled at round t and are stored in set $\mathcal{A}$. We sample all those arms and update total counts and expected rewards lines 23-25. Depending on the nature of the duel of arm a, checked line 27, we update the pseudo-count line 28, run an update of the portfolio selection algorithm `ALG` line 29 and update the unlikelihood of optimality for the couple (a, ℓ) line 30. The OIMED algorithm is derived from this one by changing the lines 15-16 corresponding to Equation 4.26 to the corresponding OIMED duel Equation 4.25. In the following, we focus on the OMED version rather than the OIMED, but the two are rather similar.

Algorithm 18: OMED: Online MED algorithm

Input: A bandit tuple $(\mathcal{A}, s, \mathcal{V})$ as in Definition 3.1.12;
A suitable portfolio selection algorithm `ALG`;

Initialize for all $a \in A$ and $\ell \in A$, initialize N_a, $N_{a,\ell}$, and $\widetilde{N}_{a,\ell}$ that are the counting functions (from non-stored history to $\mathbb{N}$) that we previously introduced. When the history is empty, all those quantities are zero.;
Initialize for all $a \in A$, μ_a the function from history that compute the empirical expected value of arm a with a zero initial value when history is empty.;
Initialize for all $a \in A$ and $\ell \in A$, $L_{a,\ell} = 0$ that will store the estimate of unlikelihood for a challenger-leader couple equal to $(a; \ell)$.;
Initialize the selection parameters $\lambda_{a,\ell} = \frac{1}{2}$. The selection parameters are the probability distributions in $\triangle_2$ introduced at the beginning of this section.;
Initialize the active set $\mathcal{A} = \emptyset$ of arms that will be pulled at round $t \in \mathbb{N}$. ;
for $t \in \mathbb{N}$ *(caution: t is the number of **rounds**)* **do**
  Compute leader candidates $\mathcal{L} = \operatorname{argmax}_{a \in A} N_a(t)$;
  Compute leader $\ell \in \operatorname{argmax}_{a \in \mathcal{L}} \mu_a(t)$, breaking ties uniformly at random;
  forall $a \in A \setminus \ell$, *compute duels* **do**
    if $\widetilde{N}_{a,\ell} \geq f(N_\ell)$ ***or*** $\mu_\ell(t) \geq \mu^{max}$ **then**
      if $\mu_a(t) \geq \mu_\ell(t)$ **then**
        Update $\mathcal{A} \leftarrow \mathcal{A} \cup \{a\}$;
        Set $Z_a = 0$;
    else
      Sample $W_a \sim \text{Bern}(\exp(-L_{a,\ell}))$;
      if $W_a = 1$ **then**
        Update $\mathcal{A} \leftarrow \mathcal{A} \cup \{a\}$;
        Set $Z_a = 1$;
  if $\mathcal{A} = \emptyset$, *i.e. the leader won all its duels* **then**
    Update $\mathcal{A} \leftarrow \mathcal{A} \cup \{\ell\}$;
    Set $Z_\ell = 0$;
  forall $a \in \mathcal{A}$ **do**
    Remove a from $\mathcal{A}$, $\mathcal{A} \leftarrow \mathcal{A} \setminus \{a\}$;
    Sample reward $r \sim \beta(a)$;
    Update expected reward μ_a;
    Update total number N_a of collected samples;
    if $Z_a = 1$ **then**
      Update pseudo-count $\widetilde{N}_{a,\ell}$, incrementing it by one;
      Update portfolio selection parameter $\lambda_{a,\ell}$ using `ALG`. The algorithm `ALG` uses $\widetilde{N}_{a,\ell}$ and $\frac{M-r}{M-\mu_\ell(t)}$;
      Update the unlikelihood of optimality $L_{a,\ell}$,
      $L_{a,\ell} \leftarrow L_{a,\ell} + \log\left(1 - \lambda_{a,\ell} \frac{r-\mu_\ell(t)}{M-\mu_\ell(t)}\right)$

4.4 Partial proof & open question

Under some additional assumptions, OMED and OIMED have an asymptotical optimal regret and its expected number of pulls can be proved to have a very similar upper bound to that of FMED and FIMED stated in Theorem 4.2.2.

Theorem 4.4.1 (Regret bound for Online MED algorithms) *Consider (A, β) a Bandit problem where the distribution of $\beta(a)$ is $F_a \in \mathfrak{F}$, a set of bounded distributions with known boundaries, $\beta(a) \sim F_a \in \mathfrak{F}$ for all $a \in A$. We denote $\mu^\star = \max_{a \in A} \mu(a)$ the maximal expected reward which, we recall, also is the maximal gain possible of the corresponding Bandit control problem. Assume the knowledge of a bound $\mu^{max} < M$ such that $\mu^\star < \mu^{max}$.*

Assume that OMED and OIMED use a portfolio selection algorithm `ALG` *that satisfies for any challenger a and leader ℓ, $(a, \ell) \in A^2$ and $n \in \mathbb{N}$ the deterministic guarantee*

$$|R_{a,\ell}(n)| = o(n) \ , \tag{4.27}$$

*that is to say, the **absolute value** of the portfolio regret is asymptotically sublinear.*

Let $T \in \mathbb{N}$ corresponds to a total number of interactions, or time horizon, $a \in A$ be the index of a suboptimal arm, and $\epsilon > 0$ be arbitrary (small) real number. Then, both OMED and OIMED satisfy the following equation

$$\mathbb{E}\left(N_a(T)\right) \leq \frac{\log(T)}{\mathcal{E}_{\mathfrak{F}}(F_a, \mu^\star) - \epsilon} + o_\epsilon(\log(T)) \ , \tag{4.28}$$

where $o_\epsilon(\log(T))$ denotes a term that is asymptotically dominated by $\log(T)$ for any fixed ϵ, but with a polynomial dependency in ϵ^{-1}. This second order term differs for OMED and OIMED.

*OMED and OIMED are therefore **asymptotically optimal** and enjoy the same theoretical guarantees than the originals MED and IMED.*

Owing to the algorithmic structure of OMED and OIMED, the regret performances of the algorithms mainly depend on the regret bounds[14] of the portfolio regret. Before sketching a proof of Theorem 4.4.1, we would like to emphasize on what we think is the most important algorithmic structure behind the success of a sequential optimization scheme, **tracking**. **Tracking** allows checking for especially bad events and take actions whenever those occur. It may not always be necessary to achieve asymptotic optimality, but it can surely help when interested in minimizing the second order terms that often depends on how fast an algorithm can recover from bad events. While the expected time to recover from a bad events may be finite or sub-logarithmic without tracking, it may be much smaller with a bit of tracking. Here, we are dealing with a fully sequential memoryless algorithmic scheme. Tracking bad events to make sure that we recover fast enough (or do not suffer) from bad events is probably a necessity.

14: Plural here because both upper and lower bounds on the portfolio regret are important.

Remark As of writing those lines, I do not think that the duelling structure is a necessary feature to stabilize the algorithmic sampling

scheme, although it does enjoy some nice properties. What I think is the most important features is this notion of **greedy** and **non-greedy** phases. This amounts to a notion of **tracking**. The algorithm *tracks* incoherences in the expected values $\mu_\ell \leqslant \mu^{\max}$ because it controls the magnitude of the gradient that is used in the update rule and the algorithm *tracks* that the sampled portfolio reward, $pr \sim_X \log(1 - (X - \mu^\star(t))\nu)$, is indeed sampled from a distribution that is close enough to the true one. In this case, close enough means that $\mu^\star(t)$ should be overwhelmingly better estimated than the current empirical mean of portfolio rewards. In this case, the tracking takes the form of a comparison between the number of samples of the leader arm and the challenger arm. However, we do think that one could replace this duelling structure by a structure closer to the original MED and IMED. In that case, one would *track* that the empirical best arm is the most sampled and make sure to sample it until this condition is reached. This tracking could be used in the original MCMC-like algorithmic idea presented at the beginning of this chapter.

Proof sketch of Theorem 4.4.1. The general proof scheme is inspired by [31], that proposed a similar duel-based approach. The first thing to remark is that, because the greedy/non-greedy duelling structures, not all the samples from challenger arms will be used to compute update of the unlikelihood estimation, $L_{a,\ell_t}(t)$. Therefore, one should find a way to prove that the number of greedy duels is not too large so that $\widetilde{N}_{a,\ell_t}(t) \simeq N_{a,\ell_t}(t)\,(\simeq N_a(t))$. This fact also explain why one should not compare $L_{a,\ell_t}(t)$ to $N_a(t)\mathcal{E}_\mathcal{F}(F_a(t), \mu_{\ell_t}(t))$, where $F_a(t)$ is the empirical distribution of arm a using all $N_a(t)$ samples, but rather to $\widetilde{N}_{a,\ell_t}(t)\mathcal{E}_\mathcal{F}(F_{a,\ell_t}(t), \mu_{\ell_t}(t))$, where $F_{a,\ell_t}(t)$ is the empirical distribution of the $\widetilde{N}_{a,\ell_t}(t)$ observations used to update $L_{a,\ell_t}(t)$. The greedy/non-greedy structure is the reason why one should not use the raw $N_{a,\ell_t}(t)$, because all those samples are not used to perform updates of the unlikelihood estimation. However, as stated, one should make sure that those are roughly equal in order to maximize information extracted from suboptimal samples.

[31]: Chan (2020), 'The multi-armed bandit problem: An efficient nonparametric solution'

The main difficulty consists in controlling the deviation of $L_{a,\ell_t}(t)$ from $\mathcal{E}_\mathcal{F}(F_{a,\ell_t}(t), \mu_{\ell_t}(t))$. Both over-estimation and under-estimation are bad. Under-estimation of the unlikelihood of optimality may lead to over-sampling suboptimal arms and therefore increase the Bandit regret. Over-estimation of the unlikelihood of optimality may prevent sufficient exploration of the true best arm when it was initially wrongly confused for a suboptimal one. The recovery time from this bad event may be too large, *i.e.* one may play a suboptimal arm thinking it is optimal for too long, therefore leading to an increase in the Bandit regret. Of course, we already know that the unlikelihood of estimation $N_a\mathcal{E}_\mathcal{F}$ defines the right sampling boundary to achieve optimal exploration-exploitation trade-off, a fact that is used by MED and IMED. This is why we also try to control the deviation of $L_{a,\ell_t}(t)$ from $\mathcal{E}_\mathcal{F}(F_{a,\ell_t}(t), \mu_{\ell_t}(t))$.

The proof relies on the following crucial decomposition. For fixed arms (a, ℓ), a sample size $n := \widetilde{N}_{k,\ell}(t)$, and any threshold μ, it holds that

$$n\mathcal{E}_\mathcal{F}(F_{a,\ell,n}, \mu) = L_{a,\ell,n} + R_{a,\ell,n} + B_{a,\ell,n}(\mu)\,, \tag{4.29}$$

where

- $R_{a,\ell,n}$ is the portfolio regret as in Equation 4.21,

- $B_{a,\ell,n}(\mu)$ is the portfolio bias with respect to a fixed threshold μ, defined in Equation 4.24, and
- $L_{a,\ell,n}$ and $F_{a,\ell,n}$ are used to denote $L_{a,\ell}(t)$ and $F_{a,\ell}(t)$ when the pseudo-count is such that $\widetilde{N}_{a,\ell}(t) = n$.

In the proofs, the portfolio regret is controlled by assumption. The term $L_{a,\ell,n}$ will control the Bandit regret similarly to the exact unlikelihood of optimality $N_a^{\mathcal{E}_{\mathcal{F}}}$, as in [31] and [12]. Therefore, the main difficulty is the control of the bias, which requires more careful examination. While the analysis of the Bandit regret term is not trivial, isolating and analyzing this portfolio bias term was one of the main theoretical contribution of the paper. To analyze it, we prove the following result. For $m \leq n$, we denote by $t_{a,\ell,m}$ the time (*i.e.*, interaction's index in the history) when the m^{th} update of $L_{a,\ell,n}$ occurred, *i.e.* the time when updating the estimate $L_{a,\ell}$ from $L_{a,\ell,m-1}$ to $L_{a,\ell,m}$.

[31]: Chan (2020), 'The multi-armed bandit problem: An efficient nonparametric solution'

[12]: Honda et al. (2011), 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'

Lemma 4.4.2 (Deviations of $B_{k,\ell,n}(\mu)$) *Let $\mu \in (0,1)$, and set*

$$C_{a,\ell,m} = \frac{M - \mu_\ell(t_{a,\ell,m})}{M - \mu} .$$

It holds that

$$B_{a,\ell,n}(\mu) \geq \sum_{m=1}^{n} \log(C_{a,\ell,m})\, \mathbb{1}(\mu \leq \mu_\ell(t_{a,\ell,m})) \tag{4.30}$$

$$B_{a,\ell,n}(\mu) \leq \sum_{m=1}^{n} \log(C_{a,\ell,m})\, \mathbb{1}(\mu \geq \mu_\ell(t_{a,\ell,m})) \tag{4.31}$$

The two inequations will always be used for a value of μ that is in-between the second and largest expected reward, $\mu \in \left(\max_{a:\mu_a<\mu^\star}, \mu^\star\right)$. It immediately implies that the two inequations will always be used for $\mu_\ell(t_{a,\ell,m}) < \mu^{\max}$ due to one of the tracking condition which ensure that the denominator of $C_{a,\ell,m}$, $M - \mu$, is always larger than the gap $\gamma = M - \mu^{\max}$.

Hence, the upper bound defined by Equation 4.31 is expected to be sublinear, and even finite, when the leader is an optimal arm, $\ell = \star$, because of the indicator which contains an event of the form $\{\mu < \mu^\star, \mu \leq \mu_\star(t)\}$. Concentration *à la* Sanov (as in the book [1]) will take care of this term. The lower bound defined by Equation 4.30 is expected to be sublinear when the leader is not optimal $\ell \neq \star$. Again, this is thanks to the indicator function that contains an event of the form $\left\{\mu > \max_{a:\mu_a<\mu^\star}, \ell \neq \star, \mu < \mu_\ell(t)\right\}$ and Concentration *à la* Sanov will take care of this term.

[1]: Dembo et al. (2010), *Large Deviations Techniques and Applications*

The rate of convergence of the indicators is important. The proof presented in this chapter work if $n = o(N_\ell(t_{a,\ell,n}))$, which justifies the greedy duels. Indeed, we need $e^{an}\mathbb{P}(B_{a,\ell,n} < -nx)$ to be small enough (for some $a, x > 0$ and large n), which allows us to obtain that $e^{an - f(n)y_x} \to 0$ (for some y_x depending on x) and f the function used to assess part of the greediness of a duel condition.

Proof. We prove Lemma 4.4.2 that is important for the forthcoming proof of regret upper bound of OMED. We introduce some notation for

convenience. First, we denote $X_{k,\ell,i}$ by X_i and $\mu^\star(t_{k,\ell,i})$ by $\mu^\star(i)$. Then, we define $Y_i = \frac{X_i - \mu^\star(i)}{B - \mu^\star(i)}$, $Z_i = \frac{X_i - \mu}{B - \mu}$, and $\lambda = \underset{\lambda \in [0,1]}{\operatorname{argmax}} \sum_{i=1}^n \log(1 - \lambda Z_i)$.

We use two elementary analysis properties. First, $\frac{1-\lambda Z_i}{1-\lambda Y_i}$ is positive and non-decreasing if $Y_i \geq Z_i$. Then, $Y_i \geq Z_i$ only if $\mu^\star(i) \leq \mu$. Otherwise, $\log(1 - \lambda Z_i) - \log(1 - \lambda Y_i) \leq 0$. Hence, it holds that

$$\begin{aligned} B_{k,\ell,n}(\mu) &\coloneqq \sum_{i=1}^n (\log(1 - \lambda Z_i) - \log(1 - \lambda Y_i)) = \sum_{i=1}^n \log\left(\frac{1 - \lambda Z_i}{1 - \lambda Y_i}\right) \\ &\leq \sum_{i=1}^n \log\left(\frac{1 - Z_i}{1 - Y_i}\right) \mathbb{1}(\mu \geq \mu^\star(i)) \, . \end{aligned}$$

We then plug the expression of Y_i and Z_i in this bound, obtaining that

$$\begin{aligned} B_{k,\ell,n}(\mu) &\leq \sum_{i=1}^n \log\left(\frac{\frac{B - X_i}{B - \mu}}{\frac{B - X_i}{B - \mu^\star(i)}}\right) \mathbb{1}(\mu \geq \mu^\star(i)) \\ &= \sum_{i=1}^n \log\left(\frac{B - \mu^\star(i)}{B - \mu}\right) \mathbb{1}(\mu \geq \mu^\star(i)) \, , \end{aligned}$$

which gives the result. Applying the exact same steps provides the other direction,

$$-B_{k,\ell,n} \leq \sum_{i=1}^n \log\left(\frac{B - \mu}{B - \mu^\star(i)}\right) \mathbb{1}(\mu^\star(i) \geq \mu) \, .$$

□

Remark: While this was the line of the paper at the time of writing, after thinking a bit more about this, I will again insist that the most important feature for the algorithm stability is, in my current opinion, the tracking and not the duels. To not update the unlikelihood estimation when the number of samples of the optimal arm is not large enough is the most important feature. When this event occurs, using modified indexes (such as the empirical expected values) indeed is a good idea to make sure that the most sampled arm corresponds to the empirical maximal arm.[15] □

15: I hope to test this idea before the oral of this thesis.

The modifications introduced in OMED and OIMED guarantee a small absolute bias with large probability. This is a good thing if we refer to the intuition that about the portfolio bias that was developed until now, see paragraph 4.24. It is natural to ask whether such structural modifications introduced in OMED and OIMED were necessary. While one could not formally prove it, our experiments suggest that the regret of OIMED/OMED may be linear without them.

Before performing numerical experiments or presenting the full proof of Theorem 4.4.1, one must decide upon the specific portfolio algorithm, denoted `ALG` in Algorithm 18, that we want to use. As stated in the introduction part of this chapter, there are several candidates for the implementation of OIMED/IMED.

Assumption on the portfolio regret Ideally, the chosen portfolio algorithm should satisfy all the necessary assumptions. Unfortunately, while upper bounds on the portfolio regrets are often studied in the portfolio optimization community, lower bounds (different from the *by definition* zero) are not an active research subject and the literature on regret *lower* bounds is scarce. Hopefully, this thesis and corresponding paper will foster research of lower bound due to this connection with the Bandit community.

[67]: Gofer et al. (2016), 'Lower bounds on individual sequence regret'

[68]: Guzmán et al. (2021), 'Best-case lower bounds in online learning'

In [67], the authors characterize algorithms with non-negative regret, but for linear losses. The paper [68] showed that the regret of FTRL algorithms is bounded as $\mathcal{O}(\sqrt{N})$ both from above and below, in a setting that encompasses portfolio selection. Unfortunately, FTRL is computationally inefficient, which makes it unsuitable for our application targeting the best numerical complexity. We used **Soft-Bayes** for its simplicity and computational efficiency, but it might not have the desired lower bound. Our problem posed by this Bandit theoretical setting seems to be among the first where a large negative regret is detrimental, and obtaining a portfolio algorithm with both small computational complexity and a regret lower bound is an open question. On the other hand, it may be that this lower bound requirement could be relaxed in the analysis, and that for example a high-probability lower bound could be enough. Our experimental study supports the hypothesis that using Soft-Bayes in OIMED gives good regret bounds for the bandit problem.

Numerical complexity We summarize numerical and theoretical performances of our OMED and OIMED algorithms in Table 4.6 that will be used to complete the original table of comparisons 4.1. This is a huge numerical improvement compared to the original MED and IMED algorithms. The time complexity indeed is reduced to $\mathcal{O}(1)$ per arm and interaction to update the index or unlikelihood of estimation. The space complexity is independent of the number of interactions and is reduced to a $\mathcal{O}(|A|)$ per arm since for all challenger/arm a, one must store $|A|$ pseudo-counts and estimation of unlikelihood of optimality, one per potential leader/arm. The total space complexity is therefore $\mathcal{O}\left(|A|^2\right)$, which again, is independent of the number of interactions.

Table 4.6: Time complexity needed to compute the next suboptimal arm a of distribution $\beta(a)$ to pull and total space complexity per arm required after T interactions.

Algorithm	Time complexity	Space complexity	Constant	Optimality
OMED (Alg. 18, Eq. 4.26)	$\mathcal{O}(1)$	$\mathcal{O}(\lvert A\rvert)$	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$ (Thm. 4.4.1)	Opt.
OIMED (Alg. 18, Eq. 4.25)	$\mathcal{O}(1)$	$\mathcal{O}(\lvert A\rvert)$	$\frac{1}{\mathcal{E}_{\mathcal{F}}(\beta(a),\mu^\star)}$ (Thm. 4.4.1)	Opt.

Proof of Theorem 4.4.1: regret analysis of OMED and OIMED

Proof. As for the proof of Theorem 4.2.2 we start by proving a first regret upper bound that holds for both algorithms, OMED and OIMED.

Lemma 4.4.1 (Generic upper bound) *For any suboptimal arm k, for any $\epsilon > 0$ it holds that*

$$\mathbb{E}[N_k(t)] \leq u_\epsilon(T) + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}(k \in \mathcal{A}_{t+1}, \ell_t = 1, \widetilde{N}_{k,1}(t) > u_\epsilon(T), Z_k(t) = 1, \mathcal{G}_k(t))\right]}_{\textit{Post-CV}}$$
$$+ 4\log(4)\underbrace{\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}(1 \notin \mathcal{A}_{t+1}, \ell_t \neq 1, Z_1(t) = 1)\right]}_{\textit{Pre-CV}} + \mathcal{O}_\epsilon(1)\ ,$$

where $u_\epsilon(T) = \frac{\log(T)}{\mathcal{E}_\mathcal{F}(F_n, \mu_1) - \epsilon}$, *and* $\mathcal{G}_k(t) = \left\{L_{k,1}(t) \geq \widetilde{N}_{k,1}(t)(\mathcal{E}_\mathcal{F}(F_k, \mu_1) - \epsilon)\right\}$.

Before proving this result, let us detail some intuitions. First, we remark that the two expectations need to be second order terms to make the algorithms asymptotically optimal. The first expectation corresponds to pulls of arm k in a *post-convergence regime*, where arm 1 is the leader and the empirical distribution of arm k is close to F_k. On the contrary, the second term is the expected number of duels lost by arm 1 as a challenger, against a suboptimal leader. For this reason, we name these two terms respectively *Post-CV* and *Pre-CV* (where CV abbreviates convergence). Note that Lemma 4.4.1 actually holds for any bandit algorithms that would use the same duel-based structure, independently of what they do during the non-greedy duels.

Proof. We start the proof by introducing some notation. Due to the specific structure of the duel-based algorithm we define for a challenger/leader pair (k, ℓ) a pseudo-count $\widetilde{N}_{k,\ell}(t)$ and a corresponding empirical cdf $\widetilde{F}_{k,\ell}(t)$: they are computed by considering only the observations of arm k that have been collected *when arm k was pulled after a non-greedy duel ($Z_k(t) = 1$) performed against the leader $\ell_t = \ell$*. Hence, $\widetilde{N}_{k,\ell}(t) \leq N_k(t)$. In several parts of the proof we also use constants $x_k \in (\mu_k, \mu_1)$, that we arbitrarily set to $\frac{\mu_k + \mu_1}{2}$. Finally, we introduce two ways to denote the rewards: we continue to call $X_{k,n}$ the n-th reward received by arm k, but we also introduce the notation $X_{k,\ell,n}$ for the n-th reward received by arm k after a non-greedy duel against the leader ℓ.

We start the analysis by considering the case when the best arm is the leader, and the alternative. For each suboptimal arm k, it holds that

$$\mathbb{E}[N_k(T)] = 1 + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(k \in \mathcal{A}_{t+1}, \ell_t = 1)\right]}_{A_1} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(k \in \mathcal{A}_{t+1}, \ell_t \neq 1\right]}_{A_2}$$

We then consider the favorable case where arm 1 the leader, splitting the cases according to the value of $Z_k(t)$.

$$A_1 \leq \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(k \in \mathcal{A}_{t+1}, \ell_t = 1, \mu_1(t) \leq \mu_k(t))\right]}_{B_1 \text{ (greedy duels)}} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(k \in \mathcal{A}_{t+1}, \ell_t = 1, Z_k(t) = 1)\right]}_{B_2 \text{ (non-greedy duels)}}\ .$$

We now upper bound B_1 using that if $\ell_t = 1$ then $N_1(t) \geq t/K$, and that if $\mu_k(t) \geq \mu_1(t)$ then either $\mu_k(t) \geq x_k$ or $\mu_1(t) \leq x_k$. By starting $N_k(t)$ at 1 we cover all possible scenarios for $Z_k(t) = 0$ (including $\mu_{\ell_t}(t) \geq \mu_{\max}$). We then obtain

$$\begin{aligned} B_1 &\leq \mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(\ell_t = 1, \mu_1(t) \leq x_k)\right] + \mathbb{E}\left[\sum_{t=1}^{T-1} \mathbb{1}(k \in \mathcal{A}_{t+1}, \mu_k(t) \geq x_k)\right] \\ &\leq \sum_{t=1}^{+\infty} \sum_{n=t/K}^{+\infty} \mathbb{P}(\mu_{1,n} \leq x_k) + \sum_{n=1}^{+\infty} \mathbb{P}(\mu_{k,n} \geq x_k) = \mathcal{O}(1) \, . \end{aligned}$$

We then consider B_2. We use that for any n,

$$\{k \in \mathcal{A}_{t+1}, Z_k(t) = 1, \widetilde{N}_{k,1}(t) = n\} = \{\widetilde{N}_{k,1}(t+1) = \widetilde{N}_{k,1}(t) + 1, \widetilde{N}_{k,1}(t) = n\} \, ,$$

and can happen only once. For any $u_\epsilon(T) \in \mathbb{N}$, we hence obtain that

$$B_2 \leq u_\epsilon(T) + \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}(\widetilde{N}_{k,1}(t+1) = \widetilde{N}_{k,1}(t) + 1, \widetilde{N}_{k,1}(t) > u_\epsilon(T))\right] \, .$$

We then upper bound B_2 by considering separately cases when $\mathcal{G}_k(t)$ holds or not,

$$\begin{aligned} B_2 \leq u + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}(\widetilde{N}_{k,1}(t+1) = \widetilde{N}_{k,1}(t) + 1, \widetilde{N}_{k,1}(t) > u, \mathcal{G}_k(t))\right]}_{\text{Post-CV}} \\ \underbrace{+ \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}(\widetilde{N}_{k,1}(t+1) = \widetilde{N}_{k,1}(t) + 1, \widetilde{N}_{k,1}(t) > u, \bar{\mathcal{G}}_k(t))\right]}_{B_2'} \, . \end{aligned}$$

We remark the first term is exactly the post-convergence term that we introduced in the lemma. Hence, we leave this expression as it is for this first result. We then consider B_2'. We use Equation (4.29), for any $\epsilon_0 > 0$ it holds that

$$L_{k,1,\widetilde{N}_{k,1}(t)} = \widetilde{N}_{k,1}(t)\mathscr{E}_{\mathcal{F}}(\widetilde{F}_k(t), \mu_1 - \epsilon_0) + R_{k,1,\widetilde{N}_{k,1}(t)} + B_{k,1,\widetilde{N}_{k,1}(t)}(\mu_1 - \epsilon_0).$$

Hence, we obtain that

$$\bar{\mathcal{G}}_k(t) \subset \left\{\max\left\{\frac{B_{k,1,\widetilde{N}_{k,1}(t)}(\mu_1 - \epsilon_0)}{\widetilde{N}_{k,1}(t)}, \frac{R_{k,1,\widetilde{N}_{k,1}(t)}}{\widetilde{N}_{k,1}(t)}, \mathscr{E}_{\mathcal{F}}(F_k, \mu_1) - \mathscr{E}_{\mathcal{F}}(\widetilde{F}_k(t), \mu_1 - \epsilon_0)\right\} \geq \frac{\epsilon}{3}\right\} \, .$$

Note that we could have stated different thresholds for each term, but we choose $\epsilon/3$ in each case for simplicity.

We use the first side of our assumption on the regret of the portfolio algorithm: if $R_n = o(n)$, then there exists $n_0 \in \mathbb{N}$ large enough such that for $n \geq n_\epsilon$, $R_n \leq \frac{n\epsilon}{3}$. Hence, by defining $u_\epsilon(T) \geq n_\epsilon$ we ensure that $\bar{\mathcal{G}}_k(t)$ is not due to a large portfolio regret.

We now consider the term involving $\mathscr{E}_{\mathcal{F}}(\widetilde{F}_k(t), \mu_1 - \epsilon_0)$, which is analogous to the CV-Emp term of the proof of Theorem 4.2.2. We hence directly write that

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left(\widetilde{N}_{k,1}(t+1)=\widetilde{N}_{k,1}(t)+1,\widetilde{N}_{k,1}(t)>u_\epsilon(T),\mathscr{E}_{\mathcal{F}}(\widetilde{F}_k(t),\mu_1-\epsilon_0)\leq\mathscr{E}_{\mathcal{F}}(F_k,\mu_1)-\frac{\epsilon}{3}\right)\right]$$
$$\leq\sum_{n=u_\epsilon(T)}^{+\infty}\mathbb{P}\left(\mathscr{E}_{\mathcal{F}}(\widetilde{F}_{k,n},\mu_1)\leq\mathscr{E}_{\mathcal{F}}(F_k,\mu_1)+\frac{\epsilon_0}{B-\mu_1}-\frac{\epsilon}{3}\right)=\mathcal{O}_\epsilon(1)\,,$$

by choosing $\epsilon_0=(B-\mu_1)\frac{\epsilon}{6}$. Finally, we use that $B_{k,1,\widetilde{N}_{k,1}(t)}(\mu_1-\epsilon_0)\geq\widetilde{N}_{k,1}(t)\epsilon/3$ is possible only if at least one empirical mean computed with a "reasonable" sample size deviates. More precisely,

$$B_{k,1,\widetilde{N}_{k,1}(t)}(\mu_1-\epsilon_0)\geq\widetilde{N}_{k,1}(t)\frac{\epsilon}{3}\Rightarrow\exists n\geq\widetilde{N}_{k,1}(t)\frac{\epsilon}{3}\;:\;\mu_1(t_{k,1,n})\leq\mu_1-\epsilon_0$$
$$\Rightarrow\exists n\geq f\left(\widetilde{N}_{k,1}(t)\frac{\epsilon}{3}\right)\;:\;\mu_{1,n}\leq\mu_1-\epsilon_0\,,$$

where we used that, thanks to our algorithm, at any time s for which the estimate $L_{k,1}(t)$ was incremented $N_1(s)\geq f(\widetilde{N}_{k,1}(s))$ was satisfied (check to decide that the duel is non-greedy). Interestingly, we now get an event that only depends on $\widetilde{N}_{k,1}(t)$. Using Lemma 4.4.2 we finally get that

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left(\widetilde{N}_{k,1}(t+1)=\widetilde{N}_{k,1}(t)+1,\widetilde{N}_{k,1}(t)>u_\epsilon(T),B_{k,1,\widetilde{N}_{k,1}(t)}(\mu_1-\epsilon_0)\geq\widetilde{N}_{k,1}(t)\frac{\epsilon}{3}\right)\right]$$
$$\leq\sum_{t=u_\epsilon(T)}^{+\infty}\mathbb{P}\left(\widetilde{N}_{k,1}(t+1)=\widetilde{N}_{k,1}(t)+1,\widetilde{N}_{k,1}(t)\geq u_\epsilon(T),\exists s\geq f\left(\widetilde{N}_{k,1}(t)\frac{\epsilon}{3}\right):\mu_{1,s}\leq\mu_1-\epsilon_0\right)$$
$$\leq\sum_{n=u_\epsilon(T)}^{+\infty}\mathbb{P}\left(\exists s\geq f\left(n\frac{\epsilon}{3}\right):\mu_{1,s}\leq\mu_1-\epsilon_0\right)$$
$$\leq\sum_{n=u_\epsilon(T)}^{+\infty}\sum_{s=f\left(n\frac{\epsilon}{3}\right)}^{+\infty}\mathbb{P}\left(\mu_{1,s}\leq\mu_1-\epsilon_0\right)=\mathcal{O}_\epsilon(1)\,,$$

thanks to Hoeffding's inequality. This last step allows to conclude that $B_2'=\mathcal{O}_\epsilon(1)$, using that $f(s)\geq s$ for any $s\in\mathbb{N}$.

Remark 4.4.1 We can see that at this step $N_1(s)\geq N_k(s)$ was sufficient, the enforcement of $N_1(s)\geq f(N_k(s))$ is necessary only in the "pre-convergence" analysis.

Upper bounding A_2, $\ell_t\neq 1$

For this part of the proof, we mainly use techniques from [31, 69]. In particular, we use that if the current leader is a suboptimal arm then either 1 has already been leader and has lost leadership or 1 has never been leader. Formally, we define the event

$$\mathcal{D}_t=\left\{\exists r\in\left[\frac{t}{4},t\right]\;:\;\ell_r=1\right\}\,.$$

Then, we first upper bound the term

$$C_1\coloneqq\sum_{t=1}^{T-1}\mathbb{E}\left[\mathbb{1}\left(k\in\mathcal{A}_{t+1},\ell_t\neq 1,\mathcal{D}_t\right)\right]\,.$$

We use that if $\mathcal{D}_t$ holds and $\ell_t \neq 1$ then arm 1 was the leader at some point between $t/4$ and t and lost its leadership. Furthermore, a change of leadership from 1 to j at time s can only happen if (1) arm j has been pulled at the previous round, (2) the two arms now satisfy $N_1(s) = N_j(s) = n$ for some $n \geq s/K \geq t/(4K)$, and (3) $\mu_{1,n} \leq \mu_{j,n}$. Thanks to these properties, we obtain with some union bounds that

$$\begin{aligned}
C_1 &\leq \sum_{j=2}^{K} \sum_{t=1}^{T-1} \sum_{s=\lceil t/4 \rceil}^{t} \sum_{n=\lceil s/K \rceil}^{t} \mathbb{E}\left[\mathbb{1}\left(j \in \mathcal{A}_s, N_1(s) = N_j(s) = n, \mu_{j,n} \geq \mu_{1,n}\right)\right] \\
&\leq \sum_{j=2}^{K} \sum_{t=1}^{T-1} \sum_{n=\lceil t/K \rceil}^{t} \mathbb{E}\left[\mathbb{1}\left(\mu_{j,n} \geq \mu_{1,n}\right) \sum_{s=\lceil t/4 \rceil}^{t} \mathbb{1}(j \in \mathcal{A}_s, N_j(s) = n)\right] \\
&\leq \sum_{j=2}^{K} \sum_{t=1}^{T-1} \sum_{n=\lceil t/K \rceil}^{t} \mathbb{P}\left(\mu_{j,n} \geq \mu_{1,n}\right) \\
&= \sum_{j=2}^{K} \sum_{t=1}^{T-1} \sum_{n=\lceil t/K \rceil}^{t} \left(\mathbb{P}\left(\mu_{j,n} \geq x_j\right) + \mathbb{P}\left(\mu_{1,n} \leq x_j\right)\right) = \mathcal{O}(1) \,.
\end{aligned}$$

We now upper bound the term

$$C_2 \coloneqq \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(k \in \mathcal{A}_{t+1}, \ell_t \neq 1, \bar{\mathcal{D}}_t\right)\right] \,.$$

Following [31], we use that if arm 1 has never been leader between $t/4$ and t, then it has necessarily lost at least $t/4$ duels in that time interval. Using Markov inequality, we hence obtain that

$$\begin{aligned}
C_2 &\leq \sum_{t=1}^{T-1} \mathbb{P}\left(\mathbb{1}\left(\sum_{s=t/4}^{t} \mathbb{1}(1 \notin \mathcal{A}_{s+1}, \ell_s \neq 1) \geq \frac{t}{4}\right)\right) \\
&\leq \sum_{t=1}^{T-1} \frac{4}{t} \sum_{s=t/4}^{t} \mathbb{E}\left[\mathbb{1}(1 \notin \mathcal{A}_{s+1}, \ell_s \neq 1)\right] \\
&\leq \sum_{s=1}^{T-1} \left(\sum_{t=1}^{T-1} \frac{4}{t} \mathbb{1}(t \in [s, 4s])\right) \mathbb{E}\left[\mathbb{1}(1 \notin \mathcal{A}_{s+1}, \ell_s \neq 1)\right] \\
&\leq 4\log(4) \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}(1 \notin \mathcal{A}_{t+1}, \ell_t \neq 1)\right] \,.
\end{aligned}$$

Thank to these simple tricks, we are back to upper bounding the total number of duels lost by arm 1 while not being the leader at the cost of a multiplicative constant, which is close to the remaining term in our statement. The last step of this first generic analysis consists in upper bounding the regret caused by $Z_1(t) = 0$ by a constant. We denote by f^{-1} the function satisfying $f^{-1}(f(s)) = s$ for any $s \in \mathbb{N}$. We use that the greedy duel

is caused by either $\mu_{\ell_t}(t) \geq \mu_{\max}$ or $\widetilde{N}_{1,\ell_t}(t) \geq f^{-1}(t/K)$,

$$\begin{aligned}
D_1 &\coloneqq \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(1 \notin \mathcal{A}_{t+1}, \ell_t \neq 1, Z_1(t) = 0\right)\right] \\
&\leq \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(\cup_{j=2}^{K}\left\{\mu_1(t) \leq x_j \cup \mu_j(t) \geq x_j, \widetilde{N}_{1,j}(t) \geq f^{-1}(t/K), N_j(t) \geq t/K\right\}\right)\right] \\
&\quad + \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(\cup_{j=2}^{K}\left\{\mu_j(t) \geq \mu_{\max}\right\}, \ell_t = j\right)\right] \\
&\leq \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}(\mu_1(t) \leq x_j, N_1(t) \geq f^{-1}(t/K))\right] + 2\sum_{j=2}^{K} \mathbb{E}\left[\mathbb{1}(\mu_j(t) \geq x_j, N_j(t) \geq t/K)\right] \\
&\leq \sum_{t=1}^{T-1} \sum_{n=f^{-1}(t/K)}^{t} \mathbb{P}(\mu_{1,n} \leq x_j) + 2\sum_{j=2}^{K}\sum_{t=1}^{T-1}\sum_{n_j=t/K}^{t} \mathbb{P}(\mu_{j,n} \geq x_j) = \mathcal{O}(1) \,,
\end{aligned}$$

where we grouped upper bounded the terms corresponding to $\mu_j(t) \geq \mu_{\max}$ by the terms corresponding to $\mu_j(t) \geq x_j$ for simplicity. Finally, the remaining term of our upper bound is exactly

$$\text{Pre-CV} \coloneqq 4\log(4)\sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(1 \notin \mathcal{A}_{t+1}, \ell_t \neq 1, Z_1(t) = 1\right)\right] \,,$$

as stated in the Lemma 4.4.1. This concludes the proof of the lemma,

$$\mathbb{E}[N_k(T)] \leq u_\epsilon(T) + \text{Post-CV} + \text{Pre-CV} + \mathcal{O}_\epsilon(1) \,.$$

□

We now prove the following lemma, that concludes the proof of Theorem 4.4.1.

Lemma 4.4.2 *OMED and OIMED both satisfy*

$$\textit{Post-CV} = \mathcal{O}_\epsilon(1) \quad \textit{and} \quad \textit{Pre-CV} = \mathcal{O}(1) \,.$$

Proof. We need to upper bound four terms. We start with the upper bounds of the two post-convergence terms, that are straightforward thanks to the tuning of $u_\epsilon(T)$ and the definition of $\mathcal{G}_k(t)$.

Upper bounding Post-CV

We start with OMED,

$$\begin{aligned}
\text{Post-CV}_{\text{OMED}} &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}(k \in \mathcal{A}_{t+1}, \ell_t = 1, \widetilde{N}_{k,1}(t) > u_\epsilon(T), Z_k(t) = 1, \mathcal{G}_k(t))\right] \\
&\leq \left[\sum_{t=1}^{T} e^{-\widetilde{N}_{k,1}(t)(\mathcal{E}_{\mathcal{F}}(F_k,\mu_1)-\epsilon)}\mathbb{1}(\widetilde{N}_{k,1}(t) > u_\epsilon(T), Z_k(t) = 1, \mathcal{G}_k(t))\right] \\
&\leq T e^{-u_\epsilon(T)(\mathcal{E}_{\mathcal{F}}(F_k,\mu_1)-\epsilon)} \\
&\leq 1 \,,
\end{aligned}$$

thanks to the definition of $u_\epsilon(T)$. For OIMED, we obtain

$$\begin{aligned}\text{Post-CV}_{\text{OIMED}} &\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}(k \in \mathcal{A}_{t+1}, \ell_t = 1, \widetilde{N}_{k,1}(t) > u_\epsilon(T), Z_k(t) = 1, \mathcal{G}_k(t))\right] \\ &\leq \left[\sum_{t=1}^{T} \mathbb{1}(\widetilde{N}_{k,1}(t)(\mathcal{E}_{\mathcal{F}}(F_k, \mu_1) - \epsilon) < \log(T), \widetilde{N}_{k,1}(t) > u_\epsilon(T))\right] \\ &= 0 \ ,\end{aligned}$$

again thanks to the definition of $u_\epsilon(T)$.

Upper bounding Pre-CV, OMED

We recall that

$$\text{Pre-CV}_{\text{OMED}} \coloneqq \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(1 \notin \mathcal{A}_{t+1}, \ell_t \neq 1, \widetilde{N}_{1,\ell_t}(t) \leq f^{-1}(N_j(t))\right)\right] \ .$$

We use the notation $p_1(t) = p_{1,j,\widetilde{N}_{1,\ell_t}(t)}$, and $p_{1,j,n} = e^{-L_{1,j,n}}$. Then, with the same arguments as for the previous regret analysis of MED and FMED we obtain that

$$\begin{aligned}\text{Pre-CV}_{\text{OMED}} &= \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(\ell_t \neq 1, \widetilde{N}_{1,\ell_t}(t) \leq f^{-1}(N_j(t))\right)(1 - p_1(t))\right] \\ &= \sum_{t=1}^{T-1} \mathbb{E}\left[\mathbb{1}\left(1 \in \mathcal{A}_{t+1}, \ell_t \neq 1, \widetilde{N}_{1,\ell_t}(t) \leq f^{-1}(N_j(t))\right)\frac{1 - p_1(t)}{p_1(t)}\right] \\ &\leq \sum_{j=2}^{K}\sum_{n=1}^{T} \mathbb{E}\left[\frac{1}{p_{1,j,n}} - 1\right] \\ &\coloneqq \sum_{j=2}^{K}\sum_{n=1}^{T} \mathbb{E}\left[e^{L_{1,j,n}} - 1\right] \ .\end{aligned}$$

Now, thanks to Equation (4.29) we can relate $L_{1,j,n}$ to $\mathcal{E}_{\mathcal{F}}(F_n, x)$ for any $x \in \mathbb{R}$. We again choose x_j for convenience, and obtain that

$$\text{Pre-CV}_{\text{OMED}} \leq \sum_{j=2}^{K}\sum_{n=1}^{T} \mathbb{E}\left[e^{(n\mathcal{E}_{\mathcal{F}}(\widetilde{F}_{1,j,n}, x_j) - R_{1,j,n} - B_{1,j,n}(x_j))_+} - 1\right] \ .$$

To relate to the proof for MED we need the portfolio regret and bias to be small enough. We use the second side of our assumption on the regret, which is that $-R_{1,j,n} = o(n)$ for any j, n. Hence, for n large enough it holds for instance that $-R_{1,j,n} \leq \frac{\delta_j}{3}n$, with

$$\delta_j = \inf_{F \in \mathcal{F}} \mathcal{E}_{\mathcal{F}}(F, \mu_1) - \mathcal{E}_{\mathcal{F}}(F, x_j) > 0 \ ,$$

where $\delta_j > 0$ is ensured by property 3. of Lemma 4.2.1. For the bias, we first obtain with Lemma 4.4.2 that

$$-B_{1,j,n}(x_j) \leq \sum_{i=1}^{n} \mathbb{1}(\mu_j(t_{1,j,n}) \geq x_j) \ .$$

We then define a "good event" under which the bias is controlled,

$$\mathcal{B}_{j,n} = \left\{ \sum_{i=1}^{n} \mathbb{1}(\mu_j(t_{1,j,n}) \geq x_j) \leq n \frac{\delta_j}{3} \right\} .$$

We then use a similar proof as for the post-convergence term. We first consider

$$P_1 \coloneqq \sum_{j=2}^{K} \sum_{n=1}^{T} \mathbb{E}\left[\left(e^{L_{1,j,n}} - 1 \right) \mathbb{1}(\mathcal{B}_{j,n}) \right] = \sum_{j=2}^{K} \sum_{n=1}^{T} \mathbb{E}\left[e^{n\mathcal{E}_{\mathcal{F}}(\widetilde{F}_{1,j,n}, x_j) + n\frac{2\delta_j}{3}} - 1 \right] .$$

Using properties 3. and 4. of Lemma 4.2.1, we obtain that

$$\begin{aligned} P_1 &\leq \sum_{j=2}^{K} \sum_{n=1}^{T} \int_0^{e^{n\mathcal{K}_j^+ + \frac{2\delta_j}{3}} - 1} \mathbb{P}\left(\mathcal{E}_{\mathcal{F}}(\widetilde{F}_{1,j,n}, x_j) > \frac{\delta_j}{3} + \frac{1}{n}\log(1+x) \right) \mathrm{d}x \\ &\leq \sum_{j=2}^{K} \sum_{n=1}^{T} en(n+1) \left(\mathcal{K}_j^+ + \frac{2\delta_j}{3} \right) e^{-n\frac{\delta_j}{3}} \\ &= \mathcal{O}(1) , \end{aligned}$$

with $\mathcal{K}_j^+ = \log\left(\frac{B-b}{B-x_j}\right)$. We then tackle the case where $\mathcal{B}_{j,n}$ does not hold, which is possible only if $\mu_{j,s} \geq x_j$ for some $s \geq (n\delta_j)^2$. Furthermore, we also use the trivial bound $-B_{1,j,n}(x_j) \leq n$ thanks to Lemma 4.4.2. We then obtain

$$\begin{aligned} P_2 &\coloneqq \sum_{j=2}^{K} \sum_{n=1}^{T} \mathbb{E}\left[\left(e^{(n\mathcal{E}_{\mathcal{F}}(\widetilde{F}_{1,j,n}, x_j) - R_{1,j,n} - B_{1,j,n}(x_j))_+} - 1 \right) \mathbb{1}(\bar{\mathcal{B}}_{j,n}) \right] \\ &\leq \sum_{j=2}^{K} \sum_{n=1}^{T} \mathbb{E}\left[\left(e^{n\mathcal{E}_{\mathcal{F}}(\widetilde{F}_{1,j,n}, x_j) + n(1+\delta_j)} - 1 \right) \mathbb{1}(\exists s \geq f(n\delta_j) : \mu_{j,s} \geq x_j) \right] \\ &\leq \sum_{j=2}^{K} \sum_{n=1}^{T} \mathbb{E}\left[\left(e^{n\mathcal{E}_{\mathcal{F}}(\widetilde{F}_{1,j,n}, x_j) + n(1+\delta_j)} - 1 \right) \frac{e^{-f(n\delta_j) I_1(x_j)}}{1 - e^{-I_1(x_j)}} \right] , \end{aligned}$$

We can conclude at this step using that $e^{n(1+\delta_j) - f(n\delta_j) I_1(x_j)} \to 0$, as we ensured that $n = o(f(n))$, so it simply holds that

$$P_1 = \mathcal{O}(1) \Rightarrow P_2 = \mathcal{O}(1) .$$

Upper bounding Pre-CV, OIMED

We again consider the two cases, depending on if $\mathcal{B}(t)$ holds or not. When this is true, we can use a similar proof to the one of IMED that we previously presented,

$$\begin{aligned}
Q_1 &\coloneqq \sum_{j=2}^{K}\sum_{t=1}^{T-1}\mathbb{E}\left[\mathbb{1}\left(\widetilde{N}_{1,j}(t)\mathcal{E}_{\mathcal{F}}(\widetilde{F}_{1,j,n}, x_j) + \log(\widetilde{N}_{1,j}(t)) \geq \log(N_j(t)) - \widetilde{N}_{1,j}(t)\frac{2\delta_j}{3}, N_j(t) \geq t/K\right)\right] \\
&\leq \sum_{j=2}^{K}\sum_{t=1}^{T-1}\sum_{n=\log(t/K)/\mathcal{K}^+}^{T-1}\mathbb{P}\left(n\mathcal{E}_{\mathcal{F}}(\widetilde{F}_{1,j,n}, x_j) \geq \log(t/K) - n\frac{2\delta_j}{3} - \log(n)\right) \\
&\leq \sum_{j=2}^{K}\sum_{t=1}^{T-1}\sum_{n=\log(t/K)/\mathcal{K}^+}^{T-1}\frac{Ken(n+2)}{t}e^{-n\frac{\delta_j}{3}} \\
&\leq K^2 e\sum_{t=1}^{T-1}\sum_{n=\log(t/K)/\mathcal{K}^+}^{T-1}\frac{n(n+2)}{t}e^{-n\frac{\delta_j}{3}}\,,
\end{aligned}$$

where we again used properties 3. and 4. of Lemma 4.2.1 to upper bound the probability. The convergence of this term is ensured by the fact that the sum on n starts at $\frac{\log(t/K)}{\mathcal{K}^+}$. To formally prove this we can for instance use that there exists a constant C_j (that typically scales in $\mathcal{O}(\delta_j^{-2})$ up to logarithm terms) such that $\forall n \in \mathbb{N}, n(n+2) \leq C_j e^{n\frac{\delta_j}{6}}$ for any $n \in \mathbb{N}$, so

$$\begin{aligned}
Q_1 &\leq C_j K^2 \sum_{t=1}^{T-1}\sum_{n=\log(t/K)/\mathcal{K}^+}^{T-1}\frac{e^{-n\frac{\delta_j}{6}}}{t} \\
&\leq C_j K^{2+\frac{\delta_j}{3\mathcal{K}^+}}\frac{1}{1-e^{-\frac{\delta_j}{6}}}\sum_{t=1}^{T-1}\frac{1}{t^{1+\frac{\delta_j}{6\mathcal{K}^+}}} \\
&= \mathcal{O}(1)\,.
\end{aligned}$$

We then consider $\bar{\mathcal{B}}(t)$, under which the bias can be up to $\widetilde{N}_{1,j}(t)$. In that case, we use the same proof techniques as for OMED, with

$$\begin{aligned}
Q_2 &\coloneqq \sum_{j=2}^{K}\sum_{t=1}^{T-1}\mathbb{E}\left[\mathbb{1}(\widetilde{N}_{1,j}(t)\mathcal{E}_{\mathcal{F}}(F_{1,j,n}, x_j) \geq \log(N_j(T)) - \widetilde{N}_{1,j}(t)(1+\delta_j), N_j(t) \geq t/K, \bar{\mathcal{B}}(t))\right] \\
&\leq \sum_{j=2}^{K}\sum_{t=1}^{T-1}\sum_{n=\log(t/K)/\mathcal{K}^+}^{T-1}\frac{K}{t}e^{-n\delta_j} \times \frac{e^{n\left(1+\frac{\delta_j}{3}\right)-(n\delta_j)^2 I_1(x_j)}}{1-e^{-I_1(x_j)}}\,.
\end{aligned}$$

We then use that the right-hand term converges to 0, so $Q_1 = \mathcal{O}(1) \Rightarrow Q_2 = \mathcal{O}(1)$. □

Remark 4.4.2 The enforcement of a sufficient sample size for the leader is justified by the analysis of the pre-convergence term. In this regime a linear bias may cost up to e^n for each sample size n. Hence, a concentration of the mean with a rate e^{-cn} is not sufficient, especially if c is small (which happens if the gaps are small). Furthermore, this kind of rate cannot be avoided without actively controlling the sample size of the leader at each update.

This concludes the proof of Theorem 4.4.1. □

Prior to benchmarking the introduced algorithms, and check if OMED and OIMED deliver the promises of the premises, one must briefly describe the Soft-Bayes portfolio algorithm that we use.

Soft-Bayes

The Soft-Bayes algorithm is proposed in [65]. We recall that the $\mathcal{E}_{\mathcal{F}}$ estimation is a portfolio optimization problem of dimension 2. At each step n, the portfolio algorithm decides an allocation $(1-\lambda_n, \lambda_n)$ between two assets, that provide a payoff $(1, Y_n)$. We use in this section Y_n to denote for simplification $\frac{B-X_{k,\ell,n}}{B-\mu^\star(t_{k,\ell,n})}$, that is used in our implementation of the estimate $L_{k,\ell,n}$ for the challenger/leader pair (k,ℓ).

[65]: Orseau et al. (2017), 'Soft-bayes: Prod for mixtures of experts with log-loss'

For the anytime version of the algorithm, a sequence of learning rates $(\eta_n)_{n\in\mathbb{N}}$ is provided as an input of the algorithm. We then define the update rule of the anytime Soft-Bayes in Algorithm 19 below.

Algorithm 19: Anytime Soft-Bayes algorithm

Input: Parameter λ_n, sequence of learning rates $(\eta_n)_{n\in\mathbb{N}}$, initial parameter $\lambda_1 = 1/2$;

1 **return** $\lambda_{n+1} = \lambda_n \times \left(1 - \eta_n + \eta_n \frac{Y_n}{1-\lambda_n(1-Y_n)}\right) + \left(1 - \frac{\eta_{n+1}}{\eta_n}\right)\lambda_1$;

In the paper [65], it is proved that defining the learning rate as $\eta_n = \sqrt{\frac{\log(2)}{4n}}$ ensures an upper bound on the portfolio regret

$$\mathcal{R}_N = 4\sqrt{\log(2)N} + (1+\log(2))\log(N+1) + \log(2)\ .$$

However, the authors do not provide a lower bound on the regret.

4.5 Empirical results

Benchmarks We numerically assess the regret and run time of the two novel approaches presented in this paper, FMED/FIMED and OMED/OIMED. We benchmark our approaches with the known asymptotically optimal algorithms: IMED [16], KL-UCB [21] and NPTS [26]; and the computationally more efficient but suboptimal UCB [5], kl-UCB [21] and IMED-kl (that denotes the binarized version of IMED). Recall that pseudo-code of those algorithms can be found in Chapter 3 of this thesis. We showcase our main findings by presenting a selection of experiments. In each experimental setting, we plot the average run time and regret of each algorithm, along with their quantiles 10%-90%. The run times are dependent of the Python implementation of our algorithms, and are thus only indicative.

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

[21]: Cappé et al. (2013), 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'

[26]: Riou et al. (2020), 'Bandit algorithms based on Thompson sampling for bounded reward distributions'

[5]: Auer et al. (2002), 'Finite-time analysis of the multiarmed bandit problem'

DSSAT experiments We first consider the crop-management optimization DSSAT problem that was presented in Figure 4.2 and the unlikelihood of optimality ratio presented in Table 4.2 were a practical justification of the work presented in this chapter. Table 4.2 illustrates the theoretical advantage of asymptotically optimal algorithms. Recall

that in this problem, a learner needs to select a planting date for maize grains among seven possible options. Numerically, for each distribution, the rewards are drawn uniformly at random among 10^4 points sampled from the DSSAT simulator, in order to emulate the distributions at a reduced cost. We plot the regret curves and average run time of tested algorithms, Figure 4.5 as well as Table 4.7 that compiles the average regret and average run time on the DSSAT experiment at the horizon 10 000.

Results displayed in Figure 4.5, show that on this problem optimal algorithms also achieve better empirical performance. IMED, NPTS and KL-UCB perform similarly, while IMED-kl and UCB achieve significantly larger regret. Furthermore, FIMED is 10 times faster than IMED for $T = 10^4$ (and as fast as kl-UCB), with almost no difference in terms of regret. As expected, OIMED is as fast as UCB and IMED-kl. It achieves better regret than those algorithms, but its performance is deteriorated compare to IMED. This can be seen as the cost of the ability to forget past observations. This experiment illustrates the respective benefits of our two novel approaches: FIMED is the fastest algorithm among those with the smallest regret, while OIMED has the smallest regret among the fastest algorithm. Reading this table, it is clear that FIMED improves the run time of IMED without compromising the regret and that OIMED improves the run time even further, close to the one of UCB, but at the cost of loosing a bit in the regret term. However, of the fastest methods (those in $\mathcal{O}(1)$ run time), OIMED clearly has the smallest regret. If the practitioner is willing to pay a bit more time complexity and space complexity, then FIMED seems to be the best algorithmic method.

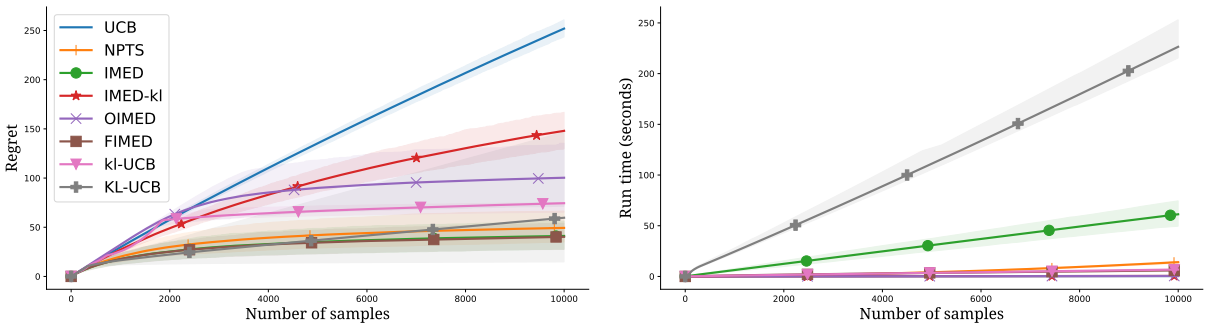

Figure 4.5: Average regret (left) and run time (right) of the algorithms on the DSSAT bandit problem

Table 4.7: Average regret and run time at horizon 10 000 on DSSAT

Algorithm	UCB	NPTS	IMED	IMED-kl	OIMED	FIMED	kl-UCB	KL-UCB
Regret	252	49	41	148	100	40	74	60
Run time (sec.)	0.38	14	61	0.51	0.47	6	6.8	226

One can also see in Table 4.7 that the empirical unlikelihood of Table 4.2 really translates into differences of performances of the corresponding algorithms. The ordering of algorithmic methods, symbolically

$$\Delta^2 < kl < \mathcal{E}_{\mathcal{F}},$$

is the same theoretically, predicted by the ratio Table 4.2, and by the experimental regret shown in Table 4.7.

Bernoulli Bandits In this second experimental setting, we consider a Bernoulli bandit with several means close to 0.5. In this setting, all optimal algorithms match their implementation with the Bernoulli kl-divergence, *i.e.* IMED-kl ~ IMED and KL-UCB ~ kl-UCB. Intuitively, sequences of 0 and 1 with high variance may lead to the most potentially confusing inputs for portfolio algorithms, so our objective is to check the performance of OIMED in that case. In this setting, the variance of samples is close to maximal and initial mistakes are frequents in the sense that based on the first samples, it is easy to mistake a suboptimal arm for an optimal arm. Strategies that are too greedy too early or cannot correct their indexes fast enough surely would be penalized by a large regret. To check the numerical soundness of OIMED, we therefore run an experiment on such a setting.

Our results, summarized in Figure 4.6, are promising: the average regret of OIMED is on par with other algorithms, while it still is among the fastest. We remark that its 90% quantile is larger than the others, but still exhibits a logarithmic shape. In that experiment only UCB is suboptimal, and performs much worse than the other algorithms. Analysis of Figure

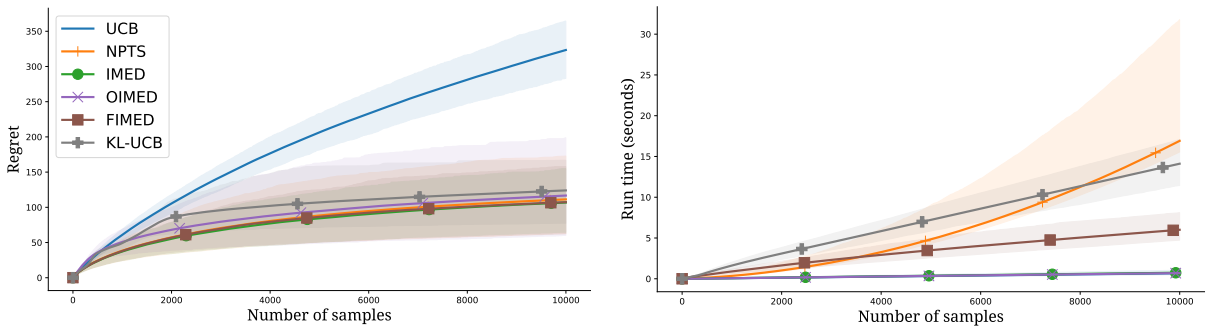

Figure 4.6: Average regret (left) and run time (right) of the algorithms on a 6-arms Bernoulli bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$.

4.6 shows that, except for UCB, regret performances are similar among tested algorithms and the analysis of run times lead to the same conclusion as the analysis of the DSSAT experiment (Figure 4.5).

We first looked at a Bernoulli bandit problem where all the means are close to 0.5, Figure 4.6, identified as a difficult problem because the variance of the distributions in maximized among those supported in $[0, 1]$. We explore two more Bernoulli bandit problems, one where the means are located close to 1 (Figure 4.7, Table 4.8) and another hereafter where means are located close to 0 (Figure 4.8, Table 4.9).

Table 4.8: Average regret and run time at horizon 10 000 on on a 6-arms Bernoulli bandit problem with means $\{0.4, 0.6, 0.7, 0.85, 0.9, 0.95\}$

Algorithm	UCB	NPTS	IMED/IMED-KL	OIMED	FIMED	kl-UCB/KL-UCB
Regret	289	33	30	43	30	95
Run time (sec.)	0.51	15	0.74	0.67	4.2	7

We recall that for Bernoulli bandit, IMED and IMED-kl are the same algorithms (we assume that IMED is implemented as IMED-kl while only 0s and 1s are observed). For FIMED we could have done the same

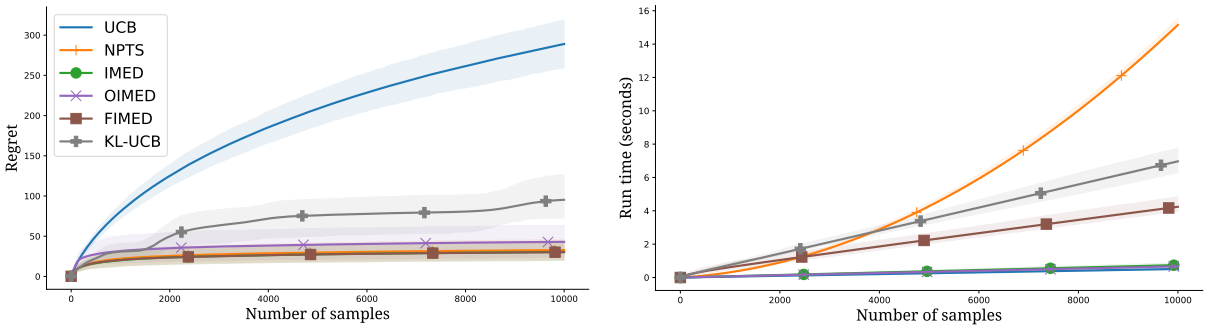

Figure 4.7: Average regret (left) and run time (right) of the algorithms on a 6-arms Bernoulli bandit problem with means $\{0.4, 0.6, 0.7, 0.85, 0.9, 0.95\}$.

thing to improve its run time without changing its regret in order to emphasize that FIMED is always faster (or as fast as) kl-UCB, even when using the Bernoulli kl. While the regret and time values change from one experiment to another, the conclusions that can be drawn from them do not, especially when considering the confidence intervals represented by the 10%-90% quantiles. Experimentally, OIMED (or OMED, see Section 4.5.1) should be the preferred $\mathcal{O}(1)$ method and FIMED (or FMED, see Section 4.5.1) should be preferred if we really target an empirically optimal regret without compromising too much on the running time.

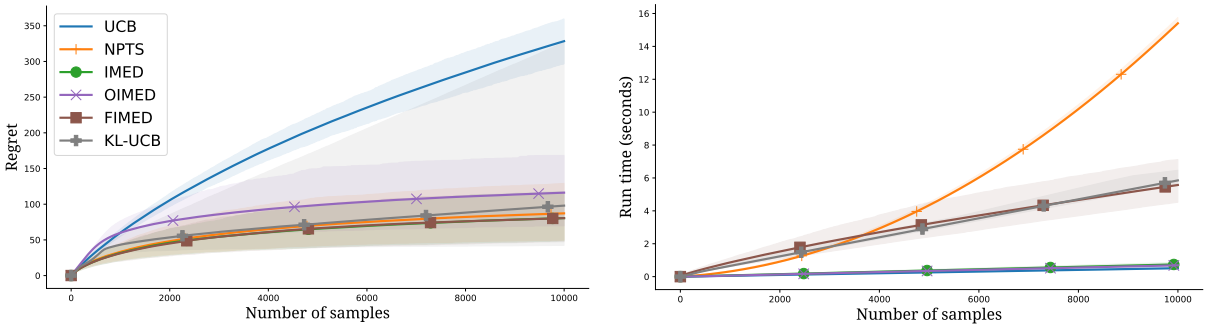

Figure 4.8: Average regret (left) and run time (right) of the algorithms on a 6-arms Bernoulli bandit problem with means $\{0.05, 0.1, 0.15, 0.2, 0.22, 0.25\}$

Table 4.9: Average regret and run time at horizon 10 000 on a 6-arms Bernoulli bandit problem with means $\{0.05, 0.1, 0.15, 0.2, 0.22, 0.25\}$

Algorithm	UCB	NPTS	IMED/IMED-KL	OIMED	FIMED	kl-UCB/KL-UCB
Regret	328	87	82	116	81	98
Run time (sec.)	0.51	15.4	0.75	0.69	5.6	5.9

Beta bandits Beta distribution of a given mean can have different shapes. In particular, a **Beta** distribution can be close to a **Bernoulli** distribution (shape parameter close to zero) with most of the density located around 0 and 1, close to a **Dirac** distribution (shape parameter significantly larger than one) with most of the density located around the mean, and close to a truncated **Gaussian** distribution (shape parameter larger than one) with the characteristic bell shape distribution around the mean.

In the previous experimental setting, we studied a Bernoulli bandit problem where all the means are close to 0.5 (Figure 4.6). Here, using the same set of means, $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$, we run two experiments, one on Beta distributions with an intermediate shape parameter of 5 (Figure 4.9) and another where Beta distributions have a large shape parameter of 50 (Figure 4.11), hence with highly piked distributions. We do so to illustrate the effect of changing the shape of distributions without changing the set of means.

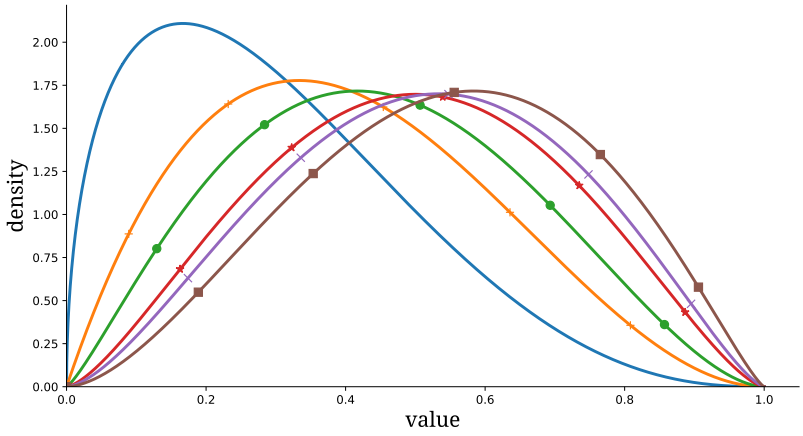

Figure 4.9: Beta bandit distributions with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 5

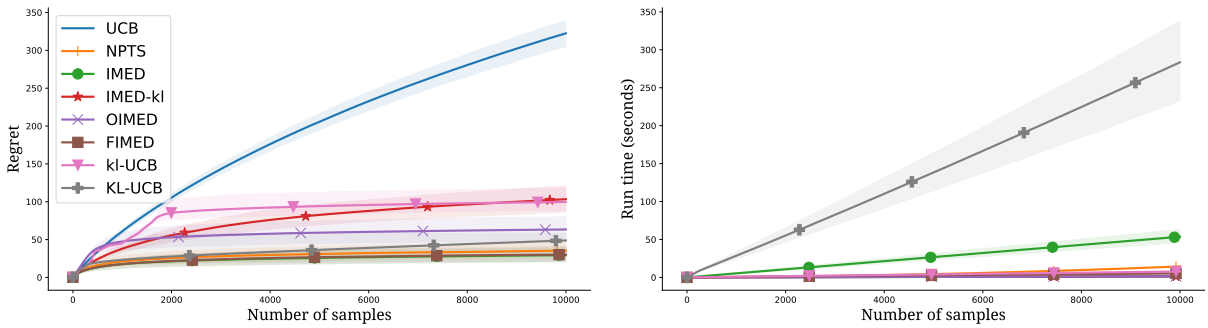

Figure 4.10: Average regret (left) and run time (right) of the algorithms on a 6-arms Beta bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 5

Table 4.10: Average regret and run time at horizon 10 000 on a 6-arms Beta bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 5

Algorithm	UCB	NPTS	IMED	IMED-KL	OIMED	FIMED	kl-UCB	KL-UCB
Regret	322	35	29	103	63	30	100	49
Run time (sec.)	1.2	14.5	53.5	1.5	1.4	5.2	7.7	284

Comparing the regret curves of Figure 4.6, Figure 4.10, and Figure 4.12, it is clear that the Bernoulli distributions does induce the larger variance on the regret curves as we identified while the Beta distributions with the largest shape parameter (more concentrated around their means) induce the smallest variance on the regret curve. This, once again, confirms our theoretical findings. While the order of regret curves is globally preserved, one can see on Figure 4.6 that, except for UCB, the Bernoulli experiment makes all curves to be very similar, which is normal since,

except for UCB, all algorithms are roughly solving the same problem (all sample are 0s and 1s). When dealing with Beta distributions, IMED is different from IMED-kl and KL-UCB is different from kl-UCB. Those

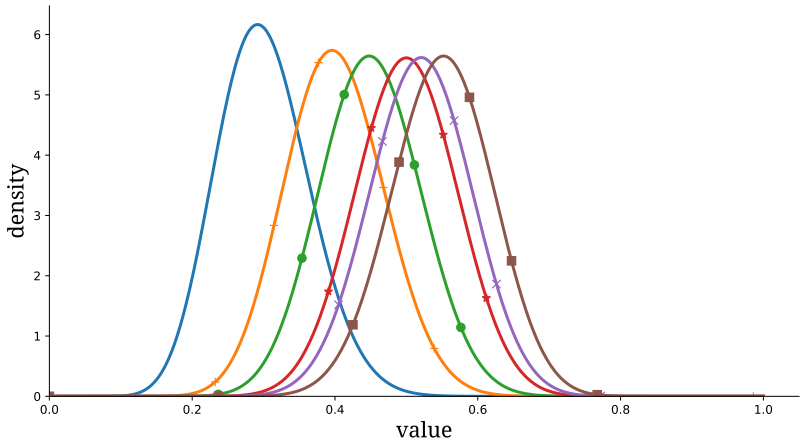

Figure 4.11: Beta bandit distributions with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 50

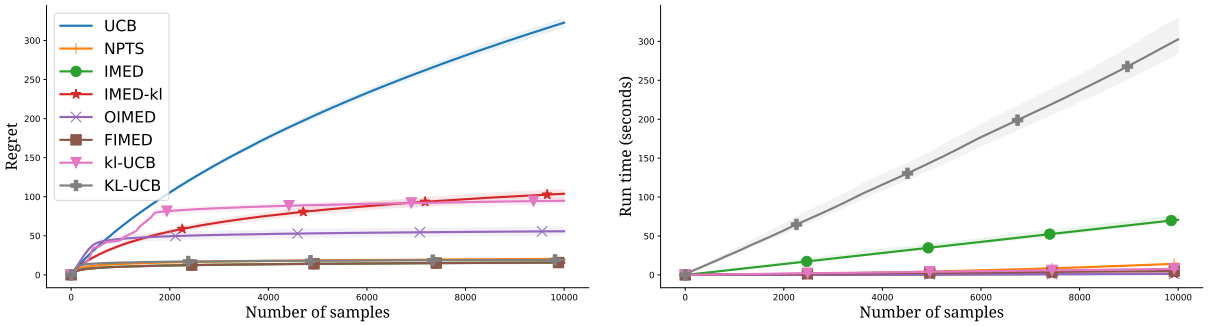

Figure 4.12: Average regret (left) and run time (right) of the algorithms on a 6-arms Beta bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 50

Table 4.11: Average regret and run time at horizon 10 000 on on a 6-arms Beta bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 50

Algorithm	UCB	NPTS	IMED	IMED-KL	OIMED	FIMED	kl-UCB	KL-UCB
Regret	323	20	15.9	103.9	55.8	15.8	95	19.2
Run time (sec.)	1.2	14.4	70.8	1.5	1.4	5	7.8	303

experiments confirm our numerical findings that OIMED (or OMED, see Section 4.5.1) should be the preferred method with $\mathcal{O}(1)$ computation time, and FIMED (or FMED, see Section 4.5.1), should be preferred if we really target an empirically optimal regret without compromising too much on the running time.

4.5.1 Comparison of MED and IMED versions

In this paper, we derived algorithmic and theoretical results for modification of both IMED and MED algorithms. In this section, we compare

IMED, its modifications FIMED and OIMED, MED, and its modifications FMED and OMED. The comparison is made on three experimental setting, the DSSAT bandit problem, centered means Bernoulli bandit problem, as well as the Beta bandit problem with an intermediate shape parameter of 5 and centered means equal to the Bernoulli setting. We show that FMED and OMED exhibit the same characteristics as FIMED and OIMED, as expected. In Figure 4.13 and Table 4.12, we observe that

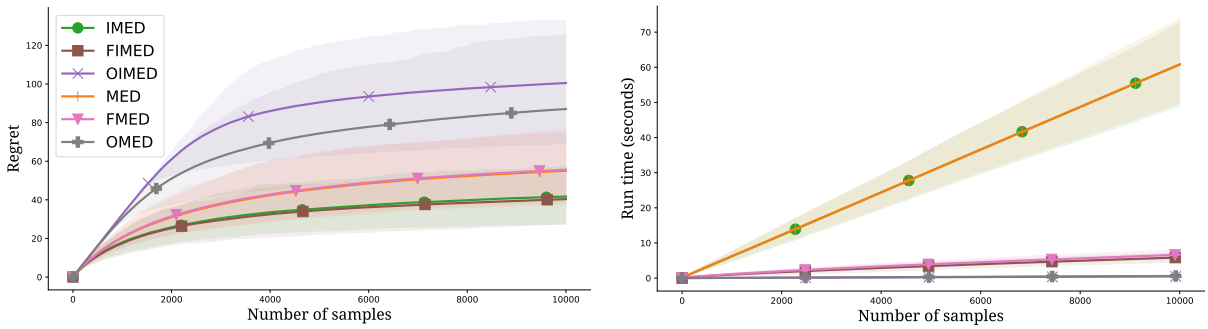

Figure 4.13: Average regret (left) and run time (right) of the algorithms on DSSAT

Table 4.12: Average regret and run time at horizon 10 000 of the algorithms on DSSAT

Algorithm	IMED	FIMED	OIMED	MED	FMED	OMED
Regret	41.7	40.4	100.5	55.1	55.6	87
Run time (sec.)	60.8	5.9	0.5	60.8	6.6	0.6

IMED slightly overperforms MED. Consequently, FIMED also slightly overperforms FMED for the regret metric. Regarding the running time, the MED versions of algorithms are a bit slower than the IMED versions due to the numerical complexity of sampling, larger than that of computing the *argmin* of IMED indexes. Interestingly, OMED overperforms OIMED in terms of average regret performance. As we can see in Figure 4.14 and Table 4.13 presenting the results of the Bernoulli experiment, this is not a general rule.

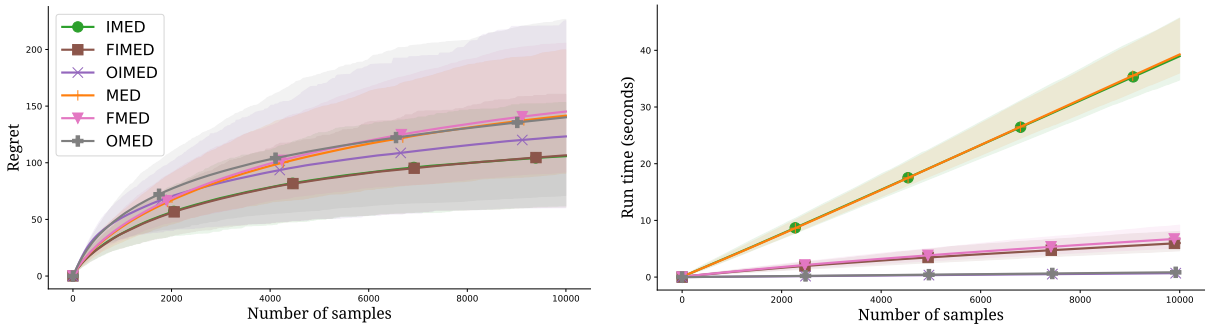

Figure 4.14: Average regret (left) and run time (right) of the algorithms on a 6-arms Bernoulli bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$

The experiment on Beta distribution, Figure 4.15 and Table 4.14, confirms that the regret loss incurred by the transition from MED to OMED is smaller than the one of transitioning from IMED to OIMED. It may be because the sampling strategy of MED is better to handle the statistical

Table 4.13: Average regret and run time at horizon 10 000 of the algorithms on a 6-arms Bernoulli bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$

Algorithm	IMED	FIMED	OIMED	MED	FMED	OMED
Regret	105.8	106.6	123.3	141.5	145.2	140.1
Run time (sec.)	39	6	0.7	39.3	6.8	0.8

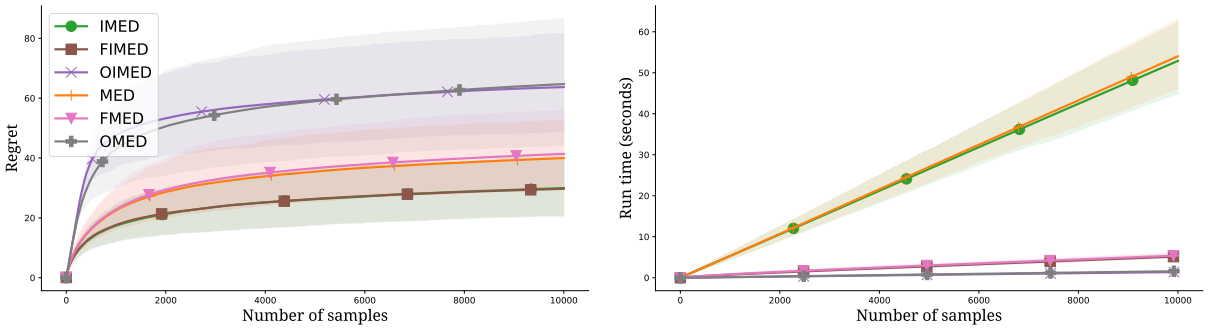

Figure 4.15: Average regret (left) and run time (right) of the algorithms on a 6-arms Beta bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 5

Table 4.14: Average regret and run time at horizon 10 000 of the algorithms on a 6-arms Beta bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 5

Algorithm	IMED	FIMED	OIMED	MED	FMED	OMED
Regret	30	29.7	63.7	40	41.4	64.7
Run time (sec.)	53	5.2	1.4	54	5.4	1.6

approximation of portfolio algorithms suggesting that the sampling strategy in MED may be a key to better handle statistical approximation induced by portfolio algorithms. A better understanding the theoretical differences between MED and IMED will surely help to understand this behavior and help craft better fully online $\mathcal{O}(1)$ time and space complexity in the future.

4.5.2 Stability of OIMED with respect to the learning rate

[65]: Orseau et al. (2017), 'Soft-bayes: Prod for mixtures of experts with log-loss'

The portfolio algorithm, Soft Bayes, behind OIMED and OMED depends on a hyperparameter, η, the learning rate. A learning rate scheme is prescribed in Theorem 3 of [65] with

$$\eta(n) = \sqrt{\frac{\log(2)}{4n}}$$

where n is the number of collected samples. In section, we test the numerical stability of our OIMED algorithm with respect to the learning rate. To do so, we will replace this original η by

$$\eta_r(n) = \sqrt{\frac{r\log(2)}{4n}}$$

where r will range from 0.01 to 100. We illustrate the stability of OIMED on three bandit settings: the DSSAT bandit problem, the centered Bernoulli problem and a Beta bandit problem where all the means are centered around 0.5 and the same as in the Bernoulli bandit.

On Figure 4.16, the average regret ranges from 103.5 for the smallest values of the parameter r to 88.5 for the largest value of r with an intermediate regret of 99.5 for our original value $r = 1$. On this same plot, it is interesting to see that despite the large range of tested parameter size, the quantile tubes remain of roughly the same size with no evident bad behavior. While the effect of changing r may seem important, in the range of tested parameter, the effect of changing r would not have affected the ranking of algorithms in the original experiment, Figure 4.5. This experiment shows the stability of OIMED with respect to the learning rate, in the sense that a multiplicative constant does not seem to be able to dramatically deteriorate its performance.

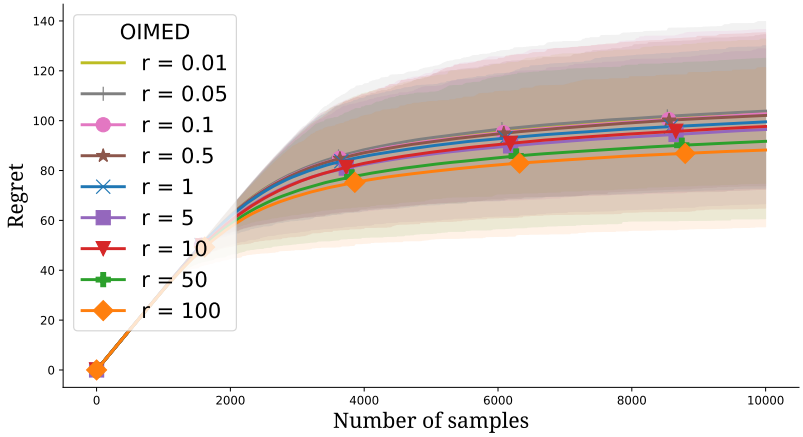

Figure 4.16: Average regret of the algorithms on DSSAT

Those findings are confirmed on a second experiment performed on a Bernoulli bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$. The regret changes from 107 to 124 when varying the parameter r from 0.01 to 100, and, while the upper quantile is larger for small value of r, we still observe a logarithmic curve which confirms the stability of OIMED.

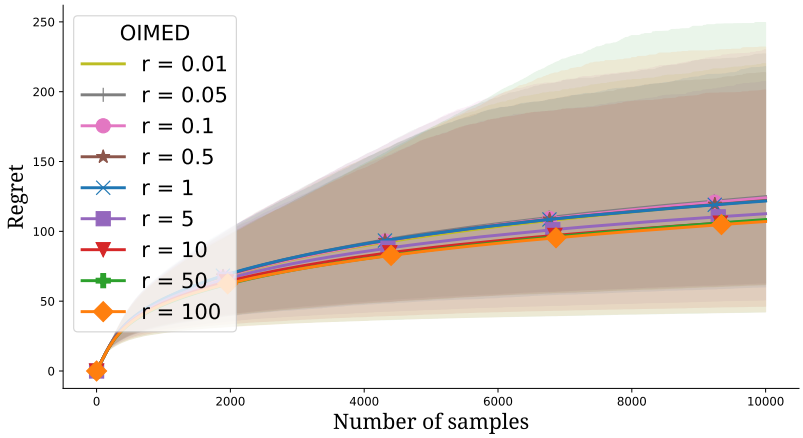

Figure 4.17: Average regret at horizon 10 000 of the algorithms on a 6-arms Bernoulli bandit problem with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$

Finally, we check the effect of the learning rate on two Beta bandit problems where means are $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and one problem has Beta distributions with shape parameter 5 and the other with shape parameter 50.

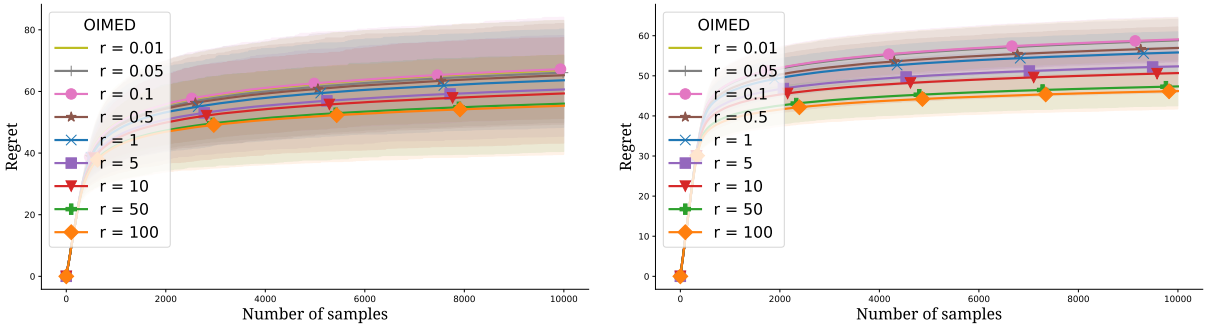

Figure 4.18: Average regret of the algorithms on two 6-arms Beta bandit problems with means $\{0.3, 0.4, 0.45, 0.5, 0.52, 0.55\}$ and shape parameter of 5 (left) and 50 (right)

When the variance of the distributions is smaller (large shape parameter), we can see that regret curves are better separated and the bonus of choosing a larger learning rate parameter r is amplified. When the variance is larger (small shape parameter), regret curves are closer to each other and more within each others quantile tubes. This last experiment confirms all our previous findings about the numerical stability of OIMED.

4.5.3 IMED with discretized rewards

In the introduction of this chapter, we introduced a known trick to reduce the time and memory complexities of MED/IMED. It consists in using algorithms designed for multinomial rewards, by using a discretization procedure on the collected rewards. We furthermore proved its inevitable suboptimality for some problems. We end this experimental section by experimenting with multinomial IMED, comparing the empirical performance and computation time of several instances using different subdivisions. In each case, we fix a number of ticks p, and use the regular subdivision $\left\{0, \frac{1}{p-1}, \ldots, \frac{p-2}{p-1}, 1\right\}$ ($\{0, 1\}$ if $p = 2$). Our objective is to identify the number of ticks needed to get a regret close to IMED, and the evolution of the computation time with the number of ticks. We first run an experiment on the DSSAT bandit problem, and then on a Beta bandit problem where means are close to zero.

Table 4.15: Average regret and run time at horizon 10 000 of the algorithms on DSSAT

Alg. (ticks)	IMED (2)	IMED (3)	IMED (5)	IMED (7)	IMED (10)	IMED (20)	IMED
Regret	151.5	75.7	50.9	45.2	44.1	40.25	40.25
Run time	32.8	33.1	36.8	39.4	42.2	44.8	61.1

Unsurprisingly, the more ticks, the finer the subdivision, the better (smaller) the regret and the larger the running time, with IMED and IMED-kl providing the range for both metrics. For this problem, $p = 20$

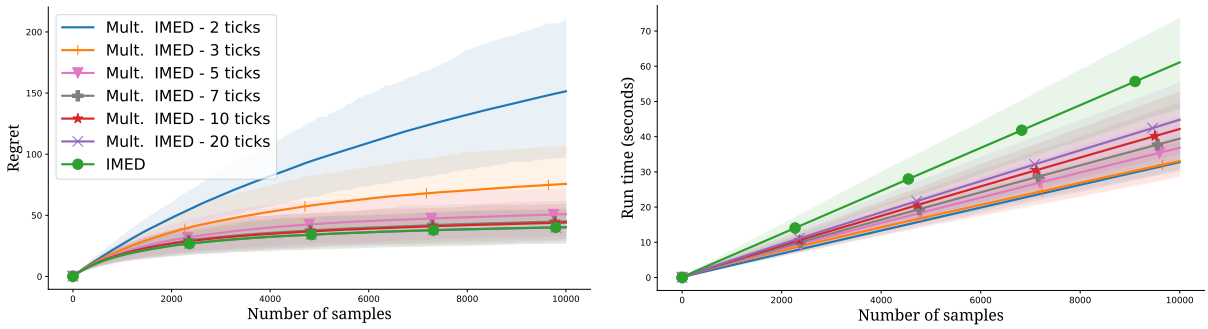

Figure 4.19: Average regret (left) and run time (right) of the algorithms on DSSAT

(intervals of length ≈ 0.05 between each tick) seems to be enough to make the algorithm using discretized rewards almost match the true IMED in terms of regret, while being 25% faster. Recall that FIMED is 10 times faster than IMED, and these experiments suggest that combining FIMED with discretization may further decrease the computation time.

We remark that the evolution of the time complexity per time step is not linear in the number of ticks. This is (likely) because it is actually dependent of the effective number of ticks used in memory, *i.e.* those that contain at least 1 samples. Getting at least 1 sample for each tick/arm, if possible, takes some time, that depends on the shape of the distributions on the $[0, 1]$ support. If the considered distribution has zero mass on a sub-interval $(i, i + 2)$ of the subdivision, then the tick $i + 1$ cannot have any associated sample. The running time is dependent on the model complexity (number of ticks *a priori*) and the problem modelled (number of ticks with positive projected mass *a posteriori*). On a recreational mathematics note, the effective number of ticks used in memory is connected to the coupon collector's problem. Therefore, on average, one cannot hope to observe the linear behavior of the complexity as a function of the number of ticks if the number of collected samples for each suboptimal arm is $p \log p$ for each tested size p. Because suboptimal arms a are sampled $c_a \log T$, it means that one should select a horizon satisfying the equation $\sum_{a \neq \star} c_a \log T = c \log T \simeq |A| p \log p$, *i.e.* $T = \exp(\frac{|A|}{c} p \log p)$ which is an exponential function of the subdivision size p for distributions dominating the Lebesgue measure. For $p = 20$ and $\frac{|A|}{c} = 1$, this means a horizon of 10^{26}.

We run a final experiments on a Beta bandit problem that confirms our findings. IMED and IMED-kl again provide the range for both computation time and regret, and for this problem $p = 20$ is also the value from which IMED and discretized IMED start to match in terms of regret, while the latter is approximately 20% faster.

Table 4.16: Average regret and run time at horizon 10 000 of the algorithms on a 6-arms Beta bandit problem with means $\{0.05, 0.1, 0.15, 0.2, 0.22, 0.25\}$ and shape parameter of 5

Alg. (ticks)	IMED (2)	IMED (3)	IMED (5)	IMED (7)	IMED (10)	IMED (20)	IMED
Regret	81.8	43.8	32.9	30.8	29.6	29.2	29.2
Run time	31.3	41.5	45.5	46.4	49.2	51.2	62.5

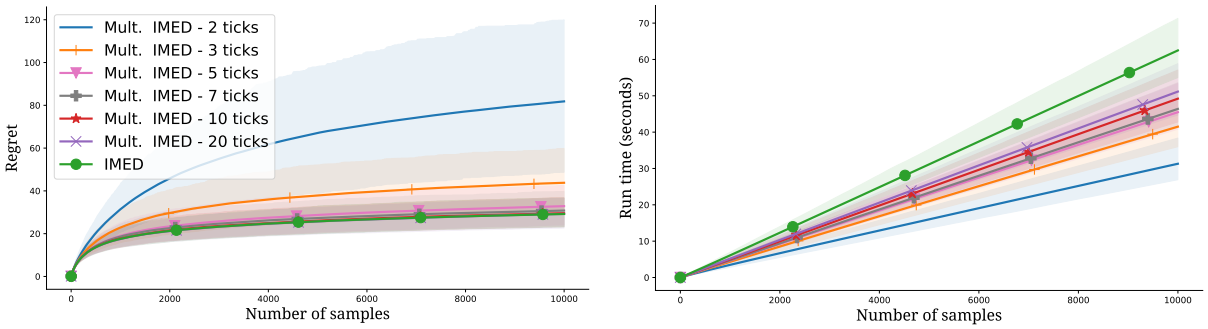

Figure 4.20: Average regret (left) and run time (right) of the algorithms on a 6-arms Beta bandit problem with means $\{0.05, 0.1, 0.15, 0.2, 0.22, 0.25\}$ and shape parameter of 5

4.6 Conclusion

In this chapter, we introduced methods to compute efficiently approximations of $\mathcal{E}_{\mathcal{F}}$ and demonstrated their use in algorithms for regret minimization in stochastic bandits. The FMED and FIMED variants have the same asymptotic optimality properties as the base MED algorithms, but have a much reduced computational complexity. OMED and OIMED push the computational gains further and are also memory efficient. They can also be asymptotically optimal, under the hypothesis that the portfolio algorithm they use satisfies a deterministic regret lower bound.

While our experiments show the good practical performance of OIMED and OMED with the Soft-Bayes portfolio algorithm, this question is however still open: can we have a portfolio algorithm which is computationally efficient, does not store all past gains in memory and has sublinear regret upper and lower bounds?

Finally, our work towards enabling the use of $\mathcal{E}_{\mathcal{F}}$ in a computationally efficient way has potential applications beyond regret minimization in stochastic bandits. First, similar quantities are used in the IMED-RL algorithms that we present in Chapter 7. Second, other bandit tasks like best arm identification also have complexities that depend on a $\mathcal{E}_{\mathcal{F}}$ quantity, and could benefit from faster variants of the algorithms that need to compute it.

Groups of similar arms | 5

When a structure is assumed on the set of considered Bandit problem ζ, information about an arm a' can be gained from playing another arm a. Structure induces a notion of problem-related dependency between the arms even if the reward random variables are still assumed independents, from a measure theoretical viewpoint. When solving the regret minimization problem in structured Bandit, the structure can *a priori* be leveraged to increase the problem-related information contained in a sample and therefore reduce the necessary exploration, hence logarithmic regret rate, compared to the unstructured Bandit setting. It is interesting to ask how the rate of information acquisition, or more precisely, the asymptotic logarithmic rate of progress per unit of interaction is improved compare to a situation in which structural information is not used.

In this chapter, based on the paper ***Stochastic bandits with groups of similar arms*** published at NeurIPS 2021 with Hassan Saber and Odalric-Ambrym Maillard, we study regret minimization in a structured Bandit setting, see [70]. We illustrate how structural information can help reduce the regret compared to the unstructured setting, how solving should not be confused with estimating, and how the expected reward criterion allows for especially greedy strategies.

[70]: Pesquerel et al. (2021), 'Stochastic bandits with groups of similar arms'

This light shed on the greediness of an optimal strategy beg the question of fairness within the class of optimal actions. While in its purest form, fairness is not a constraint of the Bandit formulation, one can envision scenario where Bandit is used to model situations in which fairness amongst the group of optimal arms is important while preserving optimality also is. For instance, preserving optimality might be important at the macro-scale, that may represent society, while fairness is important at the micro-scale of optimal arms, that may represent persons or industries amongst which a policymaker must choose from.

Information about the problem & information about a solution

When a learner interact with a Bandit problem, its aim is to minimize its cumulated regret over the long run or, similarly, to maximize its cumulated reward. It does so by sampling as often as possible an action with maximal expected reward. The main problem the learner faces is that it does not know, prior to the interaction, the reward distributions of the arms it can interact with. However, it does assume that the distributions belong to known family. Such prior assumption on the family of distribution help to understand and control how much information is gained about the expected value of an arm when a sample from that arm is drawn. We say that the learner adaptively solve the Bandit control problem as it try to maximize its number of interaction with a solution of the control problem. **Solving an unknown problem should not be confused with estimating the unknown problem.** While gaining information about the unknown Bandit problem is a way to adaptively solve it, it is not the only or smallest-regret way. Rather than **gaining**

information about the problem, a learner should aim to **gain information about a solution to the problem**. It is obvious when we remark that the best strategy to gain information about the problem is to play uniformly on the set of arms, which is obviously not an optimal Bandit strategy in the general case.

Remark on optimal information gain The uniform sampling strategy is optimal to gain information about the problem is optimal when all the arms are from the same family of distribution. Otherwise, the uniform strategy may not be optimal from an information gain point of view. For instance, in the extreme case of a two arms Bandit problem where one of the arm is assumed to be Dirac distribution and the other a $(0,1)$-bounded distribution, then it is absurd to sample the Dirac distribution more than once. Rather, one should the optimal allocation scheme will sample the Dirac arm once and allocate all the samples to the other arm (from an information gain viewpoint).

Exploiting & Estimating a structure

In this chapter, we use a structured Bandit problem to illustrate a similar fact. **One can exploit a structure without actively estimating it**. At first, this may not be obvious. One could think that estimating the structure, in a way similar to how we estimate the reward distributions, may be a desirable thing to do in order to exploit it a and suffer a smaller regret. However, we show in this chapter that this need to be the case.

5.1 Structured Bandit

General case

Compared to an unstructured Bandit problem (A, β) where all distributions $\beta(a)$ belong to a set $\mathcal{F}_a$ independently of the distribution of other arms, adding a structure to the set ζ that Bandit problems belong to will only reduce the achievable optimal logarithmic regret rate because the set ζ is strictly included in $\bigotimes_a \mathcal{F}_a$, $\zeta \subsetneq \bigotimes_a \mathcal{F}_a$, which reduces the size $\zeta(\beta)$ of problems in the **control-indistinguishable optimal alternative** set that is defined in Definition 3.1.25. This can be seen from the general allocation-constrained optimality viewpoint on the lower bound. We restate Theorem 3.1.6 in its general form. The smaller the set of constraints, the more we gain information about an arm while playing another, the more $KL(\beta(a), \beta'(a))$ are strictly positive in the allocation constraints and the smaller the logarithmic sampling rates of optimal arm can be, see Equation 5.1.

Theorem 5.1.1 (**Regret lower bound**: allocation-constrained optimality viewpoint) *Let A be a finite set of arms, $\zeta = \bigotimes_{a \in A} \mathcal{F}_a$ be a class of Bandit problems and $\nu = (A, \beta) \in \zeta$ be a Bandit problem within that class.*

Then, for all uniformly fast convergent policy π, and therefore uniformly maximal converging rate policy, the growth rate of the regret $\mathcal{R}_\nu(n; \pi)$ is

lower bounded,

$$\liminf_{n\to\infty} \frac{\mathcal{R}_\nu(n;\pi)}{\log n} \geq \inf_{(\eta_a)_{a\in A}} \sum_{a\in A} \eta_a \left(\mu^\star - \mu(a|\beta)\right) \quad \text{s. t. } \eta_a \geq 0 \ \forall a \in A \ , \ \inf_{\beta'\in\zeta(\beta)} \sum_{a\in A} \eta_a KL(\beta(a),\beta'(a)) \geq 1 \quad , \tag{5.1}$$

where the set $\zeta(\beta)$ is defined using $A^\star(\beta) = \text{argmax}_{a\in A}\, \mu(a|\beta)$, the set of maximal arms in a Bandit problem (A,β), as

$$\zeta(\beta) = \left\{ \beta' \in \beta' \in \zeta \mid \begin{array}{l} \mathcal{A}^\star(\beta) \cap \mathcal{A}^\star(\beta') = \emptyset, \\ \forall a \in \mathcal{A}^\star(\beta) \ , \ KL(\beta(a),\beta'(a)) = 0 \end{array} \right\} . \tag{5.2}$$

$\zeta(\beta)$ is the set of Bandit problem $\beta' \in \zeta$ with different optimal set and such that the distribution of optimal arm in β is unchanged in β', i.e. only the distribution of suboptimal arms is allowed to be changed and at least one must be transformed into an optimal arm within β'.

The art of deriving a practical and useful lower bound when interested in crafting a related algorithm lies in rewriting the set $\zeta(\beta)$ in a way that is formally handy to deal with. Sometimes, one can greatly reduce the "size" of $\zeta(\beta)$ by extracting a subset of $\zeta(\beta)$ such that the actual constraints only depend on this subset. This is what we did in the unstructured case to extract the per-arm logarithmic rate of sampling, remarking that one can consider Bandit problems where arms are made optimal one at a time. In structured Bandit, one cannot, in general, make arms optimal one at a time without moving other arms (in the KL sense). Therefore, finding a good subset of $\zeta(\beta)$ that can be formally handled is usually not an easy task. However, there are some cases, where intuition can help.

Unimodal Bandit

This is for instance the case of unimodal Bandit, see [53, 71, 72]. In unimodal Bandit, the set of arms A is assumed to be ordered, *i.e.* A is isomorphic to $(\{1,2,\dots,|A|\},\geq)$ endowed with its comparison structure $\geq$. Then for all unimodal Bandit problem β, the expected reward function $\mu(\cdot|\beta) : a \in A \mapsto \mu(a|\beta)$ is assumed to be a concave function. To exploit the structure, one can simply remark that *(discrete) derivative is a* ***local property***. If we were to find the best arm in this setting, one could use dichotomy which requires three measures of μ, the left side, the right side and the middle point of the dichotomy. This is certainly an interesting method that won't generalize well to other graph-like structures. Instead of exploiting this local property of derivative, one could also, from a more real analysis viewpoint, consider a set of three adjacent arms, compute the local derivative, and perform a gradient ascent until the optimal arm is reached. To exploit the structure of the unimodal Bandit, we can certainly perform what would look like a stochastic gradient ascent. At each time step, one only need to consider the set of arms that are adjacent to the current optimal arms and, from there play an unstructured Bandit algorithm. If the best empirical arm's index shift to the right or left, then this shift the set of local arms that we consider. Implicitly, we are performing a Bandit-stochastic gradient ascent. In the end, the regret

[53]: Combes et al. (2014), 'Unimodal Bandits: Regret Lower Bounds and Optimal Algorithms'
[71]: Yu et al. (2011), 'Unimodal Bandits.'
[72]: Saber et al. (2021), 'Indexed Minimum Empirical Divergence for Unimodal Bandits'

bound only depends on the arms that are adjacent to the optimal arms. Furthermore, One would have very little knowledge about the arms that are not in this local neighborhood around the best arm. Indeed, an agent would never need to sample them since, by the unimodal assumption, those arms have smaller expected reward that the (at most) two arms that are directly to the left and right from the optimal one. This idea of performing a Bandit-stochastic gradient ascent along the edges of a partially ordered graph is key to the IMED-RL [73] algorithm that we present in Chapter 7. Using this idea of local discrete gradient allows to reduce the complexity of making a choice from the cardinal of all possible actions to the cardinal of the local subset of actions required to compute all components of the "gradient".

[73]: Pesquerel et al. (2022), 'IMED-RL: Regret optimal learning of ergodic Markov decision processes'

5.2 Group of similar arms

We now focus on the presentation of the work done in the paper *Stochastic bandits with groups of similar arms* [70]. In a nutshell, we consider Bandits with groups of similar arms, *i.e.* groups of **arms with the same expected reward**. We prove a **lower bound on the regret** that involves a **combinatorial optimization problem**. We craft an **efficient algorithm**, IMED-EC, that uses a **relaxation** of the combinatorial optimization problem. We prove that the algorithmic logarithmic regret rate and regret lower bound **do not differ by a factor larger than 2**. We run experiments to prove the **numerical efficiency of IMED-EC**.

[70]: Pesquerel et al. (2021), 'Stochastic bandits with groups of similar arms'

A group-like structure

Motivated by various practical reasons, one may want to restrict to a subset $\mathcal{B} \subset \bigotimes_{a\in A} \mathcal{F}_a$ of allowed bandit configurations instead of the full set $\mathcal{F}^{\mathcal{A}}$. In this chapter, we assume that the reward function, $\mu : a \in \mathcal{A} \to \mu_a$, satisfies a cluster-like structural property which we call the **q-equivalence property**.

Definition 5.2.1 (q-equivalence property) *A Bandit ν is said to satisfy the **q-equivalence property** if for every arm $a \in \mathcal{A}$, there are at least $q-1$ distinct arms having the same expected value:*

$$\forall a \in \mathcal{A}, \quad |\{a' \in \mathcal{A} | \mu_{a'} = \mu_a\}| \geq q.$$

This property is illustrated in Figure 5.1, depicting a problem that can be described as a Bandit with the 3-equivalence property but also the 2-equivalence property and even the 1-equivalence property. The 1-equivalence property corresponds to the unstructured Bandit case in which, who knows, there may or may not have non-trivial group of similar arms. From this example, we note that if a Bandit satisfies the q-equivalence properties, then it satisfies the k-equivalence properties for all $k \leq q$. If we only assume that groups of similar arms may exist, then Bandit problems with trivial groups of size one belong to the set $\zeta(\beta)$ which will not be different from the unstructured one. One must assume to know a non-trivial lower bound on the size of the groups to effectively change the set of constraints in the lower bound and thus change the

lower bound. Only then it is possible to exploit the information of the existence of groups.

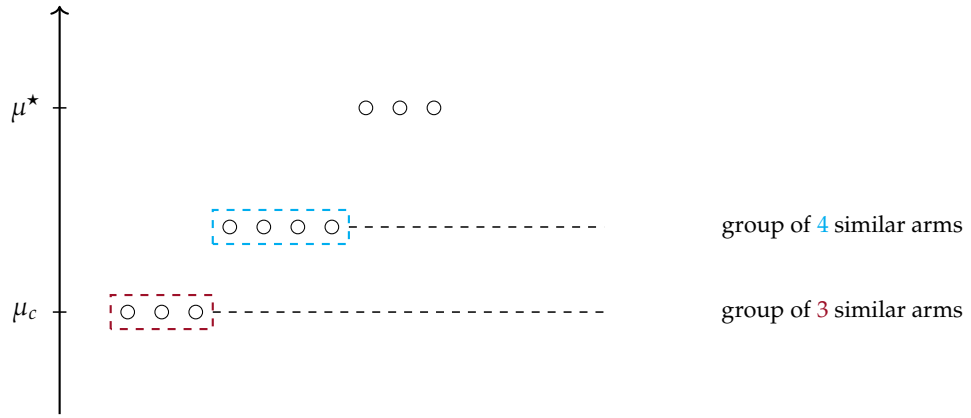

Figure 5.1: Graphical illustration of a 10-arms Bandit problem with groups of similar arms

Assuming the set of arms $\mathcal{A}$ and base distributions $(\mathcal{F}_a)_a$ are known to the learner, we denote by $\mathcal{B}_q$ the set of bandit configurations having the q-equivalence property, without explicit reference to the sets $\mathcal{F}_a$. Note that the sets $\mathcal{F}_a$ should be such that there exists at least one Bandit problem in $\bigotimes_a \mathcal{F}_a$ with the q-equivalence property. This is not the case if, for instance, all the reward distributions all have mutually disjoints supports. We also denote by $\mathcal{B}_q(\mu)$ the set of all expected values in $\mathcal{B}_q$. Formally, $\mathcal{B}_q(\mu)$ is the image of $\mathcal{B}_q$ under the μ mapping. When two arms in a Bandit ν share the same expected value, we will say that those two arms are *equivalent* and belong to the same *equivalence class*.

Definition 5.2.2 (Arm equivalence and equivalence class) *Given a bandit configuration ν, two arms $a, a' \in A$ are said to be equivalent if their associated distributions have the same expected values:*

$$a \sim a' \Leftrightarrow \mu_a = \mu_{a'}.$$

An equivalence class c in ν is a maximal subset of arms in A having the same mean, i.e., for all arms a, a' in c, $\mu_a = \mu_{a'}$ and for all arm $a \in c$ and $a' \in A \setminus c$, $\mu_a \neq \mu_{a'}$.

While the equivalence class may have different cardinality and do not refer to an equivalence relation, this naming convention relate to the origin of the project were the equivalence of actions originated from symmetries in the system. If we imagine the task of exploring the universe in the quest of finding information about some physical phenomenon, *e.g.* life, then under the assumption that the laws of the universe are invariant by translation and rotation the universe roughly is an isotropic environment. If the available actions are the direction in which looking or launching a spatial probe, then from probabilistic standpoint of finding information about the phenomenon we study, there is an invariance by rotation in the direction of the probing. When groups are created from invariance by symmetries, the computed groups of arms really are equivalent classes under an equivalence relation. Hence, our naming convention. Hopefully, there are other practical reasons to consider groups of similar arms that do not stem from symmetries.

Grouping actions

Redundant descriptors

This structure typically appears in practical situations when each arm can be described with a list of categorical attributes, and the (unknown) mean reward function only depends on a subset of them, the others being redundant. In this case, q is naturally linked to the number of attributes considered redundant (or useless descriptors), and the number of categories of each attribute. Formally, on good "basis of descriptors" (*i.e.* it is an orthogonal-like description), an arm a could be described using the following representation,

$$a = \left(\underbrace{d_1^a, \cdots, d_k^a}_{\text{Useful descriptors}} \quad , \quad \underbrace{d_{k+1}^a, \cdots, d_n^a}_{\text{Redundant descriptors}} \right),$$

and the set of arms could be described as the product

$$\underbrace{\prod_{l=1}^{k} D_l}_{\text{Useful descriptors}} \quad \times \quad \underbrace{\prod_{l=k+1}^{n} D_l}_{\text{Redundant descriptors}} \quad ,$$

where D_l is the number of categories for attribute l. Thus, in this good "basis of descriptors", $q = \prod_{l=k+1}^{n} |D_l|$, product of the cardinal redundant descriptors' sets.

The learner may know that there exists such a structure while not knowing a closed form formula mapping the list of categorical attributes to the significant subset. In this case, q might be a lower bound on the sizes of the class since the set of redundant descriptors might not be the largest possible one or because the number of redundant attributes depends on the number of relevant attributes. In all cases, the smallest possible number of redundant attributes can be naturally linked to q.

Paths

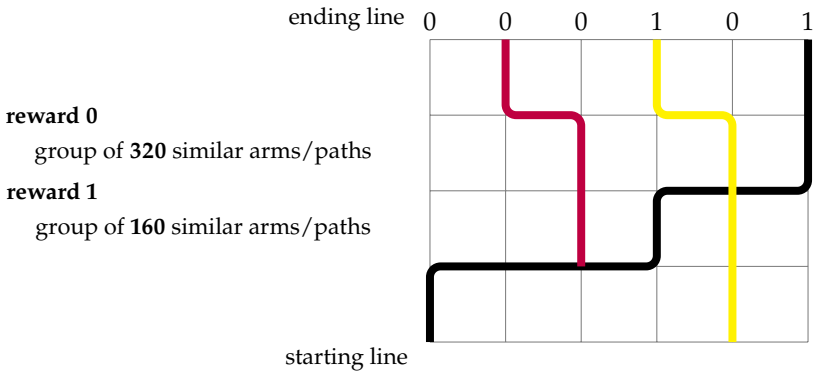

Figure 5.2: Groups of similar arms: the path perspective

There is another example where groups of similar arms appear naturally, depicted in Figure 5.2. We consider a grid-like environment, as depicted in the figure, and identify an upward path with an arm. By upward path, we mean a path going from the bottom line to the upper line, consisting of pieces of segments aligned with the grid. Staring from a vertex of the bottom line, a path can only go up, left or right, must respect the boundary conditions and cannot use an edge that was already used. There are no loop nor possibility to have a piece of the path going in the up-to-bottom direction. To each vertex of the upper line, a reward distribution is associated and in Figure 5.2 we show the expected reward of an instance of such a problem. This problem can be viewed as a Bandit problem in which each path corresponds to an arm, and a sample collected from an arm/path is a sample from the reached upper vertex.

On can see that groups of similar arms appear naturally since only the end point matters and not the specific path. If we consider the full description of a path in terms of a finite sequence of elements in the set $\{\text{up}, \text{left}, \text{right}\}$ (such a sequence must be a valid path), then one can clearly see that $\beta : a \to \beta(a)$ is not at all an injection (it may not be the case in an unstructured Bandit, but the difference here is that we know from the structure of the problem that it is not the case). Therefore, there is redundant information in the full description of a path and only the number of *lefts* and number of *rights* matter to describe the reward distribution associated to a path. This problem is linked to the previous redundant descriptors problem. If an agent, prior to the interaction know that there is such a structure but cannot compute in advance the set of similar arms, then it can still exploit the knowledge of a lower bound q on the number of similar paths thanks to the work we present in this chapter.

How to exploit this structure?

Goal For the set of structured Bandit $\mathcal{B}_q$ having the q-equivalence property, we show in Theorem 5.4.1, that the logarithmic growth rate of regret is lower bounded by a term $\mathfrak{C}_{\mathcal{B}_q}(\nu)$ which unfortunately makes appear in general a combinatorial optimization problem, due to the constraint set $\mathcal{B}_q(\beta)$ for $\beta \in \mathcal{B}_q$. It means that computing an optimal allocation scheme *a priori* involves solving a combinatorial optimization problem. Thus, resorting to OSSB [58] or any strategy targeting exact asymptotic optimality is daunting task for the practitioner because of the numerical complexity of combinatorial problems. Our goal is to provide a computationally efficient strategy adapted to the structure $\mathcal{B}_q$, that is able to reach optimality up to controlled error term.

[58]: Combes et al. (2017), 'Minimal exploration in structured stochastic bandits'

How to? In Section 5.3, we first consider without technical details the problem of structured Bandits with group of similar arms and assume the knowledge of the groups. Despite what it may look at a first glance, this simple problem will teach us what can probably be targeted by an algorithm without the knowledge of the groups as well as other intuitions. Then, we will move on to the problem of groups of similar arms where the groups are unknown but a lower bound on their sizes is known. In section 5.4, we derive a lower bound on the regret for the

structured set of bandit configurations $\mathcal{B}_q$. This bound makes appear two components, one that we call *non-combinatorial* as optimizing it can be done efficiently, and a second one that we term *combinatorial* as it involves solving a combinatorial problem. Interestingly, using in Lemma 5.4.2 and Theorem 5.6.1 it can be shown that the contribution of the combinatorial part of the lower bound can be controlled. Owing to this key insight, we introduce in section 5.5, IMED-EC, an adaptation of the IMED strategy from [74] to the structured set $\mathcal{B}_q$ and inspired by the non-combinatorial part of the lower bound we derive. Thus, in Section 5.6, we prove that IMED-EC achieves a controlled asymptotic regret that matches the non-combinatorial part of the lower bound and is at most (less than) a factor of 2 times the optimal regret bound. Furthermore, IMED-EC enjoys a small numerical complexity that is comparable to that of IMED in the unstructured setting. At each time step, the complexity of computing the next arm to be pulled by IMED-EC is no more than the one of sorting a list of $|A|$ elements once the IMED indexes of all arms have been computed, which is only $\log |A|$ times larger than looking for the minimal IMED index. Last, we illustrate the benefit of the IMED-EC over its unstructured version in section 5.7, where it shows a substantial improvement. Our experiments also highlights the robustness of the algorithm to a misspecified parameter q, which is a desirable feature for the practitioner.

[74]: Honda et al. (2015), 'Non-Asymptotic Analysis of a New Bandit Algorithm for Semi-Bounded Rewards'

5.3 Knowledge of the groups

Assuming the knowledge of the classes would probably be a too strong of an assumption to considered useful for the practitioner. Yet, this problem has some non-triviality that may be overlooked if we are not careful enough about our assumption. In particular, one should be aware of our assumption on the set of distributions $\mathcal{F}_a$, whether those are equal within a group or not, because it can change a lot of things. We denote $\mathcal{C}$ the set of all groups and $\Delta_c = \mu^\star - \mu(a)$ the suboptimality gap associated with group c and defined using an arm $a \in c$, chosen arbitrarily since all arms in the group c share the same expected value by definition.

In a group, all distributions are from the same family

Assume that within a group $c \in \mathcal{C}$ all the families $\mathcal{F}_a$ are the same set $\mathcal{F}_c$. The sets $\mathcal{F}_c$ could be different. Usually, that would be the set of $[0, 1]$-bounded distributions. Within a group, *a priori* nothing distinguish an arm from another prior to the interaction, but this could not be true after the beginning of the interaction if, within a group, reward distributions are not independent and identically distributed (i.i.d.).

In a group, all distributions are i.i.d.

Furthermore, let's make the even stronger assumption that, within a group, all the arms are i.i.d. If we were to **know the groups**, then we could identify a group of arms as one arm of a modified bandit problem and use an unstructured bandit algorithm on this new Bandit problem.

This new Bandit problem would be built by choosing a representative arm within each group, effectively creating a new Bandit problem where there are as many arms as there were groups in the original Bandit problem. This procedure is depicted in Figure 5.3 where the 10-arms Bandit problem of Figure 5.1 (left) is transformed into a 3-arms Bandit problem (right).

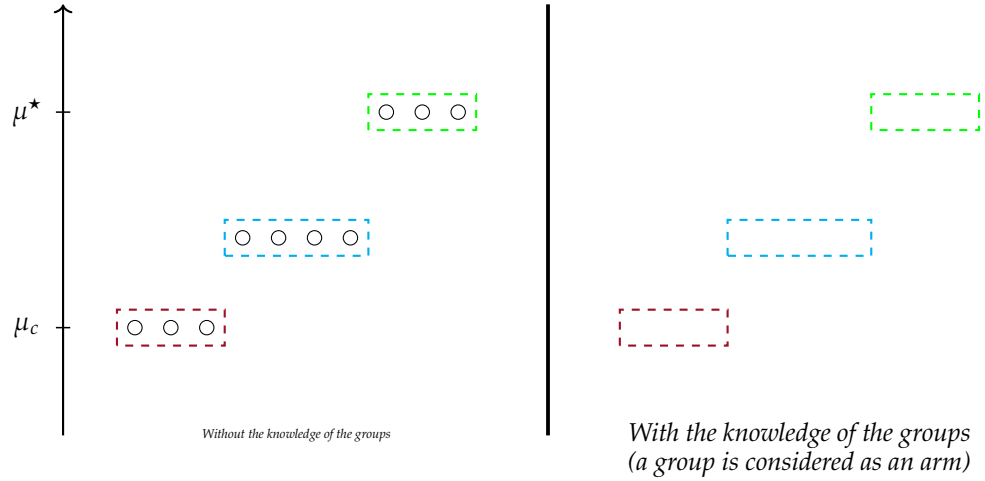

Figure 5.3: Identification of a group of arms with an arm in a transformed Bandit problem thanks to the knowledge of the groups

In this case, we have that $\mathcal{E}_{\mathcal{F}_c}(F_a, \mu^\star) = \mathcal{E}_{\mathcal{F}_c}(F_{a'}, \mu^\star)$ for all arms a, a' in a group c of equivalent arms. For each group $c \in \mathcal{C}$, one can denote F_c a representative distribution corresponding to that group. In the new Bandit problem the logarithmic growth rate of regret is

$$\sum_{c \in \mathcal{C}} \frac{\Delta_c}{\mathcal{E}_{\mathcal{F}_c}(F_c, \mu^\star)},$$

while it was

$$\sum_{a \in \mathcal{A}} \frac{\mu^\star - \mu(a)}{\mathcal{E}_{\mathcal{F}_a}(F_a, \mu^\star)} = \sum_{c \in \mathcal{C}} \sum_{a \in c} \frac{\Delta_c}{\mathcal{E}_{\mathcal{F}_c}(F_c, \mu^\star)} = \sum_{c \in \mathcal{C}} |c| \frac{\Delta_c}{\mathcal{E}_{\mathcal{F}_c}(F_c, \mu^\star)},$$

in the original unstructured Bandit that does not assume the knowledge of the groups. From two equations above, we can see that the regret in the modified problem is such that the contribution of each group c has been divided by $|c|$. The regret is therefore at least divided by $\min_{c \in \mathcal{C}} |c|$. If we further add the knowledge that all groups are of size at leas q, *i.e.* $\min_{c \in \mathcal{C}} |c| \geq q$, then one can state that **the regret has been divided by at least q**.

In a group, distributions are not assumed having the same law

We remove the assumption that, within a group, all the arms are identically distributed while they are still independent. Within a group, the arms are no more indistinguishable and there is a break in the "symmetry" of the arms within a group. Compared to the previous situation, arms index can no longer be considered permutation invariant within a group.

To restore some symmetry, instead of choosing a representative arm in a deterministic fashion, we sample a representative arm from a group $c \in \mathscr{C}$ uniformly at random. Thus, with the knowledge of the groups, we can identify a group of arms as one arm of a modified bandit problem and use an unstructured bandit algorithm on this new Bandit problem. This is very similar to what was done above except that we specify that the new Bandit problem is built by sampling a representative arm within each group, effectively creating a new Bandit problem where there are as many arms as there were groups in the original Bandit problem. Such a procedure is similarly depicted in Figure 5.3 where the 10-arms Bandit problem of Figure 5.1 (left) is transformed into a 3-arms Bandit problem (right).

In the unstructured Bandit problem the logarithmic growth rate of regret is still

$$\sum_{c\in\mathscr{C}}\sum_{a\in c}\frac{\Delta_c}{\mathscr{E}_{\mathscr{F}_c}(F_a,\mu^\star)} = \sum_{c\in\mathscr{C}}|c|\left(\frac{1}{|c|}\sum_{a\in c}\frac{\Delta_c}{\mathscr{E}_{\mathscr{F}_c}(F_a,\mu^\star)}\right),$$

while the expected logarithmic growth rate of regret, that depends on the choice of the representative arms chosen uniformly at random with probability $\frac{1}{|c|}$, is

$$\sum_{c\in\mathscr{C}}\frac{1}{|c|}\sum_{a\in c}\frac{\Delta_c}{\mathscr{E}_{\mathscr{F}_c}(F_a,\mu^\star)}.$$

The main distinction with the previous case is that instead of considering $F_c \in \mathscr{F}_c$ the common law of the class $c \in \mathscr{C}$, we must consider each $F_a \in \mathscr{F}_c$ of arms $a \in c$. From two equations above, we can see that the regret in the modified problem is such that the contribution of each group c has been divided by $|c|$. The regret is therefore, on average, at least divided by $\min_{c\in\mathscr{C}}|c|$. If we further add the knowledge that all groups are of size at leas q, *i.e.* $\min_{c\in\mathscr{C}}|c| \geq q$, then one can state that **the regret has been divided by at least q**.

However, one can doubt that it is the best we can do. There are experiments where the arm $a \in c$ that is drawn corresponds to $\min_{a\in c}\mathscr{E}_{\mathscr{F}_c}(F_a,\mu^\star)$ with minimal unlikelihood of optimality (*i.e.* large regret contribution), while in other experiments, the arm $a \in c$ that is drawn corresponds to $\max_{a\in c}\mathscr{E}_{\mathscr{F}_c}(F_a,\mu^\star)$ with maximal unlikelihood of optimality (*i.e.* small regret contribution). All other situations are equally likely due to the uniform sampling of a representative arm within a class c, and the unlikelihood of optimality is averaged out, in the sense of the harmonic mean.[1] The best one could therefore hope to achieve in terms of logarithmic growth rate of regret is

$$\sum_{c\in\mathscr{C}}\frac{\Delta_c}{\max_{a\in c}\mathscr{E}_{\mathscr{F}_c}(F_a,\mu^\star)}.$$

Since one cannot choose such representatives at the beginning of the interaction without prior information, it means that we should be able to sample an $\operatorname{argmax}_{a\in c}\mathscr{E}_{\mathscr{F}_c}(F_a,\mu^\star)$ representative of class c most of the time we sample within that class. Therefore, we cannot choose a representative at the beginning of the interaction but adaptively find the best representative within each class without oversampling each class, that is with a logarithmic sampling rate smaller than the harmonic-averaged one of our previous scheme.

1: I stumbled upon the harmonic mean more times than what can be ignored during this PhD thesis. I think that it would be interesting to investigate this way of averaging things when it comes to adaptive learning and its potential applications.

It is possible (and I would say likely), that getting or trying to exploit this information about the maximal $\mathcal{E}_{\mathcal{F}_c}$ is too costly in terms of the sampling power required and that trying to do so while sampling within the class c at a logarithmic rate $\frac{1}{\max_{a\in c}\mathcal{E}_{\mathcal{F}_c}(F_a,\mu^\star)}$ is impossible. I would be happy to investigate this problem further.

In a group, all distributions are not necessarily from the same family

We remove the assumption that all the families $\mathcal{F}_a$ of arms $a \in c$, where $c \in \mathcal{C}$ is a group of equivalent arms, are all equal to the same set $\mathcal{F}_c$. *A priori*, those sets could all be different, which further break the symmetry between the arms of a group. However, this situation is not that different from the previous one, were distributions of groups were independent, *a priori* different and from the same set $\mathcal{F}_c$. Indeed, one can still use the knowledge of the groups to sample a representative arms for each group, uniformly at random, and use those to build a new Bandit problem following the lines of what was described previously.

In the unstructured Bandit problem the logarithmic growth rate of regret is now

$$\sum_{c\in\mathcal{C}}\sum_{a\in c}\frac{\Delta_c}{\mathcal{E}_{\mathcal{F}_a}(F_a,\mu^\star)} = \sum_{c\in\mathcal{C}}|c|\left(\frac{1}{|c|}\sum_{a\in c}\frac{\Delta_c}{\mathcal{E}_{\mathcal{F}_a}(F_a,\mu^\star)}\right),$$

which differs from the previous case by the fact that one must use $\mathcal{E}_{\mathcal{F}_a}$ instead of the $\mathcal{E}_{\mathcal{F}_c}$ that was generic to a class.

On the other hand, the expected logarithmic growth rate of regret of the group-created Bandit, that depends on the choice of the representative arms chosen uniformly at random with probability $\frac{1}{|c|}$, is

$$\sum_{c\in\mathcal{C}}\frac{1}{|c|}\sum_{a\in c}\frac{\Delta_c}{\mathcal{E}_{\mathcal{F}_a}(F_a,\mu^\star)}.$$

The main distinction with the previous case is that instead of considering $F_a \in \mathcal{F}_c$ the common class $\mathcal{F}_c$, we must consider each $\mathcal{F}_a$ and modified optimization problems $\mathcal{E}_{\mathcal{F}_a}$. For a specific draw of representative arms, we denote them $(a_c)_{c\in\mathcal{C}}$, the regret is

$$\sum_{c\in\mathcal{C}}\frac{\Delta_c}{\mathcal{E}_{\mathcal{F}_{a_c}}(F_{a_c},\mu^\star)}.$$

Those regrets are, as previously, averaged out due to the sampling scheme.

As in the two other presented cases, we can see that the regret in the modified problem is such that the contribution of each group c has been divided by $|c|$. The regret is therefore, on average, at least divided by $\min_{c\in\mathcal{C}}|c|$. If we further add the knowledge that all groups are of size at leas q, *i.e.* $\min_{c\in\mathcal{C}}|c| \geq q$, then one can state that **the regret has been divided by at least q**.

For similar reason than the previous case, one can wonder whether this is the best one can do. Perhaps, a clever design would allow taking

advantage of the knowledge of the classes in a clever way (combining the samples of several arms, projecting empirical distributions $\hat{F}_a$ to distributions in $\mathcal{F}_{a'}$ where $a \sim a'$, using other mixing strategies, etc.).

In this paper, we show that, with only the knowledge of a lower bound q on the size of the classes, but without the knowledge of the classes, one can also devise a strategy dividing the regret by a factor q compared to solving the same Bandit in an unstructured setting.

Fairness

What is fairness in our setting?

In the above exposition of the problem of structured Bandits with groups of similar arms where the groups are known, there is a property that may be desired for the practitioner that is not satisfied by the proposed method, that is to sample a representative element within each group. This property is sometimes called **fairness** but suffer the problem of having multiple related definitions depending on the considered work in the literature.

In this chapter, and related to the problem of exploiting the structure of groups of similar arms, we can say that **fairness** would be achieved if all the arms from the optimal class are sampled asymptotically linearly with respect to the number of interactions. This is already quite a weak notion of fairness since we don't say that, for instance, the ratio of samples $\frac{N_a(t)}{N_{a'}(t)}$ of two arms from the optimal group must be bounded. This stronger notion of fairness probably is possible with the knowledge of the class. Why would such a notion be useful and called fairness? If we imagine that the agent is a policymaker that must sequentially choose between companies, people, or services to solve some problems and that there is a natural notion of known group, then from the policy make point view, regret minimization is important but from the "arms" point of view, the situation is unfair if only one optimal "arm" is always picked and the other optimal "arms" are never.

Fairness with known groups

The strategy of sampling a representative arm and playing a modified Bandit problem where only one arm per group is considered lead to the largest *unfairness* possible. One optimal arm is sampled a linear amount of time while the other optimal arms are sampled zero times, which worse than any increasing, even sub-logarithmic, function of the number of interactions. When the groups are known, one can easily think of the following scheme to enforce fairness in the group of optimal arms. We consider as previously the group-constructed Bandit problem made from picking one representative arm per group of similar actions. When the optimal (representative) arm is supposed to be played in this new Bandit setting, we sample all $|c|$ arms that are in the same class as the empirical optimal representative arms that we are supposed to sample in the modified Bandit problem. Only the collected sample from the representative arm is used to update its empirical distribution of probability since *a priori*, the arms from the group do not have the same

distribution. Of course, under additional assumptions, such as the i.i.d. hypothesis, one can mix the samples. For all empirical suboptimal arms, we do as previously since doing otherwise would only add (with high probability that empirical arm is the optimal arm) to the regret. With this method, fairness in the optimal group is ensured while we do not compromise on the reduced logarithmic growth rate. In fact, one can even imagine proving that the number of samples of arms within the optimal group differ at most by a logarithmic function of the interaction. Non-representative optimal arms would be sampled each time the optimal representative is sampled except when the representative arm is not empirically optimal. Such a situation cannot happen for a large number of interactions by design. In this setting, the representative arm would always be the most sampled arm of its group.

Fairness without the knowledge of the groups

In this chapter, we do not assume the knowledge of the class but only the weaker assumption of a lower bound on the number q of elements per group. A weaker notion of fairness than that of sampling all arms in the optimal group linearly would therefore be that at least q arms of the optimal groups are sampled linearly. We did not originally consider or study this or another fairness constraint but now think that this would make for a nice addendum to the original work. However, the fairness constraint is *a priori* far more difficult to handle than the one where groups are known. Furthermore, while one can enforce fairness without compromising on the logarithm growth rate of regret when the classes are known, it is not obvious that the regret guarantees of our proposed IMED-EC algorithm will be preserved when trying to enforce, even weak, fairness in the group of similar optimal arms. In the final section of this chapter, we first present the unfairness of IMED-EC, a result that was already presented in the Appendix D of our paper [70], and then try to modify the original design of our IMED-EC algorithm to enforce fairness.

[70]: Pesquerel et al. (2021), 'Stochastic bandits with groups of similar arms'

Bandits with the q-equivalence property

We hereafter consider the learner only knows a lower bound

$$q \geq \min_{c \in \mathcal{C}} |c|$$

on the number of elements per group of similar arms. Still, we would like to exploit the prior knowledge of this structure in a way that would allow us to reduce the logarithmic growth rate of regret, hopefully by a factor q as it is possible with the knowledge of the groups. First, we derive a lower bound on the logarithmic growth rate of regret that is then used to devise a sampling strategy which is tested and compared to other strategies. Afterward, we investigate numerically the fairness of our strategy.

5.4 Regret lower bound

In this section, we derive a lower bound on the number of pulls of suboptimal arms that involves a combinatorial optimization problem. The proof of Theorem 5.4.1 is based on the set of **control-indistinguishable optimal alternative** of a Bandit problem $\beta \in \mathcal{B}_q$, that we denote $\mathcal{B}_q(\beta)$, and that we introduced in Definition 3.1.25.

Confusing instances

We recall the intuition that from a learning perspective, Bandit problems in this set $\mathcal{B}_q(\beta)$ cannot be distinguished from β by only playing action in $A^\star(\beta)$, *i.e.* by being greedy on the computed information, while problems in this set can make the learner suffer a linear regret. This set appears in the constraint of the optimization problem involved to compute the instance-dependent regret lower bound of a problem β belonging to the class of Bandit problems $\mathcal{B}_q$, where this set is the prior information we have on the problem we aim to solve. For this reason, we call elements of this set, *i.e.* Bandit problems in $\mathcal{B}_q$, **confusing instances** of problem β in class $\mathcal{B}_q$. Confusing instances allow to assess the intrinsic difficulty of a bandit problem and allow computing lower bounds on the number of times suboptimal arms are pulled, even if implicitly. The lower bound informs us on the minimal amount of exploration one needs to do to solve a bandit problem. More formally, a confusing instance ν' associated to a suboptimal arm a for a bandit problem ν is a bandit instance with the same set of arms as the original one, but in which μ_a has been changed to $\mu'_a > \mu^\star$. We refer to Definition 3.1.25 and Theorem 3.1.6 for a more detailed exposition of the different ideas and intuitions. In this chapter, we focus on the intuitions of this set when the prior knowledge about the Bandit problems is the set of Bandits satisfying the q-equivalence property.

A good sampling strategy, *i.e.* one that does not sample suboptimal arms too much in the sense that it is uniformly fast convergent as in Definition 3.1.20, should behave very differently on the original problem and on a confusing instance $\beta' \in \mathcal{B}_q(\beta)$. The difference should be even greater for an optimal policy, *i.e.* a policy with uniformly maximal convergence rate as in Definition 3.1.21. Studying this difference of sampling behaviors, we can compute the minimal amount of exploration performed by an optimal strategy on arm a in the original problem ν, at least implicitly (without an immediate closed form formula). Doing so for all suboptimal arms allows to bound the logarithmic sampling rate of suboptimal arms and therefore to characterize the intrinsic complexity of a bandit instance ν.

In this structured setting, a Bandit in $\mathcal{B}_q$ has to respect the structure to be called a confusing instance. In our case, it means that a confusing instance cannot have a class with less than q arms. We will therefore consider confusing instances associated to classes rather than individual arms. We formally introduce the set of confusing instances in this specific setting with notation that are adapted to the exposition of this problem. Those notations are adapted from the one already introduced, *e.g.* in Definition 3.1.25.

Definition 5.4.1 (Confusing instance) *Given a Bandit configuration $\nu = (\beta, A) \in \mathcal{B}_q$, a real number λ and a subset $c_q \subseteq A$ of q equivalent arms in ν, we denote by $\mathcal{B}_q\left(\nu, c_q, \lambda\right)$ the set of all bandit configurations having the same set of arms as ν and such that for all $\nu' \in \mathcal{B}_q\left(\nu, c_q, \lambda\right)$, $\nu' \in \mathcal{B}_q$ and for every arm a in c_q, $\mu'_a \geq \lambda$,*

$$\mathcal{B}_q\left(\nu, c_q, \lambda\right) = \left\{\nu' = (\beta', A) \in \mathcal{B}_q \mid \forall a \in c_q,\ \mu'(a) \geq \lambda\right\} ,$$

where $\mu'(\cdot) = \mu\left(\cdot|\beta'\right)$.

*When $\lambda > \mu^\star$, and c_q is a subset of a suboptimal class, a bandit configuration in $\mathcal{B}_q\left(\nu, c_q, \lambda\right)$ is called a **confusing instance** of ν.*

We will use the notation $\mathcal{B}_q\left(\mu, c_q, \lambda\right)$ to specify the set of expected rewards, $\{\mu\left(a|\beta'\right)\}_{a\in A}$ of bandit configurations in $\mathcal{B}_q\left(\nu = (\beta, A), c_q, \lambda\right)$.

Using this fine-tuned Definition 5.4.1 of confusing instance, we derive the asymptotic logarithmic growth rate of regret for the class $\mathcal{B}_q$. As in Theorem 3.1.6 that showcase the general lower bound from the optimal constrained allocation viewpoint, we will need to use the unlikelihood of optimality, see Definition 3.1.23, to derive a proper formula. Such a quantity requires assuming the knowledge of set of distributions $\mathcal{F}_a$ for each arm. Of course, such sets should allow both the existence of the structure and the existence of uniformly fast convergent strategies for our usual tools to apply. In this chapter, we assume that all the sets $\mathcal{F}_a$ are the same set $\mathcal{F}$. We make the standard assumption that the set $\mathcal{F}$ is the set of Bounded distributions in $[0, 1]$ with expected values all strictly smaller than the upper bound of the support, here one. This assumption is the same we made in the previous Chapter 4 and is a standard assumption in the Bandit community. Since our IMED-EC algorithm is based on the IMED algorithm, one could also assume that $\mathcal{F}$ is the set of semi-bounded distributions with support in $]-\infty, M]$ and expected values in $]-\infty, M[$ with existence of the moment generating function in a neighborhood of zero.

Upon deriving a lower bound for the class $\mathcal{B}_q$ of Bandit problems with q-equivalence property, we need to use a quantity that we denote $\mathcal{W}_{\mathcal{F}}$ and call the **unlikelihood of grouping**. The quantity is coined this way directly in reference to the **unlikelihood of optimality** that we introduced in Definition 3.1.23. The unlikelihood of grouping will relate to *how unlikely* it is for an arm to be confused, from a discriminating-sampling rate viewpoint, with a distribution targeting an exact value of expected reward.

Definition 5.4.2 (Unlikelihood of grouping) *Let $\beta(a) \in \mathcal{F}$ corresponds to a reward distribution of an arm a in a Bandit problem. We denote $\mathcal{W}_{\mathcal{F}}\left(\beta(a), \lambda\right)$ and call **unlikelihood of grouping** the quantity defined as*

$$\mathcal{W}_{\mathcal{F}}(\kappa, \lambda) = \inf_{F\in\mathcal{F}} \left\{\mathrm{KL}(\kappa, F) \mid \mathbf{E}_G(X) = \lambda\right\} , \tag{5.3}$$

where KL denotes the Kullback-Leibler divergence between two distributions. When the set of constraints is empty, the quantity is defined as $+\infty$.

The quantity $\mathcal{W}_{\mathcal{F}}$ which is very similar to $\mathcal{E}_{\mathcal{F}}$ in its definition except that

the *greater than* is replaced by an *equality* in the constraint. The reason for this quantity in the forthcoming lower bound is that we need to consider "moving" groups of similar arms "together" to create confusing instances. If we want to create a confusing instance where suboptimal arm a is now optimal with expected reward λ such that $\lambda \geq \mu^\star$ (we use the $\mathcal{E}_\mathcal{F}$), then we need to make at least $q-1$ other suboptimal arms optimal with the same expected reward λ which is now an equality constraint in order to create a group (we use the $\mathcal{W}_\mathcal{F}$). Therefore, $\mathcal{W}_\mathcal{F}$ will help relates to the unlikelihood of optimality of arm given the grouping constraint that is imposed by the q-equivalence property of the Bandit problem we consider.

Regret lower bound

Theorem 5.4.1 (Asymptotic lower bound) *Let $q \in \mathbb{N}_*$ be a positive integer and $\nu \in \mathcal{B}_q$ be a bandit configuration having the q-equivalence property. Let $c \subset \mathcal{A}$ be a suboptimal equivalence class in ν. Then, on the class of uniformly fast converging learner, we have the following lower bound relating logarithmic sampling rate of suboptimal arms in a class c,*

$$\liminf_{T\to\infty} \frac{1}{\log T} \min_{c_q \subseteq c} \left(\begin{array}{l} \displaystyle\sum_{a\in c_q} \mathbb{E}_\nu(N_a(T))\, \mathcal{E}_\mathcal{F}\left(\nu_a, \mu^\star\right) \; + \\ \displaystyle\inf_{\mu'\in\mathcal{B}_q\left(\mu, c_q, \mu^\star\right)} \sum_{a\notin c_q \cup A^\star} \mathbb{E}_\nu(N_a(T))\, \mathcal{W}_\mathcal{F}(\nu_a, \mu'_a) \end{array} \right) \geq 1\,, \tag{5.4}$$

where c_q is any subset of c having q distinct arms within it.

This lower bound is based on the set of constraints $\mathcal{B}_q(\beta)$ that can be built from a Bandit problem $\beta \in \mathcal{B}_q$. This lower bound relates the number of pulls of arms of suboptimal arms in a group c in the following way. The first term $\sum_{a\in c_q} \mathbb{E}_\nu(N_a(T))\, \mathcal{E}_\mathcal{F}\left(\nu_a, \mu^\star\right)$ consider a subgroup $c_q \subseteq c$ of q arms in the group c and, using the unlikelihood of optimality, "*move them all*" to an optimal position. This is possible thanks to the continuity of the $\mathcal{E}_\mathcal{F}$ under the assumption that $\mathcal{F}$ is the set of bounded distribution (and common to all arms) and the fact that the *infimum* constrained KL is in fact a minimum attained in $\mu^\star$ as proved in the paper of J. Honda and A. Takemura [16]. It means that all the terms $\mathcal{E}_\mathcal{F}\left(\nu_a, \mu^\star\right)$ correspond to unlikelihood of optimality of "moving" arms to an equivalent group. The second term $\inf_{\mu'\in\mathcal{B}_q\left(\mu, c_q, \mu^\star\right)} \sum_{a\notin c_q \cup A^\star} \mathbb{E}_\nu(N_a(T))\, \mathcal{W}_\mathcal{F}(\nu_a, \mu'_a)$ deal with the *remainder* of the arms that are in c but not in c_q. This set of remainders, $c \setminus c_q$ could be made of less than q arms and thus makes the newly constructed Bandit problem violate the q-equivalence property structural constraint. The lower bound tells us that those arms must be moved (exactly) to other existing groups in the created Bandit problem, *i.e.* in the set $\mathcal{B}_q\left(\mu, c_q, \mu^\star\right)$. Those can be moved together, completely dispatched or anything in-between.

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

Sketch of proof

In this chapter, we only provide sketch of the proofs because those rely on standard mathematical tools and most of the novelty is in the algorithmic

exploitation of the structure. The proofs techniques that we use and can be found in the paper [70] this chapter is based on can be found seminal work of [17] which present the constrained-allocation lower bound and the more recent proof techniques of Garivier et al. [14] which is probably closer in spirit to this manuscript as it has a strong focus on tools from information theory such as the data processing inequality. Therefore, we only present sketch of proofs which we think is enough to get all the good intuitions. In this manuscript, we favored the exposition of the proofs in the previous section dealing with the FMED and OMED algorithms since those were based on non-standard tools for the Bandit community, in particular the usage of portfolio algorithms.

[70]: Pesquerel et al. (2021), 'Stochastic bandits with groups of similar arms'

[17]: Graves et al. (1997), 'Asymptotically efficient adaptive choice of control laws incontrolled markov chains'

[14]: Garivier et al. (2016), 'Explore first, exploit next: The true shape of regret in bandit problems'

Sketch of proof Following the intuition that was given about the set of confusing instances, those are used in the proof of Theorem 5.4.1. We use Figure 5.4 to illustrate the construction of a confusing instance and the different situations that may appear. In Figure 5.4, the arrow $\longrightarrow$ (associated to $\mathscr{E}_{\mathcal{F}}$) means that we want to shift distributions $\boxed{}$ from a class c in order to make them optimal and the arrow $\longrightarrow$ (associated to $\mathscr{W}_{\mathcal{F}}$) means that we have to move distributions in order to satisfy the q-equivalence property. In this figure, we assume that the known lower bound q on the number of element per group is equal to 3, *i.e.* we consider $\mathscr{B}_3$. The combinatorial nature of the lower bound will come from the

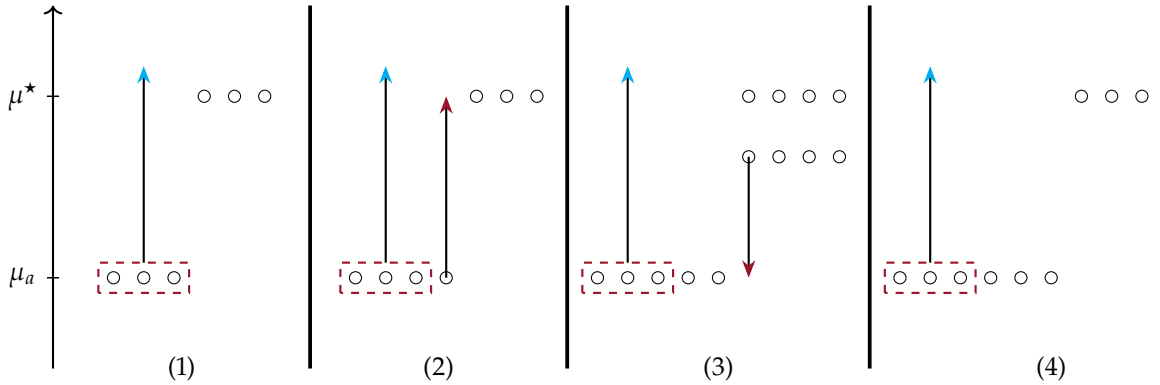

Figure 5.4: Illustration of the computation of a confusing instance depending on the situation. In this illustration, we assume that the known lower bound q on the number of element per group is equal to 3, *i.e.* we consider $\mathscr{B}_3$.

$\binom{|c|}{q}$ ways of grouping elements from a class c to be moved together. We consider confusing instances in which q arms from a suboptimal class c are moved above the optimal one (*w.r.t.* the mean). If there are q arms in the class, then there are no remaining arms to move. From left to right, this is the first situation (1) depicted in Figure 5.4. If the number of arms in class c is greater than q, then several situations can occur. If there are more than $2q$ arms in c, then moving q arms creates a remainder of size larger than q meaning that the crafted confusing instance respects the equivalence structure. This is the situation depicted in (4) of Figure 5.4. However, if there are between $q+1$ and $2q-1$ arms, then the remainder is of size larger than 1 but strictly smaller than q, as in situation (2) and (3) of Figure 5.4. The confusing instance built from only moving arms in c_q does not respect the equivalence structure, and we have to deal with the arms in the remainder (the *infimum* of Equation 5.4). One can either move a remainder arm in c to another existing group as in (2) of Figure 5.4 or

one can complete the remainder by moving an arm from another class (but not the optimal one because of the zero-KL constraint on $A^\star(\beta)$ in the definition of $\mathcal{B}_q(\beta)$) to the group c, as in (3) of Figure 5.4. There are $|c|$ choose q possible choices to move q arms from class c (the *minimum* of Equation 5.4) and all in all, the lower bound involves a combinatorial optimization problem.

While this lower bound involves a combinatorial optimization term, one can distinguish between two regimes depending on the size of the suboptimal class. This different regimes even show in the sketch of proof we just did. One regime will correspond to cases (1) and (4), the *non-combinatorial regime*, and one regime will correspond to cases (2) and (3), the *combinatorial regime*.

Non-combinatorial regime

For a suboptimal class c, if $|c| = q$ or $|c| \geq 2q$, then the lower bound reduces to

$$\liminf_{T\to\infty} \frac{\min_{c_q \subseteq c} \sum_{a\in c_q} \mathbb{E}_\nu\left(N_a(T)\right) \mathcal{E}_{\mathcal{F}}\left(\nu_a, \mu^\star\right)}{\log T} \geq 1,$$

because the remainder is of size larger than q and the *infimum* from Theorem 5.4.1 disappears. Indeed, the *infimum* is always 0 as this quantity can be obtained by choosing $\mu'_a = \mu_a$ for all $a \in c \setminus c_q$. When $c \setminus c_q$ is empty, this corresponds to (1) of Figure 5.4 and if $|c| \geq 2q$, this corresponds to case (4) of Figure 5.4. In full generality the $\min_{c_q \subset c}$ that is on all q-partitions of c involves a combinatorial optimization problem. When $c_q = c$ because the class c has exactly q elements, then the problem is trivial since there is only one possibility. When $|c| \geq 2q$, then the lower bound amount to computing the minimum possible value of a sum of q elements chosen from a list of $|c|$. This is no longer a combinatorial problem. The minimum over all q-partitions of c is the sum of the q smallest elements of $\{\mathbb{E}_\nu\left(N_a(T)\right) \mathcal{E}_{\mathcal{F}}\left(\nu_a, \mu^\star\right)\}_{a\in c}$. The search amongst all the q-partitions of c amounts to researching the q smallest elements which is not more complex than sorting a list of $|c|$ elements. While the complexity of sorting a list of $|c|$ elements is $|c| \log |c|$, finding all the q smallest element can be done with a min-heap at $|c| \log q$ cost. More on that later when we present the IMED-EC algorithm. Hence, the problem is no more a combinatorial optimization one, and we call this case the *non-combinatorial* regime.

In the non-combinatorial regime the regret lower bound is modified as follows.

Lemma 5.4.2 (Non-combinatorial regime lower bound) *Let $\nu \in \mathcal{B}_q$ be a bandit configuration having the q-equivalence property. Let c be a suboptimal class in the non-combinatorial regime, then,*

$$\liminf_{T\to\infty} \frac{\sum_{a\in c} \mathbb{E}_\nu\left(N_a(T)\right)}{\log T} \geq \frac{|c|}{q} \frac{1}{\mathcal{E}_{\mathcal{F}}\left(\nu_a, \lambda\right)}. \tag{5.5}$$

While we do not have information about individual number of times an arm in a class has been sampled, Lemma 5.4.2 roughly tells us than on average, the lower bound on the minimal amount of exploration of an arm in a suboptimal class has been divided by q.

Lemma 5.4.3 *If all suboptimal classes are in the non-combinatorial regime, the regret may be asymptotically lower bounded by*

$$\liminf_{T\to\infty} \frac{\mathcal{R}(\nu, T)}{\log T} \geq \frac{1}{q} \sum_{a\in A\setminus A^\star} \frac{\mu^\star - \mu_a}{\mathcal{E}_{\mathcal{F}}(\nu_a, \lambda)}. \tag{5.6}$$

Lemma 5.4.3 informs us that in the non-combinatorial regime, the classical lower bound on the regret has been divided by q.

Combinatorial regime

For a suboptimal class c to be in the *combinatorial* regime, we need $q < |c| < 2q$, since the remainder is such that $0 < |c \setminus c_q| < q$ and the infimum in Theorem 5.4.1 is not 0. In that case, the lower bound (5.4) involves a combinatorial optimization problem. Two difficulties arise from the term

$$\inf_{\mu'\in\mathcal{B}_q(\mu, c_q, \lambda)} \sum_{a\notin c_q\cup A^\star} \mathbb{E}_\nu(N_a(T))\, \mathcal{W}_{\mathcal{F}}(\nu_a, \mu'_a).$$

First, while we could have thought that summing on the remainder $c \setminus c_q$ would be enough, the summand has to be on $a \notin c_q \cup A^\star$ as a whole. Indeed, the residual $c \setminus c_q$ may be of size $q-1$ meaning that it might cost less (in information terms, as measured by the KL) to move an arm from another class to the residual in order to complete it rather than moving all the remainder. This situation is depicted in (3) of Figure 5.4. Second, while we could have thought that moving elements from one class of ν to another might be enough, the *infimum* has to be taken on $\mathcal{B}_q(\mu, c_q, \lambda)$. Indeed, the residual $c \setminus c_q$ may be of size $q-1$ and the *nearest* class might be of size exactly q. In this case, it may cost less to move all the $2q-1$ distributions in between the two classes and create a new one rather than merging one of the two with the other.

Lemma 5.4.4 *Let $\nu \in \mathcal{B}_q$ be a bandit configuration having the q-equivalence property and c be a suboptimal class in the combinatorial regime, then*

$$\liminf_{T\to\infty} \frac{\sum_{a\in c} \mathbb{E}_\nu(N_a(T))}{\log T} \geq \frac{1}{\frac{q}{|c|}\mathcal{E}_{\mathcal{F}}(\nu_a, \mu^\star) + \frac{|c|-q}{|c|}\min_{\kappa\in\nu} \mathcal{W}_{\mathcal{F}}(\nu_a, \kappa)}, \tag{5.7}$$

$$\liminf_{T\to\infty} \frac{\sum_{a\in c} \mathbb{E}_\nu(N_a(T))}{\log T} \geq \frac{1}{2q} \sum_{a\in c} \frac{1}{\mathcal{E}_{\mathcal{F}}(\nu_a, \mu^\star)}. \tag{5.8}$$

Those equations can be compared to the Equation 5.5 from the non-combinatorial regime. We emphasize the fact that the lower bounds given by Equations 5.7 and 5.8 are not the *largest* possible lower bound and hence do not provide as much information about the algorithmically achievable regret as the largest one given by Equation 5.4. However,

together with a regret upper bound on the algorithm IMED-EC, those quantities will help us control the asymptotic discrepancy between IMED-EC's regret and the asymptotic lower bound given by Theorem 5.4.1.

5.5 IMED-EC

IMED-EC: IMED for Bandits with equivalence class

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

As the name suggest, the algorithm we introduce, IMED-EC, is built on the IMED algorithm introduced by J. Honda and A. Takemura in [16], an algorithm that the reader should be familiar with at this point of the manuscript. We recall that this algorithm was introduced in Section 3.3 and was used to craft FIMED Algorithm 14 and OIMED Algorithm 18 that we introduced in the previous Chapter 4. The work that is done in Chapter 4 can therefore be connected with the work presented in this chapter. This is why we preferred and in-depth presentation of the ideas and proofs techniques of Chapter 4, because we think that those ideas could have consequences on a lot of other works, such as the one presented in this chapter but also in the next two chapters, as well as basically all the algorithms relying on an efficient computation of the unlikelihood of optimality $\mathcal{E}_{\mathcal{F}}$.

Description

At each interaction t, for each arm $a \in A$, one can compute its IMED index as

$$I_a(t) = N_a(t)\mathcal{E}_{\mathcal{F}}\left(\hat{\mu}_a(t), \hat{\mu}^\star(t)\right) + \log N_a(t),$$

where we recall that $\hat{\mu}^\star(t) = \max_{a\in A} \hat{\mu}_a(t)$ and for each arm $a \in A$, $\hat{\mu}_a(t)$ is the empirical mean of arm a computed with samples from this arm collected up to time t,

$$\hat{\mu}_a(t) = \frac{1}{N_a(t)} \sum_{s=1}^{t} X_s \mathbb{1}\left\{a_s = a\right\} \ ,$$

where X_s is the sample collected by the algorithm at the s^{th} interactions. As previously, we denote by $\mathcal{A}^\star(t) = \arg\max_{a\in\mathcal{A}} \{\hat{\mu}_a(t)\}$ the set of empirical optimal arms at interaction t. We will denote by $A_q(t)$ the set of arms having the q smallest IMED indexes. Ties are broken using the number of samples and privileging arms that have been sampled the least and if ties still occurs after that step, they are further broken uniformly at random. Ties are broken until the set $A_q(t)$ is exactly of size q The IMED-EC Algorithm 20 will consider and compute at each time t the two following quantities:

$$I^\star(t) = \min_{a\in\mathcal{A}^\star(t)} I_a(t) = \min_{a\in\mathcal{A}^\star(t)} \log N_a(t),$$
$$I(t) = \min_{\substack{\mathcal{A}'\subset\mathcal{A} \\ |\mathcal{A}'|=q}} \sum_{a'\in\mathcal{A}'} I_{a'}(t) = \sum_{a'\in\mathcal{A}_q(t)} I_a(t).$$

The first quantity $I^\star(t)$ is related to the optimal arms. The second quantity, $I(t)$, is related to the collective unlikelihood of optimality of the group of smallest IMED indexes. While the smallest index may be small enough that it would be sampled by the unstructured IMED, collectively, the q arms that are the less unlikely to be optimal are more unlikely to be optimal.

The quantity $I(t)$ can be computed efficiently by summing the q smallest elements of the list of IMED indexes. Finding the q smallest elements can be done by sorting the list of $|A|$ indexes which cost $O(|A| \log |A|)$. However, this is not the fastest we can do. Instead, one can build a heap as the IMED indexes are computed, costing $O(|A|)$, and extract the q smallest from the heap, which cost $O(|A| \log q)$. This new cost is linear in $|A|$, and will make IMED-EC almost as fast as the original IMED algorithm. In big O notation, there will be a factor of $\log q$ because of the computation of $I(t)$.

Algorithm

Having introduced all the necessary quantities, one can present the IMED-EC Algorithm 20.

Algorithm 20: IMED-EC algorithm

Input: A bandit tuple $(\mathcal{A}, s, \mathcal{V})$ as in Definition 3.1.12;
The IMED index function $\mathcal{I} : (a, H) \to (a, H)$

1 Initialize history H as $H = \emptyset$;
2 **for** $t \in \mathbb{N}$ **do**
3 **forall** $a \in A$ **do**
4 Compute index $I_a = \mathcal{I}(a, H)$;
5 Compute $I^\star(t) = \min_{a \in \mathcal{A}^\star(t)} \log N_a(t)$;
6 Compute $I(t) = \sum_{a' \in \mathcal{A}_q(t)} I_a(t)$;
7 **if** $I^\star(t) \leq I(t)$ **then**
8 Compute $a \in \operatorname{argmin}_{a \in \mathcal{A}^\star(t)} N_a(t)$;
9 **else**
10 Compute $a \in \operatorname{argmin}_{a \in A} I_a(t)$
11 Sample a reward $r \sim \beta(a)$;
12 Update history, $H \leftarrow H \cup \{(a, r)\}$;

Based on the comparison, line 7, of $I^\star(t)$ and $I(t)$, computed lines 5 and 6, it is decided to sample an empirically optimal arm, line 8, or not, line 10. When a suboptimal arm is sampled because $I(t) < I^\star(t)$, the one that is sampled is the one with minimal IMED index, *i.e.* the one the original unstructured IMED algorithm would have sampled. While probably obvious, it should be noted that $I(t)$ is always greater than the minimal IMED index, in fact at least q times larger, which means that IMED-EC always perform less exploration than the original IMED algorithm. Interestingly, when $q = 1$, the IMED-EC algorithm coincide exactly with the IMED algorithm, which is a nice property since in the case $q = 1$, the problem is unstructured and an optimal Bandit algorithm is IMED.

IMED-EC can there be seen as a class of algorithm $(IMED - EC(q))_{q\in\mathbb{N}}$ (with $q \leq |A|$) from which IMED is the element corresponding to index $q = 1$.

While the original problem involves combinatorial quantities, those are not involved in the IMED-EC algorithm. From a time complexity viewpoint, this makes this algorithm on par with other popular algorithms such as UCB, KL-UCB, and IMED algorithm. On the contrary, the general structure algorithm OSSB involves solving a combinatorial optimization problem at each time step, which makes it numerically inefficient. We are not aware of any general relaxation method for this algorithm that we could compare IMED-EC with.

Intuition

It should be noted that the original idea of the index $I(t)$ emerged from the non-combinatorial reduction that we discussed above. While it does not correspond to the terms appearing in the equations from the analysis, its form is highly reminiscent of these equations.

We recall that the IMED algorithm can be understood as an algorithm that computes an index for each arm a such that the index is comparison (a difference) between the *posterior log-probability of optimality* $N_a(t)\mathcal{E}_{\mathcal{F}}(\hat{\mu}_a(t), \hat{\mu}^\star(t))$ and the *log-frequency* $\log \mathbb{N}$ (up to additional constant $\log t$ common to all arms) at which the arm has been empirically sampled. When the IMED index is too small, it means that an arm has been sampled less than its empirical log-probability of optimality (or unlikelihood of optimality), *i.e.* $\frac{\mathbb{N}}{t} < \frac{\exp\left(-N_a(t)\mathcal{E}_{\mathcal{F}}\left(\hat{\mu}_a(t),\hat{\mu}^\star(t)\right)\right)}{Z(t)}$ where $Z(t)$ is the normalization constant.

In our setting, there is at least q elements in each group. It therefore makes sense to test for the optimality of a group rather single elements. Since all arms are independent, it makes sense to sum the *log-probability of optimality* or *empirical unlikelihood of optimality* on all the q-partitions of the set of arms. Since we have the intuition that this first part is the logarithm of a product of probability, we may compare it to the product of the empirical sample frequencies (and thus sum of the log-frequencies). Therefore, we get that important quantities are the sum of IMED indexes for each q partition of the arms, seen as a comparison between the optimality of this group of q elements and the associated frequency of play of that group. The minimal IMED index is the one whose frequency of play is the lowest compared to its *unlikelihood of optimality*, similarly for the sum of IMED indexes.

Other algorithmic ideas, the speed at which we learn

At the beginning of this project, a lot of methods were first tested, some with numerical success but without theoretical guarantees. All of those ideas were based on the idea of **aggregating similar arms**. This meant finding ways to cluster arms and aggregate information from them since those arms were supposed to share the same expected reward. The form of the algorithm I investigated the most was that of a UCB algorithm and used the idea of Z-tests to aggregate arms. By aggregating samples,

the hope was to benefit from a better estimation of the mean and the number of associated samples was replaced by a larger pseudo-number of samples computed from the aggregation of other arms.

Another natural idea was to use a q-nearest neighbor instead of Z-tests and aggregate samples of the q arms that are the closest in terms of expected values. One could obviously weight the samples using a kernel to attribute more weight to the closest neighbors and less to the farthest. The pseudo-number of samples would then be the weighted sum of the number of samples of the aggregated arms.

The main problem with these aggregation methods is that the gain of information we hope for thanks to the aggregation is nullified by the risk of performing a wrong aggregation, especially when the number of samples of suboptimal are supposed to be logarithmic at most. To estimate the groups correctly enough in order to exploit that information as if we had the knowledge of the groups, we may need (I couldn't prove otherwise) a number of samples that is linear in the number of interactions. If this number of samples is not linear, then it seems like we cannot exploit the computed groups because of the too large probability of aggregating arms that are not similar.

The solution we found was to understand that **estimating the problem** is different from **solving the problem**. As IMED-EC proves in the next sections, concerned about its regret and numerical efficiency, having knowledge of the groups is not necessary to exploit the structure. In fact, IMED-EC exploit the other fact that in our setting, we are interested in expected regret and that there is no additional notion of risk. Since all the arms are replicated (from an expected reward point of view) at least q times, then why not being q times more greedy in its selection. Coincidentally, by being q times more greedy, we will see that IMED-EC tends to sample only one arm from the optimal class and ignore others, a way of doing thing that is reminiscent of the approach we took in the introduction of this chapter with the random selection of one arm per group when we assumed the knowledge of the groups.

5.6 Regret of IMED-EC

In this section, we state the upper bound on the logarithmic growth rate of IMED-EC whose prove is given in the paper this chapter is based on, [70]. The proof of Theorem 5.6.1 is based on the proof of IMED [16] and is given the companion paper [70] of this chapter. The regret bound is derived from a logarithmic growth rate of sample of suboptimal arms. Confirming our intuition from the introduction of this chapter, we derive that the expected number of pulls of a suboptimal arm is a fraction q of what it would be in the unstructured setting thus making the regret q times smaller than what it could be in the unstructured setting. This is on par with what we would have with the knowledge of the groups and replacing all the cardinals of groups $|c|$ by their lower bound q, as we already mentioned in Section 5.3.

[70]: Pesquerel et al. (2021), 'Stochastic bandits with groups of similar arms'

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

[70]: Pesquerel et al. (2021), 'Stochastic bandits with groups of similar arms'

Theorem 5.6.1 (Upper bound on the number of pulls) *Under the IMED-*

EC algorithms, the number of pulls of a suboptimal arm a is upper bounded by:

$$\mathbb{E}_\nu(N_a(T)) \leq \frac{\log T}{q\mathcal{E}_\mathcal{F}(\nu_a, \mu^\star)}(1+\alpha(\epsilon)) + f(\epsilon), \tag{5.9}$$

where $0 < \epsilon < \frac{1}{3}\min_{a\in A\setminus A^\star}(\mu^\star - \mu_a)$, f is function that depends on concentration properties on $\mathcal{F}$, and α tends to 0 as ϵ tends to 0.

From Theorem 5.6.1, we deduce that the asymptotic logarithmic sampling rate of a suboptimal arm a is such that

$$\liminf_{t\to+\infty} \frac{\mathbb{E}_\nu(N_a(T))}{\log T} \leq \frac{1}{q\mathcal{E}_\mathcal{F}(\nu_a, \mu^\star)}, \tag{5.10}$$

and the asymptotic logarithmic growth rate of regret of IMED-EC on a problem $\nu = (\beta, A)$ is such that,

$$\liminf_{T\to\infty} \frac{\mathcal{R}_\nu(T; \text{IMED-EC})}{\log T} \leq \frac{1}{q}\sum_{a\notin A^\star} \frac{\mu^\star - \mu(a)}{\mathcal{E}_\mathcal{F}(\beta(a), \mu^\star)},$$

exactly what could have been wanted and intuited from Section 5.3. On expectation, suboptimal arms are therefore sampled q times less than in the unstructured setting. In Section 5.3, where groups were known, a suboptimal arm was sampled on average $|c|$ times less than in the unstructured setting, where c is the class the considered suboptimal arm belongs to. Given that only knowledge we have on c is that its cardinal $|c|$ is lower bounded by q, $|c| \geq q$, one can say that IMED-EC do a good job at exploiting the structure. Indeed, if replace $|c|$ by q in the sentence *with knowledge of the groups suboptimal arms are sampled on average $|c|$ times less than in the unstructured setting*, we get the one that is written for IMED-EC. Contrary to an algorithm with the knowledge of the groups, IMED-EC achieve this theoretical performance by being q times more greedy than the original IMED algorithm, and sampling suboptimal arms q times less often on average.

Upper and lower bound comparison

This upper bound shows that in particular, the number of pulls of a suboptimal class, $\sum_{a\in c}\mathbb{E}_\nu(N_a(T))$ is asymptotically no more than $\frac{|c|}{q\mathcal{E}_\mathcal{F}(\nu_a,\mu^\star)}\log T$, thus matching the lower bound in the *non-combinatorial regime*. When class c is in the *combinatorial regime*, along with Equation 5.8, this sampling upper bound shows that

$$\frac{|c|}{q\mathcal{E}_\mathcal{F}(\nu_a,\mu^\star)} \geq \liminf_{T\to\infty}\sum_{a\in c}\frac{\mathbb{E}_\nu(N_a(T))}{\log T} \geq \frac{1}{2}\cdot\frac{|c|}{q\mathcal{E}_\mathcal{F}(\nu_a,\mu^\star)},$$

proving that the regret of the proposed IMED-EC does not differ from the optimal lower bound by a factor more than 2. This is a striking result. Equation 5.8 can be used to have an even more precise control on the discrepancy to the optimal regret bound, as it shows the factor 2 can be actually replaced with $1 + \frac{|c|-q}{q}\frac{\min_{\kappa\in\nu}\mathcal{W}_\mathcal{F}(\nu_a,\kappa)}{\mathcal{E}_\mathcal{F}(\nu_a,\mu^\star)}$. Since the factor $\frac{|c|-q}{q}\frac{\min_{\kappa\in\nu}\mathcal{W}_\mathcal{F}(\nu_a,\kappa)}{\mathcal{E}_\mathcal{F}(\nu_a,\mu^\star)}$ is strictly between 0 and 1 in the combinatorial regime that we are studying, the discrepancy between the lower bound and the regret of IMED-EC indeed is always bounded by 2. On the other hand,

this refined error measurement is problem dependent while the factor of 2 is universal.

We now move on to the experimental section to test whether IMED-EC is numerically efficient. Given the efficiency of the IMED algorithm it is based on, one can certainly hope so.

5.7 Experiments

In this section, we support the theoretical analysis by conducting three sets of experiments. The Python code used to perform those experiments is available on github. We support our empirical evidences using plots of cumulative regrets. In this section, all the experiments are conducted using truncated Gaussian distributions with supports in $[0, 1]$ and means strictly upper bounded by 1. Along with the mean regret, we plot the quantiles 0.1 and 0.9, as well as the tube that this two curves form, and median regret curve (quantile 0.5) to give an idea of the spread the regret curves. The number of experiments and horizon depend on the considered problem as well as the number of tested algorithms.

We distinguish several relevant cases to assess the performances of IMED-EC. First we consider the case where the classes are balanced, *i.e.* of the same cardinality, and give IMED-EC the exact value of this cardinal, *i.e.* perfect knowledge for IMED-EC. Then, still considering balanced groups, we give IMED-EC a strict lower bound on the cardinal of the classes. This lead us to the experiment that test, in the balanced setting, the influence of parameter q on the regret. We run such an experiment where q varies from one to the value of the cardinal of the classes. Afterward, we run a similar set of experiments with unbalanced groups of arms. In this scenario, a tight bound can no longer be tight for all the groups at the same time, it will be tight for the smallest of the groups.

Balanced classes

Knowledge of a tight q

In this set of experiments, see Figure 5.5, we focus on the bandit configurations in which all equivalence classes have the same cardinality and assume that IMED-EC is given the number of elements per class as its parameter q. This setting is interesting for two reasons. First, one can compute the theoretical lower bound without solving a combinatorial optimization problem. Second, the theoretical analysis shows that IMED-EC is asymptotically optimal in this case. This setting will thus allow us to numerically grasp what happens in the most structured case of our structured setting. We compare IMED-EC to unspecialized bandit algorithm, UCB, IMED and KL-UCB. To make the comparison fairer we also compare IMED-EC to OSSB, an algorithm specialized in structured bandit. Since OSSB has to solve a combinatorial optimization problem at each time step, we cannot carry experiments on large sets of arms while comparing IMED-EC to it. In the Figure 5.5, we plot the result of an experiment run a Bandit problem with 9 arms, distributed in 3 groups of 3 arms. The groups expected values are 0.3, 0.5, and 0.9. In this

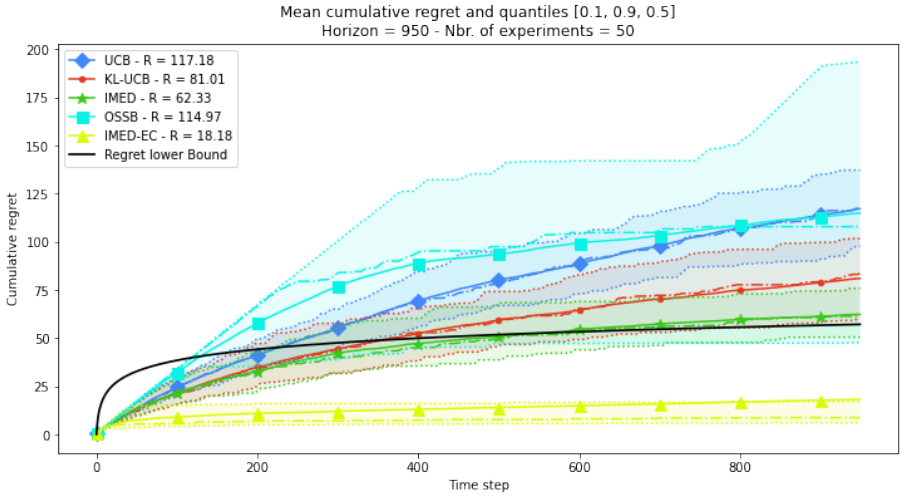

Figure 5.5: 3 classes, 3 distributions per class - set of means = $\{0.3, 0.5, 0.9\}$

particular setting, we see that while OSSB and IMED-EC are provably asymptotically optimal, IMED-EC numerically performs better in finite time horizon. We recall that it is furthermore numerically more efficient since it does not involve any combinatorial optimization. Unsurprisingly, IMED-EC also outperforms unspecialized algorithm.

Knowledge of a strict lower bound q

In the experiment plotted Figure 5.6, the parameter q that IMED-EC is given, it is 2, is now a strict lower bound on the size of the classes, which is 8, while the classes are still balanced. We compare IMED-EC to unspecialized bandit algorithm, IMED and KL-UCB. We drop OSSB from our test bed due to the computational burden of solving a combinatorial optimization problem at each time step. In the Figure 5.6, we plot the

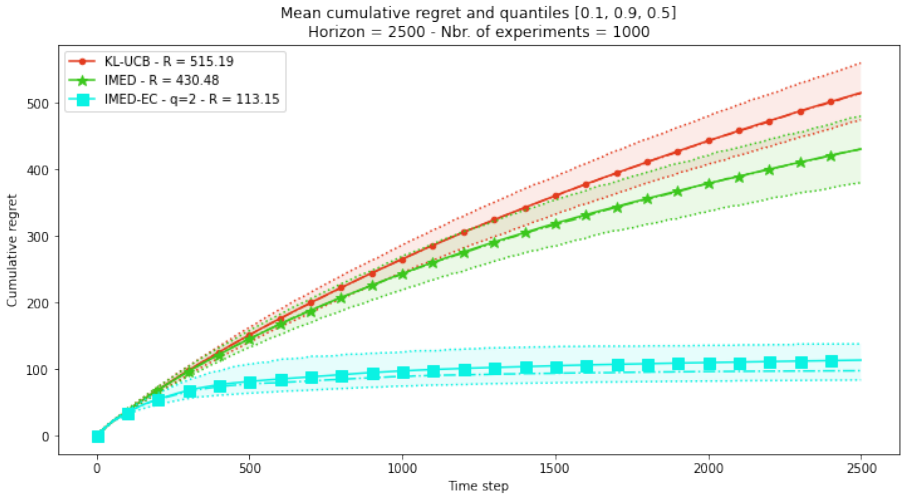

Figure 5.6: 7 classes, 8 distributions per class - set of means = $\{0.1, 0.3, 0.4, 0.5, 0.6, 0.75, 0.9\}$

result of an experiment run a Bandit problem with 7 groups of 8 arms. The groups expected values are 0.1, 0.3, 0.4, 0.5, 0.6, 0.75, and 0.9. We can

see that the finite time cumulative regret of IMED-EC indeed is much smaller than the regret of the unspecialized algorithms.

Influence of the parameter q

Here we show the numerical robustness of IMED-EC with respect to the lower bound parameter q on the number of elements per classes. On the same bandit problem made of 4 classes and 10 arms per group, we compare different instances of IMED-EC where different values of q are used. In the legend, *opt.* stands for optimal and corresponds to the largest valid lower bound on the number of elements per class, *i.e.* the minimal number of elements in a class, which is 4 in this case. The expected values of the groups are 0.1, 0.3, 0.6, and 0.9. While q increases

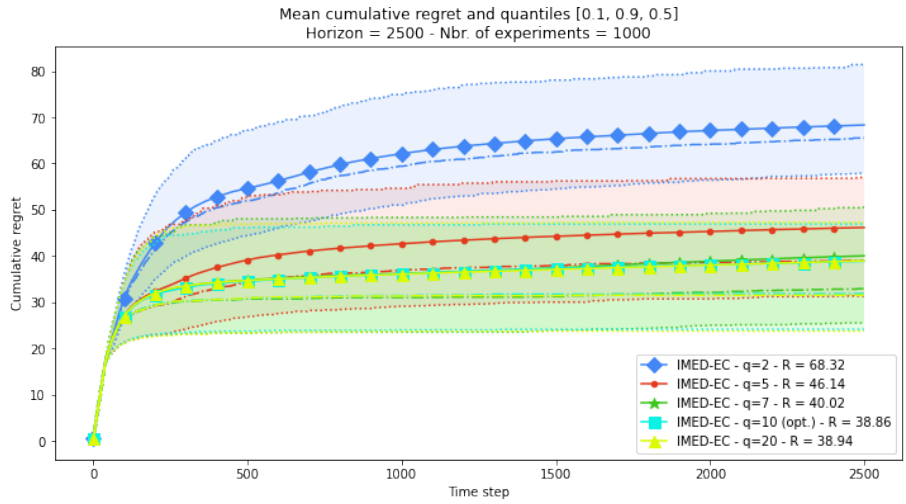

Figure 5.7: 4 classes, 10 distributions per class - set of means = $\{0.1, 0.3, 0.6, 0.9\}$

up to the minimum cardinality of a class, we see on Figure 5.7 that the performances of IMED-EC increases, which is expected. It is rather remarkable that once we go beyond that theoretical threshold of 4, the performances of IMED-EC do not seem to deteriorate. Therefore, we see that IMED-EC is robust to misspecification of parameter q in the setting of balanced classes of arms. In the context of unbalanced classes, we will see that IMED-EC is still robust to misspecification of q but that it is possible to find instances in which the quantile 0.9 is of linear shape.

Unbalanced classes

Comparison with IMED and KL-UCB

In this experiment, see Figure 5.8, we focus on the bandit configurations in which equivalence classes have different cardinals and assume that IMED-EC is given a strict lower bound as its parameter q. In the next experiment Figure 5.9, we study the influence of q on the performances of IMED-EC on the same problem and therefore consider the case where a tight lower bound is given to IMED-EC. We consider a bandit problem with 7 classes and an uneven number of distributions per class. The smallest class has 4 elements and the largest 23. In the experiment

plotted Figure 5.6, the parameter q that IMED-EC is given, it is 2, a strict lower bound on the size of the classes, and indicate the set of expected rewards in the legend. We compare IMED-EC to unspecialized bandit algorithm, IMED and KL-UCB. In the Figure 5.8, we plot the results of

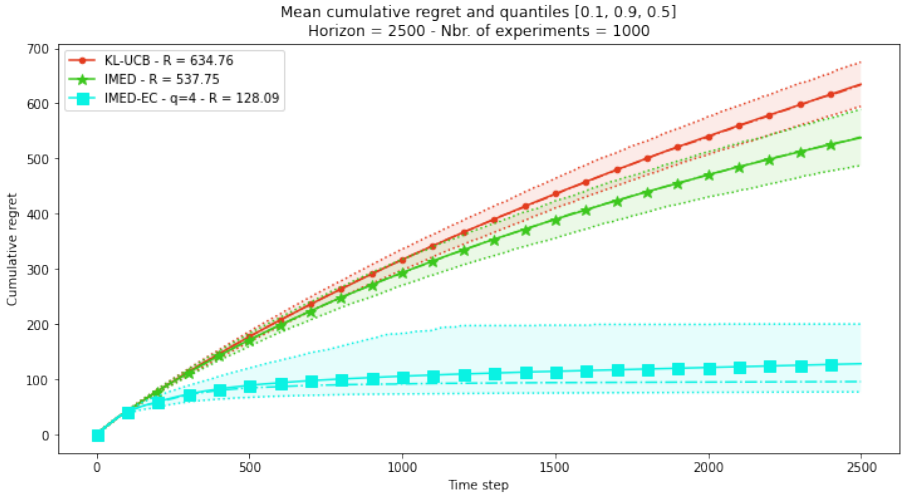

Figure 5.8: 7 classes, unbalanced - set of means = $\{0.1, 0.3, 0.4, 0.5, 0.6, 0.75, 0.9\}$

the experiment, and it can be seen that finite time cumulative regret of IMED-EC indeed is much smaller than the regret of the unspecialized algorithms. The control on the quantile tube and median also show that IMED-EC is a good algorithm that enjoy some of the original properties of IMED. Furthermore, on this experiment and all the other, the median regret is very close if not equal to the expected regret, which is also a nice experimental property.

Influence of the parameter q

Here we show the numerical robustness of IMED-EC with respect to the lower bound parameter q on the number of elements per classes. The experiment Figure 5.9 is performed on a bandit problem with 7 classes and an uneven number of distributions per class. The smallest class has 4 elements and the largest 23. While q increases up to the minimum cardinality of a class, we see that the performances of IMED-EC increases. It is rather remarkable that once we go beyond that theoretical threshold, the expected and median performances of IMED-EC do not deteriorate. We even found it difficult to find settings to deteriorate them at all. While the expected regret does not seem to deteriorate, we sometimes see that the tails of the regret widen as it can be seen on the plot Figure 5.9 for $q = 7$ and $q = 20$ since the 0.9 quantile curves are so large for those values of q. We attribute part of this robustness to the fact that the relaxation induced in IMED-EC makes the algorithm over explore compared to what the true lower bound suggests. Increasing q reduces the exploration and therefore may improve the performances of the algorithm. However, this robustness is observed even in the case where the classes are balanced. This interpretation thus does not explain everything about the numerical robustness of IMED-EC. There is also the possibility that we could not find the right setting and right horizon to make the algorithm fail. All in all, those experiment confirms the efficiency of IMED-EC.

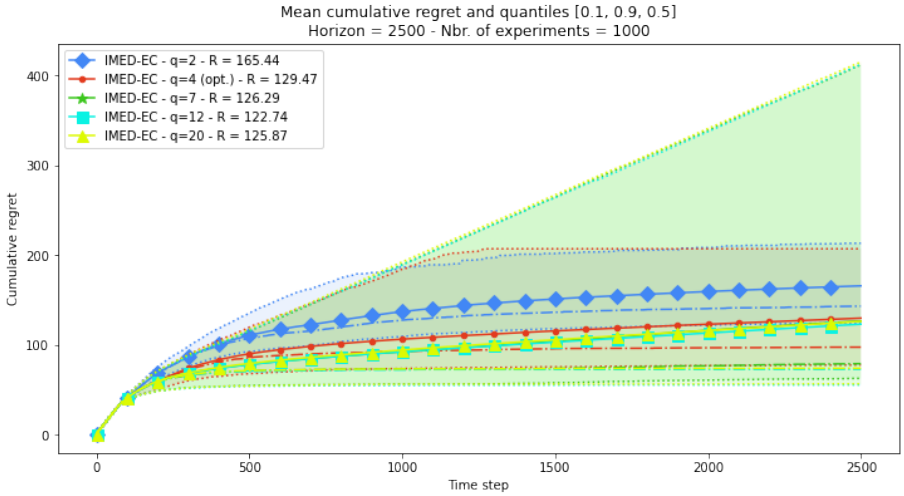

Figure 5.9: 7 classes, unbalanced - set of means = $\{0.1, 0.3, 0.4, 0.5, 0.6, 0.75, 0.9\}$

5.8 Fairness

In this section, we numerically explore the *fairness* of the IMED-EC algorithm. By studying *fairness*, we mean to compare the discrepancy in the number of pulls within a class. In particular, we are interested in the behavior of the different sampling strategies within the optimal class. We compare the *fairness* of IMED-EC to the one of KL-UCB and IMED. In the considered Bandit settings, for each algorithm, we report for each class the histogram accounting for the number of times distributions within each class has been pulled. Specifically, at the end of an episode, we sort arms within each class by the number of times they have been sampled. This gives an ordered statistic of the number of pulls of each arm within a class. In the fairness experiment, we are interested in the empirical histograms of the ordered statistics of number of samples within each group. The histograms are therefore built using those sorted number of pulls. Error bars corresponds to the standard deviations and have been clipped to not go below the x-axis.

Balanced problem

On a bandit problem whose number of classes is 4 with expected rewards 0.1, 0.3, 0.6, and 0.9 and 10 distributions per group. The chosen horizon is 2000, and we run 1000 independent experiments. IMED-EC is assumed to be given the tight bound $q = 10$. When running other experiments where IMED-EC was not given a tight bound, we did not find any significant differences in the histogram. The main difference was in the size of the standard deviation in the second-best class where it was higher for lower q, indicating a slight higher level of exploration. We consider the histogram in order, from the most suboptimal class to the optimal one. In the legend of the figures, the considered class is represented by a bold font.

For the most suboptimal class, Figure 5.10, not much can be said since the number of pulls is very small. Still, one can see that the progression of the order statistics for KL-UCB and IMED is somewhat *linear* while it

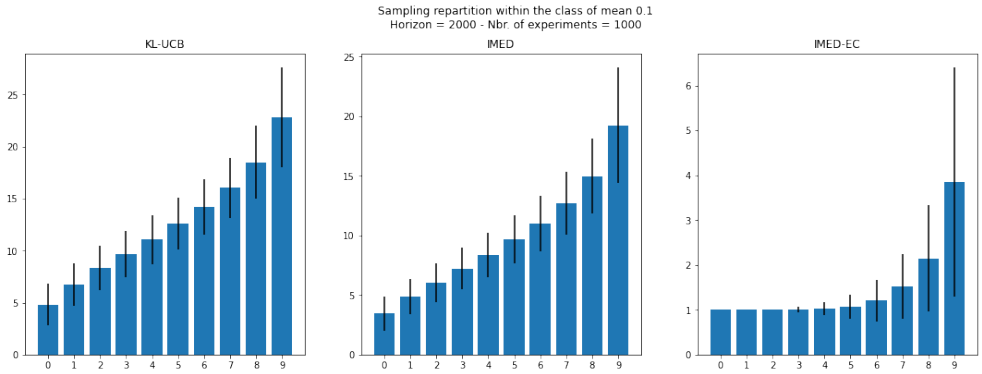

Figure 5.10: 4 classes, 10 distributions per class - set of means = {**0.1**, 0.3, 0.6, 0.9} - class of mean 0.1

seems more *exponential* for IMED-EC. (Note that we use these terms here informally.)

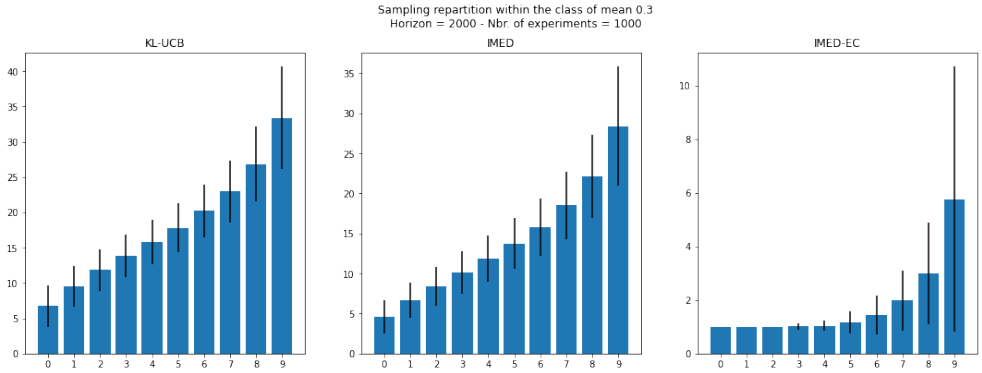

Figure 5.11: 4 classes, 10 distributions per class - set of means = {0.1, **0.3**, 0.6, 0.9} - class of mean 0.3

The same linear versus exponential apparent behavior can be seen on Figure 5.11.

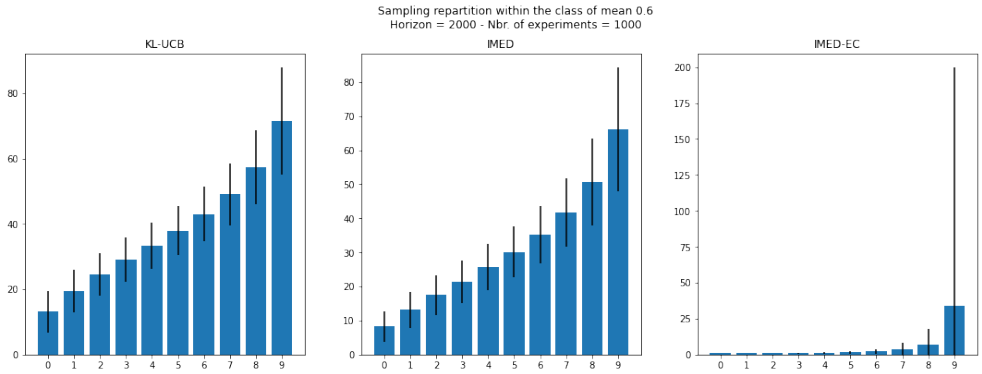

Figure 5.12: 4 classes, 10 distributions per class - set of means = {0.1, 0.3, **0.6**, 0.9} - class of mean 0.6

On Figure 5.12, one can clearly a difference in the behavior of KL-UCB and IMED, that have *small* error bars, and IMED-EC that have a large error bar for the most pulled arm within the least suboptimal class. We can clearly see how risky it might be to reduce the exploration from this error bar. However, this risk is compensated by the fact that there is at least q similar distributions. This can be read from the fact that the sum of all the number of pulls within this class for KL-UCB and IMED is above

200 which is roughly the maximal number of pulls of IMED-EC within this class computed using the upper bounds (given by the maximal value of the standard deviation).

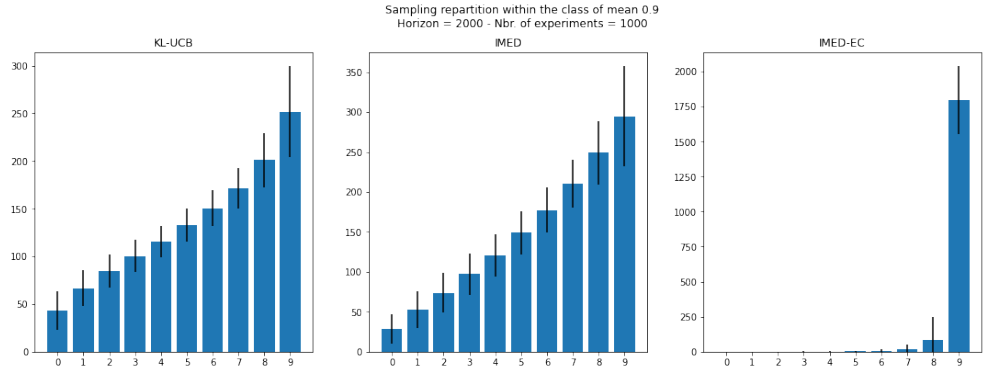

Figure 5.13: 4 classes, 10 distributions per class - set of means = $\{0.1, 0.3, 0.6, \mathbf{0.9}\}$ - class of mean 0.9

Finally, Figure 5.13 enables to compare the behaviors of the algorithms within the optimal class. It seems clear that, at least numerically, IMED-EC is not a fair algorithm in finite time. IMED-EC does not equally distribute the pulls between arms from the same class and seems to leverage the lower bound on the number of elements per class to play a more risky strategy, and benefits from it, from an expected regret viewpoint. Again, we observe the linear versus exponential progression in the order statistics of the number of pulls. In this regard, KL-UCB and IMED seems fairer algorithm within the optimal class.

Unbalanced problem

In this final experiment, we consider a bandit problem whose number of classes is 7 with means 0.1,0.3,0.4,0.5,0.6,0.75, and0.9 and an uneven number distributions within each class. The chosen horizon is 2000, and we run 1000 independent experiments. We assume that IMED-EC does not have a tight lower bound on the number of elements per class, and we use 3 as the lower bound parameter of IMED-EC. For four classes, including the optimal one, we report the histograms of the corresponding ordered statistics. As above, error bars corresponds to the standard deviations and have been clipped to not go below the x-axis.

The comments that can be made about those plots are similar to the one that were already made. We included them to show that the behavior of IMED-EC (and also the behavior of IMED and KL-UCB) is consistent across multiple settings. In particular, the algorithm IMED-EC exhibits the same aforementioned behavior for the least suboptimal class, as it can be seen by comparing Figure 5.16 to the corresponding Figure 5.12. In a nutshell, IMED-EC is not a fair algorithm while it seems that, at least for the optimal group, IMED and KL-UCB seem to exhibit linear-shaped ordered statistics. On the other hand, IMED-EC is indeed q times more greedy which benefit the agent from an expected regret viewpoint but contribute to the absence of fairness within the optimal class.

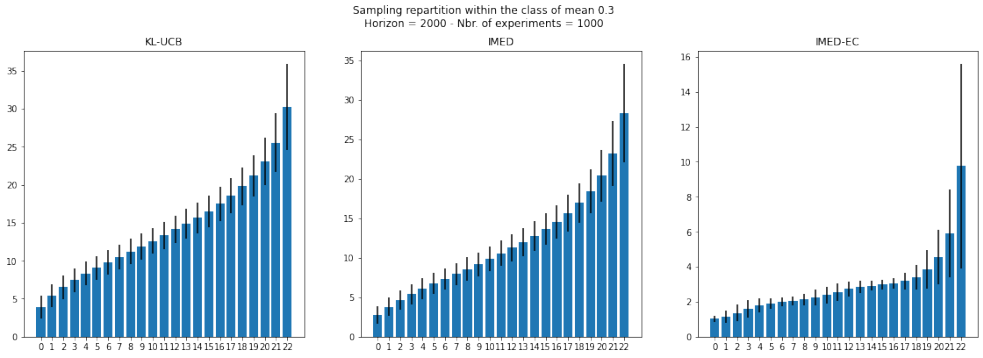

Figure 5.14: 7 classes - unbalanced - set of means = $\{0.1, \mathbf{0.3}, 0.4, 0.5, 0.6, 0.75, 0.9\}$ - mean 0.3

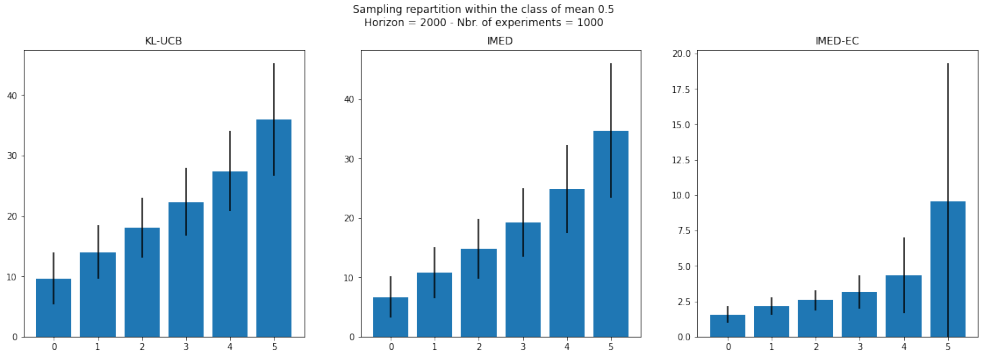

Figure 5.15: 7 classes - unbalanced - set of means = $\{0.1, 0.3, 0.4, \mathbf{0.5}, 0.6, 0.75, 0.9\}$ - mean 0.5

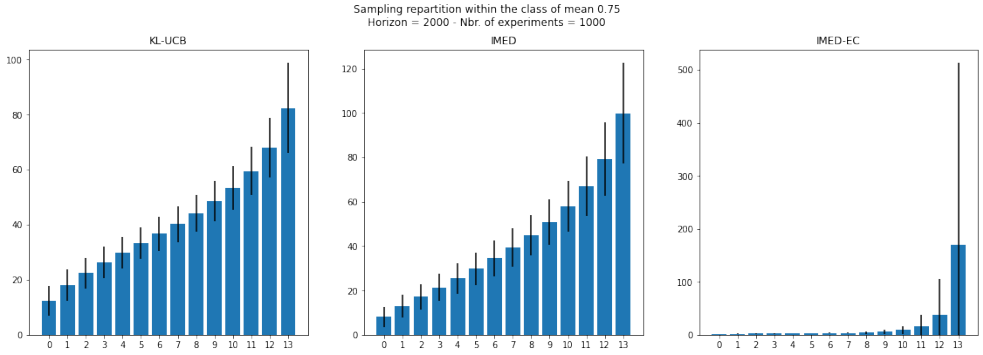

Figure 5.16: 7 classes - unbalanced - set of means = $\{0.1, 0.3, 0.4, 0.5, 0.6, \mathbf{0.75}, 0.9\}$ - mean 0.75

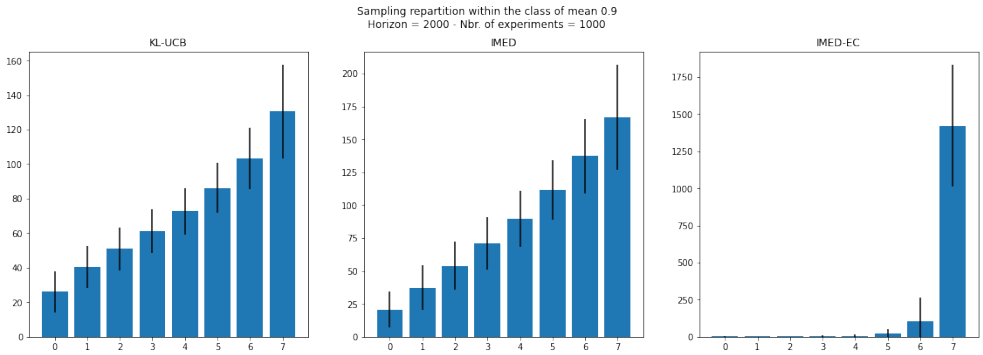

Figure 5.17: 7 classes - unbalanced - set of means = $\{0.1, 0.3, 0.4, 0.5, 0.6, 0.75, \mathbf{0.9}\}$ - mean 0.9

5.9 Conclusion

In this chapter, we introduced IMED-EC, a numerically efficient algorithm to solve a structured bandit problem for which we derived a lower bound involving a combinatorial optimization problem. While not being asymptotically optimal, the asymptotic regret of IMED-EC is always smaller than the unstructured one and that we can control the discrepancy with respect to the structured regret lower bound by a factor of at most 2. It is possible for IMED-EC to benefit from the $\mathscr{E}_{\mathcal{F}}$ approximation scheme introduced in the previous chapter 4 and in the future, it may be possible to use an approach *à la* IMED-EC to tackle Reinforcement Learning problem in which symmetries in the action space emerge from physical constraint thus creating equivalence classes of actions or policies. This would complete the introductory example that we gave where paths in a grid could be seen as actions in a structured Bandit with groups of similar arms.

Reinforcement Learning

A group of choices: reinforcement learning

6

6.1 A first order model of decision-making

After introducing the Bandit model in the previous chapter, we more briefly present the second sequential learning model that we studied in this thesis. Most of the built intuitions in the previous chapter can be applied to understand this more complex setting. Compared to the Bandit learning problem, Reinforcement Learning is a less mature research field. For instance, there are no instance dependent regret lower bound in the general case and little is known about how to efficiently navigate the problem, where navigation is a problem that is specific to Reinforcement Learning and was not present in the Bandit learning problem.

We model a **first order model** of **sequential decision-making**. This model is called a **Markov Decision Process**. In this model, as in the previous, an **agent**, will have to **sequentially** make **decisions**. Often, the agent is an **algorithm**. Mathematically, the **sequential** aspect of the decision-making process is modeled by the fact that the agent **must** make a decision sequentially at **abstract time steps** that belong to an **ordered set**. Mathematically, **making a decision** is modeled thanks to a **set** whose elements are interpreted as **decisions** or **actions**. The consequence of making a decision from a set is made known to the agent thanks to a **numerical signal** belonging to the set of real numbers, $\mathbb{R}$. The main difference with the previous Bandit case is that the set of decision the agent can choose from is dependent on another set, the **state space**. Upon making a decision in a given state, the agent is transported in another state according to a transition probability that depends on the current state and chosen action in that state. We are interested to model only those situations in which the agent can observe the consequences of its decisions and only those. A **Markov decision problem** emerges when the agent has an objective that is a function of the Bandit model. The agent will have to solve its objective using gathered information about the model through interaction with it. Of course, its interaction is modeled through the sequence of decisions and associated numerical rewards. Therefore, we can see that some additional structure might be necessary on the decision-making model to model how the agent **measure the consequences of its decision with respect to its internal objective**.

Remark. In science, order of approximation is way to referring to the accuracy of an approximation. It can be informal, as it is here, or formal, particularly when referring to the Taylor expansion of a function.

Figure 6.1: Andreï Andreïevitch Markov (1856-1922)

We first present the **Markov control model** and then explain the average-reward **Reinforcement Learning** (RL) model. The presentation hereafter is much shorted than the previous one for two reasons. First, a lot of the intuitions about pointers, information, decision, etc. apply to this framework. They were specifically introduced to build intuition in the previous easier setting. The second is that the theory of RL is far less understood than the Bandit theory. For instance, there is no known generic lower bound for the regret minimization problem in Reinforcement Learning, and therefore even less "optimal" algorithm. One of the main contribution of the thesis was to develop an Algorithm 21, IMED-RL, that tackle the problem of average-reward Reinforcement Learning in a

setting called ergodic, one of the few RL setting for which a lower bound is known.

Markov Control Model

In this thesis, we study Reinforcement Learning with an unknown finite Markov Decision Problem (MDP) under the average-reward criterion in which a learning algorithm interacts sequentially with the dynamical system, without any reset, in a single and infinite sequence of observations, actions, and rewards while trying to maximize its total accumulated rewards over time. Prior to defining the learning problem, we present the Markov control problem.

In a Markov decision model the agent must be able to **measure** the consequences of its decision. By measuring, we mean that it must be able to evaluate the consequences of its decision. This motivates the following definition of a **Markov control model**.

Definition 6.1.1 (Markov Control Model) *A **Markov Decision Process** (MDP) $\boldsymbol{M}$ is a five tuple*

$$(\mathcal{S}, A, \{A(x)|x \in \mathcal{S}\}, \boldsymbol{p}, \boldsymbol{r})$$

consisting of

1. *a **Borel space** $\mathcal{S}$, called the **state space** and whose elements are referred to as **states**. We denote $\mathfrak{X}$ the σ algebra on $\mathcal{S}$;*
2. *a **Borel space** A, called the control or **action set** i.e. A is a couple $(A, \mathscr{A})$ where $\mathscr{A}$ is a σ algebra on A;*
3. *a family $\{A(x)|x \in \mathcal{S}\}$ of non-empty measurable subsets $A(x)$ of A i.e. $A(x) \in \mathscr{A} \forall x \in \mathcal{S}$. $A(x)$ denotes the **set of feasible controls** or **actions** when the system is in state $x \in \mathcal{S}$. Furthermore, the set*

$$\mathfrak{X}_{\boldsymbol{M}} = \{(x, a)|x \in \mathcal{S}, a \in A(x)\}$$

   *of **feasible state-action pairs** is a measurable subset of $\mathcal{S} \times A$. $\mathcal{S} \times A$ is endowed with the σ algebra $\mathcal{G}$ generated by $\mathfrak{X}$ and $\mathscr{A}$, $\sigma(\mathfrak{X} \times \mathscr{A})$;*
4. *a **stochastic kernel** P on $\mathcal{S}$ given $\mathfrak{X}_{\boldsymbol{M}}$ called the **transition law**;*
5. *a **measurable function** $\mathbf{r} : \mathfrak{X}_{\boldsymbol{M}} \to \mathcal{P}(\mathbb{R})$ called the **reward per stage** (or one stage reward) distribution;*

*Furthermore, we denote $\boldsymbol{m} : \mathfrak{X}_{\boldsymbol{M}} \to \mathbb{R}$ the **measurable function** called the **reward per stage** (or one stage reward) function that corresponds to the expected reward, $\boldsymbol{m}(s, a) = \mathbb{E}_{R \sim \boldsymbol{r}(s,a)}(R)$.*

When the transition kernel and reward distribution known, the problem of finding a policy that maximizes its expected cumulated reward is called a control model. By known reward distribution, it is meant that As for the Bandit setting, one can replace the reward function $\mathbf{m}$ by reward distributions $\mathbf{r} : \mathfrak{X}_{\mathbf{M}} \to \mathcal{P}(\mathbb{R})$ from which we assume that the reward function $\mathbf{m}$ is computable. If $\mathbf{m}$ can only be computed from sampling the system, *i.e.* interaction with it, then we are in a learning setting, even when the transition kernel is known, as in Chapter 8.

The noun **Markov** comes from the assumption of the transition kernel. The transition kernel is such that the distribution over the state space only depend on the current considered state and chosen action, not on the history of visited state-action pairs. This dependency to only the immediate past is a feature of the model. Of course, an agent is not necessarily Markovian in the sense that its decisions can depend on more than the immediate past. In a Markov decision model, the agent takes decisions **sequentially**, it is allowed to remember past information. Similarly to Bandit, we introduce a notion of history of information.

Definition 6.1.2 (Markov Control History) *Given a Markov control model* $(\mathcal{S}, A, \{A(x)|x \in \mathcal{S}\}, \boldsymbol{p}, \boldsymbol{m})$ *and its set* $\mathfrak{X}_{\boldsymbol{M}}$ *of feasible controls, for all* $t \in \mathbb{N}$*, one can define the set of* ***admissible histories*** *up to time* t

$$\begin{aligned} H_0 &= \mathcal{S} \\ H_t &= \mathfrak{X}_{\boldsymbol{M}}^t \times \mathcal{S} = \mathfrak{X}_{\boldsymbol{M}} \times H_{t-1} \qquad \forall t \in \mathbb{N}_* \end{aligned}$$

An element $h_t \in H_t$ *is called an admissible t-history or a* ***t-history*** *and is a vector of the form*

$$h_t = (x_0, a_0, \cdots, x_{t-1}, a_{t-1}, x_t)$$

with $(x_k, a_k) \in \mathfrak{X}_{\boldsymbol{M}}$ *for* $k \in [0, t-1]$ *and* $x_t \in \mathcal{S}$*.*

For all t*, the space* H_t *is equipped with a canonical* σ*-field inherited from the set of feasible state-action pairs i.e.,* H_t *is a measurable space with* σ*-field of* $\mathfrak{X}_{\boldsymbol{M}}^{\otimes t}$*.*

The decision-making process of an agent is modelled thanks to the notion of **control policy**. A control policy can be deterministic, stochastic, history-agnostic or not.

Definition 6.1.3 (Control Policy) *A* ***randomized control policy****, or* ***policy****, is a sequence* $\pi = (\pi_t)_{t\in\mathbb{N}}$ *of stochastic kernels* $\pi_t \in \mathcal{P}(A|H_t)$ *on the control set* A *given* H_t *satisfying the constraint*

$$\pi_t(A(x_t)|h_t) = 1 \ \forall h_t \in H_t, t \in \mathbb{N}$$

The set of all policies is denoted by Π*.*

As previously, it is important to assume that it is possible for an agent to make decision in the first place.

Assumption 6.1.1 *We will assume that* $\mathfrak{X}_{\boldsymbol{M}}$ *contains the graph of a measurable function from* $\mathcal{S}$ *to* A *i.e. there is a measurable function* $d : \mathcal{S} \to A$ *such that for all* $x \in \mathcal{S}$*,* $d(x) \in A(x)$*.*

Definition 6.1.4 (Admissible control) *We denote by* Φ *the set of all stochastic kernels* $\phi \in \mathcal{P}(A|\mathcal{S})$ *such that* $\phi(A(x)|x) = 1$*. We denote by* Δ *the set of all measurable functions* $d : \mathcal{S} \to A$ *satisfying that* $d(x) \in A(x)$ *for all* $x \in \mathcal{S}$*.*

One may identify a function $d \in \Delta$ as a stochastic kernel $\phi_f \in \Phi$ for

which $\phi(\cdot|x) = \delta_{f(x)}(\cdot)$ is the Dirac measure at $f(x)$ for all $x \in \mathcal{S}$ *i.e.*

$$\forall x \in \mathcal{S},\ \forall C \in \mathcal{A},\ \phi_f(C|x) = 1_C\{f(x)\} = 1\{f(x) \in C\}$$

where 1_C is the indicator function of the set C.

Policies can be categorized according to the same typology we described at length in Definition 3.1.5.

Definition 6.1.5 *A **policy** $\pi = (\pi_t)_t \in \Pi$ is said to be a*

- *randomized Markov policy if there exists a sequence $(\phi_t)_t$ of stochastic kernels $\phi_t \in \Phi$ such that*

$$\forall h_t \in H_t,\ t \in \mathbb{N},\ \pi_t(\cdot|h_t) = \phi_t(\cdot|x_t)$$

- *randomized (Markov) stationary policy if there exists a stochastic kernel $\phi \in \Phi$ such that*

$$\forall h_t \in H_t,\ t \in \mathbb{N},\ \pi_t(\cdot|h_t) = \phi(\cdot|x_t)$$

The set of all randomized Markov policies is denoted Π_{RM}. The set of all randomized stationary policies is denoted Π_{RS}. By definition of stationarity, a randomized stationary policy necessary is Markov. Indeed, if all ϕ_t are the same kernel ϕ then they all must be kernel on A given a common space H. The largest common subspace to all histories is $H_0 = \mathcal{S}$. Note that $\Pi_{RS} \subseteq \Pi_{RM} \subseteq \Pi$.

- *deterministic (or pure) policy if there exists a sequence $(g_t)_t$ of measurable functions $g_t : H_t \to A$ such that for all $h_t \in H_t$ and $t \in \mathbb{N}$, $g_t(h_t) \in A(x_t)$ and $\pi_t(\cdot|h_t)$ is a Dirac concentrated at $g_t(h_t)$ i.e.*

$$\forall C \in \mathcal{A},\ \ \pi_t(C|h_t) = 1_C(g_t(h_t))$$

- *deterministic Markov policy if there is a sequence $(f_t)_t$ of functions $f_t \in \mathbb{F}$ such that $\pi_t(\cdot|h_t)$ is concentrated at $f_t(x_t) \in A(x_t)$ for all $h_t \in H_t$ and $t \in \mathbb{N}$.*
- *deterministic (Markov) stationary policy if there is a function $f \in \mathbb{F}$ such that $\pi_t(\cdot|h_t)$ is concentrated at $f(x_t) \in A(x_t)$ for all $h_t \in H_t$ and $t \in \mathbb{N}$.*

The set of all deterministic policies is denoted Π_D. The set of all deterministic Markov policies is denoted Π_{DM}. The set of all deterministic stationary policies is denoted Π_{DS}. Note that $\Pi_{DS} \subseteq \Pi_{DM} \subseteq \Pi_D \subseteq \Pi$.

We also have that $\Pi_{DS} \subseteq \Pi_{RS}$. Thus, Π_{DS} is contained is all the aforementioned set of policies. To ensure that those are non-empty, it suffices to assume that Π_{DS} is non-empty which the same as our main assumption, that is to say, assuming that there exist a measurable function $f : \mathcal{S} \to A$ such that $f(x) \in A(x)$ for all $x \in \mathcal{S}$.

Assumption 6.1.2 (Finite Rewards) *The Markov control model $(A, \boldsymbol{m})$ satisfies the **finite reward** assumption if for all actions $(x, a) \in \mathcal{X}_{\boldsymbol{M}}$ we have $-\infty < \boldsymbol{m}(x, a) < +\infty$.*

On an MDP $\mathbf{M}$, each stationary deterministic policy $\pi : \mathcal{S} \to \mathcal{A}_s$ defines

a Markov reward process, *i.e.* a Markov chain on $\mathcal{S}$ with kernel

$$\mathbf{p}_\pi : s \in \mathcal{S} \mapsto \mathbf{p}(\cdot|s, \pi(s)) \in \mathcal{P}(\mathcal{S}) \ ,$$

together with rewards

$$\mathbf{r}_\pi : s \in \mathcal{S} \mapsto \mathbf{r}(s, \pi(s)) \in \mathcal{P}(\mathbb{R}) \ ,$$

and associated mean rewards

$$\mathbf{m}_\pi : s \in \mathcal{S} \mapsto \mathbf{m}(s, \pi(s)) \in \mathbb{R} \ .$$

The t-steps transition kernel of policy π on $\mathbf{M}$ is denoted $\mathbf{p}_\pi^t$. We denote

$$\overline{\mathbf{p}}_\pi = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{p}_\pi^{t-1} : \mathcal{S} \to \mathcal{P}(\mathcal{S})$$

the Cesàro-average of $\mathbf{p}_\pi$. This Cesàro-average of the transition kernel of the Markov $\mathbf{M}_\pi$ allows us to correctly define quantity related to average behavior because it erases periodic features. As such, it is a quantity that is more related to asymptotic properties of a policy rather than finite time, but so is the concept of average reward.

A learning agent is executing a sequence of policies $\pi_t \in \Pi(\mathbf{M})$, $t \geq 1$, where π_t depends on past information $h_t = (s_{t'}, a_{t'}, r_{t'})_{t'<t}$. With a slight abuse of notation, a sequence of identical decision rules, $\pi_t = \pi$ for all t, is also denoted π.

Definition 6.1.6 (Cumulated Reward) *Let*

$$\boldsymbol{M} = (\mathcal{S}, A, \{A(x) | x \in \mathcal{S}\}, \boldsymbol{p}, \boldsymbol{m})$$

*be a Markov control model satisfying the finite reward 6.1.2 and optimal feasibility 6.1.1 Assumptions, and let $\pi \in \Pi$ be a control policy. The **n-stage cumulated reward** of the policy π on the MDP $\boldsymbol{M}$ starting at state s is defined as*

$$V_s(n; \pi, \boldsymbol{M}) = \mathbb{E}_{s,\boldsymbol{M},\pi} \left(\sum_{t=1}^{n} \boldsymbol{m}(s_t, a_t) \right). \tag{6.1}$$

*The **average reward** of the policy π on the Markov control model $\boldsymbol{M}$ is defined as*

$$\boldsymbol{g}_\pi(s, \boldsymbol{M}) = \liminf_{n \to \infty} \frac{V_s(n; \pi, \boldsymbol{M})}{n}. \tag{6.2}$$

As we can see, this definition is very similar to the one of Bandit and the average-reward setting is the direct generalization of the average-reward Bandit criterion. One of the most important theorem of Markov control theory states that, there exists an optimal policy that is deterministic and stationary. The average reward of an optimal policy is maximal and independent of the initial state s as long as the MDP is communicating.

Definition 6.1.7 (Communicating MDP) *The MDP $\boldsymbol{M}$ is communicating, if*

$$\forall s, s', \exists \pi, \exists t \in \mathbb{N} : \boldsymbol{p}_\pi^t(s'|s) > 0 \ .$$

In a communicating MDP, any two states can be joined with positive probability in finite time. This definition differ from the notion of ergodic MDP by the quantifier $\exists$ in front of the policy, which is replaced by a $\forall$ in the definition of ergodic MDP. We write the definition here to emphasize the difference between the two notions.

Definition 6.1.8 (Ergodic MDP) *The MDP $\boldsymbol{M}$ is ergodic, if*

$$\forall s, s', \forall \pi, \exists t \in \mathbb{N} : \boldsymbol{p}_\pi^t(s'|s) > 0 \, .$$

In an ergodic MDP, whatever the played policy, all states are connected in finite time with positive probability. In an ergodic MDP, all policies share the same set of recurrent states, which is equal to the whole state space. This is an important property that can be exploited to learn without caring too much about dynamic exploration as will be seen in Chapter 7.

[75]: Puterman (1994), *Markov Decision Processes — Discrete Stochastic Dynamic Programming*

[76]: Hernández-Lerma et al. (1996), *Discrete-Time Markov Control Processes*

A proof of the existence of an optimal deterministic stationary policy can be found in [75] for the finite state and action space case, while another proof, more general, can be found in [76].

Theorem 6.1.1 (Optimal Deterministic Stationary policy) *Let $\boldsymbol{M}$ be a Markov control model with finite state space and finite action space, i.e. $\mathfrak{X}_{\boldsymbol{M}}$ is finite, then there exists an optimal policy that is deterministic and stationary.*

The existence of such a policy allows to define a **learning criterion**, measuring the speed at which an agent that do not know the reward distributions $\mathbf{r}$, or the transition kernel $\mathbf{p}$, or even both, can converge to a such a policy. Note that there exists other framework than the average reward criterion such as the discounted or finite horizon settings. These two settings are presented in the reference books [75, 76], but we do not consider them in this thesis.

We note that the average-reward can be expressed after rewriting the cumulative reward (value). After T interactions, starting from an initial state s_1 of policy $\pi = (\pi_t)_t$ is formally given by, the cumulative reward is formally given by

$$\begin{aligned} V_{s_1}(\mathbf{M}, \pi, T) &= \mathbb{E}_{\pi,\mathbf{M},s_1}\left[\sum_{t=1}^{T} r_t\right] \\ &= \mathbb{E}_{\pi,\mathbf{M},s_1}\left[\sum_{t=1}^{T} \mathbf{m}(s_t, a_t)\right] \\ &= \sum_{t=1}^{T}\left(\prod_{t'=1}^{t-1} \mathbf{p}_{\pi_{t'}}, \mathbf{m}_{\pi_{t'}}\right)(s_1) \, . \end{aligned}$$

For $\pi \in \Pi(\mathbf{M})$, the average-reward $\frac{1}{T} V_{s_1}(\mathbf{M}, \pi, T)$ tends to $\left(\overline{\mathbf{p}}_\pi \mathbf{m}\right)(s_1)$ as $T \to \infty$. The gain of policy $\pi \in \Pi(\mathbf{M})$, when starting from state s_1 can therefore be written as

$$\mathbf{g}_\pi(s_1) = (\overline{\mathbf{p}}_\pi \mathbf{m})(s_1) \, .$$

Another way of writing the gain is in terms of state-action pair, which can

be useful. Starting from state-action pair $c_1 = (s_1, a_1)$, it can be written

$$\mathbf{g}_{c_1,\pi} \coloneqq (\overline{\mathbf{p}}_\pi \mathbf{m})(c_1) = \sum_{c \in \mathscr{C}} \overline{\mathbf{p}}_\pi(c_1, c) \cdot \mathbf{m}(c),$$

where $\overline{\mathbf{p}}_\pi(c_1, c)$ is the visit probability mass the pair $c \in \mathscr{C}$ under policy π started in pair $c_1 \in \mathscr{C}$, and where $\mathbf{m}(c)$ is the average reward of the pair c. That is to say, the gain can be written as an average of rewards that depends on the stationary distribution of an agent following policy π. The optimal gain, staring from s_1 is defined as $\mathbf{g}^\star(s_1) = \max_{\pi \in \Pi(\mathbf{M})} \mathbf{g}_\pi(s_1)$. The set of policies achieving maximal gain on $\mathbf{M}$ starting from state s is defined as

$$\mathcal{O}_s(\mathbf{M}) = \{\pi \in \Pi : \mathbf{g}_\pi(s) = \mathbf{g}^\star(s)\} .$$

Similarly, one can define state-action pair related notions of optimal sets. Given a *finite* set of stationary policies Π, we denote $\mathbf{g}_c^\star = \max_{\pi \in \Pi} \mathbf{g}_{c,\pi}$ the optimal gain starting from c, and $\Pi_c^\star = \{\pi \in \Pi : \mathbf{g}_{c,\pi} = \mathbf{g}_c^\star\}$ the set of policies achieving the optimal gain. Those definitions are mostly here for theoretical and notational purpose since, under the communicating hypothesis, $\Pi_c^\star$ is independent of c.

We recall that the optimal gain and therefore, set of optimal policies are independent of s in a communicating, and a fortiori ergodic, MDP. There are several notions of regret, and we will use the following.

Definition 6.1.9 (Regret) *The regret at time T of a learning policy $\pi = (\pi_t)_t$ starting at state s on an MDP $\boldsymbol{M}$ is defined with respect to any $\pi^\star \in \mathcal{O}_s(\boldsymbol{M})$, as*

$$\mathscr{R}_{\pi,s}\left(\boldsymbol{M}, T; \pi^\star\right) = V_s(\boldsymbol{M}, \pi^\star, T) - V_s(\boldsymbol{M}, \pi, T) . \tag{6.3}$$

This corresponds to a finite macroscopic viewpoint on the regret where the regret is defined at the policy level. Formally, one could have defined another notion of T-stage regret $\mathscr{R}_{\pi,s}(\mathbf{M}, T)$ as

$$\mathscr{R}_{\pi,s}(\mathbf{M}, T) = T\mathbf{g}^\star(s) - V_s(\mathbf{M}, \pi, T)$$

which correspond to an asymptotic macroscopic viewpoint. The regret is defined at the policy level using the asymptotic notion of gain, *i.e.* the average reward per unit of interaction.

Similarly to what we did for Bandit, one can define the set of what is considered "good" learner. To do so, we need to consider an assumption on the reward distribution. This assumption is formalized by saying that the learner will now, prior to interaction, that the MDP belong to set ζ. This set of learner, we call it the set of **Uniformly fast convergent policies**.

Definition 6.1.10 (Uniformly fast convergent policies) *Given a set ζ of MDPs, a policy π is uniformly fast convergent on ζ if for all MDP $\boldsymbol{M} \in \zeta$, the regret $T \mapsto \mathscr{R}_{\pi,s}(\boldsymbol{M}, T)$ is negligible compared to T^α for all state s and $\alpha > 0$, i.e.*

$$\forall \boldsymbol{M} \in \zeta, \ \mathscr{R}_{\pi,s}(\boldsymbol{M}, T) = o\left(T^\alpha\right) .$$

The word uniform is attached to the set ζ and the word fast to the highly sublinear growth rate of the regret function $o(T^\alpha)$ for all positive α.

Among the subset of policies of interest, here the subset of uniformly fast convergent polices, some may converge faster than others on some problems. Some may converge faster than any other policies on all problem in the set ζ. Those policies have the maximal convergence rate and achieve this maximality criterion uniformly on ζ.

Definition 6.1.11 (Uniformly maximal convergence rate policies) *Given a set ζ of MDPs, a policy $\pi^\star$ is said to have uniformly maximal convergent rate on ζ if for all MDPs $\boldsymbol{M} \in \zeta$, and all uniformly fast convergent policy π the regret $T \mapsto \mathcal{R}_{\pi^\star,s}(\boldsymbol{M}, T)$ is asymptotically smaller than the regret $T \mapsto \mathcal{R}_{\pi,s}(\boldsymbol{M}, T)$, i.e.*

$$\forall \nu \in \zeta, \ \limsup_{n\to\infty} \frac{\mathcal{R}_{\pi^\star,s}(\boldsymbol{M}, T)}{\mathcal{R}_{\pi,s}(\boldsymbol{M}, T)} \leq 1 \ .$$

We consider the lim sup of the ratio because the limit may not exist and the lim sup is more restrictive than a lim inf and better suited to define a sound notion of optimality. Note how this definition corresponds to the one we introduce as a lim inf on the gain. It can be read as the fact that we want to control the worst adherent point of the sequence of regret ratio. After defining such a learning class and optimality criterion, we are already more in the realm of research than that of knowledge. Little is known about the existence of uniformly fast convergent policies in general communicating MDPs. In this thesis, we investigate the restricted case of ergodic MDPs, for which a regret lower bound is known. Prior to presenting this lower bound and reviewing the relevant literature, we briefly present the intuition behind policy iteration, that we use in both Chapter 7 and Chapter 8.

6.2 Planning

In a Markov Decision Process, the space of policies is highly structured. Indeed, the average reward criterion (or discounted one for any discount factor) induces a partial ordering of the policies on the MDP. Furthermore, this partial ordering has the property that there exists a greatest elements called **optimal control policies**. This is a much nicer property than simple existence of maximal elements since those cannot be compared with all elements of the partially ordered set while it is the case for greatest elements. The existence of a greatest element means that, for all policy π_0, there exist a path $(\pi_0, \pi_1, \ldots, \pi_n)$ in the policy space, made of policies of *increasing order*, *i.e.* $\pi_{k+1} \geq_{MDP} \pi_k$, where all the inequalities can be chosen to be strict and π_n can be made equal to a greatest element, *i.e.* an optimal control policy. Owing to the fact that, given a policy π, one can compute a *neighborhood* of policies $\mathcal{V}_{MDP}(\pi)$ in which one is guaranteed to find a strictly larger policy (except if π is a greatest element), then starting from one policy on the MDP, one can theoretically performer a *gradient ascent*. If the set $\mathcal{V}_{MDP}(\pi)$ is small enough, then a *gradient ascent*-like algorithm could be tractable. While one could always pick $\mathcal{V}_{MDP}(\pi)$ to be the space of all deterministic stationary policy and converge in one step, this would make any *gradient ascent*-like algorithm useless. There is therefore a trade-off between the cardinal of $\mathcal{V}_{MDP}(\pi)$ we consider at

each time step and the number of steps until convergence to a greatest element, *i.e.* an optimal control policy.

Action iteration

In Chapter 3, we described a very simple Algorithm 1 that we called action-iteration. This algorithm simply iterates through the actions of a Bandit control problem, keeping tab of the best action encountered so far. We highlighted that an assumption for the algorithm to converge is the optimal feasibility. For the algorithm to compute an optimal action, it must be able to select such an action. A direct naive generalization of that algorithm to the case of MDP control would be to iterate through the whole set of policies, iteratively computing the gains and keeping tab of the best policy and gain encountered so far. While trivial, this algorithm would be a valid description of a **policy iteration** algorithm. However, it would, in some sense not be very efficient. It should be noted that what is usually called policy iteration is guaranteed to converge to an optimal policy, but there is no guarantee on the number of iterations until convergence is reached. *A priori*, the number of iterations could be exponential in the number of states, see [77]. Empirically, the problems studied by the practitioners and the RL community in general seem to be far more structured than the cases constructed to make policy iteration run for such a large number of iterations. The existence of such MDPs that make the worst case complexity horrendous should not prevent practitioner and theoretician alike to create and study algorithms that works well on the average MDP, *i.e.* with empirically polynomial complexity in the number of sates.

[77]: Hollanders et al. (2012), 'The complexity of Policy Iteration is exponential for discounted Markov Decision Processes'

Policy Iteration

A major result from the control theory of MDPs is the **policy improvement** theorem and its associated gradient ascent-like algorithm, the **policy iteration** algorithm. Indeed, for the average reward criterion, the partial ordering $\leq_{MDP}$ is defined using the reward and bias function, and partial order $\succcurlyeq_{\mathbb{R}^S}$ on $\mathbb{R}^S$, where S is the cardinal of the sate space. Under the discounted reward criterion, the partial ordering is defined using the reward and value functions, and partial order $\succcurlyeq_{\mathbb{R}^S}$ on $\mathbb{R}^S$. The policy improvement theorem state that for all suboptimal control policy π, one can search for a strict improvement of that policy in a neighborhood $\mathcal{V}_{MDP}(\pi)$ of $S \times A$ policies, a number that is far smaller than the size of policy space, A^S. This neighborhood consists in all the policies that are a modification of π in one state,

$$\mathcal{V}_{MDP}(\pi) = \{\pi' \in \Pi_{DS}(\mathbf{M}) \mid \exists! s \in \mathcal{S}, \pi'(s) \neq \pi(s)\}$$

where $\Pi_{DS}(\mathbf{M})$ is the set of deterministic stationary policies on $\mathbf{M}$ and $\mathcal{S}$ its state space. From the point of view of this thesis, it means that **there is enough information** in this set, to iteratively compute an optimal control policy. A natural question is therefore to ask whether this set can be exploited in the Reinforcement Learning setting, where the dynamics and rewards are unknown. Is it possible, using this set and partially ordered structure of the policies to craft a **stochastic gradient ascent**-like

algorithm? In the Markov control model, the answer is positive. It is interesting to first remark that the set $\mathcal{V}_{MDP}(\pi)$ can be expressed in terms of $\mathcal{X}_{\mathbf{M}}$, which emphasizes the locality of the changes on the considered policy. The policy iteration algorithm relies on the computation of a function called the bias function (known as the value function in the discounted reward setting) which is a function $\mathbf{b}_\pi : \mathcal{S} \to \mathbb{R}$, that can be seen as a vector in $\mathbb{R}^{|\mathcal{S}|}$. The partial order on the space of policies, $\leq_{MDP}$, that we mentioned above is defined using the canonical one $\geqslant$ on $\mathbb{R}^{|\mathcal{S}|}$ and bias function. In Reinforcement Learning, we will be interested in finding an optimal learning policy, *i.e.* one that converges to an optimal control policy. The convergence will be characterized thanks to the notion of regret, similarly to what we presented in the Bandit chapter.

6.3 Reinforcement Learning

Which policy should be played to gain enough information and not suffer too much regret? In the Chapter 7, we explore a setting, ergodic MDPs, such that this set is "accessible enough" for the learner to perform an optimal stochastic gradient ascent on the policy space. We will see that thanks to the ergodic property, whatever the policy played, all states in which a policy improvement is possible are regularly accessed *automatically thanks to the hypothesis*. This allows to perform a stochastic gradient ascent in the policy space using this very small subset of policies defined by local changes. In the Chapter 8, we explore another setting when a learner cannot *a priori* access some part of the state space by playing any policy and must actively explore some region to check for possible policy improvement in unvisited states.

Average-reward reinforcement learning

To summarize the last section in one paragraph, we consider a finite MDP $\mathbf{M} = (\mathcal{S}, \mathcal{A}, \mathbf{p}, \mathbf{r})$ where $\mathcal{S}$ is the finite set of states, $\mathcal{A} = (\mathcal{A}_s)_{s\in\mathcal{S}}$ specifies the set of actions available in each state, and we introduce the set of pairs $\mathcal{X}_{\mathbf{M}} = \{(s, a) : s \in \mathcal{S}, a \in \mathcal{A}_s\}$ for convenience. Further, $\mathbf{p} : \mathcal{X}_{\mathbf{M}} \to \mathcal{P}(\mathcal{S})$ is the transition distribution function and $\mathbf{r} : \mathcal{X}_{\mathbf{M}} \to \mathcal{P}(\mathbb{R})$ the reward distribution function, with corresponding mean reward function denoted by $\mathbf{m} : \mathcal{X}_{\mathbf{M}} \to \mathbb{R}$. An agent interacts with the MDP at discrete time steps $t \in \mathbb{N}^*$ and yields a random sequence $(s_t, a_t, r_t)_t$ of states, actions, and rewards in the following way. At each time step t, the agent observes the current state s_t and decides the action a_t to take based on s_t and possibly past information, *i.e.* previous elements of the sequence. After playing a_t, it observes a reward $r_t \sim \mathbf{r}(s_t, a_t)$, the current state of the MDP changes to $s_{t+1} \sim \mathbf{p}(\cdot|s_t, a_t)$ and the agent proceeds sequentially. In the *average-reward setting*, one is interested in maximizing the limit, $\frac{1}{T}\sum_{t=1}^{T} r_t$, when $T \to \infty$, providing it exists. This setting is a popular framework for studying sequential decision-making problems; it can be traced back to seminal papers such as those of [17] and [78] This theoretical framework allows to study the *exploration-exploitation* trade-off that arises from the sequential optimization problem a learner is trying to solve while being uncertain about the very problem it is optimizing.

Literature review

Early papers like [17, 79] mostly presented regret bounds for ergodic MDPs and with an asymptotic flavor. In Chapter 7 of this manuscript, inspired by the work of [79], we will craft a new algorithm, IMED-RL, that is asymptotically optimal in ergodic MDP but do enjoy a finite time regret bound whose first order term matches the asymptotic lower bound on logarithmic growth rate of regret. Because there is no current instance dependent lower bound on the logarithmic growth rate of regret for the general case of communicating MDP, most recent work focused on worst case regret bounds. The modern works presented in the non-exhaustive list of papers [80–86], reported non-asymptotic regret guarantees for the bigger class of communicating MDPs. The majority of recent literature on learning in MDPs, following [80], report worst-case regret bounds growing as $\mathcal{O}(\sqrt{T})$ after T steps. We note that this is also matching the worst-case regret bounds that exist in the Bandit literature. In contrast, comparatively there exists little work that present logarithmic and instance-dependent regret bounds for average-reward MDPs. The most notable exceptions include [80], which reports a logarithmic regret bound for UCRL2 (albeit with a large mixing-time related additive term), and more recent paper by [87], which only consider ergodic MDPs. In order to better understand the problem of MDPs that are communicating only, some papers such as [88, 89] derived logarithmic regret bounds derived for the simpler setting of MDPs with deterministic transitions. In Chapter 8, we also derive a lower bound for a simpler case of communicating MDPs with known transition kernel. We also derive an algorithm, IMED-KD, for which a logarithmic regret upper bound can be proved.

Some papers consider regret minimization in MDPs in the *episodic* setting, with a fixed and known horizon; *e.g.* [90–92], where the latter work presents a problem-dependent, logarithmic regret bound. However, the proof machinery used in episodic RL often fails to work in average-reward RL due to relying on the fixed episode length and resetting of the state.

[17]: Graves et al. (1997), 'Asymptotically efficient adaptive choice of control laws incontrolled markov chains'
[79]: Burnetas et al. (1997), 'Optimal adaptive policies for Markov decision processes'

[80]: Jaksch et al. (2010), 'Near-optimal regret bounds for reinforcement learning'
[81]: Filippi et al. (2010), 'Optimism in Reinforcement Learning and Kullback-Leibler Divergence'
[82]: Talebi et al. (2018), 'Variance-Aware Regret Bounds for Undiscounted Reinforcement Learning in MDPs'
[83]: Fruit et al. (2018), 'Near Optimal Exploration-Exploitation in Non-Communicating Markov Decision Processes'
[84]: Zhang et al. (2019), 'Regret minimization for reinforcement learning by evaluating the optimal bias function'
[85]: Wei et al. (2020), 'Model-free reinforcement learning in infinite-horizon average-reward Markov decision processes'
[86]: Bourel et al. (2020), 'Tightening exploration in upper confidence reinforcement learning'

[87]: Gopalan et al. (2015), 'Thompson sampling for learning parameterized markov decision processes'

[88]: Ortner (2009), 'Online Regret Bounds for Markov Decision Processes with Deterministic Transitions'
[89]: Tranos et al. (2021), 'Regret analysis in deterministic reinforcement learning'

[90]: Osband et al. (2013), '(More) efficient reinforcement learning via posterior sampling'
[91]: Azar et al. (2017), 'Minimax regret bounds for reinforcement learning'
[92]: Simchowitz et al. (2019), 'Non-asymptotic gap-dependent regret bounds for tabular mdps'

RL algorithms

Had the MDP only one state, it would be a bandit problem. Lower bound on the bandit regret and algorithms matching this lower bound, sometimes up to a constant factor, are well studied in the bandit literature. Therefore, bandit sampling strategies with known theoretical guarantees have inspired RL algorithms, even in the absence of lower bound for the RL setting. The KL-UCB algorithm [7], has inspired the strategy of the seminal paper of [78], as well the more recent KL-UCRL strategy [82, 93]. Inspired by the UCB algorithm [94, 95], a number of strategies implementing the optimism principle have emerged such as UCRL [96], UCRL2 [97] and UCRL3 [86] (and beyond, [91, 98] for the related episodic setup). The strategy PSRL [90] is inspired by Thompson sampling [99].

Among the modern RL algorithmic literature, [81] introduce KL-UCRL, which is a variant of UCRL that uses the KL divergence to define confidence bounds. Similarly to UCRL2, KL-UCRL achieves a regret of $\widetilde{\mathcal{O}}(DS\sqrt{AT})$ in communicating MDPs. Interestingly, a more refined regret

bound for KL-UCRL in ergodic MDPs is presented in [82]. [100] present REGAL and report a $\widetilde{\mathcal{O}}(D'S\sqrt{AT})$ regret with high probability in the larger class of weakly communicating MDPs, provided that the learner knows an upper bound D' on the span of the optimal bias function of the true MDP. [101] present SCAL, which similarly to REGAL works in weakly communicating MDPs, but admits an efficient implementation. A similar algorithm called SCAL$^+$ is presented in [102]. Both SCAL and SCAL$^+$ admit a regret bound scaling as $\widetilde{\mathcal{O}}\left(D\sqrt{\sum_{s,a} K_{s,a}T}\right)$. In a recent work, [84] present EBF achieving a regret of $\widetilde{\mathcal{O}}\left(\sqrt{HSAT}\right)$ assuming that the learner knows an upper bound H on the span of the optimal bias function of the true MDP. We remark that the universal constants of the leading term here are fairly large. However, EBF does not admit a computationally efficient implementation. Another related line of works considers posterior sampling methods such as [90] inspired by Thompson sampling [24]. For average-reward RL, existing works on these methods report Bayesian regret bounds, except [103], whose corrected regret bound, reported in [104], scales as $O(DS\sqrt{AT}\log^3(T))$ and is valid for $T \geq S^4A^3$.

[100]: Bartlett et al. (2009), 'REGAL: a regularization based algorithm for reinforcement learning in weakly communicating MDPs'

[101]: Fruit et al. (2018), 'Efficient Bias-Span-Constrained Exploration-Exploitation in Reinforcement Learning'

[102]: Qian et al. (2019), 'Exploration bonus for regret minimization in discrete and continuous average reward MDPs'

[24]: Thompson (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'

[103]: Agrawal et al. (2017), 'Optimistic posterior sampling for reinforcement learning: worst-case regret bounds'

[104]: Agrawal et al. (2017), 'Posterior sampling for reinforcement learning: worst-case regret bounds'

6.4 Summary of contributions

Learning in ergodic Markov decision processes

NeurIPS **2022**

In the paper ***IMED-RL: Regret optimal learning of ergodic Markov decision processes*** and published at NeurIPS 2022 with Odalric-Ambrym Maillard, We consider reinforcement learning in a discrete, undiscounted, infinite-horizon Markov Decision Problem (MDP) under the average reward criterion, and focus on the minimization of the regret with respect to an optimal policy, when the learner does not know the rewards nor the transitions of the MDP. In light of their success at regret minimization in multi-armed bandits, popular bandit strategies, such as the optimistic UCB, KL-UCB or the Bayesian Thompson sampling strategy, have been extended to the MDP setup. Despite some key successes, existing strategies for solving this problem either fail to be provably asymptotically optimal, or suffer from prohibitive burn-in phase and computational complexity when implemented in practice. In this work, we shed a novel light on regret minimization strategies, by extending to reinforcement learning the computationally appealing Indexed Minimum Empirical Divergence (IMED) bandit algorithm. Traditional asymptotic problem-dependent lower bounds on the regret are known under the assumption that the MDP is *ergodic*. Under this assumption, we introduce IMED-RL and prove that its regret upper bound asymptotically matches the regret lower bound. We discuss both the case when the supports of transitions are unknown, and the more informative but a priori harder-to-exploit-optimally case when they are known. Rewards are assumed light-tailed, semi-bounded from above. Last, we provide numerical illustrations on classical tabular MDPs, *ergodic* and *communicating* only, showing the competitiveness of IMED-RL in finite-time against state-of-the-art algorithms. IMED-RL also benefits from a light complexity.

Regret in communicating MDPs with known dynamics

ACML **2023**

In the paper ***Logarithmic regret in communicating MDPs: Leveraging known dynamics with bandits*** and published at ACML 2023 with Hassan Saber, Mohammad Sadegh Talebi and Odalric-Ambrym Maillard, we study regret minimization in an average-reward and communicating Markov Decision Process (MDP) with known dynamics, but unknown reward function. Although learning in such MDPs is a priori easier than in fully unknown ones, they are still largely challenging as they include as special cases large classes of problems such as combinatorial semi-bandits. Leveraging the knowledge on transition function in regret minimization, in a statistically efficient way, appears largely unexplored. As it is conjectured that achieving exact optimality in generic MDPs is NP-hard, even with known transitions, we focus on a computationally efficient relaxation, at the cost of achieving order-optimal logarithmic regret instead of exact optimality. We contribute to filling this gap by introducing a novel algorithm based on the popular Indexed Minimum Empirical Divergence strategy for bandits. A key component of the proposed algorithm is a carefully designed stopping criterion leveraging the recurrent classes induced by stationary policies. We derive a non-asymptotic, problem-dependent, and logarithmic regret bound for

this algorithm, which relies on a novel regret decomposition leveraging the structure. We further provide an efficient implementation and experiments illustrating its promising empirical performance.

IMED RL 7

In this chapter, based on the paper ***IMED-RL: Regret optimal learning of ergodic Markov decision processes*** published at NeurIPS 2022 with Odalric-Ambrym Maillard, we study regret minimization of the average-reward criterion in an ergodic MDP with unknown dynamics and reward functions.

In the considered setting, the learning agent interacts with the MDP without any reset. The minimal assumption would be to allow the agent to come back with positive probability from any initial mistake in finite time, so that the agent is not stuck in a suboptimal area of the system. This is assuming that the MDP is *communicating*, that is $\forall s, s', \exists \pi, t \in \mathbb{N} : \mathbf{p}_\pi^t(s'|s) > 0$. However, in the literature, lower bounds on the regret are stated for MDPs satisfying a stronger assumption, *ergodicity*. Since one is interested in crafting an algorithm matching a lower bound, we consider this stronger assumption.

Assumption 7.0.1 (Ergodic MDP) *The MDP* $\boldsymbol{M}$ *is ergodic, that is* $\forall s, s', \forall \pi, \exists t \in N_a(T) : \boldsymbol{p}_\pi^t(s'|s) > 0$.

Intuitively, this means that for all policies and all couples of states, there exists a finite trajectory of positive probability between the states. Interestingly, the ergodic property can be assumed on the MDP or on the set of policies in which we seek an optimal one. For instance, in any communicating MDP all ϵ-soft policies[1] are ergodic; more in the Experiment section 7.7.

1: A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A}_s)$ is ϵ-soft if $\pi(a|s) \geq \epsilon/|\mathcal{A}_s|$ for all s and a.

We build on the IMED strategy ([74]), a bandit algorithm that benefits from practical and optimal guarantees but has never been used by the RL community. We fill this gap by proposing the IMED-RL algorithm which we prove to be asymptotically optimal for the average-reward criterion. We revisit the notion of skeleton (Equation 7.10) introduced in the seminal work of [78], with a subtle but key modification that prevents a prohibitive burn-in phase, see hereafter for a detailed explanation. Further, this novel notion of skeleton enables IMED-RL to remove any tracking or hyperparameter and mimic a *stochastic-policy-iteration-like* algorithm. [2] Further, this skeleton scales naturally with the studied MDP as it does not explicitly refer to absolute quantities such as the time. We prove that our proposed IMED-RL is asymptotically optimal and show its numerical competitivity.

2: The skeleton in [78] is sometimes empty at some states, when t is too small, this causes the strategy to work well only after t is large enough to ensure that the skeleton contains at least one action in each state.

We would like to note that the original definition of ergodic MDP is that of a MDP that is both recurrent and aperiodic while our definition is of ergodicity corresponds to the recurrent condition only. The reason we define ergodicity this way is that this definition is somehow common in some modern work about average-reward reinforcement learning while the original one is still more prevalent in the theoretical oriented work. Because we wanted our original paper to also reach the application minded community, we decided to define ergodicity as in our IMED-RL paper. We could also have followed the step of [105] and use the,

[105]: Burnetas et al. (1997), 'Optimal adaptive policies for Markov decision processes'

perhaps more appropriate, term of *irreducible*. Indeed, the condition we essentially want is that for all policy, the associated Markov chain does not contain any proper closed subset other than the state space. The periodic condition can essentially be dealt with by using Cesàro average of transition kernel's powers instead of simple transition kernel's powers. In this manuscript, in order to keep the terminological coherence with the paper this chapter is based upon and use the term *ergodic MDP* as defined in Assumption 7.0.1. For future work, I think that I will try to be more mindful and less prone to follow community trend, thus using *irreducible* where it is meant and using *ergodic* with its original definition. For the sake of completeness, we state that an MDP $\mathbf{M}$ is ergodic if $\exists t \in \mathbb{N} \forall s, s', \forall \pi, : \mathbf{p}_\pi^t(s'|s) > 0$ (please note the $\exists$ and $\forall$ quantifier inversion compared to the previous definition).

Ergodic assumption

While many recent works focused on worst-case regret bounds only (e.g. [106–108] and citations therein), studying problem-dependent optimal regret bounds has been somewhat overlooked. Being more general is always more appealing but the restriction from communicating MDPs to ergodic MDPs allows us to target exact asymptotic optimality ; not just bound, not just worst-case bound. Ergodic MDPs is the only case in which explicit problem-dependent lower bounds are known and hence can be directly used to build a strategy. Indeed, the main challenge towards problem-dependent optimality is that existing lower bounds for exploration problems in MDPs are usually written in terms of non-convex optimization problems. This *implicit* form makes it hard to understand the actual complexity of the setting and, thus, to design optimal algorithms. Existing proof strategies for state-of-the-art algorithms (UCRL, PSRL, *etc.*) ensure a regret for communicating MDPs but fail to provide optimality guarantees even in the ergodic case. We believe that deriving a sharp result in the ergodic case might prove to be insightful to pave the way towards the communicating case. From a theoretical standpoint, related to UCRL type strategy, modern analysis of KL-UCRL by [82] also makes the ergodic assumption. This hypothesis has also been used in the theoretical work of [109] and the work of [110] that concerns structured MDPs. Related to this assumption are works that are interested in identification and sample complexity. [111] introduced a primal-dual method to compute an ϵ-optimal policy and bound the number of sample transitions to reach this goal. [112] relaxed the ergodic hypothesis by using a mixing hypothesis that implies the uniqueness of recurrent class for each policy. In this setting, the authors also derive a bound on the number of samples to compute an ϵ-optimal policy.

In this chapter, one is interested in developing a sampling strategy that is *optimal* amongst strategies that aim at maximizing the average-reward, *i.e.* balancing exploration and exploitation in an optimal way. To assert optimality, we state a *regret lower bound* with the purpose of defining a theoretically sound notion of optimality that is *problem-dependent*. While *regret* defines the discrepancy to optimality of a learning strategy, a *problem-dependent regret lower bound* will formally assess the minimal regret that any uniformly fast learning algorithm must incur on a given MDP problem by computing a minimal rate of exploration. Because

this minimal rate of exploration depends on the problem, it is said to be problem-dependent, as opposed to worst case regret study that can exist in the MDP literature (*e.g.* [97]). Regret lower bounds currently exist in the literature when the MDP $\mathbf{M}$ is assumed to be *ergodic*. Hence, we hereafter make this assumption, in order to be able to compare the regret of our algorithm to an optimal bound. Similarly, to ensure fast enough convergence of the empirical estimate of the reward to the true mean, an assumption controlling the rate of convergence to the mean is necessary.

Assumption 7.0.2 (Light-tail rewards) *For all $x \in \mathcal{X}_{\boldsymbol{M}}$, the moment generating function of the reward exists in a neighborhood of 0: $\exists \lambda_x > 0, \forall \lambda \in \mathbb{R}$ such that $|\lambda| < \lambda_x, \mathbb{E}_{R\sim \boldsymbol{r}(x)}[\exp(\lambda R)] < \infty$.*

Building on IMED, we make an additional assumption on the reward that is less restrictive than the common bounded reward hypothesis made in the RL community.

Assumption 7.0.3 (Semi-bounded rewards) *For all $x \in \mathcal{X}_{\boldsymbol{M}}$, the support of the reward distribution $r(x)$ is bounded from above by a known constant. There exists a known quantity $m_{\max}(x) \in \mathbb{R}$ such that for all $x \in \mathcal{X}$, the support $\mathsf{Supp}(\boldsymbol{r}(x))$ of the reward distribution is semi-bounded from above, $\mathsf{Supp}(\boldsymbol{r}(x)) \subset]-\infty, m_{max}(x)]$, and its mean satisfies $\boldsymbol{m}(x) < m_{\max}(x)$.*

Of course, one can also assume that the reward distributions have bounded support.

The rest of this chapter is organized as follows. We introduced the known lower bound for ergodic MDPs, a lower bound on the logarithmic regret growth rate of regret that is highly reminiscent of the one that exist in the Bandit literature. Afterward, we introduce the algorithm IMED-RL, IMED for Reinforcement Learning, that we presented in the paper [73], and state its regret upper bound. We end this chapter by a series of numerical experiments showcasing the impressive performances of IMED-RL. Furthermore, we test IMED-RL in non-ergodic environment and find out that this algorithm is still competitive. While we cannot be sure because there are no known lower bound in the communicating case, see Chapter 8, we may suppose that there is room for improving and slightly modify the IMED-RL algorithm so that it benefits theoretical guarantees in the communication-only setting. In the future, it would be a good research project to mix the ideas presented in this chapter and the ones presented in the next Chapter 8 based on a very recently written paper. Hopefully, this manuscript will help connect the dots.

[73]: Pesquerel et al. (2022), 'IMED-RL: Regret optimal learning of ergodic Markov decision processes'

7.1 Regret lower bound

In this section, we recall the regret lower bound for ergodic MDPs and provide a few insights about it.

Characterizing optimal policies Relying on classical results that can be found in the books of [75] and [76], we give a useful characterization of

optimal policies that is used to derive a lower bound under the ergodic assumption. Under the ergodic Assumption 7.0.1 of MDP $\mathbf{M}$, for all policy $\pi \in \Pi(\mathbf{M})$, the gain is independent of the initial state, *i.e.* $\mathbf{g}_\pi(s) = \mathbf{g}_\pi(s')$ for all states s and s', and we denote it $\mathbf{g}_\pi$. Similarly, the set of optimal policies $\mathcal{O}(\mathbf{M})$ is state-independent since $\mathcal{O}_s(\mathbf{M}) = \mathcal{O}_{s'}(\mathbf{M})$. Any policy π satisfy the following fixed point property

$$\textit{(Poisson equation)} \qquad \mathbf{g}_\pi + \mathbf{b}_\pi(s) = \mathbf{m}_\pi(s) + (\mathbf{p}_\pi \mathbf{b}_\pi)(s)\,, \tag{7.1}$$

where $\mathbf{b}_\pi : \mathcal{S} \to \mathbb{R}$ is called the bias function and is defined up to an additive constant. For aperiodic MDP, it can be expressed as $\mathbf{b}_\pi(s) = \left(\sum_{t=1}^{\infty} (\mathbf{p}_\pi^{t-1} - \overline{\mathbf{p}}_\pi)\mathbf{m}_\pi\right)(s)$. For irreducible only MDP, a similar formula can be used with Cesàro averages. However, to algorithmically compute a bias, we rely in practice on the value iteration algorithm or, sometimes, on linear programming. We highlight that bias plays a role similar to the value function in the discounted reward setting in which the gain is always zero and Equation 7.1 reduces to the Bellman equation, giving a direction in which extend our results to this other RL setting.

Interestingly, for any communicating and a fortiori ergodic MDP, the span $\mathbb{S}(\mathbf{b}_\pi) = \max_{s\in\mathcal{S}} \mathbf{b}_\pi(s) - \min_{s\in\mathcal{S}} \mathbf{b}_\pi(s)$ of the bias function of any policy is bounded, which allows to decompose the regret in the useful following way.

Lemma 7.1.1 (Regret decomposition) *Under the ergodic assumption 7.0.1, for all optimal policy $\star \in \mathcal{O}(\boldsymbol{M})$, the regret of any policy $\pi = (\pi_t)_t$ can be decomposed as*

$$\mathscr{R}_{\pi,s_1}(\boldsymbol{M}, T; \star) = \sum_{x\in\mathfrak{X}_{\boldsymbol{M}}} \mathbb{E}_{\pi,s_1}\left[N_x(T)\right]\Delta_x(\boldsymbol{M}) + \underbrace{\left(\left[\prod_{t=1}^{T} \boldsymbol{p}_{\pi_t} - \boldsymbol{p}_\star^t\right] b_\star\right)(s_1)}_{\leq \mathbb{S}(\boldsymbol{b}_\star)}, \tag{7.2}$$

where $N_{s,a}(T) = \sum_{t=1}^{T} \mathbb{1}\{s_t = s, a_t = a\}$ counts the number of time the state-action pair (s,a) has been sampled and $\Delta_{s,a}(\boldsymbol{M})$ is the suboptimality gap of the state-action pair (s,a) in $\boldsymbol{M}$,

$$\begin{aligned}\Delta_{s,a}(\boldsymbol{M}) &= \boldsymbol{m}(s,a) + \boldsymbol{p}_a \boldsymbol{b}_\star(s) - \boldsymbol{m}_\star(s) - \boldsymbol{p}_\star \boldsymbol{b}_\star(s) \\ &= \boldsymbol{m}(s,a) + \boldsymbol{p}_a \boldsymbol{b}_\star(s) - \boldsymbol{g}_\star - \boldsymbol{b}_\star(s)\end{aligned} \tag{7.3}$$

with $\boldsymbol{p}_a = \boldsymbol{p}(\cdot|s,a)$ by a slight abuse of notation. Action $a \in \mathcal{A}_s$ is optimal if and only if $\Delta_{s,a}(\boldsymbol{M}) = 0$, otherwise, it is said suboptimal.

This result can be found in [75].

Under the ergodic Assumption 7.0.1 of MDP $\mathbf{M}$, all optimal policies satisfy a Poisson equation while some are also being characterized by the optimal Poisson equation (see [76]), used to compute the optimal gain and a bias function associated to an optimal policy,

$$\mathbf{g}^{\mathbf{M}} + \mathbf{b}^{\mathbf{M}}(s) = \max_{a\in\mathcal{A}_s} \left\{ \mathbf{m}(s,a) + \sum_{s'\in\mathcal{S}} \mathbf{p}(s'|s,a)\mathbf{b}^{\mathbf{M}}(s') \right\}. \tag{7.4}$$

Lower bound To assess the minimal sampling complexity of a suboptimal state action pair, one must compute how far a suboptimal state-action pair is from being optimal from an information point-of-view. A suboptimal state-action pair $(s,a) \in \mathcal{X}_{\mathbf{M}}$ is said to be *critical* if it can be made optimal by changing reward $\mathbf{r}(s,a)$ and transition $\mathbf{p}(\cdot|s,a)$ while respecting the assumptions on the rewards and transitions. Formally, let $\phi_{\mathbf{M}} : \mathcal{P}(\mathbb{R} \times \mathcal{S}) \to \mathbb{R}$,

$$\phi_{\mathbf{M}}(\nu \otimes q) = \mathbb{E}_{R\sim\nu}[R] + q\mathbf{b}^{\mathbf{M}} \tag{7.5}$$

denotes the potential function of $\nu \otimes q$ in $\mathbf{M}$, where $\nu \otimes q$ is a notation highlighting the reward ν and transition q marginal distributions of the reward-transition distribution. A pair $(s,a) \in \mathcal{X}_{\mathbf{M}}$ is *critical* if it is suboptimal and there exists $\nu \in \mathcal{F}_{s,a}$ and $q \in \mathcal{P}(\mathcal{S})$ such that

$$\phi_{\mathbf{M}}(\nu \otimes q) > \gamma_s(\mathbf{M}) \quad \text{where } \gamma_s(\mathbf{M}) \overset{def}{=} \mathbf{g}^{\mathbf{M}} + \mathbf{b}^{\mathbf{M}}(s). \tag{7.6}$$

Note that $\gamma_s(\mathbf{M}) = \max_{a\in\mathcal{A}_s} \phi_{\mathbf{M}}(\mathbf{r}(s,a) \otimes \mathbf{p}(s,a))$ by the optimal Poisson Equation 7.4.

Definition 7.1.1 (Sub-optimality cost) *The* ***suboptimality cost*** *of a suboptimal state-action pair $(s,a) \in \mathcal{X}_{\boldsymbol{M}}$ is defined as*

$$\mathcal{E}_{\mathcal{F}}^{s,a}(\boldsymbol{M}) \overset{def}{=} \mathcal{E}_{\mathcal{F}}^{s,a}(\boldsymbol{M}, \gamma_s(\boldsymbol{M}))$$

where

$$\mathcal{E}_{\mathcal{F}}^{s,a}(\boldsymbol{M},\gamma) = \inf_{\substack{\nu\in\mathcal{F}_{s,a}\\ q\in\mathcal{P}(\mathcal{S})}} \left\{ \mathrm{KL}\left(\boldsymbol{r}(s,a)\otimes\boldsymbol{p}(\cdot|s,a), \nu\otimes q\right) \,:\, \phi_{\boldsymbol{M}}(\nu\otimes q) > \gamma \right\}, \tag{7.7}$$

and KL *denotes the Kullback-Leibler divergence between distributions.*

A lower bound on the regret may now be stated for a certain class of learner, the set of uniformly consistent learning algorithm, Interestingly, it will be those policies $\pi = (\pi_t)_t$ such that $\mathbb{E}_{\pi,\mathbf{M}}(N_{s,a}(T)) = o(T^\alpha)$ for all suboptimal state-action pair (s,a) and $0 < \alpha < 1$ (see [19]).

Theorem 7.1.2 (Regret lower bound [78]) *Let $\boldsymbol{M} = (\mathcal{S}, \mathcal{A}, \boldsymbol{p}, \boldsymbol{r})$ be an MDP satisfying Assumptions 7.0.2, 7.0.1, 7.0.3. For all uniformly consistent learning algorithm π,*

$$\liminf_{T\to\infty} \frac{\mathbb{E}_{\pi,\boldsymbol{M}}[N_{s,a}(T)]}{\log T} \geq \frac{1}{\mathcal{E}_{\mathcal{F}}^{s,a}(\boldsymbol{M})} \tag{7.8}$$

with the convention that $1/\infty = 0$. The regret lower bound is

$$\liminf_{T\to\infty} \frac{\mathcal{R}_\pi(\boldsymbol{M},T)}{\log T} \geq \sum_{(s,a)\in\mathcal{C}(\boldsymbol{M})} \frac{\Delta_{s,a}(\boldsymbol{M})}{\mathcal{E}_{\mathcal{F}}^{s,a}(\boldsymbol{M})} \tag{7.9}$$

where $\mathcal{C}(\boldsymbol{M}) = \left\{(s,a) : 0 < \mathcal{E}_{\mathcal{F}}^{s,a}(\boldsymbol{M}) < \infty\right\}$ is called the set of critical state-action pairs. Those are the state-action pairs (s,a) that could be confused for an optimal one if we were to change their associated rewards and transitions distributions at the displacement cost of $\mathcal{E}_{\mathcal{F}}^{s,a}(\boldsymbol{M})$.

We note how the ergodic hypothesis allows to isolate the logarithmic sampling rate of each suboptimal state-action pairs. Given the fact that, under an ergodic policy, all states will be visited a linear amount of time with high probability, it really hints us into using a Bandit strategy at each state of the MDP.

7.2 From Bandit to Reinforcement Learning

In this section, we show how Reinforcement Learning with ergodic MDPs can be understood from a Bandit perspective and how it can serve as a bridge between Bandits and the general RL setting with communicating only MDPs. We then loop back to Bandit by explaining at a high level, how Bandit algorithms can be used in the studied setting.

7.3 The IMED-RL Algorithm

In this section, we introduce the IMED-RL algorithm, whose regret matches this fundamental lower bound and extends the IMED strategy from [74] to ergodic MDPs. We talk about *extending* IMED and not *using* IMED because when the considered MDP has only one state, thus ergodic and akin to a Bandit problem, IMED-RL reduces to the IMED algorithm. This is the second time in this manuscript that we safeguard our idea by checking that in absence of structure, we retrieve a known optimal solution, the first time being in Chapter 5 where we studied Bandit with groups of similar arms. In future work, it would be nice to *extend* IMED-RL to the case of communicating only MDPs. This way, we would have a continuum of IMED-related algorithms, starting from the optimal IMED algorithm for Bandit, then with the optimal IMED-RL for ergodic MDPs, and continuing with the yet to be discovered algorithm for communicating only MDPs. We could then easily enough transfer the work we did in Chapter 4, where we studied approximation of the $\mathcal{E}_{\mathcal{F}}$, to those MDP settings. Similarly, one could possibly adapt the IMED-EC algorithm from Chapter 5 to the case of ergodic MDPs with groups of similar state-action pairs by transforming the IMED-RL algorithm.

In light of their success at regret minimization in multi-armed bandits, popular bandit strategies, such as the optimistic UCB, KL-UCB or the Bayesian Thompson sampling strategy, have been extended to the MDP setup. Despite some key successes, existing strategies for solving this problem either fail to be provably asymptotically optimal, or suffer from prohibitive burn-in phase and computational complexity when implemented in practice. IMED-RL shed a novel light on regret minimization strategies.

IMED-RL index

Empirical MDP

IMED-RL is a *model-based* algorithm that keeps empirical estimates of the transitions $\mathbf{p}$ and rewards $\mathbf{r}$ as opposed to *model-free* algorithm such

as Q-learning. We denote by $\hat{\mathbf{r}}_t(s,a) = \hat{\mathbf{r}}(s,a;N_{s,a}(t))$ and $\hat{\mathbf{p}}_t(s,a) = \hat{\mathbf{p}}(s,a;N_{s,a}(t))$ the empirical reward distributions and transition vectors after t interactions, *i.e.* using $N_{s,a}(t)$ samples from the distribution $\mathbf{r}(s,a)$. Initially, $\hat{\mathbf{p}}(s,a;0)$ is the uniform probability over the state space and $\hat{\mathbf{p}}(s,a;k) = (1-1/k)\hat{\mathbf{p}}(s,a;k-1) + (1/k)\mathbf{s}_k$, where $\mathbf{s}_k$ is a vector of zeros except for a one at index s_k, the k^{th} samples drawn from $\mathbf{p}(\cdot|s,a)$. This defines at each time step t an empirical MDP $\widehat{\mathbf{M}}_t = \left(\mathcal{S}, \mathcal{A}, \hat{\mathbf{p}}_t, \hat{\mathbf{r}}_t\right)$.

Skeleton

On this empirical MDP, for each state, some actions have been sampled more than others and their empirical quantities are therefore better estimated. We call *skeleton* at time t the subset of state-action pairs that can be considered sampled enough at time t. It is defined by restricting $\mathcal{A}_s$ to $\mathcal{A}_s(t)$ for all state $s \in \mathcal{S}$, with

$$\mathcal{A}_s(t) = \left\{ a \in \mathcal{A}_s \;:\; N_{s,a}(t) \geq \log^2\left(\max_{a' \in \mathcal{A}_s} N_{sa'}(t)\right)\right\}. \tag{7.10}$$

Since $x > \log^2 x$, $\mathcal{A}_s(t) \neq \emptyset$, hence $\mathcal{A}(t) = (\mathcal{A}_s(t))_s$ contains at least one deterministic policy.

We note that the MDP $\mathbf{M}(\mathcal{A}(t)) \overset{def}{=} (\mathcal{S}, \mathcal{A}(t), \mathbf{p}, \mathbf{r})$ defined by restricting the set of actions to $\mathcal{A}(t) \subseteq \mathcal{A}$ is an ergodic MDP. The restricted empirical MDP $\widehat{\mathbf{M}}_t(\mathcal{A}(t)) \overset{def}{=} (\mathcal{S}, \mathcal{A}(t), \hat{\mathbf{p}}_t, \hat{\mathbf{r}}_t)$ also is ergodic thanks to the ergodic initialization of the estimate $\hat{\mathbf{p}}$. This skeleton is carefully crafted to satisfy property allowing the IMED-RL algorithm to make the most of available information, *i.e.* to maximize its progress per unit of interaction. After introducing the IMED-RL index and IMED-RL algorithm, so that the reader has a complete view of IMED-RL, we come back to this very important piece of the IMED-RL scheme.

IMED-RL index

Inspired by IMED and the logarithmic growth rate of state-action pairs sample for ergodic MDPs, we define the IMED-RL index.

Definition 7.3.1 (IMED-RL index) *For all state-action pairs $(s,a) \in \mathcal{X}_{\boldsymbol{M}}$, let us define $\mathbf{K}_{s,a}(t) \overset{def}{=} \mathcal{C}_{\mathcal{F}}^{s,a}\left(\widehat{\boldsymbol{M}}_t(\mathcal{A}(t)), \hat{\gamma}_s(t)\right)$ with empirical threshold $\hat{\gamma}_s(t) \overset{def}{=} \max_{a \in \mathcal{A}_s} \phi_{\widehat{\boldsymbol{M}}_t(\mathcal{A}(t))}(\hat{\boldsymbol{r}}(s,a) \otimes \hat{\boldsymbol{p}}(s,a))$ Then, the IMED-RL index of (s,a) at time t, $\mathbf{H}_{s,a}(t)$, is defined as*

$$\mathbf{H}_{s,a}(t) \quad = \quad N_{s,a}(t)\mathbf{K}_{s,a}(t) + \log N_{s,a}(t)\,. \tag{7.11}$$

Note that $\hat{\gamma}_s(t) \neq \gamma_s(\hat{\mathbf{M}}_t(\mathcal{A}(t)))$ as the maximum is taken over all $a \in \mathcal{A}_s$ and not just $a \in \mathcal{A}_s(t)$. Similarly to the IMED index we already discussed at length in the previous chapters, the IMED-RL index relates the unlikelihood of optimality of a state action pair to its sample frequency. The main difference with IMED is that the quantity $\mathcal{C}_{\mathcal{F}}^{s,a}\left(\widehat{\mathbf{M}}_t(\mathcal{A}(t)), \hat{\gamma}_s(t)\right)$ cannot exactly be described as an **unlikelihood of optimality**. Rather, it

is better described as an **unlikelihood of policy-improvement**. On an MDP $\mathbf{M}$, we consider a subset $\mathscr{A}(t)$ of action and restricted MDP $\mathbf{M}(\mathscr{A}(t))$ (at least one action per state). On this restricted MDP, we compute an optimal policy. Then, $\mathscr{C}_{\mathcal{F}}^{s,a}(\mathbf{M}(\mathscr{A}(t)), \gamma_s(\mathbf{M}))$ is exactly the unlikelihood of policy-improvement of state-action pair (s, a) for the best policy in the restricted MDP when considering all available actions in the unrestricted MDP. With all those intuitions we should partially insist again on, the IMED-RL algorithm comes quite naturally.

Remark 7.3.1 (Known support of transitions) Were the support of transition known, the infimum in suboptimality cost $\mathscr{C}_{\mathcal{F}}^{s,a}$ defined by Equation 7.7 would be redefined as one over the set

$$\{q \in \mathcal{P}(\mathcal{S}) : \mathtt{Supp}(q) = \mathtt{Supp}(\mathbf{p}(\cdot|s,a))\},$$

modifying both the lower bound and IMED-RL index.

IMED-RL algorithm

The IMED-RL algorithm consists in playing at each interaction t, an action a_t of minimal IMED-RL index at the current state s_t. The intuition behind the IMED-RL index is similar to the one of the IMED index for bandits and stems from an information theoretic point-of-view of the lower bound. It uses the intuition we built about the notion of unlikelihood of optimality and relates it to the different terms present in the lower bound of the logarithmic growth rate of the regret. More specifically, the form of the lower bound decompose as a sum on some state-action pairs that is reminiscent of the Bandit lower bound. The main challenge of RL compared to Bandit is that one must find a way to quickly converge to the *right problem*, that is to say, one must find a way top converge to the sub-MDP $\mathbf{M}(\pi^\star)$ so that the considered unlikelihood of policy improvement IMED-RL consider correspond to the unlikelihood of policy improvement that appear in the lower bound.

The intuition is mathematically that at a given time t, the **frequency of play** $\frac{N_{s,a}(t)}{N_s(t)}$ of action $a \in \mathcal{A}_s$ in state $s \in \mathcal{S}$, should be larger than or equal to its posterior **probability of being a policy-improving action** in that state, for the considered empirically optimal policy in the correctly-enough-estimated restricted MDP, $\mathbf{M}(\mathscr{A}(t))$ defined thanks to the skeleton, $\exp(-N_{s,a}(t)\mathbf{K}_{s,a}(t))$, that is to say $\frac{N_{s,a}(t)}{N_s(t)} \geq \exp(-N_{s,a}(t)\mathbf{K}_{s,a}(t))$. Taking the logarithm and rearranging the terms, this condition rewrites $\mathbf{H}_{s,a}(t) \geq \log N_s(t)$ at each time step t. The action that is the closest to violate this condition or that violates this condition the most is the one of minimal IMED-RL index, $\arg\min_a \mathbf{H}_{s,a}(t)$, the one IMED-RL decides to play.

Thanks to the ergodic assumption, all states will be such that $N_s(t) \propto t$ with high probability. Thus, if a policy improving action exist, thanks to the optimality of IMED, it should not take too long before that action is sampled a lot, that is to say, a number of time that is linear with respect to the number of new visits since the restricted MDP changed. Other actions will be sampled a logarithmic number of the new visit but could still have a number of samples that is much larger than $\log t$ due to the existence of

previous restricted problems, prior to the one the agent is solving, where those actions might have been policy-improving. What can be stated for sure is that if an action that was not in the super-logarithmic regime, $N_{s,a}(t) \lessapprox \log N_s(t)$, suddenly makes it to this super-logarithmic regime, $N_{s,a}(t) \geq (\log N_s(t))^{1+\epsilon}$, for $\epsilon > 0$ arbitrarily small, then it means that in the current restricted MDP, it is likely to be a policy-improving action. Such an action switch from a logarithmic or sub-logarithmic sampling regime to a super-logarithmic one are those actions that, in the current problem defined by the current skeleton are likely to be policy-improving thanks to the optimality of IMED. Therefore, as soon as those actions are detected as entering the super-logarithmic sampling regime, they should be added to the skeleton in order to create a new problem from which one can make a new policy improvement step, if possible.

Algorithm 21: IMED-RL

Input: A Markov Decision Process $\mathbf{M} = (\mathcal{S}, \mathcal{A}, \mathbb{r}, \mathbf{p})$;
IMED-RL index $\mathbf{H}_{s,a} : H \to \mathbb{R}$;
Observe initial state s_0;
Initialize history H as $H = \{s_0\}$;

1 **for** $t \in \mathbb{N}_+$ **do**
2 **forall** $a \in \mathcal{A}_{s_t}$ **do**
3 Compute index $I_a = \mathbf{H}_{s,a}(H)$;
4 **Compute** $a_t \in \operatorname{argmin}_{e \in \mathcal{A}_{s_t}} I_e$;
5 **Receive** a sampled state-reward $s_t, r_t \sim \mathbf{p} \otimes \mathbf{r}(s_t, a_t)$;
6 Update history, $H \leftarrow H \cup \{(a_t, s_t, r_t)\}$;

Control In control theory, we assume that both the expected rewards and transitions probabilities of an MDP $\mathbf{M}$ are known. Policy iteration (see [75], [113]) is an algorithm that computes a sequence $(\pi_n)_n$ of deterministic policies that are increasingly strictly better until an optimal policy is reached. In the average-reward setting and under the ergodic assumption, a policy π is strictly better than another policy π' if $g_\pi(\mathbf{M}) > g_{\pi'}(\mathbf{M})$. The policy iteration algorithm computes the sequence of policies recursively in the following way. Initially, an arbitrary deterministic policy π_0 is chosen. At step $n+1 \in \mathbb{N}^*$, it computes $\mathbf{m}_{\pi_n}$ and $\mathbf{b}_{\pi_n}$ then swipes through the states $s \in \mathcal{S}$ in an arbitrary order until it reaches one state s such that there exists $a \in \mathcal{A}(s)$ with $\mathbf{m}(s,a) + \mathbf{p}(\cdot|s,a)\mathbf{b}_{\pi_n} > \mathbf{m}_{\pi_n}(s) + \mathbf{p}_\pi(s)\mathbf{b}_{\pi_n}$. If such an s does not exist, then it returns π_n as an optimal policy. Otherwise, π_{n+1} is defined as $\pi_{n+1}(s') = \pi_n(s')$ for all $s \neq s'$ and $\pi_{n+1}(s) \in \arg\max \{\mathbf{m}(s,a) + \mathbf{p}(\cdot|s,a)\mathbf{b}_{\pi_n}\}$. Such a step is called a policy improvement step. Policy iteration is guaranteed to finish in a finite number as the cardinal of $\Pi(\mathbf{M})$ is finite. At each step $n \in \mathbb{N}^*$, $\phi_{\mathbf{M}(\pi_n)}$ is a local function that takes into account the whole dynamic of the MDP and allows computing, *via* an *argmax*, an optimal choice of improvement (or optimal action) based on local information; $\phi_{\mathbf{M}(\pi_n)}(\mathbf{r}(s,a) \otimes \mathbf{p}(\cdot|s,a)) = \mathbf{m}(s,a) + \mathbf{p}(s,a)\mathbf{b}_{\pi_n}$. IMED-RL uses $\phi_{\widehat{\mathbf{M}}(\mathcal{A}(t))}$ and improves the skeleton similarly to policy iteration as it can be seen in the analysis 7.4.

Bandit control A degenerate case of MDP would be one where there is only one state s with $\phi_{\mathbf{M}(\phi)}(\mathbf{r}(s,a)) = \mathbf{m}(s,a)$ by choosing the bias func-

tion to be zero*. Playing optimally consists in playing an action with the largest expected reward at each time step t, $a_t \in \arg\max_{a \in \mathcal{A}_s} \mathbf{m}(s,a)$.

Bandit Learning occurs when rewards are unknown; this is the bandit problem. In that case, a lower bound on the regret similar to 7.1.2 exists. Under some assumptions on the reward distributions, optimal algorithms whose regret upper bounds asymptotically match the lower bound can be derived. IMED [74], KL-UCB [34, 114] are two such examples that use indexes, *i.e.* computes a number $I_{s,a}(t)$ at each time step and play $a_t \in \arg\min I_{s,a}(t)$. Such indexes are crafted to correctly handle the *exploration-exploitation* trade-off.

RL in Ergodic MDPs The delayed rewards caused by the dynamic of the system is the main source of difficulty arising from having more than one state. IMED-RL combines control and bandit theory in the following way. At each time step t, a restricted MDP $\widehat{\mathbf{M}}_t(\mathcal{A}(t))$ is built from the empirical one $\widehat{\mathbf{M}}_t$. If the condition to belong to the skeleton is selective enough, then the potentials on the restricted empirical MDP $\widehat{\mathbf{M}}_t(\mathcal{A}(t))$ may become close to those of the restricted true MDP $\mathbf{M}(\mathcal{A}(t))$, that is $\|\phi_{\widehat{\mathbf{M}}_t(\mathcal{A}(t))} - \phi_{\mathbf{M}(\mathcal{A}(t))}\|_\infty$ is small. We want to make policy improvements by finding, at each state s an action $a' \in \arg\max \phi_{\mathbf{M}(\mathcal{A}(t))}(\mathbf{r}(s,a) \otimes \mathbf{p}(\cdot|s,a))$, play it enough that it belongs to the skeleton which will modify ϕ and repeat until $\phi_{\mathbf{M}(\mathcal{A}(t))} = \phi_{\mathbf{M}}$. Using ϕ, the global dynamic is reduced to a local function so that at each state, the agent is presented a bandit problem. This bandit problem is well estimated if $\|\phi_{\widehat{\mathbf{M}}_t(\mathcal{A}(t))} - \phi_{\mathbf{M}(\mathcal{A}(t))}\|_\infty$ is small. As opposed to the control setting, the learning agent cannot choose the state in which to make the policy improvement step, and it may be possible that no policy improvement step is possible at state s_t. However, thanks to the ergodic assumption 7.0.1 the agent is guaranteed to visit such a state in finite time, if it exists. There is a trade-off between the adaptivity of the skeleton, *i.e.* how quickly one can add an improving action to define a new ϕ, and concentration of statistical quantities defined on the restricted MDP.

7.4 Regret of IMED-RL

In this section we state the main theoretical result of this chapter, which consists in the IMED-RL regret upper bound. We then sketch a few key ingredients of the proof. We refer to the paper [73] this chapter is based on for a full proof of regret of IMED-RL. In this manuscript, we prefer to focus on the proof of the next Chapter 8 that deal with communicating MDP since those proofs introduce new concepts, ideas, and technique that could later on be used to tackle the more challenging problem of communicating MDPs, the full RL setting. The proof of IMED-RL, while long, combines the proofs techniques presented in [16] and [78].

[73]: Pesquerel et al. (2022), 'IMED-RL: Regret optimal learning of ergodic Markov decision processes'

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

[78]: Burnetas et al. (1997), 'Optimal adaptive policies for Markov decision processes'

Theorem 7.4.1 (Regret upper bound) *Let* $\boldsymbol{M} = (\mathcal{S}, \mathcal{A}, \boldsymbol{p}, \boldsymbol{r})$ *be an MDP*

* recall that the bias function is defined up to an additive constant

satisfying assumptions 7.0.2, 7.0.1, 7.0.3. Let

$$0 < \epsilon \leq \frac{1}{3} \min_{\pi \in \Pi(\boldsymbol{M})} \min_{(s,a) \in \mathfrak{X}_{\boldsymbol{M}}} \{ |\Delta_{s,a}(\boldsymbol{M}(\pi))| \, : \, |\Delta_{s,a}(\boldsymbol{M}(\pi))| > 0 \} \, .$$

The regret of IMED-RL is upper bounded,

$$\mathscr{R}_{IMED-RL}(\boldsymbol{M}, T) \leq \left(\sum_{(s,a) \in \mathscr{C}(\boldsymbol{M})} \frac{\Delta_{s,a}(\boldsymbol{M})}{\mathscr{C}_{\mathscr{F}}^{s,a}(\boldsymbol{M}) - \epsilon \Gamma_s(\boldsymbol{M})} \right) \log T + O(1), \tag{7.12}$$

where $\Gamma_s(\boldsymbol{M})$ is constant that depends on the MDP $\boldsymbol{M}$ and state s.

From this Theorem 7.4.1, one can immediately deduce the optimality of IMED-RL, stated in the following Theorem 7.4.2.

Theorem 7.4.2 (Asymptotic Optimality) *IMED-RL is asymptotically optimal, that is,*

$$\lim_{T \to +\infty} \frac{\mathscr{R}_{IMED-RL}(\boldsymbol{M}, T)}{\log T} \leq \sum_{(s,a) \in \mathscr{C}(\boldsymbol{M})} \frac{\Delta_{s,a}(\boldsymbol{M})}{\mathscr{C}_{\mathscr{F}}^{s,a}(\boldsymbol{M})}. \tag{7.13}$$

This Theorem 7.4.2 proves the optimality of IMED-RL since the upper bound on the logarithmic growth rate of regret matches the lower bound of Theorem 7.1.2. In other words, Theorems 7.4.1 and 7.4.2 prove that IMED-RL is achieving an optimal exploitation-exploration trade off in ergodic MDPs. Such a bound was asymptotically matched by the algorithm proposed by [78] which suffer the previously mentioned practical problem. On the other hand, the current state-of-the-art algorithms UCRL3 and PSRL, while having some theoretical guarantees, have not been proved to match the regret lower bound. The theoretical guarantees of IMED-RL are problem-dependent rather than worst-case. Comparing to the $\log T$ bound derived for UCRL in Theorem 4 of [97], less known than the $\sqrt{T}$ bound, shows the benefit of our analysis for each instance, as we improve the constant factors in the leading terms: their dependency is $34 D^2 S^2 A / \Delta$, where Δ is a suboptimality gap and D the diameter of the MDP. On the practical side, Q-learning is often used without much theoretical guarantee because of its usually strong practical performances. It should be noted that Q-learning and other "practical" algorithms are often used in combination of a simple exploration scheme called ϵ-greedy. While the ergodic hypothesis can be criticized for its moderate scope of application, similar critics can be made to the ϵ-greedy exploration scheme. Indeed, the space of all policies that are ϵ-greedy can also be seen as a modification on the considered MDP that makes it ergodic.

[78]: Burnetas et al. (1997), 'Optimal adaptive policies for Markov decision processes'

[97]: Jaksch et al. (2010), 'Near-optimal Regret Bounds for Reinforcement Learning'

Communicating MDPs and ϵ-soft policies

The ergodic assumption can be limiting in practice, since most common MDPs are not ergodic but only communicating. Interestingly, in a communicating MDP, every stochastic policy $\pi : s \in \mathcal{S} \mapsto \pi(\cdot|s) \in \mathcal{P}(\mathcal{A}_s)$, with full-support (that is $\texttt{Supp}(\pi(\cdot|s)) = \mathcal{A}_s$ for each $s \in \mathcal{S}$) is ergodic. In particular, the uniform policy is ergodic. Also, ϵ-soft policies, that satisfy $\pi(a|s) \geq \epsilon$ for all s, a, are ergodic. When restricting to the class of ϵ-soft policies in a communicating MDPs, it seems that modifying

IMED-RL to be also ϵ-soft should lead to a strategy competitive with an optimal ϵ-soft policy. For $\epsilon < 1/|\mathcal{A}_s|$, the modification is to sample the chosen action with probability $1 - (|\mathcal{A}_s| - 1)\epsilon$ and any other action with probability ϵ. Now, a precise analysis of this modification is postponed to further work, and going beyond this case to handle the full-blown communicating assumption seem to require other ideas, especially since the lower bound for non-ergodic MDPs is expected to be much different from that of ergodic MDPs.

Therefore, while the ergodic assumption is not fully satisfying, the practitioner should not completely reject this working hypothesis. In the experiments, we will compare IMED-RL to those three algorithms, UCRL3, PSRL, and Q-learning. We will see that, even in communicating only MDPs, IMED-RL is very competitive.

Sketch of proof

[73]: Pesquerel et al. (2022), 'IMED-RL: Regret optimal learning of ergodic Markov decision processes'

Though a full proof is given in paper [73], we sketch here the main proof ideas that follow directly from the intuitions behind the IMED-RL conception. There are two main components in the design of IMED-RL. The first is the skeleton, which is used to define the policy from which IMED-RL should aim to improve. The second is the Bandit algorithm that is used to decide, at each time step, based on the current sub-problem defined by the skeleton, which action to sample. Interestingly, we can already see that, while IMED was used for that part, there should not be anything too specific about the IMED Bandit algorithm. Indeed, we think that a proof of a similar algorithm can be obtained using for instance UCB or NPTS (and appropriate corresponding assumption on the reward distributions).

The regret is decomposed into two terms, the **bandit** term when the local bandit problems defined by $\phi_{\widehat{\mathbf{M}}_t(\mathcal{A}(t))}$ is well estimated, and the **skeleton improvement** term that controls the probability that the local bandit problem is not well estimated. The main Theorem 7.4.1 follows from the following proposition. Recall from Lemma 7.1.1 that for all state-action pair $x \in \mathfrak{X}_{\mathbf{M}}$, $N_x(T) = \sum_{t=1}^{T} \mathbb{1}\{(s_t, a_t) = x\}$ counts the number of time the state-action pair x has been sampled.

Proposition 7.4.3 *For all state-action pair $x \in \mathfrak{X}_{\boldsymbol{M}}$, for all $\epsilon > 0$,*

$$N_x(t) \leq B_x(T) + S(T), \tag{7.14}$$

where we introduced the bandit term, $B_x(T)$,

$$B_x(T) = \sum_{t=1}^{T} \mathbb{1}\left\{ \begin{array}{l} x_t = x, \\ \mathcal{O}\left(\widehat{\boldsymbol{M}}_t\left(\mathcal{A}(t)\right)\right) \subseteq \mathcal{O}\left(\boldsymbol{M}\right), \\ \|\boldsymbol{b}^{\widehat{\boldsymbol{M}}_t(\mathcal{A}(t))} - \boldsymbol{b}^{\boldsymbol{M}}\|_\infty \leq \epsilon \end{array} \right\}, \tag{7.15}$$

and the skeleton improvement term, $S(T)$,

$$S(T) = \sum_{t=1}^{T} \mathbb{1}\overline{\left\{ \begin{array}{l} \mathcal{O}\left(\widehat{\boldsymbol{M}}_t\left(\mathscr{A}(t)\right)\right) \subseteq \mathcal{O}\left(\boldsymbol{M}\right), \\ \|\boldsymbol{b}^{\widehat{\boldsymbol{M}}_t(\mathscr{A}(t))} - \boldsymbol{b}^{\boldsymbol{M}}\|_\infty \leq \epsilon \end{array} \right\}}. \tag{7.16}$$

Furthermore, $\mathbb{E}\left(S(T)\right) = O(1)$, $\mathbb{E}\left(B_x(T)\right) = O(1)$ for a non-critical state-action pair, while for a critical state-action pair x,

$$\mathbb{E}\left(B_x(T)\right) \leq \frac{\Delta_x\left(\boldsymbol{M}\right)}{\mathcal{C}_{\mathcal{F}}^{x}\left(\boldsymbol{M}\right) - \epsilon\Gamma_s\left(\boldsymbol{M}\right)} \log T + O(1)\,.$$

Using the ergodic hypothesis

Central to the proof is the ergodic hypothesis. Thanks to this hypothesis, one can state that, whatever the strategy, there exists a constant $\lambda_{\mathbf{M}} > 0$ such that for all $\lambda < \lambda_{\mathbf{M}}$, the probability of the event

$$\{\exists s \in \mathcal{S}, N_s(t) < \lambda t\} \tag{7.17}$$

decreases exponentially fast to zero, *i.e.*

$$\mathbb{P}\left(\exists s \in \mathcal{S}, N_s(t) < \lambda t\right) \leq \exp\left(-\tau(\lambda, \mathbf{M})t\right)$$

where τ depends on the chosen λ and transition kernel, *i.e.* the MDP $\mathbf{M}$. At its core, this proposition state that in an ergodic MDP, one can consider that all states are visited a linear amount of the interactions. Of course, this linearity depends on the specific MDP and what we can call its mixing properties. Therefore, in the proof, one can consider that for all state, $N_s(t)$ is lower bounded by a linear function, $t \mapsto \lambda t$. Similarly, $\max_a N_{s,a}(t) \geq \frac{N_s(t)}{|\mathcal{A}_s|}$ can be considered lower bounded by a linear function, $t \mapsto \frac{\lambda}{A}t$. This property endure two things. First, it ensures that each state-bandit problems, were the agent must pick the best policy improving action, are accessible enough to the learner since the number of visits for each problem can be considered linear. Second, it ensures that the skeleton is made of state-action pairs that can be considered to have been visited a super-logarithmic number of the interactions, that is $\log^2 t$. Indeed, the skeleton is, at each state s, made of actions that are visited more than $\log^2 \max_a N_{s,a}(t)$, a quantity that can be considered with very high probability has larger than $\log^2 frac\lambda At$. Because of the concentration properties of the rewards (semi-bounded with moment generating function) and the transitions (multinomial distributions on the finite state space) estimates, that have exponential convergence of quantities of interests, it can be said that on the empirical MDP $\hat{\mathbf{M}}(\mathscr{A}(t))$ that is made of state-action pairs in the skeleton, gain and bias are well estimated and also enjoy concentration properties. Thanks to the concentration hypothesis, as long as the number of samples is super-logarithmic, *i.e.* larger than $\log^{1+\epsilon}(t)$, we are able to control the concentration of the gain and bias.

The bandit term

The bandit term Equation 7.15 counts the number of times the critical state-action pair $x \in \mathcal{X}_{\mathbf{M}}$ is sampled, $x_t = x$, when the sub-MDP extracted from the skeleton $\mathcal{A}(t)$ contains an optimal policy, $\mathcal{O}\left(\widehat{\mathbf{M}}_t\left(\mathcal{A}(t)\right)\right) \subseteq \mathcal{O}(\mathbf{M})$, and the bias of that optimal policy is well estimated, $\|\mathbf{b}^{\widehat{\mathbf{M}}_t(\mathcal{A}(t))} - \mathbf{b}^{\mathbf{M}}\|_\infty \leq \epsilon$. In the proof, ϵ is chosen small enough to discriminate all policies on $\mathbf{M}$, *i.e.* adding or subtracting ϵ to potential functions cannot exchange values of the biases. In the situation that $\mathcal{O}\left(\widehat{\mathbf{M}}_t\left(\mathcal{A}(t)\right)\right) \subseteq \mathcal{O}(\mathbf{M})$ and $\|\mathbf{b}^{\widehat{\mathbf{M}}_t(\mathcal{A}(t))} - \mathbf{b}^{\mathbf{M}}\|_\infty \leq \epsilon$, IMED-RL is mostly solving S Bandit problems, one at each of the S states, and the Bandit term $B_{s,a}$ is focused on counting what locally happen at state s. In that state, because IMED-RL is using the optimal IMED strategy on a Bandit problem that corresponds to the exact problem up to a variation of ϵ, the number of pulls of action a in state s is guaranteed to be a logarithmic number of time the agent visit state s. The number of time the agent visit state s is a linear function of the number of interaction thanks to the ergodic hypothesis. Hence, the control of this term mostly relies on the proof of IMED [16].

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

The skeleton improvement term

The skeleton improvement term Equation 7.16 is defined using the complementary of an event considering that, the optimal bias function on the skeleton is well estimated, $\|\mathbf{b}^{\widehat{\mathbf{M}}_t(\mathcal{A}(t))} - \mathbf{b}^{\mathbf{M}}\|_\infty \leq \epsilon$, and an optimal policy is included in the skeleton, $\mathcal{O}\left(\widehat{\mathbf{M}}_t\left(\mathcal{A}(t)\right)\right) \subseteq \mathcal{O}(\mathbf{M})$. We use the decomposition

$$\overline{A \cap B} = \left(\bar{A} \cap B\right) \cup \bar{B}$$

with $A = \mathcal{O}\left(\widehat{\mathbf{M}}_t\left(\mathcal{A}(t)\right)\right) \subseteq \mathcal{O}(\mathbf{M})$ and $B = \|\mathbf{b}^{\widehat{\mathbf{M}}_t(\mathcal{A}(t))} - \mathbf{b}^{\mathbf{M}}\|_\infty \leq \epsilon$.

Control of $\bar{B}$ The probability that the optimal bias on the sub-MDP $\mathbf{M}(\mathcal{A}(t))$ is not well estimated, $\|\mathbf{b}^{\widehat{\mathbf{M}}_t(\mathcal{A}(t))} - \mathbf{b}^{\mathbf{M}}\|_\infty > \epsilon$ is controlled by the fact that for all state-action pair $(s,a) \in \mathcal{A}(t)$, the number of associated sample is super-logarithmic in the number of visits in state s by definition of the skeleton, $N_{s,a}(t) \geq \log^2(max_{a'} N_{s,a'}(t))$. Since $max_{a'} N_{s,a'}(t) \geq \frac{N_s(t)}{|\mathcal{A}_s|}$ and using the ergodic property as mentioned previously, one can control the event $\bar{B}$ by an exponentially decreasing in t probability. We can now consider the control of $\bar{A} \cap B$.

Control of $\bar{A} \cap B$ Under the event B, the MDP restricted on the skeleton is well estimated and thanks to the properties of IMED, it should not take that long to add a policy improving action to the skeleton. Under B, this new skeleton will also be well estimated, and we therefore should not wait too long for another policy improving action to be added to the skeleton. Such a process repeat until, eventually, $\bar{A}$ cannot be true anymore. The longer the time t since the beginning of interaction, the less likely $\bar{A}$ holds true. Indeed, using the ergodic property, the entry barrier for a state-action pair to belong to the skeleton is that it is sampled a super-logarithmic number of time, $N_{s,a}(t) \geq \log^2 t$. When the optimal bias on

the skeleton is well estimated, *i.e.* B holds, then the IMED strategy (or any other good Bandit strategy), will start to sample a policy improving action in a state where it is possible (such a state exist because $\bar{A}$ holds true) a linear number of time of the new visits. This is because IMED is an optimal Bandit algorithm that sample optimal actions at a linear sampling rate. Because of the ergodic assumption, the number of visit in each state is linear and therefore the number of samples of such an action quickly become larger than $\log^2 t$. If the last policy improving action occurred at t_0 and the policy improving action was never sampled, then one must wait and additional time δ that is roughly such that $\log^2(t_0 + \delta) < \delta$, *i.e.* $\delta > \log^2 t_0$ since δ is positive. Thanks to the property of IMED, the added action to the skeleton is indeed a policy improving action with high probability. Since there is a finite number of deterministic stationary policies $|\Pi|$, one can divide the interval $[0, t]$ into several phases. One first phase of length $\log^2 t$ ensure that B holds with high probability, the bias is well estimated. We set $t_0 = \log^2 t$. Then we build a sequence of $|\Pi|$ intervals $[t_{k-1}, t_k]$ of length roughly equal to $\log^2 t$, $t_k = \log^2 t + k \log^2 t$. After each sub-interval, there is a high probability that a policy improvement occurred because a policy improving action surely would have been sampled at a linear rate, *i.e.* at a much faster rate than $\log^2 t$. Then, after $|\Pi|+1$ sub-intervals of length $\log^2 t$, there is a high probability that $\bar{A}$ does not hold and the remaining time $t - (|\Pi|+1)\log^2 t$ of the interval is spent under the event $A \cap B$. This remaining time is linear. The entry barrier to the skeleton cannot be smaller than $\log t$ since it is the rate at which an optimal algorithm such as IMED will sample suboptimal actions. To identify that an action is a statistically significant candidate for improving the skeleton/policy, it should be at least in a super-logarithmic regime, $\log^{1+\epsilon} t$, in which only optimal actions can be, under large probability. Together, those arguments help to upper bound $\mathbb{E}(S(T))$ by a convergent series.

7.5 Skeleton and finite time performances

Our notion of skeleton is built on the work of [78]. We improve on their original notion of skeleton by correcting some troubles happening in the small number of samples regime. In particular, this forces the authors to introduce some forcing mechanism. We discuss in this section the issues of the original definition and improvement induced by ours. **One key point of our definition is that the skeleton is defined using only empirical quantities, the number of samples, and does not depend on some arbitrary reference, such as the absolute time.**

We discuss the subtle but key modification that we made to the notion of *skeleton* introduced in the seminal paper of [78] and defined, for each state s and time t, by,

$$\mathcal{A}_s^{BK}(t) = \left\{ a \in \mathcal{A}_s : N_{s,a}(t) \geq \log^2 \left(N_s(t) \right) \right\}. \tag{7.18}$$

In contrast, the skeleton used in IMED-RL is defined replacing the sum $N_s(t) = \sum_{a' \in \mathcal{A}_s} N_{sa'}(t)$ with a maximum as follows

$$\mathcal{A}_s(t) = \left\{ a \in \mathcal{A}_s : N_{s,a}(t) \geq \log^2 \max_{a' \in \mathcal{A}_s} \left(N_{sa'}(t) \right) \right\}.$$

Correctness

The restricted MDP defined by IMED-RL, $\mathbf{M}_{\mathcal{A}(t)}$, is well-defined in the sense that, at each time, at least one action is available in each state, *i.e.* for all t, for all s, $\mathcal{A}_s(t) \neq \emptyset$. On the other hand, especially at the beginning, $\mathcal{A}_s^{BK}(t)$ could very well be empty. Indeed, in all state $s \in \mathcal{S}$, whenever there is an action $a \in \mathcal{A}_s$ with zero samples, such an action will be sampled by the algorithm (ours and that of [79]). Suppose that there are 4 actions in a state s. After the first 3 visits in s, whatever the current time t, $N_s(t) = 3$, $N_{s,a}(t) \leq 1$; it is 0 for the only unsampled action and 1 for the three others. Because $\log^2 3 \simeq 1.2 > 1$, the skeleton at state s is hence empty and therefore, no action belong to the skeleton, as per the definition of [79]. This situation does not happen when using our definition of skeleton used by IMED-RL, since $x \geq \ln^2(x)$ for all $x \geq 0.5$ and at least one action must be sampled ($N_{s,a}(t) \geq 1$). In this case, the behavior of the algorithm presented in the paper of [78] is undefined as it is not specified how to compute the bias and gain on the lacking restricted MDP. Now, in and MDP with a larger number of actions, say 100, the same argument shows that between the 3^{rd} and 100^{th} visit of state s, the skeleton at s is empty and the behavior undefined. This means that if there are only 20 states in the MDP, the behavior of the algorithm is undefined for at least about $|\mathcal{S}| \times (A - \log^2(A)) \simeq 2000$ steps (and possibly much more, since one would need all states to be visited about $A - \log^2(A)$ time, and it is unlikely that all states are visited equally often).

Incoherence

The skeleton as defined in (7.18) is "incoherent" in the sense that actions may be removed from it for no "justified" reason. In the worst case, all actions may be removed in one step. Assume a state s with 2 actions, one having been sampled 3 times and the other 2 times, *i.e.* $N_s(t) = 5$. Because $2 < \log^2 5 \simeq 2.6 < 3$, one action belongs to the skeleton and the other does not. Assume that the action that have been sampled 2 times is now sampled at time $k > t$. Then both actions have been sampled 3 times, $N_s(k) = 6$ and the skeleton at s is now empty since $\log^2 6 \simeq 3.2 > 3$. While this kind of behavior disappear for large number of samples, it is not desirable in finite time and introduces incoherence that makes the algorithm undefined and the learning less efficient if we were to resolve undefined behavior by random choices.

Forced exploration

Because of their definition of skeleton, forced exploration is necessary in the analysis of [78] meaning that their algorithm is not purely based on a computed index. While forced exploration and tracking is not inherently an unwanted feature, we think that it should be avoided when possible, hence leaning towards our IMED-RL skeleton.

Measuring accuracy

The skeleton is used to build a restricted MDP on which the gain and bias can be controlled. This control is due to the fact that, on the skeleton, state-action pairs have been sampled enough. In each state, we are interested in actions with a large enough number of pulls amongst all available actions. The most sampled action in each state should therefore obviously belong to the skeleton. Furthermore, it seems natural that the skeleton at a state does not change if the maximal precision in that state, given by the action that has been sampled the most in that state, does not change. This is mainly the rationale behind our subtle but key modification of the notion of skeleton. When the number of actions in a state is n and all actions have been sampled once, the number of visits in that state is n. However, the precision threshold used to build the skeleton of actions that are considered sampled enough should not be n since all actions have been sampled only once. The threshold should be related to the maximal possible "sampling precision" in the state, in this case 1. To put it more in a physics sentence, the perceived precision should not scale directly with the number of actions but with the number of samples per state-action pairs. However, the number of visit in a state is a linear function of the number of actions and therefore an extensive quantity. If we "merge" two identically sampled MDP by fusing states but adding actions, the original skeleton of [79] changes while ours does not. Our skeleton therefore rely more on an intensive property. This is very similar to the remark we made about KL-UCB that should not be using $\log t$ in its design but rather $\log \max N_a(t)$.

All of these explains why our algorithm display good numerical performances where finite time regret is important, as shown in Section 7.7

7.6 Computing the IMED-RL index

Prior to presenting the experimental benchmark, we recall that the IMED index is computable, which we know from the paper [16], and we intensively studied in the Chapter 4 devoted to a numerically sound approximation of the IMED index. In this section, we also present other aspects related to the computation of IMED-RL indexes.

[16]: Honda et al. (2015), 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.'

Computing the IMED-RL indexes is not that different from computing the IMED indexes, but it requires some extra steps. At each interaction, we run the value iteration algorithm on $\widehat{\mathbf{M}}_t(\mathcal{A}(t))$ to compute the optimal bias and the associated potential function $\phi_{\widehat{\mathbf{M}}_t(\mathcal{A}(t))}$. This task is standard. Once done, one must compute the value of the optimization problem $\mathbf{K}_{s,a}(t)$ which belongs to the category of convex optimization problem with linear constraint. Such problems have been studied under the name of *partially-finite convex optimization*, *e.g.* in [115]. It is possible to compute $\mathbf{K}_{s,a}(t)$ by considering the Legendre-Fenchel dual and one does not need to compute the optimal distribution to know the value of the optimization problem. This analytic form, we recall here, was studied in Chapter 4 and introduced by J. Honda and A. Takemura in [16].[3]

[115]: Borwein et al. (1991), 'Duality relationships for entropy-like minimization problem'

3: We mention that it was also previously studied by the same authors for discrete distributions in [15] then for bounded distribution in [12] before being extended to the semi-bounded setting in [16].

Proposition 7.6.1 (Index computation) *For $(s, a) \in \mathfrak{X}_{\boldsymbol{M}}$, we denote*

$$M = m_{max}(s, a) + \max_{s' \in \mathcal{S}} \boldsymbol{b}^{\boldsymbol{M}}(s)$$

the upper bound on the support of potential of state-action pair (s, a) and denote γ a generic threshold larger than the potential of state-action pair (s, a),

$$\gamma > \phi_{\boldsymbol{M}}(\boldsymbol{r}(s, a) \otimes \boldsymbol{p}(\cdot|s, a)).$$

When $\gamma > M$, then the unlikelihood of optimality is infinite When γ is smaller than M, then the IMED-RL index can be computed using the fact that $\mathcal{C}_{\mathcal{F}}^{s,a}(\boldsymbol{M}, \gamma)$, in its dual form, is the solution to a convex optimization problem,

$$\mathcal{C}_{\mathcal{F}}^{s,a}(\boldsymbol{M}, \gamma) = \max_{0 \leq x \leq \frac{1}{M-\gamma}} \mathbb{E}_{R,S \sim \boldsymbol{r}(s,a) \otimes \boldsymbol{p}(\cdot|s,a)} \left[\log \left(1 - \left(R + \boldsymbol{b}^{\boldsymbol{M}}(S) - \gamma \right) x \right) \right]$$

If $\gamma \leq \phi_{\boldsymbol{M}}(\boldsymbol{r}(s, a) \otimes \boldsymbol{p}(\cdot|s, a))$, then the unlikelihood of optimality is by definition equal to zero, $\mathcal{C}_{\mathcal{F}}^{s,a}(\boldsymbol{M}, \gamma) = 0$.

Computational complexity

In terms of state space size S and action space size A, the complexity of IMED-RL at each time step scales as $O(S^2A)$, the complexity of value iteration. Indeed, at each time step, IMED-RL runs value iteration using actions available in the skeleton, then computes the indexes of the available actions at the current state, and finally pick an *argmin*. The complexity of value iteration is $O(S^2A)$, the complexity of computing the A necessary indexes is $O(A \times S \log S)$, and the complexity of picking an *argmin* amongst those A indexes is $O(A)$. Therefore, the per-time-step complexity of IMED-RL scales as $O(S^2A)$. However, this scaling is mainly an upper-bound as value iteration is run with actions that are within the skeleton. By the theoretical design of the skeleton, we experimentally observe that, after some time, the skeleton contains one action per state (the optimal one). It means that, at some point, the computation of value iteration is reduced to the complexity of policy evaluation, $O(SA)$. At this point, the expected rewards and transitions of the state-action pairs that are in the skeleton does not fluctuate much between two samples. If the skeleton does not change, and fluctuations of skeleton-related distributions are small, the computed optimal bias, *i.e.* a result of policy evaluation, should not fluctuate a lot. Therefore, by initializing value iteration with the previously computed bias, one could hope for quick convergence. However, this behavior would only be observed in the "final" regime. Prior to that, it is possible that the skeleton has multiple actions per state within it. On the other hand, when the skeleton does not change and the fluctuations are controlled, one can think that most of the relevant quantities would not change by significant margin, at least from a numerical standpoint. This give the idea of **lazy updates**, and idea that we may combine with FIMED, presented in Chapter 4. Instead of performing a "quick" update, why cannot we not update at all? Another reason that motivates this notion of lazy update is the comparison of the IMED-RL running time with other algorithms. While IMED-RL is optimal in terms of progress per unit of interaction, it would be great if it

were close to optimal per unit of computation. This quest of optimally extracting information from samples at the minimal computational cost[4] is similar to one we started in Chapter 4 where we search for an efficient online computation of the $\mathcal{E}_{\mathcal{F}}$.

4: Formally, it could look like

$$\inf_{\text{cost}} \sup_{\text{progress}} \text{Problem}_t(X_{t+1}),$$

where the cost is related to a model of computation, progress is measured in terms of a regret rate or rate at which we interact with a solution of the problem.

It should be noted that the above complexity analysis is based on the usual value iteration algorithm where a fixed precision ϵ is used as a threshold to stop the computation and that this precision parameter ϵ is hidden in the $\mathcal{O}$ notation. If we were to *exactly* compute the bias function, then we would need to perform a $S \times S$-matrix inversion that would make the complexity per time step scale with S^3. However, this matrix inversion also depends on a precision parameter ϵ' so that it could be argued that we are simply being numerically more efficient by directly handling the precision parameter $\epsilon = \epsilon(\epsilon')$ in a numerically more efficient procedure. In the usual value iteration algorithm, the approximation of the matrix inversion is computed thanks to a convergent series, *i.e.*, the matrix $(I - P)^{-1}$ is approximated by $\sum_{k=0}^{n} P^k$, where the number n is related to the desired precision ϵ.

Lazy updates

Numerically, IMED-RL benefits from this fast computation and the fact that it employs a Value Iteration in lieu of an Extended Value Iteration for instance used in UCRL3. On the other hand, IMED-RL *a priori* updates its policy at each time step, unlike UCRL3 that proceeds into episodes. On our numerical experiments, the overall running time of IMED-RL is only about 5 times that of UCRL3, despite updating its policy at each time step. Still, for the tested horizon, it is one average 5 times slower. This foster the search for numerical optimization of the algorithm. As we just mentioned, it may be possible to further reduce the numerical complexity of IMED-RL by performing *lazy* computation of the indexes after some time. Indeed, by design, with high probability, the potential function $\phi_{\widehat{\mathbf{M}}(\mathcal{A}(t))}$ is not destined to change nor to be much different from the true $\phi_{\mathbf{M}}$ once an optimal policy belongs to $\mathcal{A}(t)$. As the number of samples increase, the magnitude of the updates decreases and $\phi_{\widehat{\mathbf{M}}(\mathcal{A}(t))}$ roughly remains the same, thus allowing the practitioner to perform value iteration every once in a while, when at least one estimate shifted by more than a fraction of the minimal suboptimality gap for instance. Of course this modification requires to update the regret analysis accordingly.

In this final regime, even more if using lazy updates, the complexity is likely to be controlled by the computation of the A indexes, for a total complexity of $O(SA)$. However, this complexity only tells half of the story since it does present the dependency in the number of collected rewards which can be at most $N_{s,a}(t)$ for a state action pair (s, a). Since $N_{s,a}(t)$ would be of order $\log t$, then the complexity of computing all the A indexes starts to be controlled by $\log t$ when $\log t >> S$ and is $O(A \times \log t \log\log t) \simeq \sum_{a \in \mathcal{A}(s)} O(\log N_{s,a}(t) \log N_{s,a}(t))$.

Link with FIMED Algorithm 14

This where the work presented in Chapter 4 can be useful. Instead of computing all the indexes at each iteration, one could use one of the

approximation schemes, in particular FIMED 14 which reduces the time complexity of computing an index to $O(1)$ per arm and therefore to $O(A)$ for all actions in the current state. We mentioned FIMED because it is one of the algorithm for which a proof of optimal regret was provided and that is likely to be transferred to the IMED-RL index. One downside of FIMED compared to OIMED is that its space complexity is still that of storing all the samples. However, from a time complexity standpoint, if we could add to the design of IMED-RL the lazy updates scheme and the FIMED schemes, then one could hope for a numerically highly competitive algorithm. Only adding the design of FIMED to create FIMED-RL could be a sufficient transformation to reduce by a non-negligible factor the time complexity of IMED-RL. Indeed, when profiling the code, we found that the computation of indexes was, at some point the bottleneck of the computational graph.[5]

5: Hopefully, some experiments will be run for the oral presentation.

Bandit index

In the IMED-RL algorithm, we used the IMED Bandit algorithm to select, at each time step, the next action to sample. However, it can be seen from the original design that there is *a priori* nothing to specific about IMED in the general structure of IMED-RL. In the IMED-RL Algorithm 21, one could *a priori* replace IMED line 3 by another Bandit algorithm, such as KL-UCB, UCB, or NPTS. However, doing so would require rederiving a proof of regret upper bound for the specific algorithm used since the proof of IMED-RL uses a decomposition of events that are adapted to the structure of IMED. Using a decomposition that is adapted to KL-UCB, *e.g. à la* A. N. Burnetas and N. Katehakis [105], could probably be enough to adapt the proof of IMED-RL to a proof of a KL-UCB-RL. Under a bounded reward assumption, one can also probably derive a proof of regret for UCB and TS-like algorithms. We experimented with those algorithms, in particular MED, to see if there were instances of communicating MDPs where IMED-RL failed but MED-RL succeed in order to find ideas for finding an algorithm with provable guarantees on communicating only MDPs. Due to the randomness of MED, we thought that it would be possible that MED-RL is better than IMED-RL on some communicating MDPs. However, we could not find such an instance and IMED-RL is on par with MED-RL. We also tested a version where IMED was replaced with UCB, which experimentally does have a logarithmic regret curve but with larger empirical regret than IMED-RL. For the practitioner to whom the speed at which indexes are computed, *i.e.* the speed at which decision are taken, it may be interesting to use the fastest Bandit algorithm that it can access (such as UCB, as shown in Chapter 4) even at the cost of a larger regret. In this chapter, we are interested in showcasing an algorithm that target asymptotic optimality while having impressive numerical performances. Therefore, we proceed with IMED-RL.

[105]: Burnetas et al. (1997), 'Optimal adaptive policies for Markov decision processes'

7.7 Numerical experiments

In this section, we discuss numerical aspects of IMED-RL. We run experiments that can be replicated using the source code hosted on github.

We consider several RL environments, having different properties making them *easier* or *harder* for some of the tested algorithms. This variety of considered environments foster a fairer comparison between algorithms and avoid the risk of crafting specialized algorithms (that may have general guarantees but perform numerically better on some type of environment). We will see that IMED-RL seems uniformly[6] good, not to say uniformly better.

6: uniformly on the tested environments

Benchmark

In different environments, we illustrate the performance of IMED-RL against the strategies UCRL3 [86], PSRL [90] and Q-learning (run with discount $\gamma = 0.99$ and optimistic initialization). From an implementation point of view, most of the complexity of IMED-RL is in the analysis rather than in the algorithm. Compared to PSRL and UCRL3, IMED-RL does not take a confidence parameter nor any hyperparameter. Also, IMED-RL uses value iteration as a routine, which is simpler (and faster as we already discussed) than the extended value iteration used in UCRL3. Q-learning technically takes an exploration parameter (ϵ-greedy exploration) or exploration scheme (ϵ_t-greedy exploration) when exploration is slowly decreased with time. We report average algorithmic regret curves computed from 256 independent experiments along with quantiles 0.1 and 0.9. The considered horizon depends on the environment as some are "easier" than others in the time required to see regret curves exhibit their logarithmic behaviors. For all experiments, we used environments with maximal expected reward 0.99 and bound $m_{max} = 1$ was used as an upper bound on the class $\mathcal{F}$ of considered reward distribution.

[86]: Bourel et al. (2020), 'Tightening exploration in upper confidence reinforcement learning'

[90]: Osband et al. (2013), '(More) efficient reinforcement learning via posterior sampling'

Theoretically, IMED-RL guarantees are only valid for ergodic MDPs. In this section we depart a bit from that hypothesis and also assess the performances of IMED-RL on communicating only MDPs. This also makes sense since we are comparing to, for instance UCRL3, algorithms that are not specifically developed for ergodic MDPs. Therefore, it could have been argued that, as an algorithm specialized for this setting, IMED-RL had an unfair numerical advantage against the other benchmarked algorithms. When testing on environment that are communicating only, we also test the same environment but with modified transitions so that its is transformed into an ergodic MDP. We recall that, any finite communicating MDP can be turned into an ergodic one, since on such MDPs, any stochastic policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A}_s)$ with full support $\texttt{Supp}(\pi(s)) = \mathcal{A}_s$ is ergodic. Hence, by mixing its transition $\mathbf{p}$ with that obtained from playing a uniform policy, formally

$$\mathbf{p}_\epsilon(\cdot|s,a) = (1-\epsilon)\mathbf{p}(\cdot|s,a) + \epsilon \sum_{a' \in \mathcal{A}_s} \mathbf{p}(\cdot|s,a')/|\mathcal{A}_s| \,,$$

for an arbitrarily small $\epsilon > 0$ one obtain an ergodic MDP. In the experiments, we consider an ergodic version of the n-state river-swim, 2-rooms and 4-rooms environment with $\epsilon = 10^{-3}$, and classical communicating versions ($\epsilon = 0$). It is interesting to see that those environments are ergodic for all values of $\epsilon > 0$ and communicating only at $\epsilon = 0$. This discontinuity in $\epsilon = 0$ is interesting for two reasons. First, as we decrease ϵ and even make it equal to $zero$, we do not observe that the experimental regret of IMED-RL (and other algorithms) change. This suggest that

second order term that depends on ϵ in the regret bound may be more controlled than thought. The second is linked to the fact that any deterministic policy mixed with a policy that assign positive probability to all actions is ergodic in the sense that its recurrent class is the whole state space. Thus, by using a stochastic version of IMED-RL such as MED-RL, one could have a sequence of ergodic policies, with *a priori* decreasing ergodicity since we will have convergence to an optimal deterministic stationary policy. It could be that if the rate at which ergodicity is reduced, one could have regret guarantees in the communicating only case. For instance, one could try to have a logarithmic rate of ergodicity that is reduced as a function of the suboptimality-information gap between the optimal policy and all other policies with distinct recurrent sets, *i.e.* policies that are such that one cannot gather all information about by only playing the optimal policy. This is why we tested MED-RL against IMED-RL in communicating MDPs and search for an instance that would make IMED-RL failed and MED-RL succeed.

RiverSwim

The *RiverSwim* environment (Figure 7.1), is an environment that is difficult to navigate and require the agent to "figure out" a difficultly attainable large reward is located at one end of the chain-like environment while a small but easy obtainable reward is located at the other end. In each

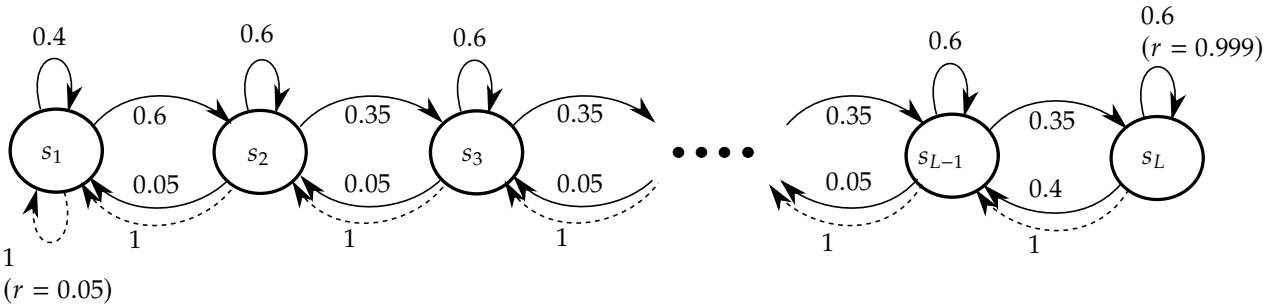

Figure 7.1: The n-states *RiverSwim* MDP

of the n states, there are two actions `RIGHT` and `LEFT`. In Figure 8.3, the `LEFT` action is represented with a dashed line and the `RIGHT` with plain line. Rewards are located at the extremities of the MDP, with a small reward in left initial state s_1 and large reward in the rightmost state s_n. Starting from state s_1, this setting has proven to be a challenging one because of the large amount of non-rewarding exploration necessary to find the optimal policy. We consider the 6-state and 25-state instances, which allows us to compare how algorithms behave depending on the amount of necessary exploration.

6-states RiverSwim

As illustrated by Figure 7.2, the performances of IMED-RL are particularly good and the regret of IMED-RL is below the regrets of all its competitors, even when the MDP is communicating only. This numerical performance grounds numerically the previous theoretical analysis. In this setting, all the tested algorithms seems to have a good enough behavior but the

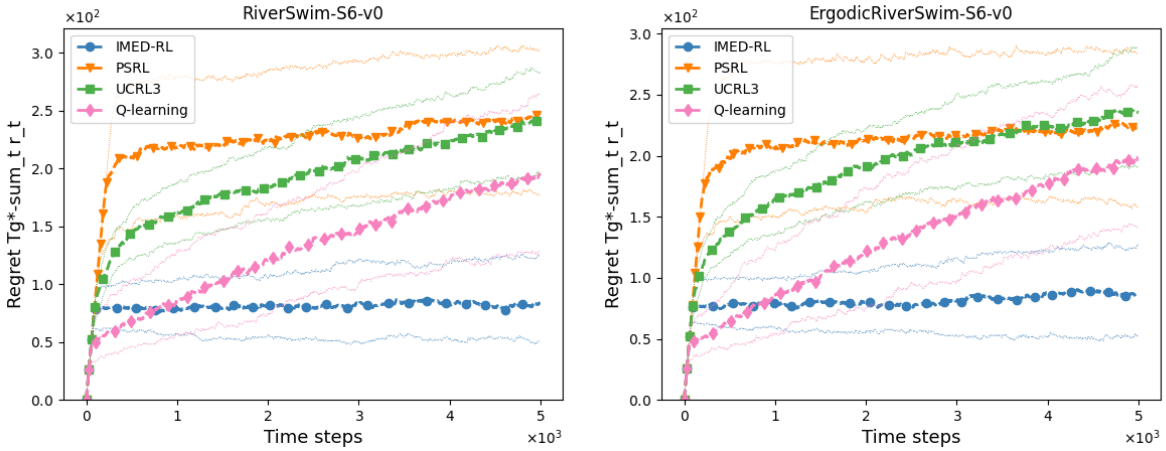

Figure 7.2: Regret on 6-states RiverSwim, communicating only (left) and ergodic version (right)

regret performance of IMED-RL stands out. Interestingly, some algorithm such as PSRL have a regret performance that does not scale well with the number of considered states in a RiverSwim environment.

25-states RiverSwim

To study the effect of the number of states on the shape of regret functions, we consider another RiverSwim experiment, this time with 25 states. RiverSwim environments are sometimes considered hard instances for strategies such as PSRL, as the reward signal is sparse. We observe in Figure 7.3 that PSRL indeed struggles in such an environment while it succeeded when the number of states was smaller. The three other strategies work well, with some statistically clear advantage for IMED-RL on the long run, as it can be seen from separations of the quantile tubes. For the sake of completeness, we present the average runtime for

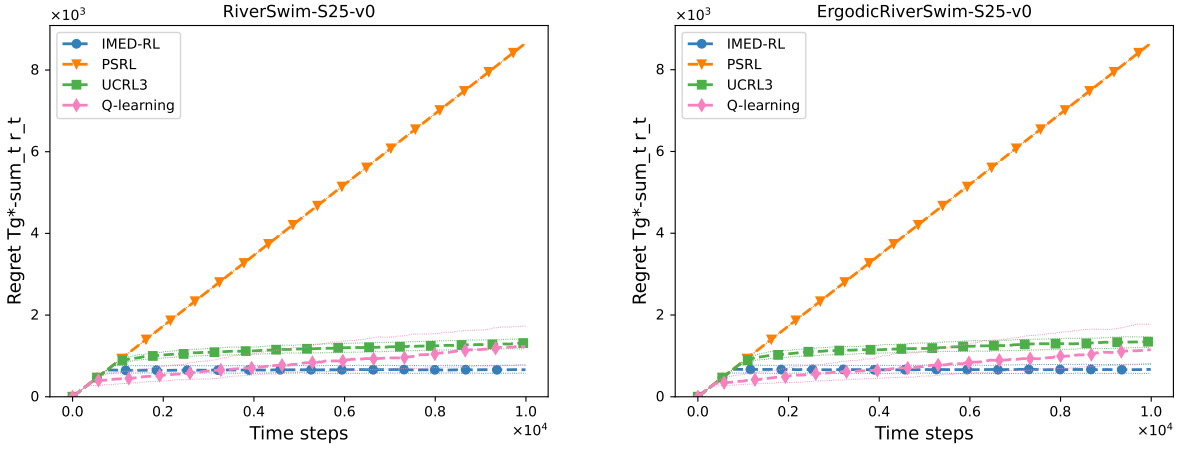

Figure 7.3: Regret on 25-states RiverSwim, communicating only (left) and ergodic version (right)

completing a trajectory of the tested algorithms on both ergodic and non-ergodic 25-states RiverSwim. Apart from Q-learning which is the fastest by a large margin, all algorithms seem to benefit a numerical boost from the ergodicity of the environment. This is perfectly coherent with the fact that Q-learning is a model-free algorithm whose per-time-step

Table 7.1: Average runtime (expressed in seconds) to complete one episode on 25-states RiverSwim

	IMED-RL	PSRL	UCRL3	Q-learning
non-ergodic	5.56	0.15	0.42	0.02
ergodic	1.45	0.04	0.23	0.02

running time does not depend on the evaluation of a function depending on the model of the transition kernel. On the other hand, IMED-RL, PSRL, and UCRL3 are all model-based and benefit from a computational boost in the ergodic case. This is coherent with the fact that, in the ergodic version of a communicating MDP, the added ϵ-transition reduced the mixing time and increased convergence rate of power laws used, for instance, by value iteration. That is not to say that those algorithms will, in general, run faster on ergodic MDPs, it is the fact that, when modifying an MDP with the "ergodicitization" process described above, we reduce mixing times and increase convergence rate of statistical quantities that depends on mixing properties of the MDP.

Reward-rich environment

In contrast with the previous experiment, where reward were spare and PSRL at disadvantage, we present an experiment where rewards are frequent and relatively high compared to the upper bound on the support of rewards, which is considered to be one. Apart from PSRL, it is also interesting to test the behavior of the algorithm in a type of environments where the reward signal is not sparse and there is not "structure" in the transition, such as the chain-like structure in RiverSwim. In the following

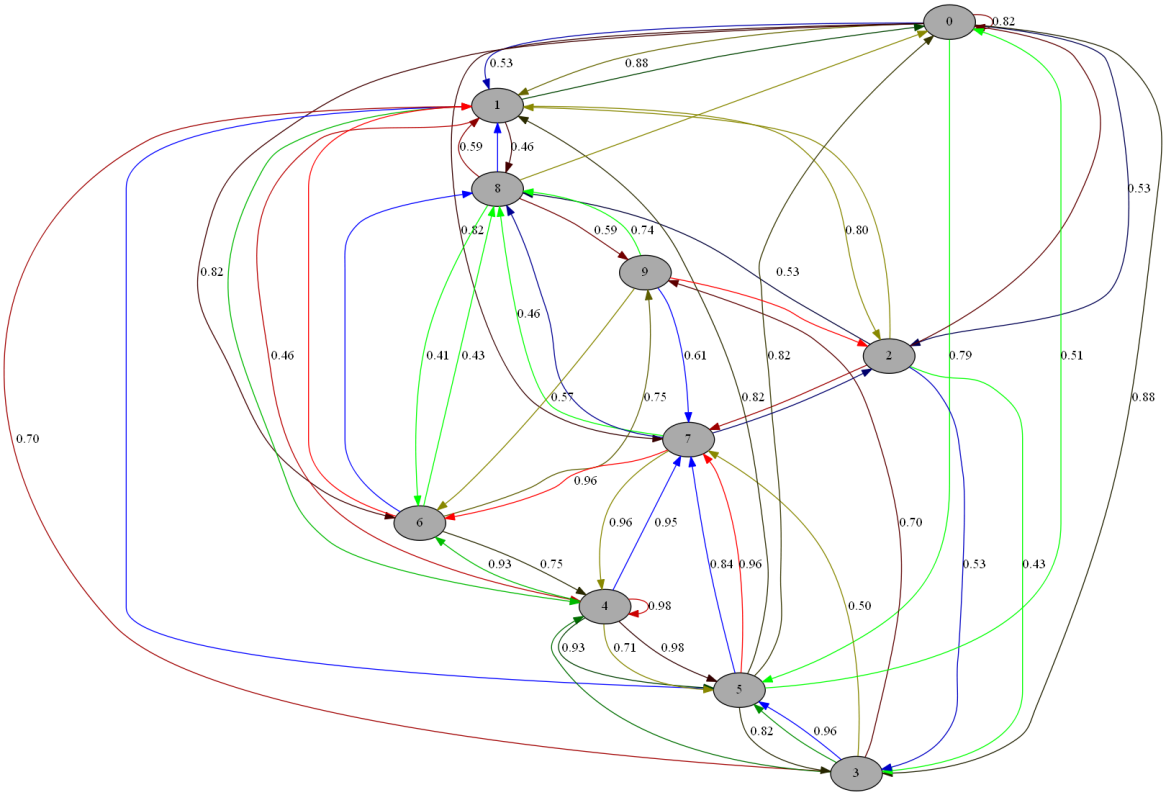

Figure 7.4: Reward-rich environment where transitions and rewards were generated at random (rewards are printed)

experiment, we consider a reward-rich environment, where about 80% of state-action pairs generate a reward of at least 0.4 (and the maximal reward is 0.99). Such environments are known to favor the PSRL strategy as well as optimistically initialized strategies, that benefit from a reduced burn-in phase thanks to their prior.

In Figure 7.5, we plot both the regret curves of the non-ergodic Reward-rich environment, left, and the regret curves of corresponding to the ergodic version of the same environment. We observe that IMED-RL

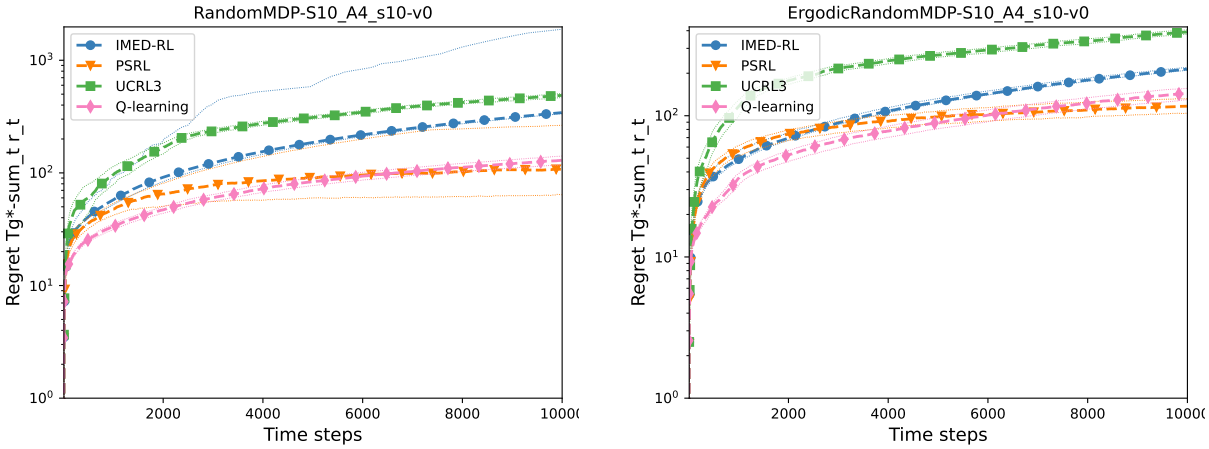

Figure 7.5: Average regret and quantiles (0.1 and 0.9) curves of algorithms (in log-scale) in a reward-rich environment (10 states, 4 actions) where 80% of state-action pairs give reward of at least 0.4. Right: regret in the ergodic version of the MDP.

outperforms the UCRL3 strategy, but is indeed beaten by PSRL (while PSRL had poor regret in reward-scarce environments, see Figure 7.3 and next experiments), as well as the Q-learning algorithm initialized with $\gamma = 0.99$ and initial value $1/(1-\gamma)$ in each state. When the MDP is modified to have minimal transition $\mathbf{p}(s'|s,a) \geq 0.01$ for each s, a, s', the performance of IMED-RL improves and becomes more stable (as well as that of other strategies), as seen in the right part Figure 7.5. By stability, we mean that its 0.9 quantile is much closer to its expected regret. This confirms that IMED-RL indeed exploit the ergodicity of its environment, at least at the level of "error recovering", when the regret is initially high, possibly because of bad first samples inducing the algorithm in errors. The time it takes IMED-RL to recover from such initial bad samples is likely to be what explain the shape of the 0.9 quantile curve in the left plot of Figure 7.5.

n-rooms

We now consider two grid-like environments, 4-*rooms* and 2-*rooms* (Figure 7.6), with sparse rewards. Such grid-like environments are classically used in RL to assess numerical performances, in particular in more numerically inclined papers, particularly deep RL which is the big absent topic of this PhD despite its very active research community. Both n-rooms environments are sparse reward with close-to-deterministic transitions. 4-rooms is a grid-like environment with 20 states and 2-rooms is an environment with 55 states. For both those grid-like environments there are 4 cardinal actions where transitions are close to deterministic with a

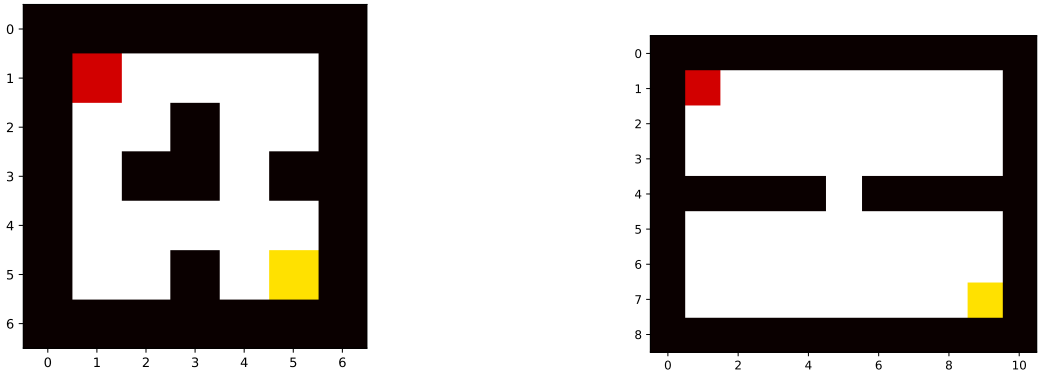

Figure 7.6: The 4-*rooms* (left) and 2-*rooms* (right) MDPs

0.9 chance of going in the intended direction and the remaining 0.1 probability mass equally dispatched between the three adjacent remaining cardinal states. Sometimes, we say that such a grid-like environment is slippery, and it is furthermore ergodic. A reward of 0.99 is located in the goal state (highlighted in yellow), while it is zero elsewhere. After reaching the goal, the agent is positioned again in the initial red-state. One can see that these environments contains some bottleneck states, which can sometimes make those considered as hard instances. In particular the 2-rooms environment where it is harder to cross between the two rooms by playing randomly. The agent shall actively try and "understand" the benefit of crossing the bottleneck state and not getting stuck in the first room. Of course, the fact that the environment are

4-rooms environment

As illustrated by Figure 7.7, the performances of IMED-RL are particularly good, in this environment. Indeed, its regret curve is significantly below

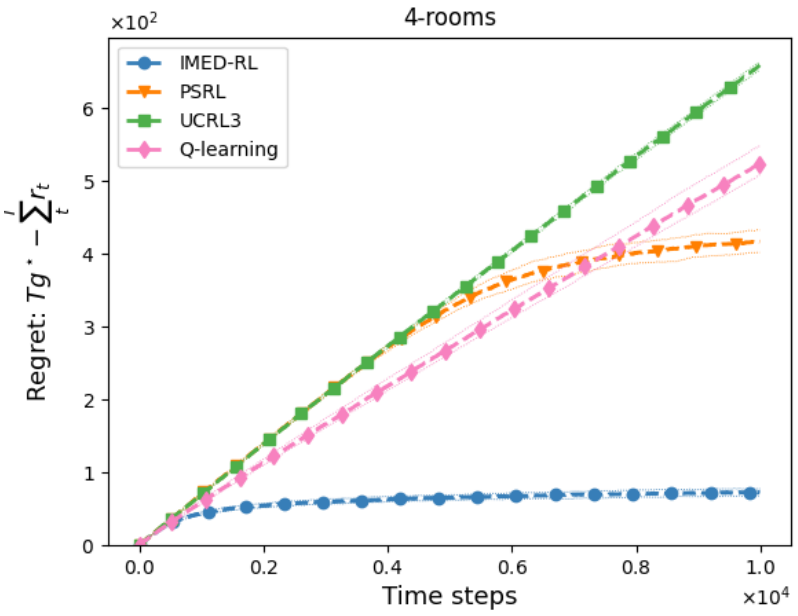

Figure 7.7: Regret curves on the 4-rooms environment

the ones of other algorithms. In this setting, for the tested horizon, only for PSRL could we see the curve bent into its logarithmic shape. Despite the scarcity of reward, PSRL is successful in this environment thanks to its optimistic initialization and the quasi-deterministic nature of transition that guarantee the algorithm to experiment a path to the unique reward of the environment with a large enough probability. However, because of its optimistic initialization, it may take some time for PSRL to actually converge on the optimal policy which is related to the shortest path from the resetting state to the reward state. If PSRL computes that some rewards may be present along other path, its average reward is reduced because it actually gets the reward at a slower frequency than it could. Once it discriminates this path from the others, its regret curve bend. For other algorithms, one cannot even state that something was learnt from the displayed regret growth rate.

2-rooms environment

The 2-rooms environment we consider now is actually a larger state-action space than the four-room MDP considered previously. Note that since the considered grid-worlds are slippery (frozen-lake style, with 0.1 probability of visiting executing nearby actions), this also means that from the bottleneck state, it is actually possible to enter the bottom room not only with action down, but also left and right. Hence, this MDP does not contain an as hard as it could be bottleneck state-action pair. In such environments, although not being "very" ergodic, we expect the IMED-RL strategy to work reasonably well, which is confirmed by the experiment in Figure 7.8. This time, and for all reasonable tested horizon,

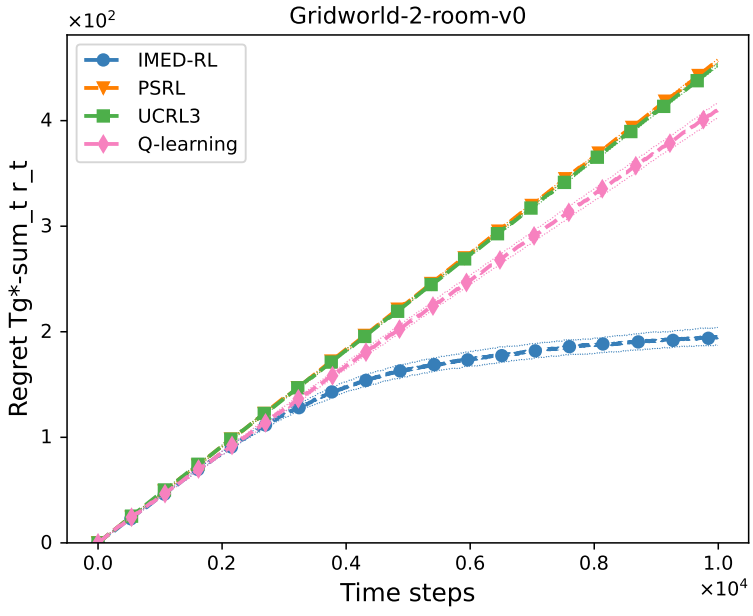

Figure 7.8: Regret curves on the 2-rooms environment

only for IMED-RL can we see the regret curve bend into a logarithmic shape.

These two n-rooms experiments are a clue that the IMED-RL strategy may still be reasonable, although not necessarily optimal in some communicating MDPs. Furthermore, the effectiveness of IMED-RL displayed

in the experiment presented until now is a proof of concept that its algorithmic design, in particular the skeleton design, guarantee strong finite time performances. We argue that this really is due to the fact that nowhere in the code of IMED-RL can be found an explicit mention of the time spent since the beginning of the experiments. Only number of samples are used to perform comparison of "precision" and the number of samples are used in a way that measured precision by IMED-RL does not depend on the number of actions. That is to say, if two actions in one state have the same number of samples, the measured precision is "independent" on the number of actions that the agent can play in that state. Precision is more like an intensive property of the system rather than an extensive property.

Random grid-worlds

In this part, we provide complementary experiments in Figure 7.9 and Figure 7.10 with randomly generated grid-worlds with a unique goal state represented in white. Black squares represent walls in the environment and red squares represent state in which the agent can be in. Whenever, the learner reach the goal state (white), it is transported in another state (red), at random. The actions are similar to the one presented in the previous part about the two n-rooms environment. The main difference is that the environments have the topology of a torus since states at a border are connected (if not a wall) to the state in regard of the opposite border.

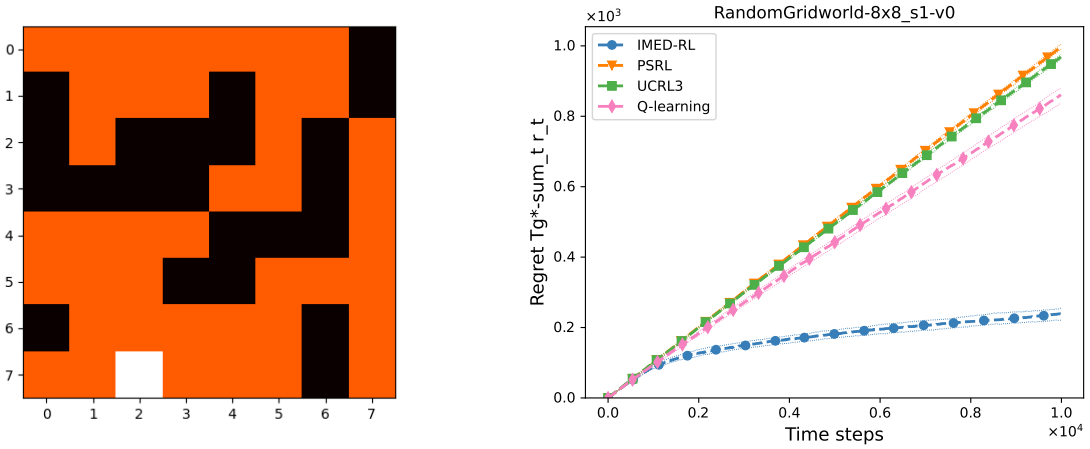

Figure 7.9: Average regret and quantiles (0.1 and 0.9) curves of algorithms (right) in a randomly generated grid-world (8x8 grid, 4 actions) with reward 0.99 in white state (right).

Once again, we can observe the striking performance of IMED-RL against the state-of-the-art UCRL3 or related PSRL and Q-learning strategies. Note that these other strategies eventually learn as well, but for larger time horizon (too large to be tested with 256 independent runs). We present in Table 7.2 the average runtime for completing a trajectory of the tested algorithms on such a grid-world environment.

Table 7.2: Average runtime (second) on 8×8 grid-world

IMED-RL	PSRL	UCRL3	Q-learning
1.82	0.75	6.36	0.03

We can see that, compared to RiverSwim, the time performances of IMED-RL and UCRL3 are exchanged. Generally, our experiments tends to show that the performances of IMED-RL are quite good on grid-worlds, both from a regret minimization viewpoint and a numerical complexity viewpoint.

In Figure 7.10, we did not report UCRL3 and PSRL as their computation time were prohibitive compare to IMED-RL and Q-learning in this setup. The problem being larger and more complex, IMED-RL learn at a slower

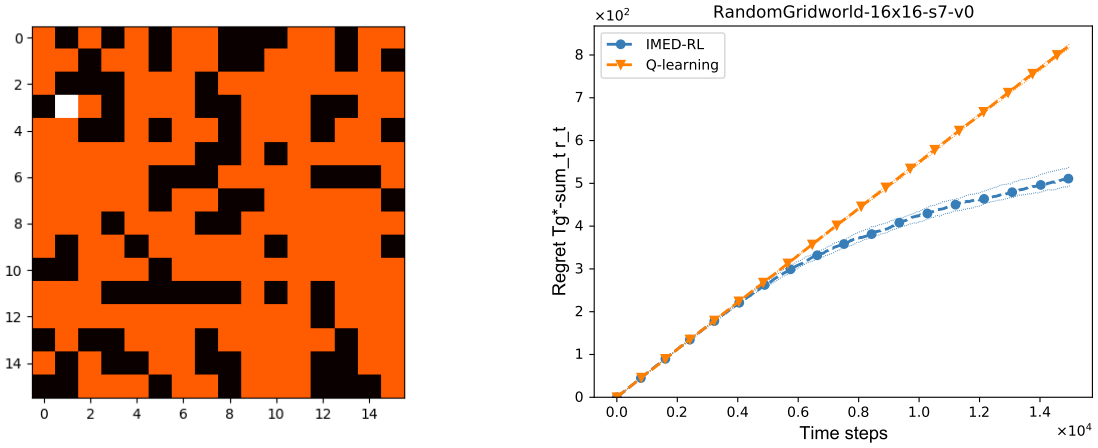

Figure 7.10: Average regret and quantiles (0.1 and 0.9) curves of algorithms (right) in a randomly generated grid-world (16x16 grid, 4 actions) with reward 0.99 in white state (right).

rate than of the other grid but still manage to display a logarithmic shape of regret which is impressive when compared to other algorithms in the previous smaller environment.

We end this experimental section by presenting an experiment that we set up originally in order to (try to) make IMED-RL fail. This is what we call, the Nasty environment.

Nasty environment

In order to better understand the limitation of the IMED-RL algorithm, we tried (but did not succeed) to craft an environment that would make the IMED-RL algorithm fail. The analysis reveals that we should consider a non-ergodic MDP for this purpose. Importantly, the index for pair (s, a) is based on building a modified MDP with unmodified reward and transitions for pairs different from (s, a), which is a feature coming from the ergodic property. However, in a non-ergodic MDP, an optimal policy and a policy playing a in state s may have different recurrent classes, say class $\star$ and $\star_a$. It is not difficult to show that, when all paths from a state in $\star$ to a state in $\star_a$ must contain (s, a), *i.e.* (s, a) is a bottleneck pair, then changing the MDP only in pair (s, a) to build a "confusing instance" isn't sound anymore, hence the construction of the IMED-RL index is *a priori* no longer justified in such cases. Inspired from this intuition, we build in Figure 7.11 a specific nasty MDP with such a bottleneck state-action pair, separating two cycles with close value. We remark that in this structure, two promising cycles at two ends of a chain with smaller rewards in between, may induce an "oscillation" of a learning agent between the two

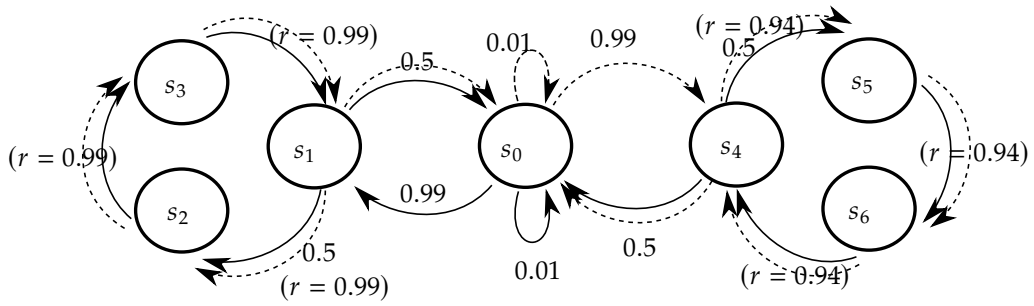

Figure 7.11: The *Nasty* MDP

cycles, paying the cost of the travel along the chain each time it "decides" to change cycle.

We observe that the quantile tube of IMED-RL is larger than before and indeed indicates more struggles but not enough that the IMED-RL fails the task. Still, we remark a small advantage of PSRL over IMED-RL in this environment. Note that the environment is reward rich with rewards close to 1, which also favors PSRL and Q-learning with optimistic initialization. This confirms the empirical result found in the non-ergodic

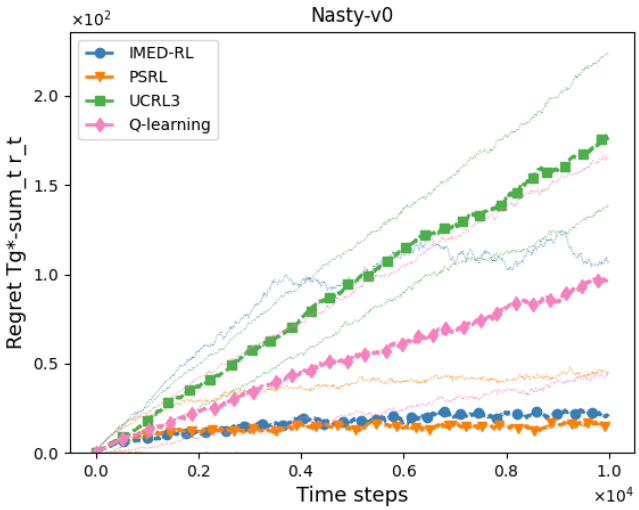

Figure 7.12: Average regret and quantiles (0.1 and 0.9) curves of algorithms (right) in a nasty environment with two cycles separated by a bottleneck action.

Reward-rich environment Figure 7.5, where a similar analysis of the 0.9 quantile indicated a "lack of stability" from IMED-RL compared to the ergodic version of the environment. It is pleasing to see that its numerical guarantees seems to go beyond the ergodic assumption with particularly good performances in grid-world.

Conclusion In this chapter, we introduced IMED-RL, a numerically efficient algorithm to solve the average-reward criterion problem under the ergodic assumption for which we derive an upper bound on the regret matching the known regret lower bound. Further, it has surprisingly good numerical performances in communicating only MDPs. The overall good performances of IMED-RL, even in this Nasty environment crafted to make it fail is promising for the future, where we will be researching

to adapt this algorithm to communicating only MDPs. Both the problem-dependent and worst-case regret bounds are interesting in this regard. We start such a quest with the IMED-KD algorithm that is introduced in the next Chapter 8, slowly but surely paving the way for an adapting IMED-RL to communicating only MDPs. Another direction we intend to explore is the adaptation of IMED-RL main ideas to *function approximation* frameworks, such as neural networks and kernel methods.

Exploiting dynamics knowledge with IMED-KD

8

In this chapter, based on the paper ***Logarithmic regret in communicating MDPs: Leveraging known dynamics with bandits*** published at ACML 2023 with Hassan Saber, Sadegh Talebi and Odalric-Ambrym Maillard, we study regret minimization in an average-reward and communicating MDP with known dynamics, but unknown reward function.

Although learning in such MDPs is a priori easier than in fully unknown ones, they are still largely challenging as they include as special cases large classes of problems such as combinatorial semi-bandits. Leveraging the knowledge on transition function in regret minimization, in a statistically efficient way, appears largely unexplored. As it is conjectured that achieving exact optimality in generic MDPs is NP-hard, even with known transitions, we focus on a computationally efficient relaxation, at the cost of achieving order-optimal logarithmic regret instead of exact optimality. This NP-hardness may be linked to the longest (simple) path problem in graphs, which is known to be NP-hard. Indeed, if we need the agent to travel from one place of the MDP to another maximizing its reward on the path, then one need to solve a problem akin to the longest path. The problem of efficient exploration in MDP, even with known dynamic is thus a non-trivial problem. This is why we target a regret that scales with the logarithm of the number of interactions but do not target the optimal logarithmic growth rate of the regret.

The main contribution of the paper this part is based on is the introduction of a novel strategy based on the IMED strategy for bandits that the reader should be familiar with at this point of the manuscript. A key component of the novel algorithm includes a carefully designed stopping criterion leveraging the recurrent classes induced by stationary policies. To this purpose, we introduce a class of strategies called *rarely-switching algorithms*. The idea is that an agent should not leave the set of states that are recurrent under an optimal policy but sometimes need to depart from it to explore the states (and actions) that are not visited by the empirical best policies. Note that this was not needed under the ergodic hypothesis since all states are recurrent under all policies. To ensure a logarithmic regret, the agent should *rarely switch* between recurrent classes, *i.e.* policies, and explore as fast as possible. Studying this problem from a theoretical and experimental standpoint is the work done in this chapter.

We derive a non-asymptotic, problem-dependent logarithmic regret bound for this strategy, which relies on a novel regret decomposition leveraging the structure. More specifically, we leverage the known dynamics with bandits, seeing the space of policies on an MDP as a highly structured set of arms. Quite obviously, playing a policy during a certain amount of time give information about all policies sharing at least one recurrent state-action pair with this policy. We show that exploiting this structure is possible in an efficient enough way that the agent suffer a logarithmic regret. Furthermore, we provide an efficient implementation and experiments illustrating its promising empirical performance.

Table of Notation

We list here, a few notations that are used a lot in this chapter for the reader to report if needed.

$\mathbf{M}$ average-reward Markov Decision Process
$\mathscr{C}$ set of state-action pairs
$\mathbf{p}$ transition distribution function on $\mathscr{C}$
$\mathbf{r}$ reward distribution function on $\mathscr{C}$
$\mathbf{m}$ mean reward function on $\mathscr{C}$
Π finite set of stationary policies
T horizon time
$\mathbf{A}$ algorithm following a policy sequence $(\pi_t)_{1\leq t\leq T} \subset \Pi$
$V_{\mathbf{M}}(\mathbf{A}, T)$ cumulative reward of an algorithm $\mathbf{A}$
$\mathbf{p}_\pi$ transition probability on $\mathscr{C}^2$ of the Markov chain induced by stationary policy π acting on $\mathbf{M}$
$\overline{\mathbf{p}}_\pi$ Cesàro-average of $\mathbf{p}_\pi$
$\mathbf{g}_{c,\pi}$ gain of stationary policy π starting from state-action pair c
$\mathbf{g}_c^\star$ maximal gain of stationary policies starting from state-action pair c
$\mathbf{g}^\star$ maximal gain of stationary policies
$\Pi_c^\star$ set of stationary policies achieving maximal gain $\mathbf{g}_c^\star$ when starting from state-action pair c
$\Pi^\star$ set of stationary policies achieving maximal gain $\mathbf{g}^\star$
$\pi^\star$ stationary policy achieving maximal gain $\mathbf{g}^\star$
$\mathscr{R}_{\mathbf{M}}(\mathbf{A}, T)$ expected regret with respect to playing a gain-maximal stationary policy (up to time T)
$\mathscr{C}_\pi^+$ set of recurrent state-action pairs (with finite return times) under stationary policy π
$\mathscr{X}_\pi$ set of disjoint recurrent cycles under stationary policy π
$\boldsymbol{\tau}_\pi(c, c')$ expected hitting time of state-action pair c' when starting from state-action pair c and following stationary policy π

From ergodic to communicating MDPs

In the previous section, we assumed that the Markov Decision Processes were **ergodic**, meaning that all policies share the same set of recurrent states, and that we can *easily* gain information about a policy while playing any other policy. This is no longer the case under the **communicating** only hypothesis where it is only assumed that for all couple of states, there exists a policy that connect the two in the desired direction with positive probability.

Under the **ergodic assumption**, all the policies are *connected* in the sense that playing policy π, one can test an action $a \neq \pi(s)$ in state s' almost whenever we want. If this action is the one played by policy π', then it means that we can gain information about the policy π' in that state, almost whenever we want. By whenever we want, it is meant that the state s is visited by policy π with positive probability due to the ergodic assumption and, therefore, the expected time before reaching this state by playing policy π is finite. Even better, the mixing in an ergodic Markov chain obey a power law. Therefore, if π is the empirical optimal policy, one can test any suboptimal policy in a state of interest at no travel cost since one can just play the optimal policy until the state if interest is reached, play the action of the suboptimal policy and resume playing the optimal policy. This is the method that was used in by the IMED-RL [73] algorithm.

[73]: Pesquerel et al. (2022), 'IMED-RL: Regret optimal learning of ergodic Markov decision processes'

Under the **communicating only assumption**, one must actively switch policies if we need to gather information on a specific part of the state space that is not in the recurrent space of the empirically optimal policy. Indeed, one can no longer rely on the ergodic assumption to wait a finite time and land on the desired state on which we wish to try a given action. This induces a non-trivial cost of travel outside the space of recurrent states of the optimal policy. On the other hand, the space of policies is heavily structured and one can still learn a lot from one policy while playing another policy. This suggests that efficient exploration outside the state support of the optimal policy can be performed efficiently and at a controlled cost, perhaps using an *as uniform as possible* exploration policy. Exploring these questions is the topic of this chapter.

8.1 Known dynamics model

Known dynamics: common in practice, uncommon in the literature

A standard assumption in most settings in RL is that the environment's dynamics is unknown, while the reward function may be known. This assumption is justified since state dynamics are not controlled by the agent, but is also in line with the argument that the difficulty of RL mainly stems from the unknown dynamics rather than unknown rewards. Consequently, the vast majority of existing regret minimizing algorithms in MDPs have some key ingredient, in design or analysis, to tackle unknown transition probabilities. In model-based algorithms (e.g., [79–81]), this is featured in the form of confidence sets around the empirical

[79]: Burnetas et al. (1997), 'Optimal adaptive policies for Markov decision processes'
[80]: Jaksch et al. (2010), 'Near-optimal regret bounds for reinforcement learning'
[81]: Filippi et al. (2010), 'Optimism in Reinforcement Learning and Kullback-Leibler Divergence'

transition distributions. In contrast, in some applications of RL, the agent has some prior knowledge on the transition function while most of the uncertainty is about the reward function. For instance, the agent may know the associated support of transitions, some transition probabilities, or the entire transition function up to some small deviation error. This could arise when the agent has access to an accurate estimate of the transition function that could be collected when performing another task on the same MDP (but with a different reward function). Access to an accurate estimate of the dynamics could also happen when the agent performs a task on an MDP, which differs from a previous task in the rewards only.

Recommender system In the context of personalized recommendation, where the rewards are given by a user based on internal evaluation of the recommendations, and the task (hence transitions) is fixed across users, it is natural to assume that based on previous interactions, the transitions of the system are perfectly known, but the rewards associated to the current user are unknown. Note that although rewards are provided by a user, this does not mean they are known, as evaluation at a point in time can be subjective and noisy.

Physical exploration Another scenario could arise in learning tasks where the dynamics are governed by some physical phenomena, which are perfectly known to the agent. This is the case for a significant portion of potential RL application. In the famous example of a garbage collector robot, transitions are in fact perfectly known but the reward of whether a bin is empty or full is unknown. The same applies to the robot that would explore Mars autonomously in order to look for life and resources. The dynamic of robotic movements, the local gravity, type of friction of the tires with the ground, etc. All of this is known or subject to little uncertainty. The uncertainty is in the reward of going some place or another, of drilling here or further away, etc. This motivates the framework of known dynamics.

In such scenarios, the following question arises naturally: ***What is the most statistically efficient way to perform exploration when the dynamics are known?***

Leveraging known dynamic with off-the-shelf algorithms

While any form of prior knowledge on the transition function do not appear directly advantageous to **model-free** algorithms, which is in line with their design principle, **model-based** algorithms can benefit directly from it. In particular, when the dynamics are perfectly known, most off-the-shelf algorithms can leverage this knowledge by simply removing the relevant confidence sets, which would lead to improved exploration, and hence, smaller regret bounds. For example, it is straightforward to show that UCRL2 [80], when equipped with the knowledge on dynamics, attains a regret bound of $\mathcal{O}(\sqrt{(SA + D)T \log(T)})$ with high probability, in any communicating MDP with S states, A actions per state, and diameter D.[1] In contrast, in the absence of prior knowledge, UCRL2 achieves a regret of $\mathcal{O}(DS\sqrt{AT \log(T)})$.

1: The diameter D of an MDP is the maximal expected time it takes to reach any state from any other state, and hence a good quantity to upper bound information gain as it gives a notion of the maximal expected time to reach a state from which we need to gain information from any other state the agent is currently at.

Despite such straightforward modifications of model-based algorithms, it still remains open as to what the best way is to incorporate such prior knowledge into algorithm design in a non-trivial manner, and whether it could lead to instance-dependent (and logarithmic) regret bounds. To our best knowledge, existing literature on learning in MDPs, albeit rich, fails to provide algorithmic ideas to leverage such prior knowledge in a statistically efficient way, and the potential gains thereof in terms of regret or sample complexity remain largely unexplored. No pun intended, **learning in MDPs with known dynamics remains largely unexplored in the literature.**

Where does this work fit?

We focus on regret minimization in communicating MDPs with known dynamics but unknown reward functions, and introduce a class of strategies called ***rarely-switching algorithms***, which provide a principled way to **leverage the connectivity structure** in the MDP through viewing the problem as a Bandit problem thanks to the prior knowledge on the dynamics. In this point of view, the arms are the policies and their expected reward are heavily related. Some policies are closer to others from an information gain point-of-view, which can be formalized in a manner that remains to be unveiled. The novel design of these introduced strategies considers **recurrent classes** induced by stationary policies as well as a carefully designed **stopping criterion** based on the said classes.

For these strategies, we present a generic regret bound, which relies on a novel regret decomposition by leveraging the structure, which could be of independent interest for learning and exploring in MDPs in general. Then, we instantiate a specific rarely-switching algorithm called IMED-KD, which uses the IMED strategy, a Bandit strategy [74] that we already encountered in this manuscript. IMED offers an interesting alternative to UCB-like strategies such as UCB or KL-UCB, and Bayesian strategies such as Thompson sampling. One of the main advantage of IMED compared to KL-UCB, both of them being proved optimal, is that for reward distributions belonging to exponential families of dimension one, the computation of the IMED index does not require solving an optimization problem unlike KL-UCB, nor sampling from Bayesian posterior, which makes it computationally appealing to the practitioner. However, we will experiment with these other possible choices in the experimental section while we do not investigate them theoretically. A key departure from existing IMED-style algorithms for MDPs, *e.g.* IMED-RL [73] that we presented in the previous chapter is to exploit the intrinsic structure of the MDP via use of a rarely-switching algorithm skeleton. In doing so, **IMED-KD relies on novel algorithmic ideas that consider recurrent classes induced by stationary policies as well as a carefully designed stopping criterion based on the said classes.**

[74]: Honda et al. (2015), 'Non-Asymptotic Analysis of a New Bandit Algorithm for Semi-Bounded Rewards'

[73]: Pesquerel et al. (2022), 'IMED-RL: Regret optimal learning of ergodic Markov decision processes'

Under some standard assumption on the reward function and MDP regularity, as well as a mild assumption on the hitting times (Assumption 8.6.1), we derive a non-asymptotic, problem-dependent, and logarithmic regret bound for IMED-KD, whose proof relies on the generic properties of rarely-switching strategies as well as proof machinery of IMED-style index but in the case of MDPs. Those proof techniques are interesting

in themselves and therefore included in this manuscript. We think that better understanding, from a statistical standpoint, the theoretical mechanisms that allows to gather information from one policy from another is key to advancing the field of Reinforcement Learning. **Per unit of interaction, we learn about a lot of policies and their connections.** Investigating ways to quantify this assertion is a rich topic that we think, deserve attention.

We further provide an efficient implementation and experiments illustrating its promising empirical performance. The source code is available on github, with the source code of all the papers presented in this manuscript.

To the best of our knowledge, IMED-KD is the first algorithm designed to leverage the structure in MDPs with known dynamics.

Related work While we presented in Chapter 6 how the work presented in this chapter fit in the existing RL literature, it is worth remarking that an MDP with known transitions, but unknown rewards may be viewed as a Bandit instance with highly structured actions, one action corresponding to a policy, in a way which is reminiscent of combinatorial Bandits [116, 117]. Despite such resemblance, the problem is more challenging as the agent is traversing an MDP without a resetting device. As a result, algorithmic ideas for combinatorial Bandits or those with generic structure [58, 118] do not directly carry over to MDPs with known dynamics. Furthermore, we highlight the fact that *a priori*, the expected reward of a policy in a communicating MDP is not immediately well-defined if we think about it as a real value. Indeed, it is a function of the state from which we start playing the policy since a policy can very well have several recurrent classes. The average reward is well-defined as a function $\mathbf{g}_\pi : S \to \mathbb{R}$. The best average reward of a policy π is then the maximum of this average reward function. It is the best one can achieve playing policy π if we were to be transported in one of its most advantageous states, $\operatorname{argmax} \mathbf{g}_\pi$. This state dependency and the inability of the learner to *reset* in an original state, as it is the case in combinatorial Bandit where the system can be thought to be brought back to its original position after each interaction, is the source of highly non-trivial technical difficulty.

[116]: Chen et al. (2013), 'Combinatorial multi-armed bandit: General framework and applications'
[117]: Combes et al. (2015), 'Combinatorial bandits revisited'
[58]: Combes et al. (2017), 'Minimal exploration in structured stochastic bandits'
[118]: Saber et al. (2020), 'Optimal Strategies for Graph-Structured Bandits'

8.2 Problem formulation

MDP under the average reward criterion

We formally introduce the problem and use this introduction to recall some of the notations.

We consider a Markov Decision Process $\mathbf{M} = (\mathcal{S}, \mathcal{A}, \mathbf{p}, \mathbf{r})$, where $\mathcal{S}$ is the set of states, and $\mathcal{A} = (\mathcal{A}_s)_{s\in\mathcal{S}}$, where $\mathcal{A}_s$ specifies the set of actions available in $s \in \mathcal{S}$. For convenience, we introduce the set of pairs $\mathcal{C} = \{(s, a) : s \in \mathcal{S}, a \in \mathcal{A}_s\}$. Further, $\mathbf{p} : \mathcal{C} \to \mathcal{P}(\mathcal{S})$ denotes the transition distribution function, and $\mathbf{r} : \mathcal{C} \to \mathcal{P}(\mathbb{R})$ the reward distribution function. We denote the corresponding mean reward function by $\mathbf{m} : \mathcal{C} \to \mathbb{R}$. Under the average reward criterion, an agent is interested in maximizing its cumulated rewards.

Policies Each stationary policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ acting on the MDP $\mathbf{M}$ induces a Markov chain $\mathbf{M}_\pi$ on $\mathscr{C}$, with corresponding transition probability $\mathbf{p}_\pi : \mathscr{C} \to \mathcal{P}(\mathscr{C})$, defined by $\mathbf{p}_\pi(s,a)(s',a') = \mathbf{p}(s'|s,a)\pi(a'|s')$. We denote $\overline{\mathbf{p}}_\pi : \mathscr{C} \to \mathcal{P}(\mathscr{C})$ the Cesàro-average of $\mathbf{p}_\pi$; formally,

$$\overline{\mathbf{p}}_\pi = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{p}_\pi^{t-1} \,.$$

This Cesàro-average of the transition kernel of the Markov $\mathbf{M}_\pi$ allows us to correctly define quantity related to average behavior because it erases periodic features. As such, it is a quantity that is more related to asymptotic properties of a policy rather than finite time, but so is the concept of average reward. Naturally, the gain of policy π, when starting from state-action pair $c_1 = (s_1, a_1)$, defined by

$$\mathbf{g}_{c_1,\pi} \coloneqq (\overline{\mathbf{p}}_\pi \mathbf{m})(c_1) = \sum_{c\in\mathscr{C}} \overline{\mathbf{p}}_\pi(c_1, c) \cdot \mathbf{m}(c),$$

where $\overline{\mathbf{p}}_\pi(c_1, c)$ is the visit probability mass the pair $c \in \mathscr{C}$ under policy π started in pair $c_1 \in \mathscr{C}$, and where $\mathbf{m}(c)$ is the average reward of the pair c. Given a *finite* set of stationary policies Π, we denote $\mathbf{g}_c^\star = \max_{\pi\in\Pi} \mathbf{g}_{c,\pi}$ the optimal gain starting from c, and $\Pi_c^\star = \{\pi \in \Pi : \mathbf{g}_{c,\pi} = \mathbf{g}_c^\star\}$ the set of policies achieving the optimal gain.

The online learning problem with known dynamic The framework of learning with known dynamics is very similar to the usual RL setting. More than the control setting. To clarify the problem and model, we quickly sum up the key points of this online learning problem. The learner interacts with MDP $\mathbf{M}$ for T time steps, starting in an initial state-action pair $(s_1, a_1) \in \mathscr{C}$. At each interaction, $t \in \mathbb{N}$, the learner is in state $s_t \in \mathcal{S}$ and chooses an action $a_t \in \mathcal{A}_{s_t}$ according to a stationary policy $\pi_t \in \Pi$, that is $a_t \sim \pi_t(s_t)$. Indeed, in our framework, a learner will be seen as playing a succession of policies, for an *a priori* random number of time, before switching. It is then possible that $\pi_t = \pi_{t+1}$ for several interactions. We assume that the learner ***does not know*** the reward distribution function $\mathbf{r}$, but ***knows*** the transition distribution function $\mathbf{p}$, and can compute $\overline{\mathbf{p}}_\pi$ for each $\pi \in \Pi$. While a theoretical sound assumption given the known dynamics hypothesis, it is not a matter to be considered lightly when it comes to numerical complexity given the size of the policy space. Fortunately, the structure of the policy space allows navigating from one policy to another at a lighter cost than what can be thought in the first place.

The learner chooses to follow the stationary policy π_t based on its observations so far. After an interaction, it receives a reward $r_t \in [0,1]$, where $r_t \sim \mathbf{r}(s_t, a_t)$ and a next state $s_{t+1} \in \mathcal{S}$ is sampled from the known transition distribution, *i.e.* $s_{t+1} \sim \mathbf{p}(\cdot|s_t, a_t)$. The sequence of chosen policies is denoted by $(\pi_t)_{t\geq 1}$, and simply by (π) when for all time step $t \geq 1$, $\pi_t = \pi$. We denote by $c_t = (s_t, a_t)$ the state-action pair at time step t. This notation $(\pi_t)_t$ is preferable to the one directly involving the sequence of state-action pairs because, as we said, we will take a Bandit approach on the problem. By directly using a notation that refers to a policy, we will be able to study event of the form $\min_{j>t} \{\pi_k \neq \pi \mid \pi_t = \pi\}$ that controls when the agent switch from one policy to another.

The learner's performance is measured through the notion of (expected) regret. Let $V_{\mathbf{M}}(\mathbf{A}, T)$ denote the cumulative reward of an algorithm $\mathbf{A}$ following a policy sequence $(\pi_t)_{t \leq T}$ up to time T:

$$V_{\mathbf{M}}(\mathbf{A}, T) = \mathbb{E}_{(\pi_t)}\left[\sum_{t=1}^{T} r_t\right] ,$$

where the expectation is implicitly over the MDP $\mathbf{M}$, *i.e.* takes into account both the transition kernel and reward distributions. When $\pi_t = \pi$ for all time step $t \geq 1$, it is simply denoted $V_{\mathbf{M}}((\pi), T)$. We recall that the (expected) regret with respect to playing a gain-optimal policy $\pi^\star$, up to time T, is defined as:

$$\mathcal{R}_{\mathbf{M}}(\mathbf{A}, T) = V_{\mathbf{M}}((\pi^\star), T) - V_{\mathbf{M}}(\mathbf{A}, T). \tag{8.1}$$

Remark on the regret and pseudo-regret While we take the point of view of Bandit on the policy space, we are still studying a Reinforcement Learning problem. Therefore, the notion of dynamics, impact the different notions of regret that can be of interest. As we saw in Chapter 6 and fortunately, these notions are asymptotically equivalent in the following sense. For each given T, the quantity $V^\star_{\mathbf{M}}(T) = \max_{\pi \in \Pi} V_{\mathbf{M}}((\pi), T)$ and set $\operatorname{argmax}_{\pi \in \Pi} V_{\mathbf{M}}((\pi), T)$ differ a priori from the cumulative reward of gain-optimal policies $(V_{\mathbf{M}}((\pi^\star), T))_{\pi^\star \in \Pi^\star}$ and set $\Pi^\star$, respectively. However,

$$\lim_{T \to \infty} V^\star_{\mathbf{M}}(T)/T = \lim_{T \to \infty} V_{\mathbf{M}}((\pi^\star), T)/T = \mathbf{g}^\star ,$$

for all gain-optimal stationary policy $\pi^\star \in \Pi^\star$. That is, the asymptotic maximal average reward coincides both with the asymptotic average reward of gain-optimal policies and the optimal gain. Since the set of considered stationary policies Π is bounded, this further implies that $\Pi^\star_T \subseteq \Pi^\star$ when horizon T is large enough, which also implies $V^\star_{\mathbf{M}}(T) - V_{\mathbf{M}}((\pi^\star), T) \overset{T \to \infty}{=} O(1)$.

Unveiling the graph-like structure between policies

On an MDP $\mathbf{M}$, each stationary policy $\pi \in \Pi$, induces a Markov chain $\mathbf{M}_\pi$ with transition kernel $\overline{\mathbf{p}}_\pi$. Those transition kernels $\overline{\mathbf{p}}_\pi$ can efficiently be grouped together by comparing their set of recurrent states and state-action pairs. The cardinality of the intersection between two sets of state-action pairs stemming from two different policies will be a good way to measure how much information can be gained on a policy from another policy.

Cycles

Definition 8.2.1 *The set of (positive) recurrent state-action pairs, i.e. pairs with finite return times under policy π is denoted $\mathcal{C}^+_\pi$ without explicit reference to the MDP $\boldsymbol{M}$ and defined as*

$$\mathcal{C}^+_\pi = \{c \in \mathcal{C} : \overline{\boldsymbol{p}}_\pi(c)(c) > 0\} . \tag{8.2}$$

The notion of recurrence allows to define an equivalence relation, denoted $\sim_\pi$, on $\mathscr{C}_\pi^+$. The relation $\sim_\pi$ such that

$$c \sim_\pi c' \Leftrightarrow \overline{\mathbf{p}}_\pi(c)(c') \cdot \overline{\mathbf{p}}_\pi(c')(c) > 0 \ .$$

Denoting $[c]_\pi$ the class of $c \in \mathscr{C}_\pi^+$ for relation $\sim_\pi$, the state-action pairs can be clustered using the quotient by the equivalence relation.

Definition 8.2.2 (Cycles) *The asymptotic cycles under policy π are denoted $\mathfrak{X}_\pi$ without explicit reference to the MDP $\boldsymbol{M}$ and defined as*

$$\mathfrak{X}_\pi = \mathscr{C}_\pi^+ / \sim_\pi = \left\{ [c]_\pi : \ c \in \mathscr{C}_\pi^+ \right\} . \tag{8.3}$$

Distinct elements of $\mathfrak{X}_\pi$ correspond to disjoint cycles. A policy π with $|\mathfrak{X}_\pi| = 1$ is called a *unichain* policy. In a unichain policy, a state is either recurrent in class $\mathfrak{X}_\pi$ or transient, *i.e.* it is visited a finite number of time and its asymptotic probability of visit is equal to zero. Interestingly, multichains policies can be decomposed into $|\mathfrak{X}_\pi|$ unichain policies, a fact that we use in our algorithm.

Remark 8.2.1 A remarkable property is that for a unichain policy π and recurrent $c' \in \mathscr{C}_\pi^+$, $\overline{\mathbf{p}}_\pi(c)(c')$ is independent on the starting pair c and equals $1/\boldsymbol{\tau}_\pi(c', c')$, where $\boldsymbol{\tau}_\pi(c', c')$ is the expected hitting time of c' when starting from c' and following policy π; see [75]. As a consequence, $\mathbf{g}_{c,\pi}$ also does not depend on c.

[75]: Puterman (1994), *Markov Decision Processes — Discrete Stochastic Dynamic Programming*

We consider the two following assumptions. The first is about the MDP regularity and the second is bout the reward distribution function.

Assumption 8.2.1 (MDP) *$\boldsymbol{M}$ is communicating, that is, $\forall c, c', \exists \pi, t \in \mathbb{N}$: $\boldsymbol{p}_\pi^t(c)(c') > 0$. Also, Π is proper, that is, the Cesàro-average $\overline{\boldsymbol{p}}_\pi$ of $\boldsymbol{p}_\pi$ exists for each $\pi \in \Pi$. There is a unique gain-optimal policy $\pi^\star \in \bigcap_{c \in \mathscr{C}} \Pi_c^\star$ that is unichain (i.e., it has a unique asymptotic cycle).*

While the unicity of the gain-optimal policy can probably be removed in the future, it prevents our algorithm (and others since it is a standard assumption) from oscillating between an optimal policy and another, possibly at the cost of non-zero traveling cost between two disjoint recurrent set of states. While not presented in this manuscript, I developed an algorithmic method based on a potential function to avoid switching too frequently between two gain-optimal policies with disjoint recurrent sets of states. Mixing gain-optimal policies in an ergodic MDP or communicating policies where all gain-optimal policies share the same set of recurrent states does not pose any problem. The unichain assumption is more of a convenient than necessary one since, as already said, one can always decompose a multichain policy into several unichain policy and ignore the original one after that.

Assumption 8.2.2 (Reward function) *For each $c \in \mathscr{C}$, the reward distribution $\boldsymbol{r}(c)$ is supported on $[0, 1]$ (in particular, it is $1/2$-sub-Gaussian), with bounded mean $\boldsymbol{m}(c) \in [0, 1)$.*

In particular, under Assumption 8.2.2, $\forall c \in \mathcal{C}, \pi \in \Pi$, the gain is bounded: $\mathbf{g}_{c,\pi} \in [0, 1)$. This assumption on reward is quite standard and allows using convenient concentration properties. Furthermore, it is not very restrictive from a practical standpoint.

Partial ordering on the graph-like structure of policies

Policy Iteration & Stochastic Gradient Ascent The idea that we want to use, while Bandit oriented in the end, is similar to the one we introduced when presenting the IMED-RL [73] algorithm. If all the expected values of the policies were independent, then, one would be considering a problem similar to the usual unstructured Bandit problem. Fortunately, expected rewards of policies are dependent and one can learn about a policy π' from a policy π that is *close enough*. In the ergodic case, close enough was all the policy π that were a modification of π' in only one state. This was because all the recurrent states of π' were reachable from such policies (and vice-versa) while being minimally modified. An interesting property was that, in this close neighborhood, there were a policy π with strictly larger gain than policy π' if the policy π' is not gain-optimal. This allowed us to perform a sort of stochastic gradient ascent on the graph of policies where two policies were connected if there were a one-state modification from each other. The complexity of computing indexes or quantities for each of the $|\Pi|$ policies was reduced to that of computing at most $S \times A$ indexes to know in which "direction" to move in order to make a policy improvement, but stochastically. In this work, we seek to exploit similar properties. To this purpose, we introduced the notion of **policy-improving neighborhood** that is inspired by the mentioned property which is also true for **unichain MDPs**, MDPs in which all policies are unichains.

[73]: Pesquerel et al. (2022), 'IMED-RL: Regret optimal learning of ergodic Markov decision processes'

Local monotony In *unichain* MDPs, there always exists a modification of a suboptimal policy in a single state having (strictly) larger gain [75]. We generalize this useful monotony property to *larger neighborhoods* as follows:

[75]: Puterman (1994), *Markov Decision Processes — Discrete Stochastic Dynamic Programming*

Assumption 8.2.3 (Policy-improving neighborhood) $\forall c \in \mathcal{C}, \pi \notin \Pi_c^\star$, $\exists \pi' \in \Pi,\ h(\pi, \pi') \leq k$, *such that* $\mathbf{g}_{c,\pi'} > \mathbf{g}_{c,\pi}$, *where* h *denotes the Hamming distance*[2] *between two policies.*

2: The Hamming distance is the number of states in which one must modify a policy to transform it into another.

Here, k is a given constant. Note that as k increases from 1 to S, Assumption 8.2.3 interpolates between (at least) all unichain MDPs, when $k = 1$, and all discrete MDPs, when $k = S$. For an MDP $\mathbf{M}$, we denote by $\mathcal{V}_\pi = \{\pi' \in \Pi :\ \mathsf{h}(\pi, \pi') \leq k\}$ the neighborhood of policy $\pi \in \Pi$ at radius k in Hamming distance h.

Remark 8.2.2 In a communicating MDP, for Π consisting of all stationary policies, the set of optimal policies does not depend on the starting pair and is simply denoted $\Pi^\star$. Moreover, an optimal policy is also (more precisely, can be made) unichain in this case. Note also that the gain of a unichain policy π does not depend on the starting state-action pair, and is simply denoted $\mathbf{g}_\pi$ (and $\mathbf{g}^\star$ for an optimal

policy); hence, we denote $\Pi_c^\star = \Pi^\star$ and $\mathbf{g}_c^\star = \mathbf{g}^\star$ for all c. Note, however, that for suboptimal policies $\pi \in \Pi$ that are not unichain, $\mathbf{g}_{c,\pi}$ may still depend on the initial state-action $c \in \mathscr{C}$.

8.3 Rarely-switching Algorithms

We choose to restrict the learner to follow a *rarely-switching* strategy, which forces the learner to keep playing the same policy until some criterion is met. The aim is to ensure a long enough visit in suboptimal state-subspaces of the MDP before returning to the optimal state-subspace of states that are recurrent under an optimal policy and avoid switching too much, not between policies, but between disjoint recurrent sets, making the agent pay the cost of travel. One cannot be sure to reset immediately within the recurrent set of the optimal policy after starting to play a suboptimal policy for exploration purposes, as it is the case under the ergodic assumption. This is why we consider learners that are **rarely-switching** as detailed in the following paragraph.

Rarely-switching learners

Interactions with the MDP are organized into episodes, numbered $k \in \{1, 2, \dots\}$, where the episode k starts at random times (indicating a policy switch) $\tau_{k-1} + 1$ and ends at random time τ_k (with $\tau_0 = 0$). The sum of the length of all the episodes until time T is equal to T. We denote $\mathcal{T}$ the sequence of switching times, *i.e.* the time index that end an episode and precede the beginning of a new one, $\mathcal{T} = (\tau_k)_{k\in\mathbb{N}}$. The end of an episode is controlled by a measurable random variable that we call `Event`. `Event` is a generic function of the current policy π and the history $h_{\tau+1:t}$ of all observations and decisions made from the beginning of the episode until the current time. Because it depends on the sequence of histories, it is more precisely described as a sequence of measurable functions, one for each size of the history. Since it informs the agent whether it should switch policy, the `Event` function is only a function from the past available information (and defined accordingly on the filtration generated by the interaction process). Therefore, for $\tau \in \mathcal{T}$, the learner starts at time step $\tau+1$ a new episode (to which we refer as "episode τ") after pulling state-action pair $c_{\tau+1} = (s_{\tau+1}, a_{\tau+1})$ and follows same policy $\pi = \pi_{\tau+1}$ until the event `Event` is triggered, after what the episode ends.

This simple procedure can be made into an Algorithm 22, of rather general form, that can then be used with different building blocks. For instance, the `Event` function and the procedure that is used line 13 to `compute a new policy` is purposefully left undefined for the moment. Rather, we prefer to ask and answer under what conditions of its building blocks, this type of algorithm satisfy desirable properties.

Algorithm 22: Rarely-switching learner

Input: $\mathbf{p}$, $(\mathbf{p}_\pi)_{\pi\in\Pi}$, $s_1 \in \mathcal{S}$ and `Event` function;

Select a starting stationary policy π;
Set $\pi_1 \leftarrow \pi$;
Start a new episode $\tau \leftarrow 0$, $\pi \leftarrow \pi_1$;
Pull action $a_1 \sim \pi(s_1)$;

for $t \in \mathbb{N}$ **do**
    Receive reward r_t;
    update history $h_t \leftarrow (s_t, \pi, a_t, r_t)$;
    if *not* `Event`$(\pi, h_{\tau+1:t})$ **then**
        Keep the same policy $\pi_{\tau+1} \leftarrow \pi$;
    else
        Start a new episode $\tau \leftarrow t$;
        Compute a new policy π;
        update $\pi_{\tau+1} \leftarrow \pi$;
    Pull action $a_{t+1} \sim \pi(s_{t+1})$;

Gathering information

Intuitively, `Event` function should be based on whether enough information was gathered using policy π before switching to another one, possibly departing from the state-subspace the agent currently is in. Remark that we use **information gather using policy** π and not *information gathered **on** policy* π. While the two can sometimes be similar, they are two different assertions. In particular the first phrasing explicitly refers to the fact that, while playing a policy π, not only do the agent gather information about that policy but also about other policies. Sometimes, one can more efficiently gather information about a policy π' by playing a policy π. This is the case if the agent want to sample a state-action pair that is recurrent under policy π' but with a small probability of visit. If we want to sample this state-action pair but fear the cost to pay due to the likely suboptimality of π', then one can play the policy π that visit the state action pair the fastest starting from the state the agent currently is in. Therefore, we see that the problem of gathering information about a policy is a different problem than the one from sampling that policy, *a priori*. Anticipating a bit, one can remark that state that visited by a policy with very small probability ϵ contributes to at most ϵ of the average reward of that policy and likely to be non-informative regarding the gain or optimality of that policy.

Since information is sampling and the `Event` function will be based on whether enough information has been collected in an episode, one should introduce some counting random variables, that we will later aggregate to form a suitable `Event` function.

Counters For a rarely-switching learner, let

$$N^{\mathrm{ini}}_{c,\pi}(0:T) = \sum_{\tau\in\mathcal{T}\cap[T]} \mathbb{1}\{\pi_{\tau+1}=\pi, c_{\tau+1}=c\}$$

denote, at the T interaction, the number of times an episode started in pair c and followed policy π. This quantity should not be confused with the possibly much larger number of visits

$$N_{c,\pi}(T) = \sum_{t\in[T]} \mathbb{1}\{\pi_t = \pi, c_t = c\}$$

of pair c by policy π until time T. In view of the introduction of `Event`, it is also convenient to introduce

$$N_c(h) = \sum_{(s,\pi,a,r)\in h} \mathbb{1}\{(s,a) = c\}$$

that counts the number of visits of pair c on the history h.

Regret decomposition of rarely-switching learners

Owing to the fact the criterion used to stop an episode is *independent* on the rewards accumulated during the episode, and using properties of the expectation, we can show the following decomposition Lemma 8.3.1, somewhat reminiscent of bandit analyses. We agree that this independence is probably a source of improvement since it would be *a priori* natural to take that information into account when deciding to switch from one policy to another. Rather, we consider the past information that is related to the number of visits and uses the counters introduced above.

The decomposition is made using the **average length** of an episode and **expected average reward** gathered by a policy π given an initial state-action pair and number of interactions. Quite naturally, if we play policy π for a number of interactions ℓ starting from state-action pair c, the expected gain from that episode will relate the aforementioned quantities.

Lemma 8.3.1 (Cumulative reward and regret decomposition) *The cumulative reward of a rarely-switching algorithm* **A** *until any last step* τ_k *of an episode* k, $T = \tau_k$, *satisfies*

$$V_M(\mathbf{A}, T) = \sum_{c\in\mathscr{C}} \sum_{\pi\in\Pi} \mathbb{E}_{(\pi_t)}\left[N^{ini}_{c,\pi}(0{:}T)\right] \cdot \mathbb{E}[\ell_{c,\pi}] \cdot G_{c,\pi}\,,$$

where

$$\ell_{c,\pi} = \min\{t > 0 : \texttt{Event}(\pi, h_{\tau+1:t}), c \in h_{\tau+1}\}$$

denotes the (random) length of the episode, and where

$$G_{c,\pi} = \mathbb{E}_{(\pi_t)}\left[\frac{1}{\ell_{c,\pi}} \sum_{t=1}^{\ell_{c,\pi}} r_t \middle| (s_1, a_1) = c\right]$$

denotes the expected average reward of an episode starting in pair c *and following policy* π.

When `Event` *further ensures an episode running policy* π *always stops in a*

same reference pair $c_\pi \in \mathcal{C}$, then writing $G^\star = G_{c_{\pi^\star},\pi^\star}$, it holds

$$\mathscr{R}_{\boldsymbol{M}}(\mathbf{A},T) = \sum_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} \mathbb{E}_{(\pi_t)}\left[N^{ini}_{c,\pi}(0:T)\right]\cdot\mathbb{E}[\ell_{c,\pi}]\cdot\left(G^\star - G_{c,\pi}\right) + \sum_{c\neq c_{\pi^\star}} \mathbb{E}_{(\pi_t)}\left[N^{ini}_{c,\pi^\star}(1:T)\right]\cdot\mathbb{E}[\ell_{c,\pi^\star}]\cdot\left(G^\star - G_{c,\pi^\star}\right). \tag{8.4}$$

Note that $N^{\text{ini}}_{c,\pi^\star}(1:T)$ excludes the first episode. Furthermore, we stress that $G^\star$ is defined using the stopping time induced by $\pi^\star$, and $G_{c,\pi}$ by the one induced by π.

The proof of Lemma 8.3.1 is provided in the next Section 8.4.

Sketch of proof

To give some intuition, Lemma 8.3.1 decomposes the cumulative reward of a learner according to *each configuration* when the current policy being played is π and the initial pair in this episode is c. Thus, it makes appear the number of times such a configuration happens, $\mathbb{E}_{(\pi_t)}\left[N^{\text{ini}}_{c,\pi}(0:T)\right]$, as well as the reward accumulated in that episode. A similar decomposition can be written for the optimal policy, and using a reference state ensures that it is $R_{\mathbf{M}}(\pi^\star,T) \simeq TG^\star$, up to the contribution of the first episode in which the episode may not start from $c_{\pi^\star}$. Combining the two cumulative reward decompositions yields the convenient form in Equation 8.4.

Further, the product form term $\mathbb{E}[\ell_{c,\pi}]\cdot G_{c,\pi}$ reveals that Lemma 8.3.1 offers a decoupling between the expected number $\mathbb{E}[\ell_{c,\pi}]$ of steps of an episode starting in c with policy π, and its averaged reward $G_{c,\pi}$ received during that episode. It is worth mentioning that the decoupling between the gain and the length of an episode holds by virtue of the Markov property and since we consider a decomposition in expectation.

Remark 8.3.1 (Simplifications) Note that $\mathbb{E}_{(\pi_t)}\left[N^{\text{ini}}_{c,\pi}(0:T)\right] = 0$ for policies π not explored by the rarely-switching algorithm. Typically, a learning algorithm will progressively focus on a few policies, and hence, the sum over all stationary policies π should effectively involve much fewer terms than A^S (*i.e.*, the number of all stationary deterministic policies). Interestingly, in the case of bandits, there is a unique state, and hence, Equation 8.4 simplifies to the classical regret decomposition, in which case the second term in the decomposition disappears

$$\sum_{c\neq c_{\pi^\star}} \mathbb{E}_{(\pi_t)}\left[N^{\text{ini}}_{c,\pi^\star}(1:T)\right]\cdot\mathbb{E}[\ell_{c,\pi^\star}]\cdot\left(G^\star - G_{c,\pi^\star}\right) = 0. \tag{8.5}$$

8.4 Regret decomposition for rarely-switching learners

As already said in the introduction, we include all the proof of the paper this chapter is based on because we think that studying how the space of policies is connected from an information gain and Reinforcement Learning standpoint will be key to tackle the problem of logarithmic regret in the full RL setting.

Regret decomposition for generic events

The proof of Lemma 8.3.1 is split into two parts. The first is concerned with the general decomposition and the second with the specific case where a reference stopping state-action pair is used for each policy.

Proof. Lemma 8.3.1

Part 1 Let us consider the (random) sequence increasing of episodes $(\tau_i)_{i=0\dots|\mathcal{T}|} \subset \mathcal{T}$. Then the duration of episode $\tau_i \in \mathcal{T}$ is $\tau_{i+1} - \tau_i$. We then define respectively the total duration and the average gain of policy $\pi \in \Pi$ started in state-action pair $c \in \mathcal{C}$ as

$$L_{c,\pi}(T) = \sum_{i:\tau_{i+1}\in[T]} \mathbb{1}\left\{c_{\tau_i+1} = c, \pi_{\tau_i+1} = \pi\right\}(\tau_{i+1} - \tau_i), \tag{8.6}$$

and

$$\hat{G}_{c,\pi}(T) = \frac{1}{L_{c,\pi}(T)} \sum_{i:\tau_{i+1}\in[T]} \mathbb{1}\left\{c_{\tau_i+1} = c, \pi_{\tau_i+1} = \pi\right\} \sum_{t=\tau_i+1}^{\tau_{i+1}} r_t,$$

Then, the cumulative reward rewrites,

$$\mathbb{E}_{(\pi_t)}\left[\sum_{t=1}^{T} r_t\right] = \sum_{c\in\mathcal{C}} \sum_{\pi\in\Pi} \mathbb{E}_{(\pi_t)}\left[L_{c,\pi}(T)\cdot \hat{G}_{c,\pi}(T)\right].$$

The Markov property implies $\mathbb{E}_{(\pi_t)}\left[\hat{G}_{c,\pi}(T)\middle|(c_\tau),(\pi_\tau),(\tau_i)\right] = G_{c,\pi}$, and from previous equality we have

$$\begin{aligned}\mathbb{E}_{(\pi_t)}\left[\sum_{t=1}^{T} r_t\right] &= \sum_{c\in\mathcal{C}} \sum_{\pi\in\Pi} \mathbb{E}_{(\pi_t)}\left[L_{c,\pi}(T)\,\mathbb{E}_{(\pi_t)}\left[\hat{G}_{c,\pi}(T)\middle|(c_\tau),(\pi_\tau),(\tau_i)\right]\right] \\ &= \sum_{c\in\mathcal{C}} \sum_{\pi\in\Pi} \mathbb{E}_{(\pi_t)}[L_{c,\pi}(T)]\, G_{c,\pi}\,. \end{aligned} \tag{8.7}$$

Similarly, the Markov property implies $\mathbb{E}_{(\pi_t)}\left[\tau_{i+1} - \tau_i\middle|(c_\tau),(\pi_\tau)\right] = \boldsymbol{\ell}_{c_{\tau_i+1},\pi_{\tau_i+1}}$. This implies for all $c \in \mathcal{C}$ and for all $\pi \in \Pi$,

$$\begin{aligned}\mathbb{E}_{(\pi_t)}[L_{c,\pi}(T)] &= \mathbb{E}_{(\pi_t)}\left[\mathbb{E}_{(\pi_t)}\left[L_{c,\pi}(T)\middle|(c_\tau),(\pi_\tau)\right]\right] \\ &= \mathbb{E}_{(\pi_t)}\left[\sum_{i\geq 0} \mathbb{1}\left\{c_{\tau_i+1} = c, \pi_{\tau_i+1} = \pi\right\}\boldsymbol{\ell}_{c,\pi}\right] \\ &= \mathbb{E}_{(\pi_t)}\left[\sum_{i\geq 0} \mathbb{1}\left\{c_{\tau_i+1} = c, \pi_{\tau_i+1} = \pi\right\}\right]\boldsymbol{\ell}_{c,\pi} \\ &= \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi}(0:T)\right]\boldsymbol{\ell}_{c,\pi}\,, \end{aligned} \tag{8.8}$$

where we recall that $N^{\mathrm{ini}}_{c,\pi}(0:T)$ is the (random) number of episodes with starting state-action pair c and followed stationary policy π.

We conclude the proof of the cumulative reward decomposition by combining Equations (8.7) and (8.8).

Part 2 We now turn to the decomposition of the regret. To this end, we apply the same decomposition to an optimal policy $\pi^\star$. Then since the event `Event` stops in the same reference state c_π for a policy π by assumption, then except possibly for the first episode, all next episodes under $\pi^\star$ start in the same reference state. This enables us to use the following Markov regeneration property from Lemma 8.5.1. Hence, we obtain that

$$\mathbb{E}_{(\pi^\star)}\left[\sum_{t=1}^{T} r_t\right] = \mathbb{E}_{(\pi^\star)}[\mathbb{1}\left\{c_1 \neq c^\star\right\} \ell_{c_1,\pi^\star} \cdot G_{c_1,\pi^\star}] + \mathbb{E}_{(\pi^\star)}\left[N^{\mathrm{ini}}_{c^\star,\pi^\star}(1{:}T)\right] \cdot \mathbb{E}[\ell_{c^\star,\pi^\star}] \cdot G^\star\,, \tag{8.9}$$

$$T = \mathbb{E}_{(\pi^\star)}[\mathbb{1}\left\{c_1 \neq c^\star\right\} \ell_{c_1,\pi^\star}] + \mathbb{E}_{(\pi^\star)}\left[N^{\mathrm{ini}}_{c^\star,\pi^\star}(1{:}T)\right] \cdot \mathbb{E}[\ell_{c^\star,\pi^\star}]\,, \tag{8.10}$$

where $G^\star = G_{c^\star,\pi^\star}$ and c_1 is generated from $\pi^\star$. In particular, both Equations (8.9) and (8.10) imply

$$\mathbb{E}_{(\pi^\star)}\left[\sum_{t=1}^{T} r_t\right] = \mathbb{E}_{(\pi^\star)}[\mathbb{1}\left\{c_1 \neq c^\star\right\} \ell_{c_1,\pi^\star} \cdot \left(G_{c_1,\pi^\star} - G^\star\right)] + T\,G^\star\,. \tag{8.11}$$

Note that, taking limits as $T \to \infty$, we recover that indeed $G^\star = \mathbf{g}^\star$.

Thus, from the cumulative reward decomposition and previous Equation (8.11), and using that, on the other hand $T = \sum_{\substack{c\in\mathcal{C}\\ \pi\in\Pi}} \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi}(0{:}T)\right] \cdot \mathbb{E}[\ell_{c,\pi}]$, we obtain

$$\mathcal{R}_{\mathbf{M}}(\mathbf{A},T) = \mathbb{E}_{(\pi^\star)}[\mathbb{1}\left\{c_1 \neq c^\star\right\} \ell_{c_1,\pi^\star} \cdot \left(G_{c_1,\pi^\star} - G^\star\right)] + \sum_{\substack{c\in\mathcal{C}\\ \pi\in\Pi}} \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi}(0{:}T)\right] \cdot \mathbb{E}[\ell_{c,\pi}] \cdot \left(G^\star - G_{c,\pi}\right). \tag{8.12}$$

At this point, focusing on the second term, we remark that

$$\begin{aligned}\sum_{\substack{c\in\mathcal{C}\\ \pi\in\Pi}} \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi}(0{:}T)\right] \cdot \mathbb{E}[\ell_{c,\pi}] \cdot \left(G^\star - G_{c,\pi}\right) &= \sum_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi}(0{:}T)\right] \cdot \mathbb{E}[\ell_{c,\pi}] \cdot \left(G^\star - G_{c,\pi}\right) \\ &\quad + \sum_{c\neq c^\star} \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi^\star}(0{:}T)\right] \cdot \mathbb{E}[\ell_{c,\pi^\star}] \cdot \left(G^\star - G_{c,\pi^\star}\right), \end{aligned} \tag{8.13}$$

from which we deduce that

$$\begin{aligned}\mathcal{R}_{\mathbf{M}}(\mathbf{A},T) &= \sum_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi}(0{:}T)\right] \cdot \mathbb{E}[\ell_{c,\pi}] \cdot \left(G^\star - G_{c,\pi}\right) \\ &\quad + \mathbb{E}_{(\pi^\star)}[\mathbb{1}\left\{c_1 \neq c^\star\right\} \ell_{c_1,\pi^\star} \cdot \left(G_{c_1,\pi^\star} - G^\star\right)] + \sum_{c\neq c^\star} \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi^\star}(0{:}T)\right] \cdot \mathbb{E}[\ell_{c,\pi^\star}] \cdot \left(G^\star - G_{c,\pi^\star}\right)\,. \end{aligned}$$

To better control the last term we first isolate the first episode from the next ones. Indeed, the first episode may be a bit special, compared to next episodes, as we do not necessary start from a reference state for the current policy. Using the definition of $N^{\mathrm{ini}}_{c,\pi^\star}(0{:}T)$, we isolate the first episode and write

$$\begin{aligned}N^{\mathrm{ini}}_{c,\pi^\star}(0{:}T) &= \mathbb{1}\left\{\pi_1 = \pi^\star, c_1 = c\right\} + \sum_{\tau\in\mathcal{T}\cap[T],\tau>0} \mathbb{1}\left\{\pi_{\tau+1} = \pi^\star, c_{\tau+1} = c\right\} \\ &= \mathbb{1}\left\{\pi_1 = \pi^\star, c_1 = c\right\} + N^{\mathrm{ini}}_{c,\pi^\star}(1{:}T)\,. \end{aligned}$$

This leads to a first interesting reduction. Indeed, we then realize that

$$\begin{aligned}\mathscr{R}_{\mathbf{M}}(\mathbf{A},T) &= \sum_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} \mathbb{E}_{(\pi_t)}\big[N^{\mathrm{ini}}_{c,\pi}(0{:}T)\big]\cdot\mathbb{E}[\ell_{c,\pi}]\cdot\big(G^\star - G_{c,\pi}\big)\\ &\quad + \mathbb{E}_{(\pi^\star)}[\mathbb{1}\left\{c_1\neq c^\star\right\}\ell_{c_1,\pi^\star}\cdot\big(G_{c_1,\pi^\star}-G^\star\big)]\\ &\quad + \sum_{c\neq c^\star}\mathbb{E}_{(\pi_t)}\big[\mathbb{1}\left\{\pi_1=\pi^\star, c_1=c\right\}\big]\cdot\mathbb{E}[\ell_{c,\pi^\star}]\cdot\big(G^\star - G_{c,\pi^\star}\big)\\ &\quad + \sum_{c\neq c^\star}\mathbb{E}_{(\pi_t)}\Big[N^{\mathrm{ini}}_{c,\pi^\star}(1{:}T)\Big]\cdot\mathbb{E}[\ell_{c,\pi^\star}]\cdot\big(G^\star - G_{c,\pi^\star}\big)\ ,\end{aligned}$$

in which the second and third term telescope, owing to the fact that since c_1 is fully determined under $\pi^\star$, then

$$\sum_{c\neq c^\star}\mathbb{E}_{(\pi_t)}\big[\mathbb{1}\left\{\pi_1=\pi^\star, c_1=c\right\}\big]\cdot\mathbb{E}[\ell_{c,\pi^\star}]\cdot\big(G^\star - G_{c,\pi^\star}\big) = \mathbb{E}_{(\pi_t)}\big[\mathbb{1}\left\{\pi_1=\pi^\star, c_1\neq c^\star\right\}\cdot\ell_{c_1,\pi^\star}\cdot\big(G^\star - G_{c_1,\pi^\star}\big)\big]\ .$$

Hence, we obtain

$$\begin{aligned}\mathscr{R}_{\mathbf{M}}(\mathbf{A},T) &= \sum_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} \mathbb{E}_{(\pi_t)}\big[N^{\mathrm{ini}}_{c,\pi}(0{:}T)\big]\cdot\mathbb{E}[\ell_{c,\pi}]\cdot\big(G^\star - G_{c,\pi}\big)\\ &\quad + \sum_{c\neq c^\star}\mathbb{E}_{(\pi_t)}\Big[N^{\mathrm{ini}}_{c,\pi^\star}(1{:}T)\Big]\cdot\mathbb{E}[\ell_{c,\pi^\star}]\cdot\big(G^\star - G_{c,\pi^\star}\big)\ ,\end{aligned}\tag{8.14}$$

thus completing the proof. □

8.5 Expected finite time average reward and gain

Gain and regeneration property

Gain One may wonder about the link between $G_{c,\pi}$ and the gain $\mathbf{g}_{c,\pi}$. Indeed, $G_{c,\pi}$ can be seen as a proxy for the gain $\mathbf{g}_{c,\pi}$ of the policy, since

$$\mathbf{g}_{c,\pi} = \lim_{T\to\infty} \mathbb{E}_{(\pi_t)}\left[\frac{1}{T}\sum_{t=1}^{T} r_t \middle| c_1 = c, \pi_1 = \pi\right],$$

that is, as $\ell_{c,\pi} \to \infty$ then $G_{c,\pi}$ indeed approaches $\mathbf{g}_{c,\pi}$. This interpretation is however valid only when $\ell_{c,\pi}$ is sufficiently large. Luckily, thanks to the regenerating properties of the chain, if we start and stop an episode in the same *recurrent* pair c_π, hence "completing a loop", then the average of the rewards received during that episode must, in expectation, equal that of infinitely many such loops. More formally, we define the **regeneration property**.

Lemma 8.5.1 (Regeneration property) *For any unichain policy π, any recurrent reference pair $c_\pi \in \mathcal{C}_\pi^+$, and any function* `Event` *ensuring that an episode always stops in c_π when we play π, then $G_{c_\pi,\pi} = \boldsymbol{g}_{c_\pi,\pi}$, that is the expected averaged reward received during an episode starting and ending at pair c_π is equal to the gain of the policy.*

Proof. Lemma 8.5.1 We denote by (π, c_π) the constant policy (π) started at reference pair c_π. (π, c_π) can be seen as a rarely-switching policy such that $\pi_{\tau+1} = \pi$ at each new episode τ. Since the episode starts and stops in c_π, its length is a multiple of the recurrent time of c_π when playing π. Note that the multiple can be larger than 1 has we may require visiting c_π several times during an episode. Also, by definition of reference pair c_π, for all pair $c \neq c_\pi$, $L_{c,\pi}(T) = 0$ and $L_{c_\pi,\pi}(T) = T$ (see Equation (8.6)).

Hence, from Equation (8.7) we have

$$\mathbb{E}_{(\pi,c_\pi)}\left[\sum_{t=1}^{T} r_t\right] = T \cdot G_{c_\pi,\pi}\,.$$

This implies in particular

$$\mathbf{g}_{c_\pi,\pi} \coloneqq \lim_{T\to\infty} \mathbb{E}_{(\pi)}\left[\frac{1}{T}\sum_{t=1}^{T} r_t \,\middle|\, c_1 = c_\pi\right] = \lim_{T\to\infty} \frac{1}{T}\,\mathbb{E}_{(\pi,c_\pi)}\left[\sum_{t=1}^{T} r_t\right] = G_{c_\pi,\pi}\,.$$

□

This motivates us to introduce for each π a reference pair

$$c_\pi \in \operatorname*{argmax}_{c\in\mathcal{C}} \overline{\mathbf{p}}_\pi(c)(c)$$

which belongs to $\mathcal{C}_\pi^+$ by construction, and define $\texttt{Event}(\pi, h_{\tau+1,t})$ to ensure that $(s_t, a_t) = c_\pi$.

Indeed, this choice of c_π also minimizes $\boldsymbol{\tau}_\pi(c,c)$ over c, hence tends to reduce the expected episode length $\mathbb{E}[\ell_{c_\pi,\pi}]$. This construction of events further yields the following useful control on the regret

Proposition 8.5.2 (Rarely-switching learners with reference pair) *Under Assumption 8.2.1, if the rarely-switching learner* $\mathbf{A}$ *specifies for each* π *to stop the episode starting with* π *in the same reference pair* $c_\pi \in \mathscr{C}_\pi^+$*, then the following bound holds almost surely:*

$$\sum_{c\neq c_{\pi^\star}} N^{ini}_{c,\pi^\star}(1{:}T) \leq \sum_{c\in\mathscr{C}} \sum_{\pi\neq\pi^\star} N^{ini}_{c,\pi}(0{:}T)\,. \tag{8.15}$$

Moreover, the cumulative regret $\mathscr{R}_{\boldsymbol{M}}(\mathbf{A},T)$ *of any such rarely-switching algorithm* $\mathbf{A}$ *with respect to the unique optimal policy* $\pi^\star$*, up to the end* T *of any episode, is upper-bounded by*

$$\mathbb{E}_{(\pi_t)}\left[\sum_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} N^{ini}_{c,\pi}(0{:}T)\right]\cdot\left(\max_{(c,\pi)\neq(c_{\pi^\star},\pi^\star)} \mathbb{E}[\ell_{c,\pi}](G^\star - G_{c,\pi}) + \boldsymbol{B}_\star\right), \tag{8.16}$$

where $\boldsymbol{B}_\star := \max_{c\neq c_{\pi^\star}} \mathbb{E}_{(\pi_t)}[\ell_{c,\pi^\star}](G^\star - G_{c,\pi^\star})$ *is a problem-dependent quantity.*

Remark 8.5.1 It holds $\mathbf{B}_\star \leq \max_{(c,\pi)\neq(c_{\pi^\star},\pi^\star)} \mathbb{E}[\ell_{c,\pi}](G^\star - G_{c,\pi})$. Further, $\mathbf{B}_\star = 0$ for bandits.

Proof. of Proposition 8.5.2 We prove an upper bound on the regret by making appear the gaps $G_{c^\star,\pi^\star} - G_{c,\pi}$, for $c\in\mathscr{C}$, $\pi\in\Pi$, using Lemma 8.3.1. We recall that $c^\star$ is the unique state-action pair of reference of the unique optimal stationary strategy $\pi^\star$ (Assumption 8.2.1).

The key property we use is that when the episode always stops in the same reference pair for a given policy, since $\pi^\star$ is unichain, then except for the first episode, an episode under $\pi^\star$ that does not start in state-action pair of reference $c^\star$ implies that the stationary policy from the previous episode differs from $\pi^\star$. This shows that

$$\sum_{c\neq c_{\pi^\star}} N^{\mathrm{ini}}_{c,\pi^\star}(1{:}T) \leq \sum_{c\in\mathscr{C}} \sum_{\pi'\neq\pi^\star} N^{\mathrm{ini}}_{c,\pi'}(0{:}T)\,. \tag{8.17}$$

Introducing $\overline{\ell} = \max_{(c,\pi)\neq(c^\star,\pi^\star)} \mathbb{E}[\ell_{c,\pi}]$ and $\overline{\Delta} = \max_{(c,\pi)\neq(c^\star,\pi^\star)} G^\star - G_{c,\pi}$. By combining Equation (8.4) and previous Equation (8.17), we prove the upper bound on the regret

$$\mathscr{R}_{\mathbf{M}}(\mathbf{A},T) \leq \mathbb{E}_{(\pi_t)}\left[\sum_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} N^{\mathrm{ini}}_{c,\pi}(0{:}T)\right] \times\left(\max_{(c,\pi)\neq(c_{\pi^\star},\pi^\star)} \mathbb{E}[\ell_{c,\pi}](G^\star - G_{c,\pi}) + \max_{c\neq c_{\pi^\star}} \mathbb{E}[\ell_{c,\pi^\star}](G^\star - G_{c,\pi^\star})\right).$$

□

Estimation and covering time

Before we specify the algorithm, let us remind that since the transitions are known, only the mean rewards need to be estimated.

Since $\mathbf{g}_{c,\pi} = \sum_{c' \in \mathcal{C}_\pi^+} \overline{\mathbf{p}}_\pi(c)(c')\mathbf{m}(c')$, where $\mathbf{m}$ is unknown, it is natural to collect observations of pairs $c' \in \mathcal{C}_\pi^+$ to estimate the corresponding $\mathbf{m}(c')$, and hence the gain $\mathbf{g}_\pi$. A natural way to ensure the estimation error reduces in each episode is to stop an episode when all pairs in $\mathcal{C}_\pi^+$ have been visited at least once: Formally, $\min_{c' \in \mathcal{C}_\pi^+} N_{c'}(h_{\tau+1,t}) > 0$, that is after *covering* the set $\mathcal{C}_\pi^+$. In order to control the resulting episode length, unfortunately, there is in general no simple control of the cover time by a policy π of its recurrent pairs. The policy could be diffusive or lazy (see next Section 8.6), yielding an arbitrarily large cover time. Formally, given $C \subset \mathcal{C}$ and $c \in \mathcal{C}$, we denote by $\pi_c^H(C)$ a policy that minimizes over policies π the expected time $\boldsymbol{\tau}_{c,\pi}^H(C)$ to reach *any element* of C starting from c and following π. Similarly, we let $\overline{\pi}_c(C)$ denote a policy minimizing over π the expected time $\overline{\boldsymbol{\tau}}_{c,\pi}(C)$ to cover *all elements* of C starting from c and following π. Recall that the diameter of a finite MDP $\mathbf{M}$ is defined as $D_{\mathbf{M}} = \max_{s \neq s'} \min_\pi \mathbb{E}[T^\pi(s, s')]$, where $T^\pi(s, s')$ denotes the number of steps it takes to reach s' starting from s and following policy π [80]. Letting $D_{\mathbf{M}}$ denote the diameter of $\mathbf{M}$, it holds: $\min_\pi \boldsymbol{\tau}_{c,\pi}^H(C) \leq D_{\mathbf{M}}$ and $\overline{\boldsymbol{\tau}}_{c,\pi}(C) \leq |C| D_{\mathbf{M}}$ for all c, C. In contrast, $\overline{\boldsymbol{\tau}}_{c,\pi}(\mathcal{C}_\pi^+)$ could be *arbitrarily large*, even for a gain-optimal policy π.

8.6 Cover times and episode lengths

In this section, we provide a few illustrative examples that highlight the challenges of having a long enough episode on the one hand, while ensuring the cover time of recurrent pairs is controlled.

Deterministic MDP and policy

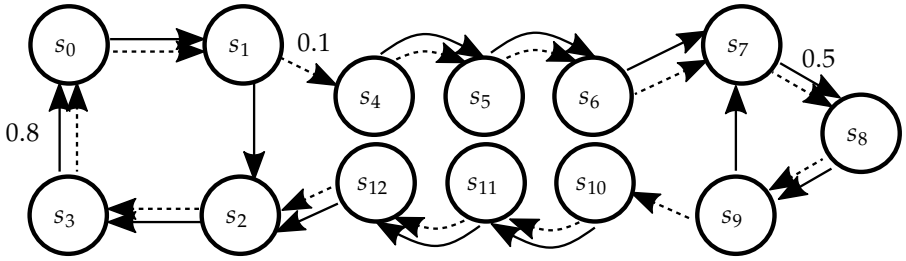

Figure 8.1: A deterministic MDP with two actions (solid/dashed line) per state, deterministic transitions, and sparse rewards.

Long enough episode To give intuition about what it means to have a sufficiently large episode length, consider the example of Figure 8.1 depicting an MDP (here deterministic for illustration purpose) with two actions (solid/dashed transitions). In this case, starting from state s_0 following solid actions yields a cycle of length 4 and dashed actions a cycle of length 13. An episode smaller than the length of the cycle will not

result in a good approximation of $\mathbf{g}_{c,\pi}$. On the other hand, the average gain is equal to the expected average reward on the cycle starting and ending in s_0 (due to the Markov property hence the regenerative property at state s_0). So, a good estimation of it can be obtained by completing exactly at least one full cycle and using the corresponding average reward to update the estimate.

Stochastic system and deterministic policy

Covering time of diffusive policies While in deterministic systems, the covering time of $\mathscr{C}_\pi^+$ by policy π would be of order the cardinal of $\mathscr{C}_\pi^+$, there is some difficulty when considering a stochastic system: Indeed, take the case of a diffusion process as illustrated in Figure 8.2. In this case, starting from s_0 with a policy π always playing up, $\mathscr{C}_\pi^+$ contains all pairs (s, up) with $s \in \mathcal{S}$. However, reaching the states s_k or s_{-k} takes time exponential in their distance k to s_0, which is undesirable. At the same time, the contribution of these states to the gain is much smaller than that of s_0 as their return frequency is also much smaller than that of s_0.

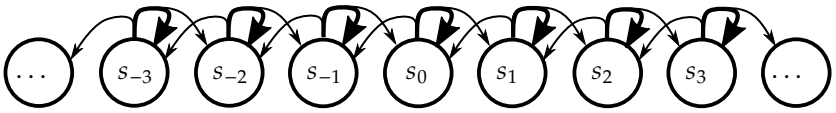

Figure 8.2: Diffusion process: Action up (solid line) has high probability $1 - 2\epsilon$ to self-loop (thick arrow), and ϵ probability (thin arrows) to go left or right, causing an arbitrary large covering time.

Covering time of lazy chains Now, let us consider the case of an MDP $\mathbf{M}$ and a deterministic policy π that induces the following chain for some small $\epsilon > 0$:

$$\mathbf{p}_\pi(s'|s) = \begin{cases} 1 - \epsilon & \text{if } s' = s \\ \frac{\epsilon}{S-1} & \text{else} \end{cases}.$$

Note that such situation can happen for an optimal, or near-optimal policy, that cycles on (a subset of) states in a lazy way. In such a chain, all states s are recurrent and asymptotically visited at same frequency, yielding $\overline{\mathbf{p}}_\pi((s, \pi(s))) = \frac{1}{S}$. Hence, $|\mathscr{C}_{c,\pi}^+(\eta)| = S$ for $\eta \in [0, 1/S)$. On the other hand, we observe that it takes about $1/\epsilon$ steps to move from one state to another one, due to high probability of self-loop $1 - \epsilon$ that makes the chain lazy. Further, it is easily shown that the expected time to cover the set, starting from any s is $O(S \ln(S)/\epsilon)$. Hence, for each recurrent c', $\overline{\boldsymbol{\tau}}_{c',\pi}(\mathscr{C}_{c,\pi}^+(\eta)) = O(S \ln(S)/\epsilon)$, which can be made arbitrarily large independently on the values of $\overline{\mathbf{p}}_\pi$. This shows that there is in general no direct control of the covering time of a set as a function of the asymptotic visiting probabilities.

Frequently recurrent pairs

Frequently recurrent pairs and restricted gain This motivates us to discard states with *too small return frequency*. Because the expected rewards are bounded, a small return frequency also mean a small contribution to the reward, hence motivating us further to discard those states if they hinder the learning of the agent without adding statistical significance to

the estimation of quantities of interest. To formalize this, we introduce a notion of gain, which we call η-*restricted gain*, defined using a parameter $\eta \in \mathbb{R}^+$. Formally, for a constant $\eta \in \mathbb{R}^+$, define the set of **frequently recurrent pairs** of a stationary policy π:

$$\mathcal{C}^+_{c,\pi}(\eta) := \{c' \in \mathcal{C} : \overline{\mathbf{p}}_\pi(c)(c') > \eta\}\,, \tag{8.18}$$

which leads to defining the corresponding η**-restricted gain** function:

$$\mathbf{g}_{c,\pi}(\eta) := \sum_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} \overline{\mathbf{p}}_\pi(c)(c') \cdot \mathbf{m}(c') / \sum_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} \overline{\mathbf{p}}_\pi(c)(c')\,. \tag{8.19}$$

We further naturally introduce

$$\mathbf{g}^\star_c(\eta) = \max_{\pi \in \Pi} \mathbf{g}_{c,\pi}(\eta)$$

and

$$\Pi^\star_c(\eta) = \operatorname*{argmax}_{\pi \in \Pi} \mathbf{g}_{c,\pi}(\eta)\,.$$

Note that for $\eta = 0$, we recover the usual definitions, for instance, $\mathbf{g}_{c,\pi}(0) = \mathbf{g}_{c,\pi}$. More generally, we have the following lemma.

Lemma 8.6.1 (Restricted-gain approximation)

$$\forall \pi, c, \eta, \quad g_{c,\pi} - g_{c,\pi}(\eta) \leq \eta m_{\max} |\mathcal{C}^+_{c,\pi} \setminus \mathcal{C}^+_{c,\pi}(\eta)|\,, \tag{8.20}$$

where $m_{\max} = \max_{c \in \mathcal{C}} m(c)$ *is the maximal state-action pair mean.*

Proof. Lemma 8.6.1 Indeed, it holds that

$$\mathbf{g}_{c,\pi} - \mathbf{g}_{c,\pi}(\eta) = \sum_{c' \notin \mathcal{C}^+_{c,\pi}(\eta)} \underbrace{\overline{\mathbf{p}}_\pi(c,c')}_{\leq \eta} \cdot \underbrace{\mathbf{m}(c')}_{\leq \mathbf{m}_{\max}} - \underbrace{\sum_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} \overline{\mathbf{p}}_\pi(c,c') \cdot \mathbf{m}(c') \frac{\sum_{c' \notin \mathcal{C}^+_{c,\pi}(\eta)} \overline{\mathbf{p}}_\pi(c,c')}{\sum_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} \overline{\mathbf{p}}_\pi(c,c')}}_{\geq 0}\,.$$

□

In particular for a given ϵ, choosing $\eta \leq \frac{\epsilon}{\mathbf{m}_{\max} |\mathcal{C}^+_{c,\pi} \setminus \mathcal{C}^+_{c,\pi}(\eta)|}$, for instance $\eta = \epsilon/(\mathbf{m}_{\max} S)$, ensures that the gain is still well-approximated by the restricted gain up to the desired precision ϵ. Hence, we can restrict to cover $C = \mathcal{C}^+_{c,\pi}(\eta)$ instead of $\mathcal{C}^+_{c,\pi}$ and define $\texttt{Event}(\pi, h_{\tau+1,t})$ accordingly. Unfortunately, $\overline{\boldsymbol{\tau}}_{c,\pi}(\mathcal{C}^+_\pi(\eta))$ can still be arbitrary in general, hence we introduce

Definition 8.6.1 (Lazyness) *A chain induced by* π *is* (B, η)*-lazy if* $\max_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} \overline{\boldsymbol{\tau}}_{c',\pi}(\mathcal{C}^+_{c,\pi}(\eta)) > B$.

To control the length of the episodes, we make the no-laziness Assumption 8.6.1 on the studied MDPs. The laziness constant B may be unknown

to the agent, or computed offline thanks to the known dynamics of the MDP.

Assumption 8.6.1 (No-laziness) *$\boldsymbol{M}$ has no (B,η)-lazy chain, where $\eta \in [0,1]$ is given.*

Covering time, information gain, structure of policy space

Structure of policies We conclude this section by showing that choosing this specific form of event further enables to revisit the decomposition of regret to better exploit structure of the policies. Indeed, while $\mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi}(0:T)\right] = 0$ for policies π not explored by a rarely-switching learner, there is more: policies are structured, in the sense that visiting one state-action pair (s,a) is not only informative about the actual policy π playing a in state s, but all such ones as well. Using Proposition 8.5.2 and the form of stopping event introduced in Lemma 8.7.1, we derive the following result, showing, remarkably, that the **s**um over all policies can be removed in favor of a maximum.

Theorem 8.6.2 (Rarely-switching learners exploiting recurrence structure) *Let* **A** *be a rarely-switching algorithm using stopping event*

$$\texttt{Event}(\pi, h_{\tau+1,t}) = \{\min_{c'\in C} N_{c'}(h_{\tau+1,t}) > 0 \, and \, (s_t,a_t) = c_\pi\}$$

where $C = \mathscr{C}^+_{c,\pi}(\eta)$ is parameterized by η.

Then,

$$\sum_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} N^{ini}_{c,\pi}(0:T) \leq |\mathscr{C}| \max_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} \boldsymbol{N}^{\eta}_{c,\pi}(T)\,, \tag{8.21}$$

where we introduced $\boldsymbol{N}^{\eta}_{c,\pi}(t) = \min_{c'\in\mathscr{C}^+_{c,\pi}(\eta)} N_{c'}(h_{1:t})$.

In particular, using Remark 8.5.1,

$$\begin{aligned}\frac{\mathscr{R}_{\boldsymbol{M}}(\mathbf{A},T)}{\mathbb{E}_{(\pi_t)}\left[\max\limits_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} \boldsymbol{N}^{\eta}_{c,\pi}(T)\right]} &\leq |\mathscr{C}|\Bigg(\underbrace{\max_{(c,\pi)\neq(c_{\pi^\star},\pi^\star)} \mathbb{E}[\ell_{c,\pi}]}_{=:\boldsymbol{L}} \underbrace{(G^\star - G_{c,\pi})}_{\in[-1,1]} + \boldsymbol{B}_\star\Bigg)\\ &\leq 2|\mathscr{C}|\boldsymbol{L}\end{aligned} \tag{8.22}$$

Proof. Theorem 8.6.2 In order to prove the upper bound on the regret we prove an upper on the total number of episodes under a suboptimal stationary strategy, that is $\sum_{c\in\mathscr{C},\ \pi\neq\pi^\star} \mathbb{E}_{(\pi_t)}\left[N^{\mathrm{ini}}_{c,\pi}(0:T)\right]$.

Let us consider the number of pulls

$$\mathbf{N}^{\eta}_{c,\pi}(t) = \min_{c'\in\mathscr{C}^+_{c,\pi}(\eta)} N_{c'}(h_{1:t})$$

associated with stationary policy $\pi\in\Pi$ started in state-action pair $c\in\mathscr{C}$.

When considering rarely-switching algorithms, due to the definition stopping event, all state-action pairs $c' \in \mathscr{C}^+_{c,\pi}(\eta)$ are visited at each episode started in state-action pair c under policy π. This implies

$$\forall c' \in \mathscr{C}, \quad \sum_{\substack{c \in \mathscr{C} \\ \pi \neq \pi^\star}} \mathbb{1}\left\{c' \in \mathscr{C}^+_{c,\pi}(\eta)\right\} N^{\mathrm{ini}}_{c,\pi}(0{:}T) \leq N_{c'}(h_{1:T})\,. \tag{8.23}$$

By considering the definition of the associated numbers of pulls and introducing the argmin set $\underline{\mathscr{C}}^+_{c,\pi}(\eta, T) = \underset{c' \in \mathscr{C}^+_{c,\pi}(\eta)}{\operatorname{argmin}} N_{c'}(h_{1:T}) \subset \mathscr{C}^+_{c,\pi}(\eta)$ in previous Equation (8.23), it holds $\forall c' \in \mathscr{C}$,

$$\begin{aligned}\sum_{c \in \mathscr{C}, \pi \neq \pi^\star} \mathbb{1}\left\{c' \in \underline{\mathscr{C}}^+_{c,\pi}(\eta, T)\right\} N^{\mathrm{ini}}_{c,\pi}(0{:}T) &\leq \sum_{\substack{c \in \mathscr{C} \\ \pi \neq \pi^\star}} \mathbb{1}\left\{c' \in \mathscr{C}^+_{c,\pi}(\eta)\right\} N^{\mathrm{ini}}_{c,\pi}(0{:}T) \\ &\leq N_{c'}(h_{1:T})\end{aligned}\,, \tag{8.24}$$

where $\mathbf{N}^\eta_{c,\pi}(T) = N_{c'}(h_{1:T})$ by construction when $c' \in \underline{\mathscr{C}}^+_{c,\pi}(\eta, T)$. This implies $\forall c' \in \mathscr{C}$,

$$\sum_{c \in \mathscr{C},\ \pi \neq \pi^\star} \mathbb{1}\left\{c' \in \underline{\mathscr{C}}^+_{c,\pi}(\eta, T)\right\} N^{\mathrm{ini}}_{c,\pi}(0{:}T) \leq \max_{\substack{c \in \mathscr{C} \\ \pi \neq \pi^\star}} \mathbf{N}^\eta_{c,\pi}(T)\,. \tag{8.25}$$

Then, summing over $c' \in \mathscr{C}$, and using that $|\underline{\mathscr{C}}^+_{c,\pi}(\eta, T)| \geq 1$, it comes

$$\begin{aligned}\sum_{c \in \mathscr{C},\ \pi \neq \pi^\star} N^{\mathrm{ini}}_{c,\pi}(0{:}T) &= \sum_{c' \in \mathscr{C}} \sum_{\substack{c \in \mathscr{C} \\ \pi \neq \pi^\star}} \frac{\mathbb{1}\left\{c' \in \underline{\mathscr{C}}^+_{c,\pi}(\eta, T)\right\}}{|\underline{\mathscr{C}}^+_{c,\pi}(\eta, T)|} N^{\mathrm{ini}}_{c,\pi}(0{:}T) \\ &\leq |\mathscr{C}| \max_{\substack{c \in \mathscr{C} \\ \pi \neq \pi^\star}} \mathbf{N}^\eta_{c,\pi}(T)\end{aligned}\,. \tag{8.26}$$

□

8.7 The IMED-KD strategy

In the previous section, we discuss in detail the `Event` function, how to efficiently and meaningfully control the length of an episode and how its is related to the regret of a rarely-switching learner. Exploiting all the assumptions about the MDP, we showed the interesting and important Proposition 8.5.2 and Theorem 8.6.2 that controls the regret upper bound of a rarely-switching algorithm exploiting the notion of reference pairs as well as the structures between the recurrent sets.

Intuition about IMED-KD

In this section, we present IMED-KD (Indexed Minimum Empirical Divergence for MDPs with Known Dynamics), which is a rarely-switching algorithm that uses an IMED-type index together with the knowledge of $\mathbf{p}$ to attain a logarithmic regret in communicating MDPs. The IMED strategy [74] has been proven asymptotically optimal in stochastic Bandits and is computationally appealing when compared with the optimistic

[74]: Honda et al. (2015), 'Non-Asymptotic Analysis of a New Bandit Algorithm for Semi-Bounded Rewards'

KL-UCB or the Bayesian Thompson sampling strategy that require, at each step, solving an optimization problem or sampling from a posterior, respectively. Although posterior sampling can be made efficient for some parametric distributions, *e.g.* Gaussian, current extensions of TS to MDPs require introducing a forced optimism mechanism [103], which makes it less appealing both from theory and computational perspectives.

[103]: Agrawal et al. (2017), 'Optimistic posterior sampling for reinforcement learning: worst-case regret bounds'

At a high level, the algorithm computes at the beginning of each episode τ an empirical best candidate policy $\hat{\pi}^\star_\tau$, as well as a best informative policy $\hat{\pi}^I_\tau$. The algorithm considers the stopping event targeting $C = \mathscr{C}^+_{c,\pi}(\eta)$ and final pair $c_0 = c_\pi$ for the policy $\pi = \hat{\pi}^I_\tau$. It runs the episode using $\pi^H_{c_\tau}(C)$ until hitting C, followed by policy π (so if $c_\tau \in C$, this reduces to running π). We now detail the computation of $\hat{\pi}^\star_\tau$, $\hat{\pi}^I_\tau$ and π^H_c.

IMED-KD algorithm

Empirical best policy

$\hat{\pi}^\star_\tau$ is computed by applying a value (or policy) iteration procedure (VI) to the MDP $\hat{\mathbf{M}}_\tau = (\mathcal{S}, \mathcal{A}, \mathbf{p}, \hat{\mathbf{r}}_\tau)$ where for each $c \in \mathscr{C}$, we introduce $\hat{\mathbf{r}}_\tau(c) = \mathcal{N}(\hat{\mathbf{m}}_\tau(c), \sigma^2)$ with $\hat{\mathbf{m}}_\tau(c) = \frac{1}{N_c(h_\tau)} \sum_{t'=1}^{\tau} r_{t'} \mathbb{1}_{c_{t'} = c}$ being the classical empirical estimate of the mean $\mathbf{m}(c)$ computed on observations received until time τ.

Informative policy

To compute $\hat{\pi}^I_\tau$, we first form $\hat{\mathbf{g}}^\star_{c,\tau}(\eta) = \hat{\mathbf{g}}_{c,\hat{\pi}^\star_\tau,\tau}(\eta)$, where for each policy π, we introduced its η-restricted gain estimate defined by

$$\hat{\mathbf{g}}_{c,\pi,\tau}(\eta) = \frac{\sum_{c' \in \mathscr{C}^+_{c,\pi}(\eta)} \overline{\mathbf{p}}_\pi(c)(c') \hat{\mathbf{m}}_\tau(c')}{\sum_{c' \in \mathscr{C}^+_{c,\pi}(\eta)} \overline{\mathbf{p}}_\pi(c)(c')}.$$

We further introduce for each policy the notation $\mathbf{N}_\pi(\tau) = \mathbf{N}^\eta_{c_\tau,\pi}(\tau)$ and the IMED-type index, inspired from [74] for Bandits,

$$I_\tau(\pi) = \mathbf{N}_\pi(\tau) \mathsf{d}\Big(\hat{\mathbf{g}}_{c_\tau,\pi,\tau}(\eta) \mid \hat{\mathbf{g}}^\star_{c_\tau,\tau}(\eta)\Big) + \log(\mathbf{N}_\pi(\tau)),$$

where $\mathsf{d}(x \mid y) = \frac{(x-y)^2}{2\sigma^2} = 2(x-y)^2$ denotes the Kullback-Leibler divergence between Gaussian distributions with respective means x and y, and identical standard deviation $\sigma = 1/2$. This is justified since under Assumption 8.2.2, all gains fall in $[0,1]$ hence can be considered $1/2$-sub-Gaussians. Finally, we let $\hat{\pi}^I_\tau$ (also written $\tilde{\pi}_{\tau+1}$) be a policy minimizing I_τ over a subset of policies $\Pi_\tau \subset \Pi$ containing $\hat{\pi}^\star_\tau$. Following Assumption 8.2.3, we introduce $\mathscr{V}_{\hat{\pi}^\star_\tau}(k) = \{\pi : \mathsf{h}(\hat{\pi}^\star_\tau, \pi) \le k\}$, and define Π_τ such that $\mathscr{V}_{\hat{\pi}^\star_\tau} \subset \Pi_\tau$. We discuss choices of Π_τ in Section 8.12.

Exploratory policy

To compute the fast hitting policy $\hat{\pi}^H_\tau = \pi^H_c(C)$ that tries to reach $C = \mathscr{C}^+_{c_\tau,\hat{\pi}^I_\tau}(\eta)$ as fast as possible starting from $c = c_\tau$ we introduce a

specific MDP $\mathbf{M}_\tau^H = (\mathcal{S}, \mathcal{A}, \mathbf{p}, \mathbf{r}_\tau^H)$ with modified reward function

$$\mathbf{r}_\tau^H(c) = \begin{cases} 1 & \text{if } c \in C \\ 0 & \text{else} \end{cases}.$$

We compute an optimal policy for this MDP, under the average reward criterion, using value iteration. This policy is used to reach the class C and ensures the hitting time is always finite.

Strategy Finally, we define IMED-KD to be the rarely-switching algorithm with update rule (line 11 of Algorithm 22) given by choosing at each new episode $\tau \in \mathcal{T}$ the policy

$$\pi_{\tau+1} = \pi^H_{c_\tau}(\mathscr{C}^+_{c_\tau, \hat{\pi}^I_\tau}(\eta)) \qquad \text{followed by } \hat{\pi}^I_\tau\,,$$

with stopping event

$$\texttt{Event}(\pi_{\tau+1}, h_{\tau+1,t}) = \{\min_{c' \in \mathscr{C}^+_{c_\tau, \hat{\pi}^I_\tau}(\eta)} N_{c'}(h_{\tau+1,t}) > 0 \text{ and } (s_t, a_t) = c_{\hat{\pi}^I_\tau}\}\,.$$

We provide the following control on the length of episodes run with IMED-KD, whose proof is given in after providing and proving Lemma 8.7.2 that is concerned about controlling the episode length for any rarely-switching algorithm using a specific stopping event.

Lemma 8.7.1 (Bound on episode lengths) *Assuming $\boldsymbol{M}$ has diameter $D_{\boldsymbol{M}}$ and no (B, η)-lazy chain, the expected length of an episode of IMED-KD started at τ satisfies*

$$\mathbb{E}[\ell_{c,\pi} | h_{1:\tau}, c_\tau = c] \leq D_{\boldsymbol{M}} + 2B. \tag{8.27}$$

In general, without further assumption on the structure of the MDP or laziness of its chains, we can prove the following.

Lemma 8.7.2 (Episode length) *Assume that for some subset $C \subset \mathscr{C}$ and target $c_0 \in C$, the stopping event is of the form $\texttt{Event}(\pi, h_{\tau+1,t}) = \{\min_{c' \in C} N_{c'}(h_{\tau+1,t}) > 0$ and $(s_t, a_t) = c_0\}$. Then, the following holds*

$$\boldsymbol{\ell}_{c,\pi}(\eta) \leq \begin{cases} \boldsymbol{\tau}_\pi(c, c_0) & \text{if } C = \emptyset \\ \sum_{c' \in C} \boldsymbol{\tau}_\pi(c, c') + \mathbb{1}\{c' \neq c_0\}\, \boldsymbol{\tau}_\pi(c', c_0) & \text{else.} \end{cases} \tag{8.28}$$

where $\boldsymbol{\tau}_\pi(c, c')$ denotes the expected first (random) passage time from state-action pair $c \in \mathscr{C}$ to state-action pair $c' \in \mathscr{C}$ in the Markov Process $M_\pi = (\mathscr{C}, P_\pi)$.

Proof. Lemma 8.7.2 Let us consider $\tau_\pi(c, c')$ the first (random) passage time from state-action pair $c \in \mathscr{C}$ to state-action pair $c' \in \mathscr{C}$ in the Markov process $M_\pi = (\mathscr{C}, P_\pi)$, whose expectation is $\boldsymbol{\tau}_\pi(c, c')$. For a set $C \subset \mathscr{C}$, the first (random) cover time of C, denoted by $\tau_\pi(c, C)$, corresponds to the first time when all elements of C have been visited at least once. Ordering the elements of C from the (random) first element $c_{(1)}$ visited in

C to the (random) last one $c_{(|C|)}$, we have $\tau_\pi(c, c_{(1)}) < \cdots < \tau_\pi(c, c_{(|C|)})$, where $\tau_\pi(c, C)$ coincides with $\tau_\pi(c, c_{(|C|)})$, that is we have $\tau_\pi(c, C) = \max_{c' \in C} \tau_\pi(c, c')$. Note that this quantity is very different from and should not be confused with $\max_{c' \in \mathscr{C}} \boldsymbol{\tau}_\pi(c, c')$, which can be much smaller than the expected covering time $\mathbb{E}[\tau_\pi(c, C)]$. Now, considering the set C to be non-empty, we thus introduce the (random) state-action pair

$$c_{\eta,\pi} = \operatorname*{argmax}_{c' \in C} \tau_\pi(c, c')$$

such that for all state-action pair $c' \in C$, $\tau_\pi(c, c') \leq \tau_\pi\big(c, c_{\eta,\pi}\big)$. By construction of the stopping event `Event`, if $c_{\eta,\pi} = c_0$ then the episode stops immediately, otherwise one has to wait to reach c_0 from $c_{\eta,\pi}$, that is $\tau_\pi(c_{\eta,\pi}, c_0)$ many steps. Hence, under any rarely-switching algorithm using such event, the expected duration of an episode started in state-action pair c under the policy π is given by the following expression

$$\begin{aligned} \boldsymbol{\ell}_{c,\pi}(\eta) &= \mathbb{E}_{(\pi)}\big[\tau_\pi\big(c, c_{\eta,\pi}\big) + \mathbb{1}\left\{c_{\eta,\pi} \neq c_0\right\} \tau_\pi\big(c_{\eta,\pi}, c_0\big)\big] \\ &\leq \mathbb{E}_{(\pi)}\left[\max_{c' \in C} \tau_\pi(c, c') + \mathbb{1}\left\{c' \neq c_0\right\} \tau_\pi(c', c_0)\right] \end{aligned}$$

Upper-bounding the maximum over c' by a sum over the possible c', and using that $\mathbb{E}_{(\pi)}[\tau_\pi(c, c')] = \boldsymbol{\tau}_\pi(c, c')$ this implies

$$\boldsymbol{\ell}_{c,\pi}(\eta) \leq \sum_{c' \in C} \Big(\boldsymbol{\tau}_\pi(c, c') + \mathbb{1}\left\{c' \neq c_0\right\} \boldsymbol{\tau}_\pi(c', c_0)\Big)\,. \tag{8.29}$$

Now, if the set C is empty then the episode stops when c_0 is reached, that is $\boldsymbol{\ell}_{c,\pi}(\eta) \leq \mathbb{E}_{(\pi)}[\tau_\pi(c, c_0)] = \boldsymbol{\tau}_\pi(c, c_0)$. □

Now, we provide the control of the length of episodes for IMED-KD, under the assumption that there are no (B, η) lazy chains.

Proof. Lemma 8.7.1 IMED-KD runs a policy that first reaches $\mathscr{C}^+_{c,\hat{\pi}^I_\tau}$ as fast as possible, but then simply run the policy $\hat{\pi}^I_\tau$. Hence, it takes at most $D_{\mathbf{M}}$ expected steps to reach $\mathscr{C}^+_{c,\hat{\pi}^I_\tau}$ but then B many steps to cover the set $\mathscr{C}^+_{c,\hat{\pi}^I_\tau}$ under Assumption 8.6.1, and at most B more steps to reach the reference pair $c_{\hat{\pi}^I_\tau}$. This proves that

$$\mathbb{E}_{(\pi_t)}[\ell_{c,\pi} | h_{1:\tau}, c_\tau = c] \leq D_{\mathbf{M}} + 2B\,.$$

□

8.8 IMED-KD: Regret upper bound

In this section, we provide performance bounds of the IMED-KD strategy. We first provide the following non-asymptotic control on the number of visits of suboptimal policies. Bounding this quantity is similar to bounding the number of time suboptimal arms have been sampled in a highly structured Bandit problem. In structured Bandits, it is sometimes easier to bound the number of times a group of arms has been sampled

rather than bounding the number of times an individual suboptimal arm has been sampled due to the structure that create a non-trivial dependency between the number of samples. This is what was done in Chapter 5 concerned about Bandits with groups of similar arms [70]. Another common proof technique in structured Bandits is to bound directly the regret without first bounding the number of pulls of suboptimal arms. However, such an approach is usually more difficult and less tractable.

[70]: Pesquerel et al. (2021), 'Stochastic bandits with groups of similar arms'

Theorem 8.8.1 (Performance bound of IMED-KD) *For an MDP $\boldsymbol{M}$ with diameter $D_{\boldsymbol{M}}$ and satisfying Assumptions 8.2.1 (communicating with unique unichain optimal policy),8.2.2 (bounded rewards),8.2.3 (a policy improvement is possible in a k-neighborhood),8.6.1 (no lazy policies), the IMED-KD strategy ensures, provided that*

$$\eta < \frac{\epsilon_{\boldsymbol{M}}(0)}{2\boldsymbol{m}_{\max} S}$$

where

$$\epsilon_{\boldsymbol{M}}(\eta) = \min_{\substack{c\in\mathscr{C}\\ \pi\notin\Pi^\star}} \left\{ \max_{\pi'\in\mathscr{V}_\pi} g_{c,\pi'}(\eta) - g_{c,\pi}(\eta) \right\} ,$$

the following bound on the number of pulls of suboptimal policies,

$$\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} \boldsymbol{N}^\eta_{c,\pi}(T) \right] \leq \max_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} \frac{(1+\alpha_{\boldsymbol{M}}(\epsilon))\log(T)}{d\big(g_{c,\pi}(\eta)\big|g^\star_c(\eta)\big)} + K_T(\epsilon,\eta)(D_{\boldsymbol{M}}+2B) ,$$

for all accuracy $0<\epsilon<\dfrac{\epsilon_{\boldsymbol{M}}(\eta)}{2}$, where $\lim_{\epsilon\to 0}\alpha_{\boldsymbol{M}}(\epsilon)=0$ and

$$K_T(\epsilon,\eta) \leq \frac{5\,|\mathscr{C}|\,e^{2\epsilon^2}}{2\epsilon^2} + |\mathscr{C}|\left(1 + c^{-1}_{\epsilon_{\boldsymbol{M}}(\eta)} + 2C_{\epsilon_{\boldsymbol{M}}(\eta)}\sqrt{\log(c_{\epsilon_{\boldsymbol{M}}(\eta)}T)}\right) ,$$

where C_ϵ and c_ϵ are constants that are independent of $\mathbf{M}$ and T.

This bound can be combined with the regret decomposition described Equation 8.22 and the fact that $G_{c,\pi} \leq 1$ from the bounded reward assumption to obtain an upper bound on the regret $\mathscr{R}_{\mathbf{M}}(\text{IMED-KD}, T)$ of IMED-KD.

Theorem 8.8.2 (Regret upper bound of IMED-KD) *For an MDP $\boldsymbol{M}$ with diameter $D_{\boldsymbol{M}}$ and satisfying Assumptions 8.2.1 (communicating with unique unichain optimal policy),8.2.2 (bounded rewards),8.2.3 (a policy improvement is possible in a k-neighborhood),8.6.1 (no lazy policies), given the result of Theorem 8.8.1, the regret of the IMED-KD strategy,*

$$\mathscr{R}_{\boldsymbol{M}}(\textit{IMED-KD}, T)$$

is upper bounded by

$$\left[\max_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} \frac{(1+\alpha_{\boldsymbol{M}}(\epsilon))\log(T)}{d\big(g_{c,\pi}(\eta)\big|g^\star_c(\eta)\big)} + K_T(\epsilon,\eta)(D_{\boldsymbol{M}}+2B) \right] \cdot 2(D_{\boldsymbol{M}}+2B)\,|\mathscr{C}|\,. \tag{8.30}$$

The regret bound in Equation 8.30 of Theorem 8.8.2 grows logarithmically with T, where the leading constant is determined by a notion of gap

with respect to η-restricted gains. In this respect, the bound bears some similarity with logarithmic regret bounds for Bandits, which is consistent with the design principle behind the rarely switching algorithms which viewed the MDP as a multi-policy Bandit. In contrast, the bound in Equation 8.30 is inversely proportional to the square of the gap terms, which stems from the technical difficulties arising in the regret analysis in the average-reward setting. We note that [80] report a logarithmic regret bound for UCRL2 that depends on a similar notion of gap term. However, their bound involves an additive term, which depends on mixing time quantities and has an implicit dependence on $\log(T)$, and hence it could grow very large. In empirical evaluation of UCRL2, it is often witnessed that the logarithmic regime in the regret actually kicks in after very long burn-in phase. While Equation 8.30 offers a bound with an optimal dependence on T, it is not clear whether the gap in terms of policy gains, appearing in both Equation 8.30 and [80], is the best one could get. Indeed, let us recall that regret *lower* bounds for average-reward MDPs, beyond the ergodic class, are open and deriving them even for the case of known dynamics is a very interesting, yet challenging, topic of future research.

[80]: Jaksch et al. (2010), 'Near-optimal regret bounds for reinforcement learning'

Adaptive laziness parameter

Given the constraint on the parameter η of Theorem 8.8.1, that is

$$\eta < \frac{\epsilon_{\mathbf{M}}(0)}{2\mathbf{m}_{\max} S}$$

where

$$\epsilon_{\mathbf{M}}(\eta) = \min_{\substack{c \in \mathscr{C} \\ \pi \notin \Pi^\star}} \left\{ \max_{\pi' \in \mathscr{V}_\pi} \mathbf{g}_{c,\pi'}(\eta) - \mathbf{g}_{c,\pi}(\eta) \right\} ,$$

a natural question is how to ensure η is small enough since $\epsilon_{\mathbf{M}}(0)$ is a priori unknown. One possible way to accommodate this is to consider near-optimality instead, with given precision $\tilde{\epsilon}$, and simply choose $\eta = \tilde{\epsilon}/2S$. In practice, we may choose η adaptively (*i.e.*, $\eta = \eta_t$); we discuss in the experimental Section 8.13 some simple adaptive choices of η, and demonstrate that they lead to promising empirical performance, though not directly covered by Theorem 8.8.1.

We now move on to the proof of the regret upper bound of the rarely-switching IMED-KD strategy.

8.9 IMED-KD: Finite Time Analysis

Outline of the proof

At a high level, the key interesting step of the proof is to realize that the considered algorithm implies empirical lower and empirical upper bounds on the numbers of pulls (see Lemma 8.9.1, Lemma 8.9.2). Then, based on concentration lemmas (see Section 8.10), the algorithm-based empirical lower bounds ensure the reliability of the estimators of interest (Lemma 8.9.4). Interestingly, this makes use of arguments based on recent concentration of measure that enable to control the concentration without

[34]: Cappé et al. (2013), 'Kullback–Leibler upper confidence bounds for optimal sequential allocation'

adding some log log bonus (such a bonus was required for example in the initial analysis of the KL-UCB strategy from [34]). Then, combining the reliability of these estimators with the obtained algorithm-base empirical upper bounds, we obtain upper bounds on the average numbers of pulls (Theorem 8.8.1). Interestingly, most of the proof is agnostic to the length of an episode (that is handled separately). We only use the property that the algorithm guarantees in each episode an increase by at least one of the number of pulls of each η-recurrent pair.

8.9.1 Notations

We recall that for an MDP $\mathbf{M}$, we denote by $\mathcal{V}_\pi = \{\pi' \in \Pi : \mathsf{h}(\pi, \pi') \leq k\}$ the neighborhood of policy $\pi \in \Pi$ at radius k in Hamming distance h. For constant $\eta \geq 0$, we let

$$\epsilon_{\mathbf{M}}(\eta) = \min_{\substack{c \in \mathscr{C} \\ \pi \notin \Pi^\star}} \left\{ \max_{\pi' \in \mathcal{V}_\pi} \mathbf{g}_{c,\pi'}(\eta) - \mathbf{g}_{c,\pi}(\eta) \right\} . \tag{8.31}$$

According to policy-improvement property 8.2.3, $\epsilon_{\mathbf{M}}(0) > 0$. Then, from Lemma 8.6.1, provided that $\eta < \epsilon_{\mathbf{M}}(0)/(2\mathbf{m}_{\max} S)$, then we also have $\epsilon_{\mathbf{M}}(\eta) > 0$. Furthermore, the following inequality holds

$$\epsilon_{\mathbf{M}}(\eta) \leq \min_{\substack{c \in \mathscr{C} \\ \pi \notin \Pi^\star}} \left\{ \mathbf{g}_{c,\pi^\star}(\eta) - \mathbf{g}_{c,\pi}(\eta) \right\} .$$

Note that this value of η also ensures that

$$\Pi^\star \coloneqq \operatorname*{argmax}_{\pi \in \Pi} \max_{c \in \mathscr{C}} \mathbf{g}_{c,\pi} = \operatorname*{argmax}_{\pi \in \Pi} \max_{c \in \mathscr{C}} \mathbf{g}_{c,\pi}(\eta) .$$

Indeed, $\epsilon_{\mathbf{M}}(0) \leq \min_{\substack{c \in \mathscr{C} \\ \pi \notin \Pi^\star}} \left\{ \max_{\pi' \in \Pi} \mathbf{g}_{c,\pi'} - \mathbf{g}_{c,\pi} \right\}$, which ensures η-reduction does not modify the best policy.

Then, there exists a function $\alpha_{\mathbf{M}}(\cdot)$ with $\lim_{\epsilon \to 0} \alpha_{\mathbf{M}}(\epsilon) = 0$ such that for all $0 \leq \epsilon < \epsilon_{\mathbf{M}}(\eta)/2$, for all state-action pair $c \in \mathscr{C}$, for all suboptimal stationary policy $\pi \notin \Pi^\star$,

$$\mathsf{d}\Big(\mathbf{g}_{c,\pi}(\eta) + \epsilon \mid \mathbf{g}^\star_c(\eta) - \epsilon\Big) \leq (1 + \alpha_{\mathbf{M}}(\epsilon))^{-1} \, \mathsf{d}\Big(\mathbf{g}_{c,\pi}(\eta) \mid \mathbf{g}^\star_c(\eta)\Big) . \tag{8.32}$$

8.9.2 Algorithm-based empirical bounds

The IMED-KD algorithm implies inequalities between the indexes that can be rewritten as inequalities on the numbers of pulls. While lower bounds involving $\log(t)$ (or $\log(\tau)$) may be expected in view of the asymptotic regret bounds, we show lower bounds on the numbers of pulls involving instead $\log\Big(\mathbf{N}^\eta_{c_{\tau+1}, \tilde{\pi}_{\tau+1}}(\tau)\Big)$ the logarithm of the number of pulls of the current index policy. We also provide upper bounds on $\mathbf{N}^\eta_{c_{\tau+1}, \tilde{\pi}_{\tau+1}}(\tau)$ involving $\log(\tau)$. We believe that establishing these empirical lower and upper bounds is a key element of our proof technique, that is of independent interest and is the motivation for including it in the manuscript.

In the sequel, for notational convenience and avoid cluttering the notations. We denote $N_\pi(\tau)$ in lieu of $\mathbf{N}^\eta_{c_{\tau+1},\pi}(\tau)$. We further write $\hat{g}_\pi(\tau)$ for $\hat{\mathbf{g}}_{c_{\tau+1},\pi,\tau}(\eta)$ and $\hat{g}^\star(\tau)$ for $\hat{\mathbf{g}}^\star_{c_{\tau+1},\tau}(\eta)$. Last, we denote by $\mathcal{T}$ the set of starting times, that is the set of time steps that start a new episode.

Remark 8.9.1 According to IMED-KD algorithm, $\hat{\pi}^\star_\tau \in \mathcal{V}_{\hat{\pi}^\star_\tau} \subset \Pi_\tau$.

Lemma 8.9.1 (Empirical lower bounds) *Under IMED-KD, for all starting time $\tau \in \mathcal{T}$, for all stationary policy $\pi \in \Pi_\tau$,*

$$\log(N_{\tilde{\pi}_{\tau+1}}(\tau)) \le N_\pi(\tau)\, d\big(\hat{g}_\pi(\tau) \mid \hat{g}^\star(\tau)\big) + \log(N_\pi(\tau)) \;, \tag{8.33}$$

and for the empirical best policy $\hat{\pi}^\star_\tau$,

$$N_{\tilde{\pi}_{\tau+1}}(\tau) \le N_{\hat{\pi}^\star_\tau}(\tau)\,. \tag{8.34}$$

Proof. For all stationary policy $\pi \in \Pi$, we have

$$I_\pi(\tau) = N_\pi(\tau)\mathsf{d}\big(\hat{g}_\pi(\tau) \mid \hat{g}^\star(\tau)\big) + \log(N_\pi(\tau))$$

by definition. Hence, by non-negativity of the first term, it comes

$$\log(N_\pi(\tau)) \le I_\pi(\tau)\,.$$

This implies, since the policy $\tilde{\pi}_{\tau+1}$ with minimum index is chosen,

$$\log(N_{\tilde{\pi}_{\tau+1}}(\tau)) \le I_{\tilde{\pi}_{\tau+1}}(\tau) = \min_{\pi\in\Pi} I_\pi(\tau) \le I_{\hat{\pi}^\star_\tau}(\tau) = \log\big(N_{\hat{\pi}^\star_\tau}(\tau)\big)\,.$$

Composing by $\exp(\cdot)$ on both side concludes the proof. □

Lemma 8.9.2 (Empirical upper bounds) *Under IMED-KD, for all starting time $\tau \in \mathcal{T}$, for all stationary policy $\pi \in \Pi_\tau$,*

$$N_{\tilde{\pi}_{\tau+1}}(\tau)\, d\big(\hat{g}_{\tilde{\pi}_{\tau+1}}(\tau) \mid \hat{g}^\star(\tau)\big) \le \log(\tau)\,. \tag{8.35}$$

Proof. By construction, since policy $\tilde{\pi}_{\tau+1}$ has minimum index, we have

$$I_{\tilde{\pi}_{\tau+1}}(\tau) \le I_{\hat{\pi}^\star_t}(\tau)\,.$$

To conclude, it remains to note that on one hand,

$$N_{\tilde{\pi}_{\tau+1}}(\tau)\mathsf{d}\big(\hat{g}_{\tilde{\pi}_{\tau+1}}(\tau) \mid \hat{g}^\star(\tau)\big) \le I_{\tilde{\pi}_{\tau+1}}(\tau)\,,$$

and on the other hand,

$$I_{\hat{\pi}^\star_\tau}(\tau) = \log(N_{\hat{\pi}^\star_\tau}(\tau)) \le \log(\tau)\,.$$

□

8.9.3 Non-reliable current best stationary policy

For accuracy $\epsilon > 0$ and stationary policy $\pi \in \Pi$ and state-action pair $c \in \mathcal{C}$, let $\mathcal{M}^{\star}_{c,\pi}(\epsilon)$ be the set of starting times $\tau \in \mathcal{T}$ such that $c_{\tau+1} = c$ and $\tilde{\pi}_{\tau+1} = \pi$ and where some of the current best stationary policy $\hat{\pi}^{\star}_{\tau}$ has not too optimistic gain and does not belong to $\Pi^{\star}$,

$$\mathcal{M}^{\star}_{c,\pi}(\epsilon) \overset{def}{=} \left\{ \tau \in \mathcal{T} : \begin{array}{ll} (1) & c_{\tau+1} = c \\ (2) & \tilde{\pi}_{\tau+1} = \pi \\ (3) & \hat{g}_{\hat{\pi}^{\star}_{\tau}}(\tau) < \mathbf{g}_{c,\hat{\pi}^{\star}_{\tau}}(\eta) + \epsilon \\ (4) & \hat{\pi}^{\star}_{\tau} \notin \Pi^{\star} \end{array} \right\}. \tag{8.36}$$

For all couple of stationary policies $(\pi, \pi') \in \Pi^2$, initial state-action pair $c \in \mathcal{C}$ and for all accuracy $\epsilon > 0$, let us further introduce $\mathcal{K}^{-}_{c,\pi,\pi'}(\epsilon)$ as the set of starting times where couple of stationary policy (π, π') shows ϵ-KL-log deviation, that is

$$\mathcal{K}^{-}_{c,\pi,\pi'}(\epsilon) \overset{def}{=} \left\{ t \in \mathcal{T} : \begin{array}{ll} (1) & c_{\tau+1} = c \\ (2) & \tilde{\pi}_{\tau+1} = \pi' \\ (3) & \hat{g}_{\pi}(\tau) \leq \mathbf{g}_{c,\pi}(\eta) - \epsilon \\ (4) & \log(N_{\pi'}(\tau)) \leq \\ & N_{\pi}(\tau)\, \mathrm{d}\Big(\hat{g}_{\pi}(\tau) \mid \mathbf{g}_{c,\pi}(\eta) - \epsilon\Big) \\ & + \log(N_{\pi}(\tau)) \end{array} \right\}. \tag{8.37}$$

The two sets are related thanks to the following result.

Lemma 8.9.3 (Relation between subsets of times) *Under the IMED-KD learning strategy, for all accuracy $0 < \epsilon < \epsilon_{\boldsymbol{M}}(\eta)/2$, for all stationary policy $\pi \in \Pi$ and starting state-action pair $c \in \mathcal{C}$,*

$$\mathcal{M}^{\star}_{c,\pi}(\epsilon) \subset \bigcap_{\pi^{+} \in \mathcal{V}^{\star}} \mathcal{K}^{-}_{c,\pi^{+},\pi}(\epsilon_{\boldsymbol{M}}(\eta)/2)\,, \tag{8.38}$$

where we introduced the set $\mathcal{V}^{\star} = \bigcup_{\hat{\pi}^{\star} \notin \Pi^{\star}} \operatorname{argmax}_{\pi' \in \mathcal{V}_{\hat{\pi}^{\star}}} \boldsymbol{g}_{c,\pi'}(\eta)$.

Proof. Let us consider $\tau \in \mathcal{M}^{\star}_{c,\pi}(\epsilon)$. Since $\hat{\pi}^{\star}_{\tau} \notin \Pi^{\star}$ is not a best stationary policy, then according to policy-improvement Assumption 8.2.3 and for a value of η ensuring that $\epsilon_{\mathbf{M}}(\eta) > 0$, for any $\pi^{+} \in \operatorname{argmax}_{\pi' \in \mathcal{V}_{\hat{\pi}^{\star}_{\tau}}} \mathbf{g}_{c_{\tau+1},\pi'}(\eta)$ we have

$$\mathbf{g}_{c_{\tau+1},\pi^{+}}(\eta) > \mathbf{g}_{c_{\tau+1},\hat{\pi}^{\star}_{\tau}}(\eta)\,. \tag{8.39}$$

Then, since $\hat{\pi}^{\star}_{\tau} \in \operatorname{argmax}_{\pi \in \Pi} \hat{g}(\tau)$ and $\mathcal{V}_{\hat{\pi}^{\star}_{\tau}} \subset \Pi_{\tau} \subset \Pi$, we have on the other hand

$$\hat{g}_{\hat{\pi}^{\star}_{\tau}}(\tau) = \hat{g}^{\star}(\tau) \geq \hat{g}_{\pi^{+}}(\tau)\,, \tag{8.40}$$

where $\pi^{+} \in \operatorname{argmax}_{\pi' \in \mathcal{V}_{\hat{\pi}^{\star}_{\tau}}} \mathbf{g}_{c_{\tau+1},\pi'}(\eta) \subset \mathcal{V}_{\hat{\pi}^{\star}_{\tau}} \subset \Pi$ according to IMED-KD algorithm. Since $\tau \in \mathcal{M}^{\star}_{c,\pi}(\epsilon)$, we have by construction

$$\mathbf{g}_{c_{\tau+1},\hat{\pi}^{\star}_{\tau}}(\eta) + \epsilon \geq \hat{g}_{\hat{\pi}^{\star}_{\tau}}(\tau)\,. \tag{8.41}$$

By combining Equations (8.40) and (8.41), it comes

$$\mathbf{g}_{c_{\tau+1},\hat{\pi}_\tau^\star}(\eta) + \epsilon \geq \hat{g}^\star(\tau) \geq \hat{g}_{\pi^+}(\tau)\,. \tag{8.42}$$

Since $\epsilon_{\mathbf{M}}(\eta) \leq \mathbf{g}_{c_{\tau+1},\pi^+}(\eta) - \mathbf{g}_{c_{\tau+1},\hat{\pi}_\tau^\star}(\eta)$, Equation (8.39) implies

$$\mathbf{g}_{c_{\tau+1},\pi^+}(\eta) > \mathbf{g}_{c_{\tau+1},\hat{\pi}_\tau^\star}(\eta) + \epsilon_{\mathbf{M}}(\eta)\,.$$

Then, since $\epsilon \leq \epsilon_{\mathbf{M}}(\eta)/2$, previous Equation (8.42) implies

$$\mathbf{g}_{c_{\tau+1},\pi^+}(\eta) - \epsilon_{\mathbf{M}}(\eta)/2 > \mathbf{g}_{c_{\tau+1},\pi^+}(\eta) + \epsilon \geq \hat{g}^\star(\tau) \geq \hat{g}_{\pi^+}(\tau)\,. \tag{8.43}$$

At this point, since $\pi^+ \in \mathcal{V}_{\hat{\pi}_\tau^\star} \subset \Pi_\tau$, empirical lower bounds (8.33) imply

$$\log(N_{\tilde{\pi}_{\tau+1}}(\tau)) \leq N_{\pi^+}(\tau)\, \mathtt{d}\big(\hat{g}_{\pi^+}(\tau) \mid \hat{g}^\star(\tau)\big) + \log(N_{\pi^+}(\tau))\,. \tag{8.44}$$

The classical monotonic properties of $\mathtt{d}(\cdot \mid \cdot)$ and Equation (8.43) imply

$$\begin{cases} \hat{g}_{\pi^+}(\tau) \leq \hat{g}^\star(\tau) < \mathbf{g}_{c_{\tau+1},\pi^+}(\eta) - \epsilon_{\mathbf{M}}(\eta)/2 \\ \mathtt{d}(\hat{g}_{\pi^+}(\tau) \mid \hat{g}^\star(\tau)) \leq \mathtt{d}\Big(\hat{g}_{\pi^+}(\tau) \mid \mathbf{g}_{c_{\tau+1},\pi^+}(\eta) - \epsilon_{\mathbf{M}}(\eta)/2\Big)\,. \end{cases} \tag{8.45}$$

Combining Equations (8.43) and (8.45), we finally get

$$\begin{cases} \hat{g}_{\pi^+}(\tau) & < \mathbf{g}_{c_{\tau+1},\pi^+}(\eta) - \epsilon_{\mathbf{M}}(\eta)/2 \\ \log(N_{\tilde{\pi}_{\tau+1}}(t)) & \leq N_{\pi^+}(\tau)\, \mathtt{d}\Big(\hat{g}_{\pi^+}(\tau) \mid \mathbf{g}_{c_{\tau+1},\pi^+}(\eta) - \epsilon_{\mathbf{M}}(\eta)/2\Big) \\ & \quad + \log(N_{\pi^+}(\tau))\,, \end{cases} \tag{8.46}$$

which means $\tau \in \mathcal{K}^-_{c,\pi^+,\tilde{\pi}_{\tau+1}}(\epsilon_{\mathbf{M}}(\eta)/2)$, hence concluding the proof. □

8.9.4 Reliable current gains and current best stationary policy

In this subsection, we characterize subsets of starting times where both the gain of current played stationary policy and the optimal gain are well-estimated.

Let us consider for an accuracy $0 < \epsilon < \epsilon_{\mathbf{M}}$, a stationary policy $\pi \in \Pi$ and a starting state-action pair $c \in \mathcal{C}$ that is suboptimal, the following set of starting times

$$\mathcal{U}_{c,\pi}(\epsilon) = \left\{ \tau \in \mathcal{T} : \begin{array}{ll} (1) & c_{\tau+1} = c \\ (2) & \tilde{\pi}_{\tau+1} = \pi \notin \Pi^\star \\ (3) & (3a) \text{ or } (3b) \text{ or } (3c) \text{ or } (3d) \text{ where} \\ (3a) & \hat{g}_\pi(\tau) \geq \mathbf{g}_{c,\pi}(\eta) + \epsilon \\ (3b) & \hat{g}_{\hat{\pi}_\tau^\star}(\tau) \geq \mathbf{g}_{c,\hat{\pi}_\tau^\star}(\eta) + \epsilon \text{ and } N_\pi(\tau) \leq N_{\hat{\pi}_\tau^\star}(\tau) \\ (3c) & \hat{g}_{\hat{\pi}_\tau^\star}(\tau) \leq \mathbf{g}_{c,\hat{\pi}_\tau^\star}(\eta) - \epsilon \text{ and } N_\pi(\tau) \leq N_{\hat{\pi}_\tau^\star}(\tau) \\ (3d) & \hat{g}_{\hat{\pi}_\tau^\star}(\tau) < \mathbf{g}_{c,\hat{\pi}_\tau^\star}(\eta) + \epsilon \text{ and } \hat{\pi}_\tau^\star \notin \Pi^\star \end{array} \right\},$$

where we recall that whenever $\tilde{\pi}_{\tau+1} = \pi$, then $N_{\tilde{\pi}_{\tau+1}}(\tau) \leq N_{\hat{\pi}_\tau^\star}(\tau)$, by (8.34). By construction of this set we have the following result.

Lemma 8.9.4 (Reliable current means) *Under the IMED-KD strategy, for all accuracy* $0 < \epsilon < \epsilon_{\mathbf{M}}(\eta)/2$*, for all stationary policy* $\pi \in \Pi$ *and starting state-action pair* $c \in \mathcal{C}$*, for all starting time* $\tau \notin \mathcal{U}_{c,\pi}(\epsilon)$ *such that* $c_{\tau+1} = c$ *and* $\tilde{\pi}_{\tau+1} = \pi \notin \Pi^\star$*,*

$$\begin{cases} \hat{\pi}^\star_\tau \in \Pi^\star_{c_{\tau+1}} \\ \hat{g}^\star(\tau) \geq \boldsymbol{g}^\star_{c_{\tau+1}}(\eta) - \epsilon \\ \hat{g}_\pi(\tau) \leq \boldsymbol{g}_{c_{\tau+1},\pi}(\eta) + \epsilon \,. \end{cases}$$

Proof. For $0 < \epsilon < \epsilon_{\mathbf{M}}(\eta)/2$, for stationary policy $\pi \in \Pi$, let us consider a starting time $\tau \notin \mathcal{U}_{c,\pi}(\epsilon)$, such that $\tilde{\pi}_{\tau+1} = \pi \notin \Pi^\star$.

Since $\tilde{\pi}_{\tau+1} = \pi \notin \Pi^\star$ and $\tau \notin \mathcal{U}_{c_{\tau+1},\tilde{\pi}_{\tau+1}}(\epsilon)$ then $\hat{g}_{\tilde{\pi}_{\tau+1}}(\tau) < \mathbf{g}_{c_{\tau+1},\tilde{\pi}_{\tau+1}}(\eta) + \epsilon$, which rewrites $\hat{g}_\pi(\tau) < \mathbf{g}_{c_{\tau+1},\pi}(\eta) + \epsilon$ (since $\tilde{\pi}_{\tau+1} = \pi$).

Likewise, since $\tilde{\pi}_{\tau+1} = \pi \notin \Pi^\star_{c_{\tau+1}}$ and $\tau \notin \mathcal{U}_{c_{\tau+1},\tilde{\pi}_{\tau+1}}(\epsilon)$, and $N_\pi(\tau) \leq N_{\hat{\pi}^\star_\tau}(\tau)$ then

$$\hat{g}^\star(\tau) = \hat{g}_{\hat{\pi}^\star_\tau}(\tau) > \mathbf{g}_{c_{\tau+1},\hat{\pi}^\star_\tau}(\eta) - \epsilon \,. \tag{8.47}$$

Likewise, we must have

$$\hat{g}^\star(\tau) = \hat{g}_{\hat{\pi}^\star_\tau}(\tau) < \mathbf{g}_{c_{\tau+1},\hat{\pi}^\star_\tau}(\eta) + \epsilon \,. \tag{8.48}$$

Since $\tilde{\pi}_{\tau+1} = \pi \notin \Pi^\star$ and $\tau \notin \mathcal{U}_{c_{\tau+1},\tilde{\pi}_{\tau+1}}(\epsilon)$, then this must in turn imply

$$\hat{\pi}^\star_\tau \in \Pi^\star \,. \tag{8.49}$$

By combining this with Equations (8.47) and (8.48), we get

$$\hat{g}^\star(\tau) > \mathbf{g}^\star_{c_{\tau+1}}(\eta) - \epsilon \,. \tag{8.50}$$

□

Size of the set $\mathcal{U}_{c,\pi}(\epsilon)$ We now want to control the expected size of the set $\mathcal{U}_{c,\pi}(\epsilon)$. To this end, we first note that, Lemma 8.9.3 enables to replace the equality in the definition of $\mathcal{U}_{c,\pi}(\epsilon)$ with an inclusion and (3d) with

$$\begin{aligned} \forall \pi^+ \in \mathcal{V}^\star, \qquad & \hat{g}_{\pi^+}(\tau) \leq \mathbf{g}_{c,\pi^+}(\eta) - \tilde{\epsilon} \qquad (8.51) \\ & \log(N_\pi(\tau)) \leq N_{\pi^+}(\tau)\, \mathrm{d}\Big(\hat{g}_{\pi^+}(\tau) \mid \mathbf{g}_{c,\pi^+}(\eta) - \tilde{\epsilon}\Big) + \log(N_{\pi^+}(\tau)) \end{aligned}$$

where $\tilde{\epsilon} = \epsilon_{\mathbf{M}}(\eta)/2 \geq \epsilon$. Further, we realize that we can decompose the set as follows

$$\mathcal{U}_{c,\pi}(\epsilon) \subset \mathcal{C}_{c,\pi}(\epsilon) \cup \overline{\mathcal{C}}_{c,\pi}(\epsilon) \cup \mathcal{K}_{c,\pi}(\tilde{\epsilon}) \,,$$

where we introduced the following convenient events

$$\begin{aligned} \mathcal{C}_{c,\pi}(\epsilon) &= \{\tau : (1), (2), \hat{g}_\pi(\tau) \geq \mathbf{g}_{c,\pi}(\eta) + \epsilon\} \\ \overline{\mathcal{C}}_{c,\pi}(\epsilon) &= \{\tau : (1), (2), \exists \pi' : |\hat{g}_{\pi'}(\tau) - \mathbf{g}_{c,\pi'}(\eta)| \geq \epsilon \text{ and } N_\pi(\tau) \leq N_{\pi'}(\tau)\} \\ \mathcal{K}_{c,\pi}(\tilde{\epsilon}) &= \{\tau : (1), (2), (8.51)\} \end{aligned}$$

Interestingly, we note that if $\tau \in \mathcal{K}_{c,\pi}(\tilde{\epsilon}) \setminus \overline{\mathcal{E}}_{c,\pi}(\epsilon)$ and since $\tilde{\epsilon} \geq \epsilon$ then for each $\pi^+ \in \mathcal{V}^\star$, on top of (8.51) we must also have $N_\pi(\tau) > N_{\pi'}(\tau)$, which motivates to introduce

$$\overline{\mathcal{K}}_{c,\pi}(\tilde{\epsilon}) \quad = \quad \{\tau : (1), (2), (8.51) \text{ and } \forall \pi^+ \in \mathcal{V}^\star, N_\pi(\tau) > N_{\pi^+}(\tau)\} .$$

Using this decomposition, we get the following control

$$\max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} |\mathcal{U}_{c,\pi}(\epsilon)| \leq \max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} |\mathcal{E}_{c,\pi}(\epsilon)| + \max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} \left|\overline{\mathcal{E}}_{c,\pi}(\epsilon)\right| + \max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} \left|\overline{\mathcal{K}}_{c,\pi}(\tilde{\epsilon})\right| . \quad (8.52)$$

We can now resort to concentration arguments in order to control the size of these sets under rarely-switching algorithms, which yields the following upper bounds. We defer the proof to the next Section 8.11, which is instructive yet hefty, and continue with the proof of the regret bound.

Lemma 8.9.5 (Bounded subsets of times) *Under all rarely-switching algorithm, for all accuracy $\epsilon > 0$, for all stationary policy $\pi \in \Pi$ and starting state-action pair $c \in \mathcal{C}$,*

$$\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} |\mathcal{E}_{c,\pi}(\epsilon)|\right] , \ \mathbb{E}_{(\pi_t)}\left[\max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} \left|\overline{\mathcal{E}}_{c,\pi}(\epsilon)\right|\right] \leq 2\,|\mathcal{C}|\,b_\epsilon ,$$

$$\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} \left|\overline{\mathcal{K}}_{c,\pi}(\epsilon)\right|\right] \leq |\mathcal{C}|\left(b_\epsilon + 1 + c_\epsilon^{-1} + 2C_\epsilon\sqrt{\log(c_\epsilon T)}\right) ,$$

where $b_\epsilon = 2\sigma^2 e^{\epsilon^2/2\sigma^2}/\epsilon^2$ with $\sigma^2 = 1/4$, considering concentration for σ-sub-Gaussian distributions, and $c_\epsilon, C_\epsilon > 0$ are the constants involved in the concentration Theorem 8.10.2.

In particular, using this lemma, it holds

$$\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} |\mathcal{U}_{c,\pi}(\epsilon)|\right] \leq 5\,|\mathcal{C}|\,b_\epsilon + |\mathcal{C}|\left(c_\epsilon^{-1} + 2C_\epsilon\sqrt{\log(c_\epsilon T)}\right) . \quad (8.53)$$

8.9.5 Upper bounds on the numbers of pulls of suboptimal policies

In this subsection, we now combine the different results of the previous subsections to prove Theorem 8.8.1 that upper bound the number of pulls of suboptimal policies. Combining this result with an upper bound on the length of an episode, controlled thanks to the no-laziness assumption, we will be able to finish the proof.

Proof. of Theorem 8.8.1 For all accuracy $0 < \epsilon < \epsilon_{\mathbf{M}}(\eta)/2$, for all stationary policy $\pi \in \Pi$, for all starting time $\tau \notin \mathcal{U}_{c,\pi}(\epsilon)$ such that $\tilde{\pi}_{\tau+1} = \pi \notin \Pi^\star$, we derive the following steps. From empirical upper bounds (8.35), we have

$$N_\pi(\tau)\,\mathsf{d}\big(\hat{g}_\pi(\tau) \mid \hat{g}^\star(\tau)\big) \leq \log(\tau) . \quad (8.54)$$

From Lemma 8.9.4, we have

$$\hat{g}_\pi(\tau) \leq \mathbf{g}_{c_{\tau+1},\pi}(\eta) + \epsilon < \mathbf{g}^\star_{c_{\tau+1}}(\eta) - \epsilon \leq \hat{g}^\star(\tau)$$

From classical monotonic properties of $\mathrm{KL}(\cdot|\cdot)$ and Equation (8.32), we have

$$\begin{aligned}\mathtt{d}\big(\hat{g}_\pi(\tau) \mid \hat{g}^\star(\tau)\big) &\geq \mathrm{KL}\Big(\mathbf{g}_{c_{\tau+1},\pi}(\eta) + \epsilon \Big| \mathbf{g}^\star_{c_{\tau+1}}(\eta) - \epsilon\Big) \\ &\geq (1 + \alpha_{\mathbf{M}}(\epsilon))^{-1} \mathrm{KL}\Big(\mathbf{g}_{c_{\tau+1}}(\eta) \Big| \mathbf{g}^\star_{c_{\tau+1}}(\eta)\Big)\end{aligned}$$

In view of Equation (8.54), and recalling that $N_\pi(\tau) = \mathbf{N}^\eta_{c_{\tau+1},\pi}(\tau)$, this implies $\forall \tau \notin \mathcal{U}_{c,\pi}(\epsilon)$ such that $\tilde{\pi}_{\tau+1} = \pi \notin \Pi^\star$,

$$\mathbf{N}^\eta_{c_{\tau+1},\pi}(\tau) \leq \frac{(1 + \alpha_{\mathbf{M}}(\epsilon))\log(\tau)}{\mathrm{KL}\big(\mathbf{g}_{c_{\tau+1},\pi}(\eta) \big| \mathbf{g}^\star_{c_{\tau+1}}(\eta)\big)}\,. \tag{8.55}$$

For state-action pair $c \in \mathcal{C}$ and for stationary policy $\pi \notin \Pi^\star$, we denote by

$$\tau_{c,\pi} = \max\{\tau \in \mathcal{T} :\ c_{\tau+1} = c, \quad \tilde{\pi}_{\tau+1} = \pi \quad \text{and} \quad \tau \notin \mathcal{U}_{c,\pi}(\epsilon)\} \tag{8.56}$$

the last starting time that does not belong to $\mathcal{U}_{c,\pi}(\epsilon)$ such that we play stationary policy π.

Now, using Equation (8.56) and that by definition, when $\tau \in \mathcal{U}_{c,\pi}(\epsilon)$, it must be that $c_{\tau+1} = c,\ \tilde{\pi}_{\tau+1} = \pi$, we obtain

$$\begin{aligned}\mathbf{N}^\eta_{c,\pi}(T) &= \min_{c'} N_{c'}(h_{1:T}) \\ &= \min_{c'} \sum_{k\in\mathbb{N}} \sum_{t=\tau_k+1}^{\tau_{k+1}} \mathbb{1}\{c_t = c'\} \\ &= \min_{c'} \sum_{k\in\mathbb{N}} \mathbb{1}\left\{\begin{array}{c}\tau+1 = c, \\ \tilde{\pi}_{\tau+1} = \pi, \\ \tau_k \notin \mathcal{U}_{c,\pi}(\epsilon)\end{array}\right\} \sum_{t=\tau_k+1}^{\tau_{k+1}} \mathbb{1}\{c_t = c'\} \\ &\quad + \sum_{k\in\mathbb{N}} \mathbb{1}\left\{\begin{array}{c}\tau+1 \neq c \text{ or} \\ \tilde{\pi}_{\tau+1} \neq \pi \text{ or} \\ \tau_k \in \mathcal{U}_{c,\pi}(\epsilon)\end{array}\right\} \sum_{t=\tau_k+1}^{\tau_{k+1}} \mathbb{1}\{c_t = c'\} \\ &\leq \min_{c'} N_{c'}(h_{1:\tau_{c,\pi}}) + \sum_{k\in\mathbb{N}} \mathbb{1}\{\tau_k \in \mathcal{U}_{c,\pi}(\epsilon)\} \sum_{t=\tau_k+1}^{\tau_{k+1}} \mathbb{1}\{c_t = c'\} \\ &\leq \min_{c'} N_{c'}(h_{1:\tau_{c,\pi}}) + \sum_{k\in\mathbb{N}} \mathbb{1}\{\tau_k \in \mathcal{U}_{c,\pi}(\epsilon)\}\,(\tau_{k+1} - \tau_k) \\ &\overset{\mathscr{L}}{=} \mathbf{N}^\eta_{c,\pi}(\tau_{c,\pi}) + \sum_{k\in\mathbb{N}} \mathbb{1}\{\tau_k \in \mathcal{U}_{c,\pi}(\epsilon)\}\,\ell_{c,\pi}\end{aligned}$$

where the last equality holds in law, using that $\tau_{k+1} - \tau_k$ is equal in law to $\ell_{c,\pi}$.

Now, from Lemma 8.7.1, we can control the expected value of $\ell_{c,\pi}$ conditionally on the past history before each episode, by the deterministic quantity

$$\mathbb{E}_{(\pi_t)}[\ell_{c,\pi} | h_{1:\tau}, c_\tau] \quad \leq \quad \mathbf{L} \overset{def}{=} \max\{(|\mathcal{C}| + 2)D_{\mathbf{M}}, D_{\mathbf{M}} + 2B\}\,.$$

Since the law of the stopping time $\ell_{c,\pi}$ is independent on other variables

before the start of an episode, we deduce that

$$\begin{aligned}
&\mathbb{E}_{(\pi_t)}\left[\sum_{k\in\mathbb{N}} \mathbb{1}\left\{\tau_k \in \mathcal{U}_{c,\pi}(\epsilon)\right\} \ell_{c,\pi}\right] \\
&\quad = \mathbb{E}_{(\pi_t)}\left[\sum_{k\in\mathbb{N}} \mathbb{1}\left\{\tau_k \in \mathcal{U}_{c,\pi}(\epsilon)\right\} \mathbb{E}_{(\pi_t)}[\ell_{c,\pi} | h_{1:\tau_k} c_{\tau_k} = c]\right] \\
&\quad \leq \mathbb{E}_{(\pi_t)}\left[\sum_{k\in\mathbb{N}} \mathbb{1}\left\{\tau_k \in \mathcal{U}_{c,\pi}(\epsilon)\right\}\right] \mathbf{L}\,.
\end{aligned}$$

Further, remarking that $\sum_{k\in\mathbb{N}} \mathbb{1}\left\{\tau_k \in \mathcal{U}_{c,\pi}(\epsilon)\right\} = |\mathcal{U}_{c,\pi}(\epsilon)|$, we deduce that

$$\mathbb{E}_{(\pi_t)}[\max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} \mathbf{N}^\eta_{c,\pi}(T)] \leq \mathbb{E}_{(\pi_t)}[\max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} \mathbf{N}^\eta_{c,\pi}(\tau_{c,\pi})] + \mathbb{E}_{(\pi_t)}[\max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}}] |\mathcal{U}_{c,\pi}(\epsilon)|]\mathbf{L}\,.$$

Combined with the inequality Equation (8.55), and using where that $\tau_{c,\pi} \leq T$, we obtain

$$\begin{aligned}
\mathbb{E}_{(\pi_t)}[\max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} \mathbf{N}^\eta_{c,\pi}(T)] \quad &\leq \mathbb{E}_{(\pi_t)}[\max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} \frac{(1+\alpha_{\mathbf{M}}(\epsilon))\log(\tau_{c,\pi})}{\mathrm{KL}\left(\mathbf{g}_{c,\pi}(\eta)\middle|\mathbf{g}^\star_c(\eta)\right)}] \\
&\quad + \mathbb{E}_{(\pi_t)}[\max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} |\mathcal{U}_{c,\pi}(\epsilon)|]\mathbf{L} \\
&\leq \max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} \frac{(1+\alpha_{\mathbf{M}}(\epsilon))\log(T)}{\mathrm{KL}\left(\mathbf{g}_{c,\pi}(\eta)\middle|\mathbf{g}^\star_c(\eta)\right)} \\
&\quad + \mathbb{E}_{(\pi_t)}[\max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} |\mathcal{U}_{c,\pi}(\epsilon)|]\mathbf{L}\,.
\end{aligned}$$

We conclude the proof using Equation (8.53) to control

$$\mathbb{E}_{(\pi_t)}[\max_{\substack{c\in\mathcal{C}\\ \pi\neq\pi^\star}} |\mathcal{U}_{c,\pi}(\epsilon)|]\,.$$

□

It remains to address the proof of Lemma 8.9.5, that controls the cardinal of some undesirable sets of episodes. Controlling this number given the fact that we can, independently of the played policy, control the length of an episode, was enough to conclude on the regret. The proof will split into two parts. Before deriving the proof, we introduce some technical results that are used in the forthcoming analysis.

8.10 Concentration inequalities

In this section, we state two concentration results used in the proof. First, a classical maximal inequality for sub-Gaussian distributions. Then, a boundary crossing probability result suitable to the analysis of an IMED strategy, adapted here to sub-Gaussian distributions.

Lemma 8.10.1 (Maximal concentration inequality) *Under assumption 8.2.2, for any $(s,a)\in\mathcal{C}$, for $x<\boldsymbol{m}_{s,a}$,and integer $n\geq 0$, we have*

$$\mathbb{P}\left(\bigcup_{\substack{t\geq 1\\ N_{s,a}(t)\geq n}} \hat{m}_{s,a}(t)<x\right)\leq \exp(-n\,d(x\mid \boldsymbol{m}_{s,a}))\,.$$

Proof. Indeed, by a Chernoff method, introducing some $\lambda>0$, and $\phi(\lambda)=\sigma^2\lambda^2/2$ where $\sigma=1/2$, we get, provided that $\lambda\epsilon\geq\phi(\lambda)$,

$$\begin{aligned}
&\mathbb{P}(\exists t, N_{s,a}(t)\geq n, \hat{m}_{s,a}(t)-\mathbf{m}(s,a)>\epsilon)\\
&= \mathbb{P}\left(\exists t, N_{s,a}(t)\geq n, \exp(\lambda\sum_{j=1}^{N} Z_j)>\exp(\lambda N_{s,a}(t)\epsilon)\right)\\
&\leq \mathbb{P}\left(\exists N\geq n, \exp(\lambda\sum_{j=1}^{N} Z_j-N\phi(\lambda))>\exp(N(\lambda\epsilon-\phi(\lambda))\right)\\
&\leq \mathbb{P}\left(\exists N\geq n, \exp(\lambda\sum_{j=1}^{N} Z_j-N\phi(\lambda))>\exp(n(\lambda\epsilon-\phi(\lambda))\right)\\
&= \mathbb{E}_{(\pi_t)}\left[\max_{N\geq n}\exp(\lambda\sum_{j=1}^{N} Z_j-N\phi(\lambda))\right]\exp\left(-n(\lambda\epsilon-\phi(\lambda)\right)\,,
\end{aligned}$$

where N denotes a random stopping time and $W_n=\exp(\lambda\sum_{j=1}^{n} Z_j-n\phi(\lambda))$ is a non-negative supermartingale bounded by 1. By Doob's maximal inequality the expectation term is thus upper bounded by 1. Optimizing over λ, we get

$$\begin{aligned}
\mathbb{P}(\exists t, N_{s,a}(t)\geq n, \hat{m}_{s,a}(t)>\mathbf{m}_{s,a}+\epsilon) &\leq \exp(-n\epsilon^2/(2\sigma^2))\\
&= \exp\left(-n\mathtt{d}(\mathbf{m}_{s,a}+\epsilon|\mathbf{m}_{s,a})\right)\,.
\end{aligned}$$

□

For reward distributions belonging to generic exponential family, concentration result is obtained in [35, Corollary 2]. Specializing this to Gaussian distributions with variance 1/2, the Kullback-Leibler between two distributions reduces to $\mathtt{d}$. When assuming regular canonical one-dimensional exponential reward distributions $\mathbf{r}(\cdot)$, we can define the reward distributions as a function their means and abusively write $\mathbf{r}(\cdot)=(\mathbf{r}_c)=(\mathbf{r}(\mathbf{m}_c))$. Using this, we obtain more precisely

Theorem 8.10.2 (Boundary crossing probabilities) *For all couple $(s,a)\in\mathcal{C}$, for all $\epsilon>0$, for all $n\geq 1$, we have for one-dimensional exponential distributions*

$$\mathbb{P}\left(\bigcup_{\substack{t\geq 1\\ \hat{m}_{s,a}(t)<\boldsymbol{m}_{s,a}-\epsilon\\ 1\leq N_{s,a}(t)\leq n}} N_{s,a}(t)KL(\boldsymbol{r}(\hat{m}_{s,a}(t)),\boldsymbol{r}(\boldsymbol{m}_{s,a}))\geq\log(n/N_{s,a}(t))\right)\leq\frac{C_\epsilon}{n\sqrt{\log(c_\epsilon n)}}\,,$$

where $\mathit{KL}(\boldsymbol{r}(\hat{m}_{s,a}(t)), \boldsymbol{r}(\boldsymbol{m}_{s,a})) = \mathsf{d}(\hat{m}_{s,a}(t) \mid \boldsymbol{m}_{s,a} - \epsilon)$ *when assuming Gaussian distributions with standard deviation* $\sigma = 1/2$*, and where* $c_\epsilon, C_\epsilon > 0$ *are explained in [35, Corollary 2].*

It is then not difficult, scrutinizing the proof, to show that the same bound still holds now for sub-Gaussian distributions. Importantly, in this case d is no longer the natural metric, but by Pinsker inequality, d always controls it, although now in a possibly loose way. More precisely, in the case of Gaussian reward distributions, we have $\mathsf{KL}(\mathbf{r}_c, \mathbf{r}'_c) = (m'_c - m_c)^2 / 2\sigma^2 = \mathsf{d}(m_c \mid m'_c)$ while for the case of reward distributions with support in $[0,1]$ that we consider (that are $\sigma = 1/2$ sub-Gaussian), by Pinsker's inequality combined with properties of total variation norm, it holds $\mathsf{KL}(\mathbf{r}_c, \mathbf{r}'_c) \geq \|\mathbf{r}_c - \mathbf{r}'_c\|_{TV}^2 / 2 \geq 2(m'_c - m_c)^2 = \mathsf{d}(m_c \mid m'_c)$. That is, concentration provides a high probability upper bound only on $\mathsf{d}(m_c \mid m'_c)$ but (of course) not on the more demanding $\mathsf{KL}(\mathbf{r}_c, \mathbf{r}'_c)$.

Discussion. The previous restriction is not problematic, as in the analysis, we use the concentration tools of Lemma 8.10.1 and Theorem 8.10.2 after we deduce from inequalities involving the gains inequalities involving the state-action means. This is what is done especially in Section 8.11 and Equation (8.67) by considering this straightforward lemma.

Lemma 8.10.3 (Mean to Gain) *For a finite set* $\mathscr{C}$*, consider aggregates* $g = \sum_{c \in \mathscr{C}} p_c \cdot m_c$ *and* $g' = \sum_{c \in \mathscr{C}} p_c \cdot m'_c$*, where* $\forall c \in \mathscr{C}, m_c, m'_c \in \mathbb{R}, p_c \in [0,1]$ *with* $\sum_{c \in \mathscr{C}} p_c = 1$*. Let* $\overline{c} \in \operatorname{argmax}_{c \in \mathscr{C}}(m'_c - m_c)$ *such that gap* $m'_{\overline{c}} - m_{\overline{c}}$ *is maximal. Then, for a given accuracy* $\varepsilon > 0$*,* $g' - g \geq \varepsilon$ *implies:*

$$(i) \quad m'_{\overline{c}} - m_{\overline{c}} \geq g' - g \geq \epsilon, \qquad (ii) \quad \left(m'_{\overline{c}} - m_{\overline{c}}\right)^2 / \sigma^2 \geq \left(g' - g\right)^2 / \sigma^2, \quad \forall \sigma > 0.$$

Reminding that $\mathsf{d}(x \mid y) = \frac{(x-y)^2}{2\sigma^2} = 2(x-y)^2$, then equation (ii) above rewrites $\mathsf{d}\left(m_{\overline{c}} \mid m'_{\overline{c}}\right) \geq \mathsf{d}(g \mid g')$ as desired for Equation (8.67) to hold. Hence, we only need a high-probability control on $\mathsf{d}\left(m_{\overline{c}} \mid m'_{\overline{c}}\right)$ to conclude and not on $\mathsf{KL}(\mathbf{r}_{\overline{c}}, \mathbf{r}'_{\overline{c}})$. Now, one may prefer to use a more refined bound. In order to use a more refined (pseudo-)metric d' such as involving $\mathsf{d}' = \mathsf{KL}$ instead of KL, one would need reversed inequalities of the form $\mathrm{KL}(g', g) = (g' - g)^2 / 2\sigma^2 \geq \kappa_{g,g'} \mathsf{d}'(g, g')$ for some $\kappa_{g,g'} \leq 1$. Whenever these are available (such as in one-dimensional exponential families), one would derive from (ii) $\mathsf{d}\left(m'_{\overline{c}} \mid m_{\overline{c}}\right) \geq \mathsf{d}(g' \mid g)$ the bound $\mathsf{d}\left(m'_{\overline{c}} \mid m_{\overline{c}}\right) \geq \kappa_{g,g'} \mathsf{d}'(g, g')$. This in turn makes appear factors of the form $1/\kappa_{g,g'}$ in the regret bounds, unless we slightly reshape the IMED-type indexes to handle such factors.

Furthermore, we note it is possible to resort to another variant of Pinsker's inequality specific to regular canonical one-dimensional exponential distributions, in lieu of the one applied to distributions with bounded support. Such distributions include Gaussian with known variance, Bernoulli or Poisson distributions as special cases, see e.g. [34] for further examples, as well as the proof of the following result.

Lemma 8.10.4 (A variant of Pinsker's inequality) *When assuming regular canonical one-dimensional exponential reward distributions* $\boldsymbol{r}(\cdot)$*, for* $m < m'$*, it holds that*

$$\mathit{KL}(\boldsymbol{r}(m), \boldsymbol{r}(m')) \geq \frac{(m' - m)^2}{2\sigma^2},$$

where $\sigma^2 = \max\left\{\mathbb{V}_{X \sim \boldsymbol{r}(m'')}(X) \colon m'' \in [m\,, m']\right\}$.

We refer to Lemma 3 in Appendix A.2.A from [34] for more insights. Thus, using this result we can assume regular canonical one-dimensional exponential reward distributions and ensure the same theoretical guarantees by replacing $\sigma^2 = 1/4$ with $\max_{m \in [m^-, m^+]} \mathbb{V}_{X \sim \mathbf{r}(m)}(X)$ in metric $\mathsf{d}(\cdot \mid \cdot)$, where m^- and m^+ are such that $\mathbf{m} \subset [m^-\,, m^+]$. This enables to replace the assumption 8.2.2 assuming bounded support with the following Assumption 8.10.1.

Assumption 8.10.1 (An alternative to Assumption 8.2.2) *We assume regular canonical one-dimensional exponential family reward distributions $\boldsymbol{r}(\cdot)$ with bounded mean $\boldsymbol{m}(\cdot) \in [0,1)$.*

8.11 Bounded subsets of times (Proof of Lemma 8.9.5)

We regroup in this section, for completeness, the proofs of the remaining lemmas used in the analysis of IMED-KD in Section 8.9.

Part 1

Proof. Lemma 8.9.5 We detail the proof to bound $\mathbb{E}_{(\pi_t)}\left[\left|\overline{\mathscr{C}}_{c,\pi}(\epsilon)\right|\right]$. The control of $\mathbb{E}_{(\pi_t)}[|\mathscr{C}_{c,\pi}(\epsilon)|]$ is similar.

We first write

$$\left|\overline{\mathscr{C}}_{c,\pi'}(\epsilon)\right| = \sum_{\tau\in\mathcal{T}} \mathbb{1}\left\{c_{\tau+1}=c,\ \tilde{\pi}_{\tau+1}=\pi', \exists \pi:\ N_{\pi'}(\tau)\le N_{\pi}(\tau),\ |\mathbf{g}_{c,\pi}(\eta)-\hat{g}_{\pi}(\tau)|\ge\epsilon\right\}. \tag{8.57}$$

Considering the stopped stopping times $\tau_n = \inf\{\tau\in\mathcal{T} : c_{\tau+1}=c,\ \tilde{\pi}_{\tau+1}=\pi' \text{ and } N_{\pi'}(\tau)\ge n\}$ for $n\ge 0$, we will rewrite the sum of indicators and use Lemma 8.10.1. We note that the set

$$\{\tau\in\mathcal{T} : c_{\tau+1}=c, \tilde{\pi}_{\tau+1}=\pi' \text{ and } n\le N_{\pi'}(\tau)<n+1\}$$

is either empty or equal to $\{\tau_n\}$. This is true for all rarely-switching algorithm (Algorithm 22), by construction of the stopping event that ensures $N_{\pi'}(\tau)$ increases by one in the corresponding episode.

$$\begin{aligned}
\left|\overline{\mathscr{C}}_{c,\pi'}(\epsilon)\right| &\le \sum_{n=0}^{T-1}\sum_{\tau\in\mathcal{T}} \mathbb{1}\left\{c_{\tau+1}=c,\ \tilde{\pi}_{\tau+1}=\pi',\ n\le N_{\pi'}(\tau)<n+1\right\} &(8.58)\\
&\quad\times\mathbb{1}\left\{\exists\pi : n\le N_{\pi}(\tau),\ |\mathbf{g}_{c,\pi}(\eta)-\hat{g}_{\pi}(\tau)|\ge\epsilon\right\}\\
&\le \sum_{n=0}^{T-1}\sum_{\tau\in\mathcal{T}} \mathbb{1}\left\{\tau=\tau_n,\ c_{\tau+1}=c,\ \tilde{\pi}_{\tau+1}=\pi',\ n\le N_{\pi'}(\tau)<n+1\right\} &(8.59)\\
&\quad\times\mathbb{1}\left\{\exists\pi : n\le N_{\pi}(\tau_n),\ |\mathbf{g}_{c,\pi}(\eta)-\hat{g}_{\pi}(\tau_n)|\ge\epsilon\right\}\\
&\le \sum_{n=0}^{T-1}\mathbb{1}\left\{\exists\pi : n\le N_{\pi}(\tau_n),\ |\mathbf{g}_{c,\pi}(\eta)-\hat{g}_{\pi}(\tau_n)|\ge\epsilon\right\},
\end{aligned}$$

where in the last line we use that $N_{\pi'}(\tau)$ does increase by one in episode τ. At this point, we make use of the fact that $\mathbf{g}_{c,\pi}(\eta) = \sum_{c'\in\mathscr{C}^+_{c,\pi}(\eta)} \tilde{\mathbf{p}}_{\pi}(c)(c')\mathbf{m}_{c'}$ and

$$\hat{g}_{\pi}(\tau) = \sum_{c'\in\mathscr{C}^+_{c,\pi}(\eta)} \tilde{\mathbf{p}}_{\pi}(c)(c')\hat{m}_{c'}(\tau)$$

with

$$\tilde{\mathbf{p}}_{\pi}(c)(c') = \frac{\overline{\mathbf{p}}_{\pi}(c)(c')}{\sum_{c'\in\mathscr{C}^+_{c,\pi}(\eta)}\overline{\mathbf{p}}_{\pi}(c)(c')}$$

so that

$$\mathbf{g}_{c,\pi}(\eta)-\hat{g}_{\pi}(\tau) = \sum_{c'\in\mathscr{C}^+_{c,\pi}(\eta)} \tilde{\mathbf{p}}_{\pi}(c)(c')\left(\mathbf{m}_{c'}-\hat{m}_{c'}(\tau)\right).$$

In particular, $|\mathbf{g}_{c,\pi}(\eta) - \hat{g}_\pi(\tau)| \geq \epsilon$ implies that $\exists c' \in \mathcal{C}^+_{c,\pi}(\eta), |\mathbf{m}_{c'} - \hat{m}_{c'}(\tau)| \geq \epsilon$, otherwise one would have $|\mathbf{g}_{c,\pi}(\eta) - \hat{g}_\pi(\tau)| < \left(\sum_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} \tilde{\mathbf{p}}_\pi(c)(c') \right) \epsilon = \epsilon$. Hence, this shows that

$$\begin{aligned}
\left|\overline{\mathcal{E}}_{c,\pi'}(\epsilon)\right| &\leq \sum_{n=0}^{T-1} \mathbb{1}\left\{\exists \pi, c' \in \mathcal{C}^+_{c,\pi}(\eta),\ n \leq N_{c'}(\tau_n),\ |\mathbf{m}_{c'} - \hat{m}_{c'}(\tau_n)| \geq \epsilon\right\} \\
&\leq \sum_{n=0}^{T-1} \mathbb{1}\left\{\exists c' \in \bigcup_\pi \mathcal{C}^+_{c,\pi}(\eta),\ n \leq N_{c'}(\tau_n),\ |\mathbf{m}_{c'} - \hat{m}_{c'}(\tau_n)| \geq \epsilon\right\}. \\
&\leq \sum_{n=0}^{T-1} \mathbb{1}\left\{\exists c' \in \mathcal{C},\ n \leq N_{c'}(\tau_n),\ |\mathbf{m}_{c'} - \hat{m}_{c'}(\tau_n)| \geq \epsilon\right\}.
\end{aligned}$$

The last inequality implies

$$\max_{\substack{c \in \mathcal{C} \\ \pi' \in \Pi}} \left|\overline{\mathcal{E}}_{c,\pi'}(\epsilon)\right| \leq \sum_{c \in \mathcal{C}} \sum_{n=0}^{T-1} \mathbb{1}\left\{n \leq N_c(\tau_n),\ |\mathbf{m}_c - \hat{m}_c(\tau_n)| \geq \epsilon\right\}. \tag{8.60}$$

Taking the expectation of Equation (8.60), it comes

$$\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c \in \mathcal{C} \\ \pi' \in \Pi}} \left|\overline{\mathcal{E}}_{c,\pi'}(\epsilon)\right|\right] \leq \sum_{(s,a) \in \mathcal{C}} \sum_{n \geq 0} \mathbb{P}\left(\bigcup_{\substack{t \geq 1 \\ N_{s,a}(t) \geq n}} |\hat{m}_{s,a}(t) - \mathbf{m}_{s,a}| \geq \epsilon \right). \tag{8.61}$$

From Lemma 8.10.1, previous Equation (8.61) implies

$$\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c \in \mathcal{C} \\ \pi' \in \Pi}} \left|\overline{\mathcal{E}}_{c,\pi'}(\epsilon)\right|\right] \leq \sum_{c \in \mathcal{C}} \sum_{n \geq 0} 2\exp(-n\, \mathtt{d}(\mathbf{m}_c - \epsilon \mid \mathbf{m}_c)). \tag{8.62}$$

From Pinsker's inequality, previous Equation (8.62) implies

$$\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c \in \mathcal{C} \\ \pi' \in \Pi}} \left|\overline{\mathcal{E}}_{c,\pi'}(\epsilon)\right|\right] \leq \sum_{c \in \mathcal{C}} \sum_{n \geq 0} 2\exp\left(-n\epsilon^2/2\sigma^2\right) = \frac{2\,|\mathcal{C}|}{1 - e^{-\epsilon^2/2\sigma^2}}, \tag{8.63}$$

where $\sigma^2 = 1/4$, assuming 1/2-sub-Gaussian reward distributions. Finally, we note that

$$\frac{1}{1 - e^{-\epsilon^2/2\sigma^2}} = \frac{e^{\epsilon^2/2\sigma^2}}{e^{\epsilon^2/2\sigma^2} - 1} \leq \frac{2\sigma^2 e^{\epsilon^2/2\sigma^2}}{\epsilon^2} = b_\epsilon.$$

□

Part 2

Proof. of Lemma 8.9.5

We now prove the upper bound on $\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c \in \mathcal{C} \\ \pi \in \Pi}} \left|\overline{\mathcal{K}}_{c,\pi}(\epsilon)\right|\right]$. By definition of the set, we have

$$\begin{aligned}
\left|\overline{\mathcal{K}}_{c,\pi'}(\epsilon)\right| &= \sum_{\tau \in \mathcal{T}} \mathbb{1}\left\{c_{\tau+1} = c,\ \tilde{\pi}_{\tau+1} = \pi',\ \forall \pi^+ \in \mathcal{V}^\star, 0 < N_{\pi^+}(\tau) < N_{\pi'}(\tau)\right\} \\
&\quad \times \mathbb{1}\left\{\hat{g}_{\pi^+}(\tau) \leq \mathbf{g}_{c,\pi^+}(\eta) - \epsilon\right\} \\
&\quad \times \mathbb{1}\left\{\log(N_{\pi'}(\tau)) \leq N_{\pi^+}(\tau)\, \mathrm{KL}(\hat{g}_{\pi^+}(\tau) | \mathbf{g}_{c,\pi^+}(\eta) - \epsilon) + \log(N_{\pi^+}(\tau))\right\}.
\end{aligned} \tag{8.64}$$

Considering again the stopped stopping times $\tau_n = \inf\{\tau \in \mathcal{T} : c_{\tau+1} = c,\ \tilde{\pi}_{\tau+1} = \pi' \text{ and } N_{\pi'}(\tau) \geq n\}$ for $n \geq 0$, we will rewrite the previous sum and use boundary crossing probabilities for one-dimensional exponential family distributions. We recall that the set

$$\{\tau \in \mathcal{T} : c_{\tau+1} = c,\ \tilde{\pi}_{\tau+1} = \pi' \text{ and } n \leq N_{\pi'}(\tau) < n+1\}$$

is either empty or equal to $\{\tau_n\}$.

$$\begin{aligned}
&\left|\overline{\mathcal{K}}_{c,\pi'}(\epsilon)\right| \\
\leq\ & \sum_{\tau\in\mathcal{T}} \mathbb{1}\left\{c_{\tau+1} = c,\ \tilde{\pi}_{\tau+1} = \pi',\ \forall \pi \in \mathcal{V}^\star, 0 < N_\pi(\tau) < N_{\pi'}(\tau),\ \hat{g}_\pi(\tau) \leq \mathbf{g}_{c,\pi}(\eta) - \epsilon\right\} \\
&\quad \times \mathbb{1}\left\{\log\left(N_{\pi'}(\tau)\right) \leq N_\pi(\tau)\,\mathrm{KL}(\hat{g}_\pi(\tau)|\mathbf{g}_{c,\pi}(\eta) - \epsilon) + \log\left(N_\pi(\tau)\right)\right\} \\
\leq\ & \sum_{n=0}^{T-1}\sum_{\tau\in\mathcal{T}} \mathbb{1}\left\{c_{\tau+1} = c,\ \tau = \tau_n,\ \tilde{\pi}_{\tau+1} = \pi',\ n \leq N_{\pi'}(\tau) < n+1\right\} \qquad (8.65) \\
&\quad \times \mathbb{1}\left\{\forall \pi \in \mathcal{V}^\star, 0 < N_\pi(\tau),\ \hat{g}_\pi(\tau) \leq \mathbf{g}_{c,\pi}(\eta) - \epsilon\right\} \\
&\quad \times \mathbb{1}\left\{\forall \pi \in \mathcal{V}^\star, \log(n) \leq N_\pi(\tau)\,\mathrm{KL}(\hat{g}_\pi(\tau)|\mathbf{g}_{c,\pi}(\eta) - \epsilon) + \log\left(N_\pi(\tau)\right)\right\} \\
\leq\ & \sum_{n=0}^{T-1} \mathbb{1}\left\{\forall \pi \in \mathcal{V}^\star, 0 < N_\pi(\tau_n),\ \hat{g}_\pi(\tau_n) \leq \mathbf{g}_{c,\pi}(\eta) - \epsilon\right\} \\
&\quad \times \mathbb{1}\left\{\forall \pi \in \mathcal{V}^\star, \log(n) \leq N_\pi(\tau_n)\,\mathrm{KL}(\hat{g}_\pi(\tau_n)|\mathbf{g}_{c,\pi}(\eta) - \epsilon) + \log\left(N_\pi(\tau_n)\right)\right\} \qquad (8.66)
\end{aligned}$$

Let us consider for stationary policy $\pi \in \Pi$ and starting time $\tau \in \mathcal{T}$,

$$c_\tau^\pi \in \operatorname*{argmin}_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} \hat{m}_{c'}(\tau) - \mathbf{m}_{c'}\,.$$

Then, the following inequality and implication holds (see Lemma 8.10.3):

$$\hat{m}_{c_\tau^\pi}(\tau) - \mathbf{m}_{c_\tau^\pi} \leq \hat{g}_\pi(\tau) - \mathbf{g}_{c,\pi}(\eta) = \sum_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} \tilde{\mathbf{p}}_\pi(c)(c')(\hat{m}_{c'}(\tau) - \mathbf{m}_{c'})\,.$$

$$\left(\hat{g}_\pi(\tau) \leq \mathbf{g}_{c,\pi}(\eta) - \epsilon\right) \quad \Rightarrow \quad \left(\hat{m}_{c_\tau^\pi}(\tau) \leq \mathbf{m}_{c_\tau^\pi} - \epsilon \text{ and } \mathrm{KL}(\hat{g}_\pi(\tau)|\mathbf{g}_{c,\pi}(\eta) - \epsilon) \leq \mathrm{KL}(\hat{m}_{c_\tau^\pi}(\tau)|\mathbf{m}_{c_\tau^\pi} - \epsilon)\right) \qquad (8.67)$$

Note also that since $c_\tau^\pi \in \mathcal{C}^+_{c,\pi}(\eta)$ and by construction $N_\pi(\tau) = \min_{c' \in \mathcal{C}^+_{c,\pi}(\eta)} N_{c'}(\tau)$, then $N_\pi(\tau_n) > 0$ implies $N_{c_{\tau_n}^\pi}(\tau_n) > 0$. In particular, Equation (8.65) and previous Equation (8.67) imply

$$\begin{aligned}
\left|\overline{\mathcal{K}}_{c,\pi'}(\epsilon)\right| \leq\ & \min_{\pi\in\mathcal{V}^\star} \sum_{n=0}^{T-1} \mathbb{1}\left\{0 < N_{c_{\tau_n}^\pi}(\tau_n),\ \hat{m}_{c_{\tau_n}^\pi}(\tau_n) \leq \mathbf{m}_{c_{\tau_n}^\pi} - \epsilon\right\} \\
&\quad \times \mathbb{1}\left\{\log(n) \leq N_{c_{\tau_n}^\pi}(\tau_n)\,\mathrm{KL}(\hat{m}_{c_{\tau_n}^\pi}(\tau_n)|\mathbf{m}_{c_{\tau_n}^\pi} - \epsilon) + \log\left(N_{c_{\tau_n}^\pi}(\tau_n)\right)\right\} \\
\leq\ & \min_{\pi\in\mathcal{V}^\star} \sum_{c' \in \mathcal{C}^+_{c_{\tau+1},\pi}(\eta)} \sum_{n=0}^{T-1} \mathbb{1}\left\{0 < N_{c'}(\tau_n),\ \hat{m}_{c'}(\tau_n) \leq \mathbf{m}_{c'} - \epsilon\right\} \\
&\quad \times \mathbb{1}\left\{\log(n) \leq N_{c'}(\tau_n)\,\mathrm{KL}(\hat{m}_{c'}(\tau_n)|\mathbf{m}_{c'} - \epsilon) + \log\left(N_{c'}(\tau_n)\right)\right\}.
\end{aligned}$$

This last inequality implies

$$\begin{aligned}
\max_{\substack{c\in\mathcal{C}\\ \pi'\in\Pi}} \left|\overline{\mathcal{K}}_{c,\pi'}(\epsilon)\right| \leq\ & \sum_{c\in\mathcal{C}} \sum_{n=0}^{T-1} \mathbb{1}\left\{1 \leq N_c(\tau_n),\ \hat{m}_c(\tau_n) \leq \mathbf{m}_c - \epsilon\right\} \\
&\quad \times \mathbb{1}\left\{\log(n) \leq N_c(\tau_n)\,\mathrm{KL}(\hat{m}_c(\tau_n)|\mathbf{m}_c - \epsilon) + \log\left(N_c(\tau_n)\right)\right\} \qquad (8.68)
\end{aligned}$$

Taking the expectation of Equation (8.68), it comes

$$\begin{aligned}\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c\in\mathscr{C}\\ \pi'\in\Pi}}\left|\overline{\mathscr{K}}_{c,\pi'}(\epsilon)\right|\right] &\leq \sum_{c\in\mathscr{C}}\sum_{n=0}^{T-1}\mathbb{P}\left(\bigcup_{\substack{t\geq 1\\ N_c(t)\geq n}}\hat{m}_c(t)<\mathbf{m}_c-\epsilon\right)\\ &\quad+\sum_{c\in\mathscr{C}}\sum_{n=2}^{T-1}\mathbb{P}\left(\bigcup_{\substack{t\geq 1\\ \hat{m}_c(t)<\mathbf{m}_c-\epsilon\\ 1\leq N_c(t)\leq n}}N_c(t)\mathsf{d}(\hat{m}_c(t)\mid\mathbf{m}_c-\epsilon)\geq\log(n/N_c(t))\right).\end{aligned}$$

From Lemma 8.10.1 and Theorem 8.10.2, previous Equation (8.69) implies

$$\mathbb{E}_{(\pi_t)}\left[\max_{\substack{c\in\mathscr{C}\\ \pi'\in\Pi}}\left|\overline{\mathscr{K}}_{c,\pi'}(\epsilon)\right|\right]\leq|\mathscr{C}|\left(\frac{2\sigma^2e^{\epsilon^2/2\sigma^2}}{\epsilon^2}+1+c_\epsilon^{-1}+2C_\epsilon\sqrt{\log(c_\epsilon T)}\right). \tag{8.69}$$

Indeed, we have

$$\begin{aligned}\sum_{n=2}^{T-1}\frac{C_\epsilon}{n\sqrt{\log(c_\epsilon n)}} &\leq 1+c_\epsilon^{-1}+C_\epsilon\sum_{n\geq 1+c_\epsilon^{-1}}^{T-1}\frac{c_\epsilon}{c_\epsilon n\sqrt{\log(c_\epsilon n)}}\\ &\leq 1+c_\epsilon^{-1}+C_\epsilon\int_{c_\epsilon^{-1}}^{T}\frac{c_\epsilon\,dx}{c_\epsilon x\sqrt{\log(c_\epsilon x)}}\\ &= 1+c_\epsilon^{-1}+2C_\epsilon\sqrt{\log(c_\epsilon T)}\,.\end{aligned}$$

□

8.12 Choice of policies

In this section, we discuss the construction of the set Π_τ. Hereafter, we consider that a set of policies to be small if its size does not exceed 10^6, somewhat arbitrarily, motivated by numerical applications.

Small set of policies

First, there are cases in which Π is small. This situation may typically happen in real-world applications when a learner must choose between a limited set of policies prescribed by experts. A typical example is that of agriculture in which policies are intervention plans carefully built by agronomists, with a few parameters, despite considering a complex system.

[75]: Puterman (1994), *Markov Decision Processes — Discrete Stochastic Dynamic Programming*

Small set of optimal policies

Then, even when Π is large, there are cases when $\Pi^\star$ is known to belong to a small set of policies. For instance in [75][Theorem 8.11.3], the author detail the case of an inventory problem when an optimal policy can be searched in a restricted set of $\binom{A+S-1}{S}$ many *non-decreasing* policies instead of all possible A^S ones. For an MDP with $S = 150$ states and $A = 4$ actions, there are over 10^{90} deterministic policies but only $585276 \simeq 10^6$ non-decreasing policies. Likewise, in goal-state MDPs, one can restrict to policies aiming at reaching (and staying) in a single state as fast as possible (they can be computed knowing the transitions of the MDP), yielding only S many policies to consider.

Considering a small enough set of policies

Finally, generic structural properties of the MDP can be used, such as restricting to stationary and unichain policies since an optimal policy satisfies both conditions. Also, when the MDP is known to be unichain, it then satisfies Assumption 8.2.3 with $k = 1$, which suggests to simply choose $\Pi_\tau = \mathcal{V}_{\hat{\pi}_\tau^\star}(1)$. More generally, one can set $\Pi_\tau = \mathcal{V}_{\hat{\pi}_\tau^\star}(k)$ provided that $|\mathcal{V}_\pi(k)| = \binom{k}{S} A^k$ is small. When k is unknown, one may choose $\Pi_\tau = \mathcal{V}_{\hat{\pi}_\tau^\star}(\tilde{k}) \cup \Gamma$ where $\tilde{k}$ satisfies $\binom{\tilde{k}}{S} A^{\tilde{k}} \leq 10^6$ and Γ is a small set of policies uniformly randomly chosen in $\Pi \setminus \mathcal{V}_{\hat{\pi}_\tau^\star}(\tilde{k})$. This indeed ensures that Π_τ contains an improving policy over $\hat{\pi}_\tau^\star$ with positive probability, which may be interesting for the practitioner. Because the set of all deterministic stationary is finite, albeit large, the expected time to wait before it belongs to the set of selected policies is therefore finite and controlled. Furthermore, while this set is large, we mention that it is not numerically costly to sample a polcy from that large set since it only consists in sampling S actions from the S sets $\mathcal{A}_s$ of cardinal at most A.

Computing the set of neighborhood policies

We now explain how we compute $\Pi_\tau(1)$ which is the 1-neighborhood of the empirical optimal policy augmented by some randomly chosen policies. Computing the indexes of policies in $\Pi_\tau(1)$ is more complicated than it seems because some polcies might be multichain, *i.e.* have multiple recurrent classes. In those cases, neither the gain nor the index is uniquely defined, and we must decompose the policy on all its recurrent classes in order to compute one gain and one index per class.

The k-neighborhood of a policy can be computed recursively from the 1-neighborhood. To compute the one neighborhood, we iterate through all states and actions to create $S \times (A-1)$ new policies. Let's denote by π_{sa} the policy that corresponds to the modification of $\hat{\pi}_\star$ where action a (different from $\hat{\pi}_\star(s)$) is chosen in state s. We compute the associated Markov chain thanks to our knowledge of transitions. If the policy is unichain, *i.e.* the associated Markov chain has only one recurrent class, then we compute the gain (unique) of the policy and add it to the 1-neighborhood pool.

If the policy is multichain, *i.e.* the associated Markov chain has $p > 1$ recurrent classes, then there are up to p gains associated to this policy, one for each recurrent class. In this case, we compute all the recurrent classes, all the associated gains (and effective number of pulls) and we register p different policies in $\Pi_\tau(1)$, one for each recurrent classes. This way, we will compute p different indexes for the policy π_{sa}, associated to each possible gain. In a sense, we derive from a multichain policy p unichain policies.

To find the different stationary distributions, we use the GTH-algorithm (Grassmann, Taksar and Heyman algorithm), [119]. The set of neighborhood policies is re-computed only when the empirical policy changes, but we still randomly sample distributions to add to the set Π_τ as described in the main part of the paper. Also, if a randomly selected policy is multichain, we apply the same decomposition procedure that we described.

[119]: Grassmann et al. (1985), 'Regenerative Analysis and Steady State Distributions for Markov Chains'

Now that we specified the `Event` function to use, how to compute the set of considered policies in the Algorithm 22 and the Bandit algorithm that we consider to specify how to choose the next policy to play at the end of an episode, *i.e.* IMED, we are ready to run experiments that will assess the soundness of our algorithmic design and theoretical analysis of IMED-KD.

8.13 Numerical experiments

3: Source code is available on github

In the final section of this chapter, we discuss the practical implementation of the IMED-KD algorithm, and present some numerical experiments[3]. We consider several environments that we already encountered in the previous Chapter 7, *Riverswim*, *Nasty*, and two gridworld-like environments, 2-*rooms* and 4-*rooms*.

Comparing IMED-KD with other RL algorithms

In those environments, we illustrate the performance of IMED-KD against the strategies UCRL3 [86], PSRL [90] and Q-learning (run with discount $\gamma = 0.99$ and optimistic initialization). PSRL and UCRL3 use a confidence parameter to control the quality of the MDP approximation, which is set to 0.05 in the experiments. The η parameter of IMED-KD plays a similar role, and we therefore use $\eta = 0.05/|\mathcal{S}|$ to ensure a fair comparison. IMED-KD uses value iteration as a routine, which is faster than the extended value iteration used in UCRL3. Q-learning takes an exploration parameter, ϵ, or exploration scheme when ϵ is slowly decreased with time. We report average algorithmic regret curves computed from 2048 independent experiments along with quantiles 0.1 and 0.9. The considered horizon depends on the environment as some are "easier" than others in the time required to see regret curves exhibit their logarithmic behaviors.

Riverswim

First, *RiverSwim* (Figure 8.3), which is difficult to navigate and require the agent to "figure out" a difficultly attainable large reward is located at one end of the chain-like environment while a small but easy obtainable reward is located at the other end. In each of the n states, there are two

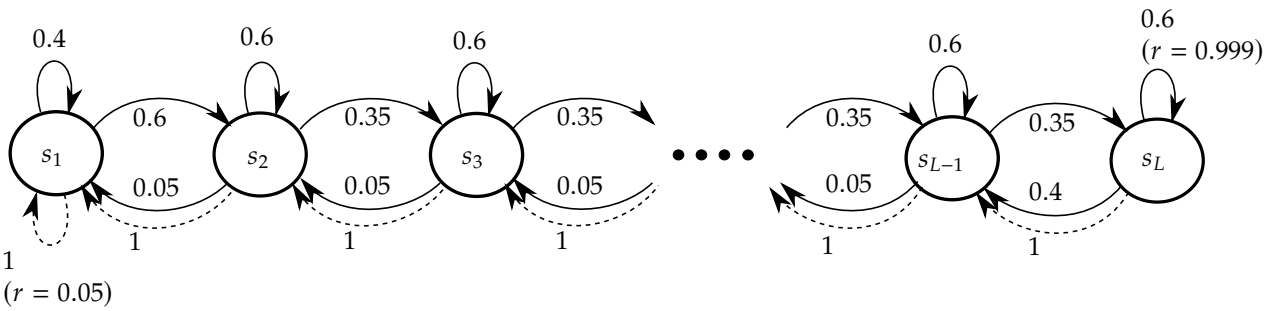

Figure 8.3: The n-states *RiverSwim* MDP

actions `RIGHT` and `LEFT`. In Figure 8.3, the `LEFT` action is represented with a dashed line and the `RIGHT` with plain line. Rewards are located at the extremities of the MDP, with a small reward in left initial state s_1 and large reward in the rightmost state s_n. Starting from state s_1, this setting has proven to be a challenging one because of the large amount of non-rewarding exploration necessary to find the optimal policy. We consider the 6-state and 25-state instances, which allows us to compare how algorithms behave depending on the amount of necessary exploration.

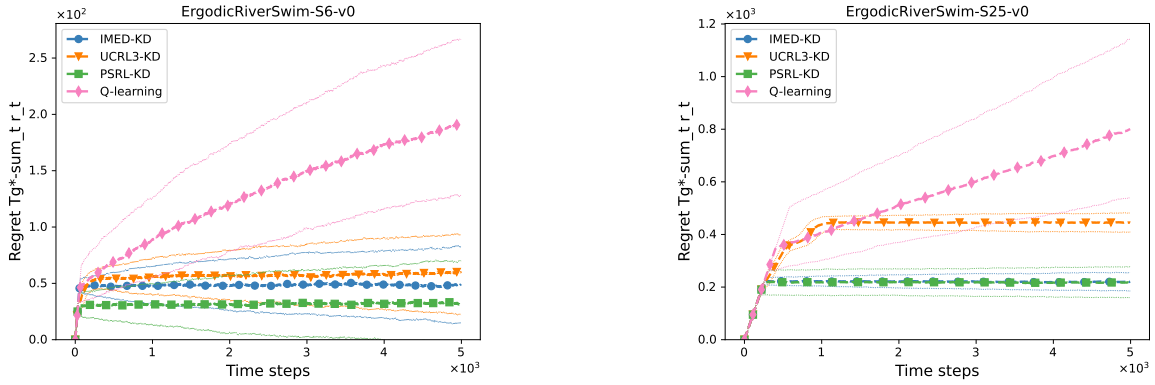

Figure 8.4: Regret on RiverSwim with 6 states (left) and 25 states (right)

Results Plots of expected regret are depicted in Figure 8.4. Q-learning is struggling despite its optimistic initialization, while IMED-KD is on par with PSRL on both experiments. The regret of UCRL3 scales differently with the number n of states than the one of IMED-KD and PSRL, although it remains controlled.

Nasty

Then we consider the *nasty* (Figure 8.5), where two high reward cycles are separated by a bottleneck action. The difficulty here is that agent must not switch too often between the cycles to avoid incurring the large cost induced by the traversal of the bottleneck. More precisely, in this

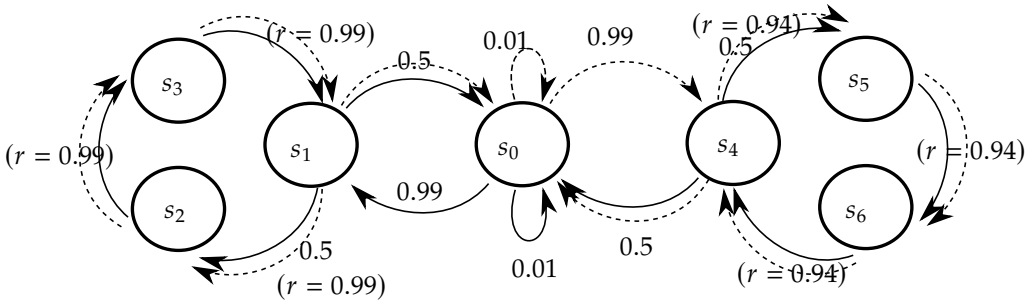

Figure 8.5: The *Nasty* MDP

setting, there are two promising cycles separated by a small chain of one bottleneck state with no associated reward, which may induce an "oscillation" of a learning agent between the two cycles, paying the cost of the travel along the chain each time it changes cycle (policy).

Results Plots of expected regret are depicted in Figure 8.6. IMED-KD and PSRL are highly competitive and perform similarly with a slight advantage of IMED-KD while UCRL3 is not doing so well, worse than Q-learning, which itself suffers a large and linear-shaped regret.

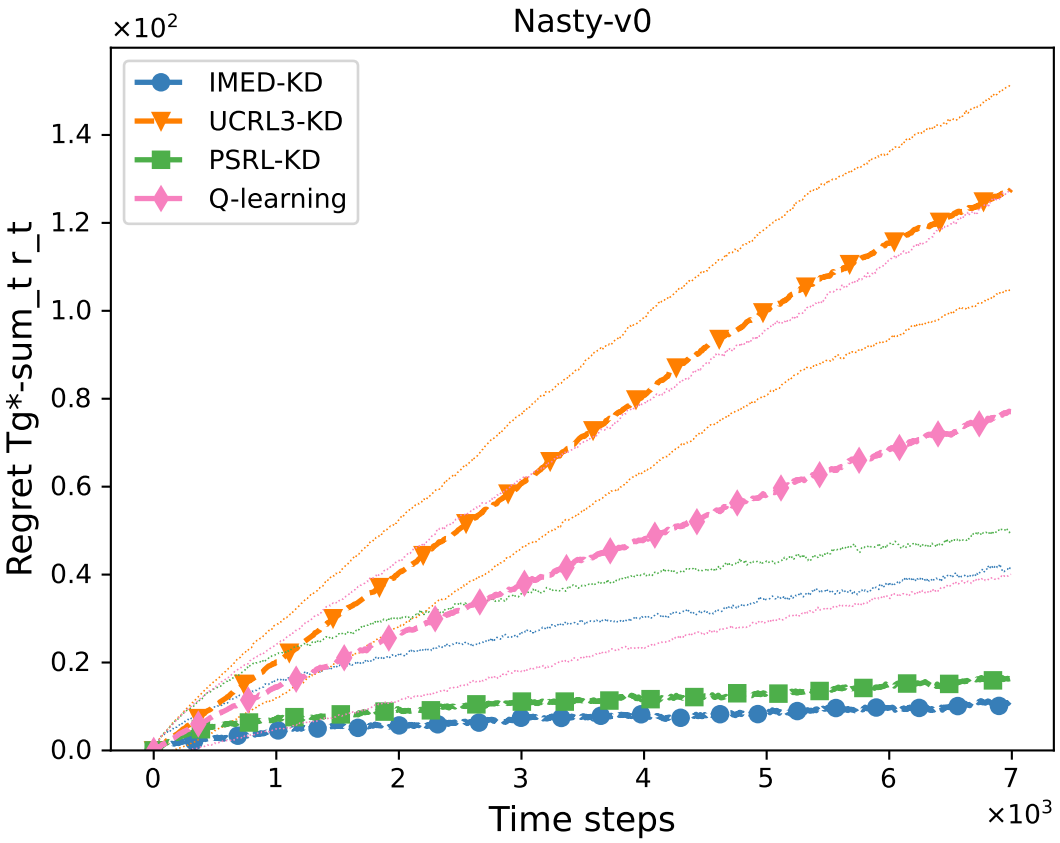

Figure 8.6: Regret curves on the *Nasty* environment

n-Rooms

Finally, we consider two grid-like environments, 4-*rooms* and 2-*rooms* (Figure 8.7). Both are sparse reward environments with close-to-deterministic transitions.

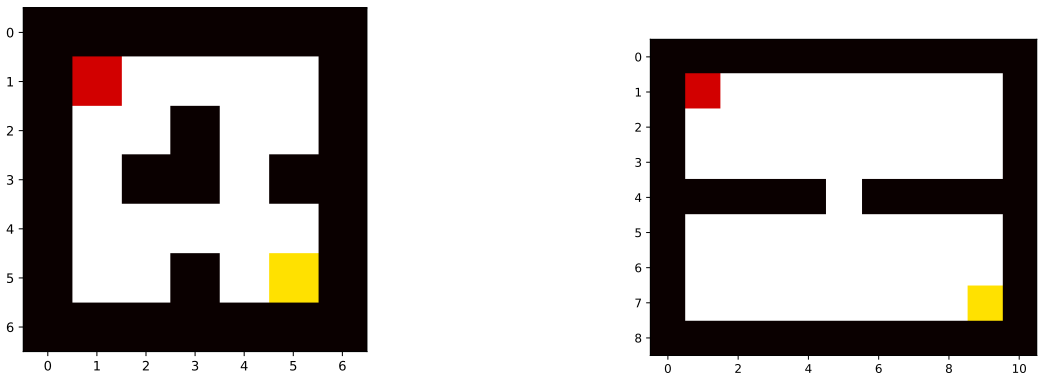

Figure 8.7: The 4-*rooms* (left) and 2-*rooms* (right) MDPs

4-rooms is a grid-like environment with 20 states and 2-rooms is an environment with 55 states. For both those grid-like environments there are 4 cardinal actions where transitions are close to deterministic with a

0.8 chance of going in the intended direction. A reward of 0.99 is located in the goal state (highlighted in yellow), while it is zero elsewhere. After reaching the goal, the agent is positioned again in the initial red-state.

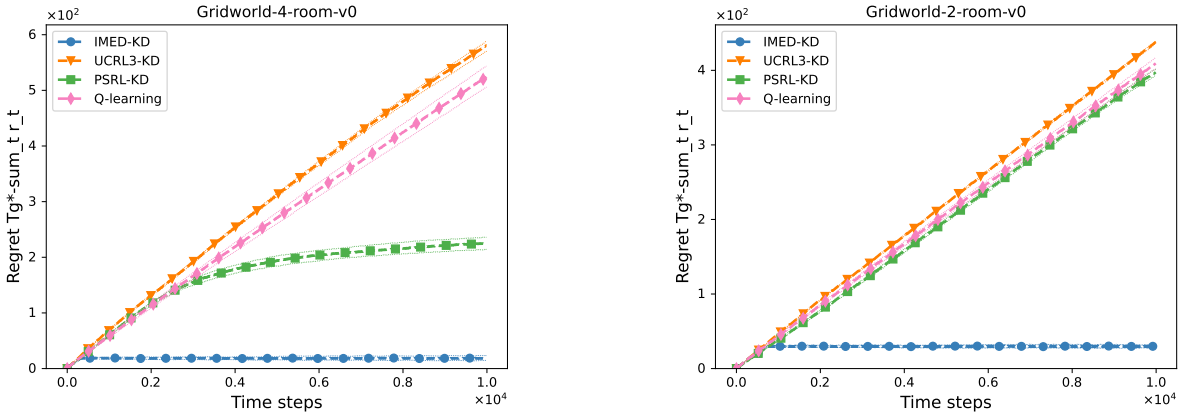

Figure 8.8: Regret on 4-rooms (left) and 2-rooms (right)

Results As shown in the left part of Figure 8.8 reporting the results on the 4-rooms environment, IMED-KD outperforms the others by a large margin. Even for horizons as large as 10^5, we cannot observe a bend in the Q-learning regret curve while it occurs around time step 6×10^4 for UCRL3. We show this in Figure 8.9 depicting the result of an experiment ran on 4-rooms with large horizon of 100 000, but only 1024 runs. The regret curve IMED-KD is then almost blend with the x-axis, but we can see the UCRL3 is indeed learning as its regret curve start to bend around time step 60 000.

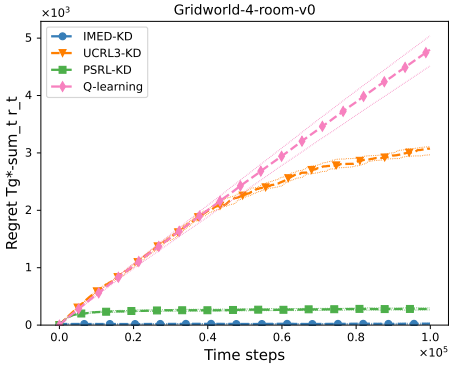

Figure 8.9: Regret curves in the 4-rooms environment with horizon of 100 000

In the right part of Figure 8.8 which report the result on the 2-rooms environment, it can be observed that IMED-KD is again, and by far, the best of tested algorithm in this environment. Even PSRL which was somewhat competitive in 4-rooms suffers a large linear regret for a very long time. In fact, we cannot find a reasonable horizon such that regret curves start bending. Those two results emphasize the effectiveness of IMED-KD in grid-like environment.

Benefit of the rarely-switching Algorithm

In this last series of experiments, we illustrate the potential benefit of Algorithm 22 itself, by experimentally comparing Algorithm 22 instantiated on others Bandits strategies. Indeed, we have combined in IMED-KD Algorithm 22 with the IMED approach, but one could in principle consider a UCB or TS approach as well. The IMED algorithm is used in line 13 of the Algorithm 22 where, IMED indexes are computed for each of the policies in the considered policy space and the IMED selection rule is applied to compute the policy to play for the next episode.

While we do not extend the theoretical analysis to other Bandit algorithms, we provide in this section, numerical experiments illustrating the performance of the rarely-switching version of TS and UCB, respectively. For these experiments, we assume Gaussian-like reward distributions. Because the reward are bounded in $[0, 1]$, we can assume that the variance is upper bounded by $\frac{1}{2}$. More precisely, TS-RS (Thompson Sampling with Rarely Switching) consists in using the following TS selection rule in line 13 of Algorithm 22. For all policy a, we sample a reward $x_a(t)$ from posterior distribution $\mathcal{N}(\hat{\mu}_a(t), \frac{1}{2\sqrt{N_a(t)}})$ and select the policy with the largest sample, $\pi_{t+1} \in \operatorname{argmax} x_a(t)$. While UCB-RS (UCB with Rarely Switching) consists in using the following UCB selection rule in line 11 of Algorithm 22. For all policy a, we compute a reward upper bound $u_a(t)$ from UCB index for $\frac{1}{2}$ sub-Gaussian distributions, $u_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{\log t}{2N_a(t)}}$ and select the policy with the largest upper bound, $\pi_{t+1} \in \operatorname{argmax} u_a(t)$.

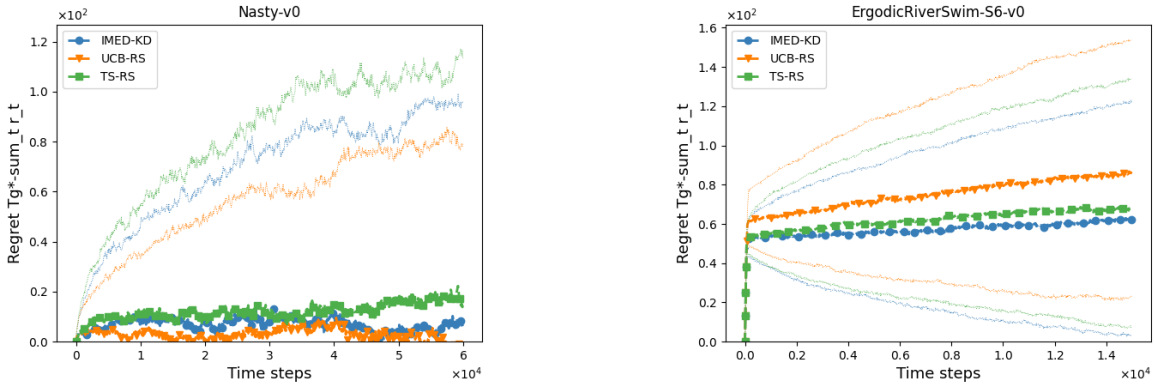

Figure 8.10: Testing and comparing Algorithm 22 with TS and UCB as based bandit strategy

We compare those strategies in the 6-states RiverSwim environment Figure 8.10 (right) where it can be seen that Algorithm 22 can indeed be used, at least experimentally, with other bandit sampling strategies with good empirical performances. Still, we observe that our original design suffer the smallest regret. We also compare the designs on the Nasty environment Figure 8.10 (left) where it can be seen that IMED-KD still is better that the other algorithms, albeit by a margin so small that its can be considered equivalent ot other designs in this experimental setting. Whatever the experiments, Algorithm 22 seems like a good enough design to be used with other bandit strategies and our preferred and studied strategy IMED-KD, seems to be empirically the best.

8.14 Conclusion

Before wrapping up this chapter, we mention the next step that could be taken to enhance the reach of the work presented. This step would be to remove the no-laziness assumption by making η adaptive.

Adaptive η

When the gap between the gain of an optimal policy and the second-largest gain achievable on an MDP, it could be that the chosen value of η impair our IMED-KD from distinguishing between the two and maybe even worse, reverse the order of the best two gains. Mathematically, it could be that the ordering of the policies according to g_π is different from the one computed from the modified gains g_π^η where only state-action pairs that are visited with a frequency larger than η are taken into account in the computation of the modified gain. Because reward are bounded, the error is controlled by η and therefore the order of policies with different gains is preserved if η is smaller than half of the smallest difference between gains.

Therefore, we know that if η is small enough, one can preserve the ordering of policies, thus making IMED-KD safe. Since we don't know the gap in advance, no fixed value of η will do in every environment. One promising solution seems to make η a decreasing function of the time T, where $\eta(T)$ tends to 0 as T tends to infinity. This way, we should be sure that, after a horizon T_0, $\eta(T_0)$ is small enough to preserve the ordering of gains. We provide below a sketch of proof.

Regret under adaptive parameter: Sketch of proof

Note that provided that $\eta < \frac{\epsilon_{\mathbf{M}}(0)}{2\mathbf{m}_{\max}S}$, then if some $\pi' \in \mathcal{V}_\pi$ satisfies, $\mathbf{g}_{c,\pi'}(\eta) > \mathbf{g}_{c,\pi}(\eta)$, then it also satisfies $\mathbf{g}_{c,\pi'} > \mathbf{g}_{c,\pi}$, hand thus a policy improvement can be obtained correctly in a neighborhood of any policy. Unfortunately, since $\epsilon_{\mathbf{M}}(0)$ is unknown, this motivates to consider a η_t decreasing with t and to introduce $T_0 = \min\{t : \eta_t < \frac{\epsilon_{\mathbf{M}}(0)}{2\mathbf{m}_{\max}S}\}$. For $\eta_t \to 0$, $T_0 < \infty$ and for all $T \geq T_0$ then the regret for subsequent time steps is controlled by $\mathscr{R}_{\mathbf{M}}(\mathbf{A}, T - T_0)$, hence $\mathscr{R}_{\mathbf{M}}(\mathbf{A}, T) \leq T_0 + \mathscr{R}_{\mathbf{M}}(\mathbf{A}, T - T_0)$. Upper-bounding $K(\epsilon, \eta)$ by $K(\epsilon, 0)$ and replacing $\mathtt{d}\big(\mathbf{g}_{c,\pi}(\eta)\big|\mathbf{g}_c^\star(\eta)\big)$ by its worst approximation $\mathtt{d}\big(\mathbf{g}_{c,\pi}(\eta_{T_0})\big|\mathbf{g}_c^\star(\eta_{T_0})\big)$, we can hence obtain the following upper bound on the regret $\mathscr{R}_{\mathbf{M}}(\mathbf{A}, T)$,

$$T_0 + \left[\max_{\substack{c\in\mathscr{C}\\ \pi\neq\pi^\star}} \frac{(1+\alpha_{\mathbf{M}}(\epsilon))\log(T-T_0)}{\mathtt{d}\big(\mathbf{g}_{c,\pi}(\eta_{T_0})\big|\mathbf{g}_c^\star(\eta_{T_0})\big) + K_T(\epsilon,0)(D_{\mathbf{M}}+2B)}\right](D_{\mathbf{M}}+2B)\,|\mathscr{C}|\ ,$$

under the only assumption that η_t decreases towards 0 with t.

To wrap up

In this chapter, we studied regret minimization in communicating MDPs with known dynamics but unknown reward functions, and introduced a

class of rarely-switching algorithms, whose design allows for leveraging the connectivity structure induced by the (known) transition function via considering the recurrent classes of the stationary policies. We presented IMED-KD, a rarely-switching algorithm that relies on an IMED-style index function. It admits an efficient implementation and significantly outperforms existing algorithms empirically. Under mild assumptions, we derived a finite-time, problem-dependent, and logarithmic regret bound for IMED-KD. Regret lower bounds for this setting (and communicating MDPs in general) are open, to our best knowledge, and deriving them is an interesting, yet challenging, direction for future work. Other interesting future directions include deriving adaptive rules to tune the parameter η (used to control the gains) and to relax the laziness assumption, even though some restrictive assumption seems required to ensure computational efficiency.

Conclusion

9 A meaningful research path

In this thesis, we modelled and studied two related problems of sequential decision-making. The first one is Bandit. We introduced in Chapter 3, where we gave a lot of intuition about the notion of progress per unit of interaction. In particular, we insisted on the class of learner algorithms, that are those algorithms with uniformly fast convergent rate. After introducing some optimal algorithms, we questioned in Chapter 4 the speed at which information can be processed by an optimal learning algorithm. We framed this question as one of numerical complexity and studied how the per-interaction time and space complexities can be reduced. This search led us to discover FMED and OMED that greatly improve the processing speed. In Chapter 5, we focused on how to exploit structural information about a Bandit problem. We showed that, in some setting, it is possible to exploit the knowledge of a structure without necessarily estimate the structure. The second setting extends the Bandit learning problem. We introduced in Chapter 6 the average-reward Reinforcement-Learning problem. We explained how a lower bound on the logarithmic growth rate of regret is unknown in general and known for the particular case of ergodic MDP. For MDPs satisfying the ergodic assumption, we presented in Chapter 7 the IMED-RL algorithm that we proved to be optimal. Furthermore, the careful algorithmic design of IMED-RL allowed it to be highly performant and suffer a very small finite time empirical regret on all tested environments, even some communicating only. In Chapter 8, we tried to depart from this restrictive ergodic hypothesis. We studied a problem that is *a priori* simpler of solving the average-reward learning problem in an MDP where the transitions are known and reward distributions unknown. The fact that the learner must actively seek information in parts of the state space that it is not guaranteed to visit by assumption make the problem highly non-trivial, even when the transition probabilities are known. As often, one we try to answer a question, many more await around the corner.

9.1 Information per unit of computation

In the Chapter 4 devoted to making the most out of every sample from a computational complexity viewpoint, we mentioned a possible improvement of the work this chapter is based on, the aAFMED Algorithm 17. In the short term, it should be possible to investigate, first empirically, next theoretically such an online method to compute indexes of an optimal Bandit algorithm. One could also add the FIMED 14 algorithmic method to our presented IMED-RL 21 and IMED-KD 22 algorithms which would improve the numerical complexity of these two algorithms. In particular IMED-RL, which is optimal in the ergodic setting would greatly benefit from such an improvement as it can be slower than other algorithms in some environment, as shown in our experiments.

In a longer time horizon, one could really think of investigating the notion of numerical complexity of optimal algorithms. For instance, is

it possible to give a lower bound on the per-time-step time complexity of a uniformly fast learning algorithm? Does this bound changes if we consider the class of optimal algorithms? Similarly, we may investigate the space complexity. How much information is it necessary to store to guarantee the existence of a uniformly fast convergent algorithm? Of an optimal algorithm? We could first try to investigate those questions in the Bandit setting then try to expand the findings and intuitions to the more general Reinforcement Learning setting.

In a much longer horizon, it would be interesting to investigate other learning settings and expand the previous work to understand better the notion of information per unit of interaction that we crafted in this thesis. For instance, consider the following problem. An agent is performing a, possibly random, walk on a bounded smooth manifold. After t time steps, how much did we learn about the manifold? Did we learn about local curvature? Did we learn about genus? Did we learn about β-skeleton? In short, what is the maximal rate of topological-information acquisition? It seems like most of the tools are ready to be used to start investigating these kinds of questions. From a sequential decision-making viewpoint, one could even hope to build a physicist assistant. Given some question that we have about nature, one could answer what is the most likely next experiment to make in order to maximize our information acquisition about the world. With such purposes in mind, first studying the numerical complexity guarantee of sequential decision-making algorithms is necessary. Furthermore, this would require developing mathematical tools suited to the study of finite time guarantees.

9.2 Structure of policy space

In an MDP, the space of policy is highly structured. It means that, by playing a policy, we can learn information about other policies. In Chapter 7, we studied quite the extreme case of ergodic MDPs where all policies share the same set of recurrent states. Basically, the need for active exploration is reduced to zero. In Chapter 8, we made a first step towards the exploitation of the structure in communicating only MDPs.

Yet a lot of work remains to be done to fully understand the structure of the policy space in MDPs and the speed at which the average-reward learning problem can be solved. In the future, investigating a lower bound for communicating only MDPs should really help the community to better understand the structure of the policy space and the cost of active exploration. While it is difficult to say how long such an investigation would last, it surely is worth the research. Once a lower bound is known, it would be much easier to craft meaningful algorithms. In the case where such a lower bound involve a combinatorial optimization problem or a NP-hard problem[1], the need for efficient relaxations and estimations would surely foster research. Following the intuition presented in Chapter 6, in the space of policy, a future research question could therefore be to understand the minimal subset of policies one need to consider performing a stochastic gradient ascent leading to an optimal policy.

1: given that the longest simple path problem is NP-hard, it is not impossible that it is the case.

Linked with the previous topic is the question of the space complexity of optimal algorithms. IMED-RL is model based, meaning that it stores an estimation of the model. Could it be that, using methods *à la* OMED, one could craft an optimal algorithm in the ergodic setting that is model-free? Given the importance of the skeleton and potential function (also known as Q-function in the discounted reward setting), I doubt that we could make the space complexity much less that $S \times A$ while it is $S \times A \times S$ in a model based method. The main problem that I see with model free methods is that, when the skeleton changes, *i.e.* when the empirical optimal policy changes[2], there is no way to propagate quickly this change to the other state-action pairs values since we don't have the knowledge of the transitions. However, after this change, all the values are wrong estimations. At the very least, a change in the empirical optimal policy should induce a reset of all the learning rates to ensure fast convergence to the new values. If it is impossible to find a model free optimal algorithm, it would be interesting to know if one can find one that has logarithmic regret. What is the best learning rate of the class of model free algorithms? Such questions could be interesting to tackle in future researches.

2: The argmax of the Q-function changes in one state

Bibliography

[1] Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*. Vol. 95. 2010 (cited on pages 38, 128).

[2] Herbert Robbins. 'Some aspects of the sequential design of experiments'. In: *Bulletin of the American Mathematical Society* 58.5 (1952), pp. 527–535 (cited on page 41).

[3] Bennett L. Fox and John E. Rolph. 'Adaptive Policies for Markov Renewal Programs'. In: *The Annals of Statistics* 1.2 (1973), pp. 334–341. DOI: 10.1214/aos/1176342370 (cited on page 41).

[4] Tze Leung Lai and Herbert Robbins. 'Asymptotically efficient adaptive allocation rules'. In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22 (cited on page 41).

[5] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 'Finite-time analysis of the multiarmed bandit problem'. In: *Machine learning* 47.2 (2002), pp. 235–256 (cited on pages 43, 62, 66, 75, 77, 85, 91, 97, 139).

[6] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020 (cited on pages 43, 49, 66).

[7] A.N. Burnetas and M.N. Katehakis. 'Optimal adaptive policies for sequential allocation problems'. In: *Advances in Applied Mathematics* 17(2) (1996), pp. 122–142 (cited on pages 44, 197).

[8] Tze Leung Lai and Herbert Robbins. 'Asymptotically efficient adaptive allocation rules'. In: *Advances in applied mathematics* 6.1 (1985), pp. 4–22 (cited on page 44).

[9] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. 'Thompson Sampling: An Asymptotically Optimal Finite-Time Analysis'. In: *Algorithmic Learning Theory - 23rd International Conference, ALT 2012*. 2012 (cited on page 44).

[10] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cited on pages 44, 46).

[11] Aurelien Garivier, Tor Lattimore, and Emilie Kaufmann. 'On Explore-Then-Commit strategies'. In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016 (cited on page 46).

[12] Junya Honda and Akimichi Takemura. 'An asymptotically optimal policy for finite support models in the multiarmed bandit problem'. In: *Mach. Learn.* (2011) (cited on pages 51, 55, 66, 68, 70, 77, 97, 128, 217).

[13] Apostolos N Burnetas and Michael N Katehakis. 'Optimal adaptive policies for sequential allocation problems'. In: *Advances in Applied Mathematics* 17.2 (1996), pp. 122–142 (cited on page 53).

[14] Aurélien Garivier, Pierre Ménard, and Gilles Stoltz. 'Explore first, exploit next: The true shape of regret in bandit problems'. In: *arXiv preprint arXiv:1602.07182* (2016) (cited on pages 53, 167).

[15] Junya Honda and Akimichi Takemura. 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models'. In: *Proceedings of the 23rd annual Conference On Learning Theory*. Ed. by Adam Tauman Kalai and Mehryar Mohri. Omnipress, 2010, pp. 67–79 (cited on pages 55, 71, 73, 217).

[16] Junya Honda and Akimichi Takemura. 'Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards.' In: *J. Mach. Learn. Res.* 16 (2015), pp. 3721–3756 (cited on pages 55, 66, 70, 75–77, 84, 87, 90, 91, 97, 100, 107, 108, 110, 115, 139, 166, 170, 173, 210, 214, 217).

[17] Todd L Graves and Tze Leung Lai. 'Asymptotically efficient adaptive choice of control laws incontrolled markov chains'. In: *SIAM journal on control and optimization* 35.3 (1997), pp. 715–743 (cited on pages 60, 167, 196, 197).

[18] Tze Leung Lai. 'Adaptive treatment allocation and the multi-armed bandit problem'. In: *The Annals of Statistics* (1987), pp. 1091–1114 (cited on pages 62, 76).

[19] Rajeev Agrawal, Demosthenis Teneketzis, and Venkatachalam Anantharam. 'Asymptotically efficient adaptive allocation schemes for controlled iid processes: Finite parameter space'. In: *IEEE Transactions on Automatic Control* 34.3 (1989) (cited on pages 62, 205).

[20] Rajeev Agrawal. 'Sample mean based index policies by O (log n) regret for the multi-armed bandit problem'. In: *Advances in Applied Probability* 27.04 (1995), pp. 1054–1078 (cited on page 62).

[21] Olivier Cappé et al. 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation'. In: *Annals of Statistics* 41.3 (2013), pp. 1516–1541 (cited on pages 62, 66, 70, 75, 77, 84, 85, 91, 92, 96, 97, 110, 113, 139).

[22] O. Cappé et al. 'Kullback-Leibler Upper Confidence Bounds for Optimal Sequential Allocation.' In: *The Annals of Statistics* (2013) (cited on page 62).

[23] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. 'On Bayesian Upper Confidence Bounds for Bandit Problems'. In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, Spain, April 21-23, 2012*. Ed. by Neil D. Lawrence and Mark A. Girolami. Vol. 22. JMLR Proceedings. JMLR.org, 2012, pp. 592–600 (cited on page 62).

[24] William R Thompson. 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'. In: *Biometrika* 25.3/4 (1933), pp. 285–294 (cited on pages 62, 67, 70, 198).

[25] William R Thompson. 'On a criterion for the rejection of observations and the distribution of the ratio of deviation to sample standard deviation'. In: *The Annals of Mathematical Statistics* 6.4 (1935), pp. 214–219 (cited on page 62).

[26] Charles Riou and Junya Honda. 'Bandit algorithms based on Thompson sampling for bounded reward distributions'. In: *Algorithmic Learning Theory*. PMLR. 2020, pp. 777–826 (cited on pages 62, 67, 70, 75, 79, 85, 91, 94, 95, 97, 107, 139).

[27] S. Agrawal and N. Goyal. 'Analysis of Thompson Sampling for the multi-armed bandit problem'. In: *Annual Conference on Learning Theory (COLT)*. 2012, pp. 39–1 (cited on page 67).

[28] S. Agrawal and N. Goyal. 'Further Optimal Regret Bounds for Thompson Sampling'. In: *Proceedings of the 16th Conference on Artificial Intelligence and Statistics*. 2013 (cited on page 67).

[29] E. Kaufmann, N. Korda, and R. Munos. 'Thompson sampling: An asymptotically optimal finite-time analysis'. In: *International Conference on Algorithmic Learning Theory (ALT)*. 2012, pp. 199–213 (cited on page 67).

[30] Daniel Russo et al. 'A Tutorial on Thompson Sampling'. In: *Foundations and Trends in Machine Learning* 11 (2018), pp. 1–96 (cited on page 67).

[31] Hock Peng Chan. 'The multi-armed bandit problem: An efficient nonparametric solution'. In: *The Annals of Statistics* 48.1 (2020), pp. 346–373 (cited on pages 71, 79, 121, 127, 128, 133, 134).

[32] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020 (cited on page 75).

[33] Junya Honda and Akimichi Takemura. 'An Asymptotically Optimal Bandit Algorithm for Bounded Support Models.' In: *COLT*. 2010, pp. 67–79 (cited on pages 75, 77, 84, 85, 91, 97, 99, 100, 107, 110, 111).

[34] Olivier Cappé et al. 'Kullback–Leibler upper confidence bounds for optimal sequential allocation'. In: *Annals of Statistics* 41.3 (2013), pp. 1516–1541 (cited on pages 76, 210, 262, 271).

[35] O-A Maillard. 'Boundary Crossing Probabilities for General Exponential Families'. In: *Mathematical Methods of Statistics* 27.1 (2018), pp. 1–31 (cited on pages 76, 270, 271).

[36] Shubhada Agrawal, Sandeep Juneja, and Wouter M. Koolen. 'Regret Minimization in Heavy-Tailed Bandits'. In: *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*. 2021 (cited on pages 77, 111, 120).

[37] Apostolos N Burnetas and Michael N Katehakis. 'Optimal adaptive policies for sequential allocation problems'. In: *Advances in Applied Mathematics* 17.2 (1996), pp. 122–142 (cited on page 77).

[38] Rajeev Agrawal. 'Sample mean based index policies with O (log n) regret for the multi-armed bandit problem'. In: *Advances in Applied Probability* (1995), pp. 1054–1078 (cited on page 77).

[39] William R Thompson. 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'. In: *Biometrika* 25.3/4 (1933), pp. 285–294 (cited on page 78).

[40] Shipra Agrawal and Navin Goyal. 'Analysis of Thompson Sampling for the Multi-armed Bandit Problem'. In: *Proceedings of the 25th Annual Conference on Learning Theory*. 2012 (cited on pages 78, 94).

[41] N. Korda, E. Kaufmann, and R. Munos. 'Thompson Sampling for one-dimensional Exponential family bandits'. In: *Advances in Neural Information Processing Systems*. 2013 (cited on page 79).

[42] Branislav Kveton et al. 'Garbage in, reward out: Bootstrapping exploration in multi-armed bandits'. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3601–3610 (cited on page 79).

[43] Branislav Kveton et al. 'Randomized Exploration in Generalized Linear Bandits'. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2066–2076 (cited on page 79).

[44] Dorian Baudry, Emilie Kaufmann, and Odalric-Ambrym Maillard. 'Sub-sampling for Efficient Non-Parametric Bandit Exploration'. In: *Advances in Neural Information Processing Systems*. 2020 (cited on page 79).

[45] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 'Improved algorithms for linear stochastic bandits'. In: *Advances in Neural Information Processing Systems*. 2011, pp. 2312–2320 (cited on page 79).

[46] Niranjan Srinivas et al. 'Gaussian process optimization in the bandit setting: no regret and experimental design'. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress. 2010, pp. 1015–1022 (cited on page 79).

[47] Audrey Durand, Odalric-Ambrym Maillard, and Joelle Pineau. 'Streaming kernel regression with provably adaptive mean, variance, and regularization'. In: *arXiv preprint arXiv:1708.00768* (2017) (cited on page 79).

[48] Tor Lattimore and Csaba Szepesvari. 'The End of Optimism? An Asymptotic Analysis of Finite-Armed Linear Bandits'. In: *Artificial Intelligence and Statistics*. 2017, pp. 728–737 (cited on page 79).

[49] Stefan Magureanu, Richard Combes, and Alexandre Proutiere. 'Lipschitz Bandits: Regret Lower Bounds and Optimal Algorithms'. In: *Machine Learning* 35 (2014), pp. 1–25 (cited on page 79).

[50] Tianyu Wang et al. 'Towards Practical Lipschitz Bandits'. In: *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference* (2020). DOI: 10.1145/3412815.3416885 (cited on page 79).

[51] Shiyin Lu et al. 'Optimal Algorithms for Lipschitz Bandits with Heavy-tailed Rewards'. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 4154–4163 (cited on page 79).

[52] Jia Yuan Yu and Shie Mannor. 'Unimodal Bandits'. In: *ICML*. 2011 (cited on page 79).

[53] Richard Combes and Alexandre Proutiere. 'Unimodal Bandits: Regret Lower Bounds and Optimal Algorithms'. In: *International Conference on Machine Learning*. 2014 (cited on pages 79, 153).

[54] Hassan Saber, Pierre Ménard, and Odalric-Ambrym Maillard. 'Forced-exploration free Strategies for Unimodal Bandits'. In: *arXiv preprint arXiv:2006.16569* (2020) (cited on page 79).

[55] Branislav Kveton et al. 'Cascading bandits: Learning to rank in the cascade model'. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 767–776 (cited on page 79).

[56] Stefan Magureanu. 'Efficient Online Learning under Bandit Feedback'. PhD thesis. KTH Royal Institute of Technology, 2018 (cited on page 79).

[57] Thibaut Cuvelier, Richard Combes, and Eric Gourdin. 'Statistically Efficient, Polynomial-Time Algorithms for Combinatorial Semi-Bandits'. In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5.1 (2021), pp. 1–31 (cited on pages 79, 80).

[58] Richard Combes, Stefan Magureanu, and Alexandre Proutiere. 'Minimal exploration in structured stochastic bandits'. In: *Advances in Neural Information Processing Systems*. 2017, pp. 1763–1771 (cited on pages 79, 157, 238).

[59] Thibaut Cuvelier, Richard Combes, and Eric Gourdin. 'Asymptotically optimal strategies for combinatorial semi-bandits in polynomial time'. In: *Algorithmic Learning Theory*. PMLR. 2021, pp. 505–528 (cited on page 80).

[60] G Hoogenboom et al. 'The DSSAT crop modeling ecosystem'. In: *Advances in crop modelling for a sustainable agriculture* (2019), pp. 173–216 (cited on page 92).

[61] Chung-En Tsai, Hao-Chung Cheng, and Yen-Huan Li. 'Online Self-Concordant and Relatively Smooth Minimization, With Applications to Online Portfolio Selection and Learning Quantum States'. In: *International Conference on Algorithmic Learning Theory*. PMLR. 2023, pp. 1481–1483 (cited on page 119).

[62] Thomas M Cover. 'Universal portfolios'. In: *Mathematical finance* 1.1 (1991), pp. 1–29 (cited on page 119).

[63] Thomas M Cover and Erik Ordentlich. 'Universal portfolios with side information'. In: *IEEE Transactions on Information Theory* 42.2 (1996), pp. 348–363 (cited on page 119).

[64] Adam Tauman Kalai and Santosh Vempala. 'Efficient algorithms for universal portfolios'. In: *Journal of Machine Learning Research* (2002), pp. 423–440 (cited on page 119).

[65] Laurent Orseau, Tor Lattimore, and Shane Legg. 'Soft-bayes: Prod for mixtures of experts with log-loss'. In: *International Conference on Algorithmic Learning Theory*. PMLR. 2017, pp. 372–399 (cited on pages 119, 120, 139, 146).

[66] Julian Zimmert, Naman Agarwal, and Satyen Kale. 'Pushing the efficiency-regret Pareto frontier for online learning of portfolios and quantum states'. In: *Conference on Learning Theory*. PMLR. 2022, pp. 182–226 (cited on page 120).

[67] Eyal Gofer and Yishay Mansour. 'Lower bounds on individual sequence regret'. In: *Machine Learning* 103 (2016), pp. 1–26 (cited on page 130).

[68] Cristóbal Guzmán, Nishant Mehta, and Ali Mortazavi. 'Best-case lower bounds in online learning'. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 21923–21934 (cited on page 130).

[69] Dorian Baudry, Patrick Saux, and Odalric-Ambrym Maillard. 'From Optimality to Robustness: Adaptive Re-Sampling Strategies in Stochastic Bandits'. In: *Advances in Neural Information Processing Systems*. 2021 (cited on page 133).

[70] Fabien Pesquerel, Hassan Saber, and Odalric-Ambrym Maillard. 'Stochastic bandits with groups of similar arms'. In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 2021 (cited on pages 151, 154, 163, 167, 173, 260).

[71] Jia Yuan Yu and Shie Mannor. 'Unimodal Bandits.' In: *ICML*. Citeseer. 2011, pp. 41–48 (cited on page 153).

[72] Hassan Saber, Pierre Ménard, and Odalric-Ambrym Maillard. 'Indexed Minimum Empirical Divergence for Unimodal Bandits'. In: *Advances in Neural Information Processing Systems* 34 (2021) (cited on page 153).

[73] Fabien Pesquerel and Odalric-Ambrym Maillard. 'IMED-RL: Regret optimal learning of ergodic Markov decision processes'. In: *NeurIPS 2022-Thirty-sixth Conference on Neural Information Processing Systems*. 2022 (cited on pages 154, 203, 210, 212, 235, 237, 242).

[74] Junya Honda and Akimichi Takemura. 'Non-Asymptotic Analysis of a New Bandit Algorithm for Semi-Bounded Rewards'. In: *Machine Learning* 16 (2015), pp. 3721–3756 (cited on pages 158, 201, 206, 210, 237, 256, 257).

[75] Martin L. Puterman. *Markov Decision Processes — Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, Inc., 1994 (cited on pages 192, 203, 204, 209, 241, 242, 276).

[76] Onésimo Hernández-Lerma and Jean-Bernard Lasserre. *Discrete-Time Markov Control Processes*. Springer New York, 1996 (cited on pages 192, 203, 204).

[77] Romain Hollanders, Jean-Charles Delvenne, and Raphaël M. Jungers. 'The complexity of Policy Iteration is exponential for discounted Markov Decision Processes'. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. 2012, pp. 5997–6002. DOI: 10.1109/CDC.2012.6426485 (cited on page 195).

[78] A.N. Burnetas and M.N. Katehakis. 'Optimal adaptive policies for Markov decision processes'. In: *Mathematics of Operations Research* (1997), pp. 222–255 (cited on pages 196, 197, 201, 205, 210, 211, 215, 216).

[79] Apostolos N. Burnetas and Michael N. Katehakis. 'Optimal adaptive policies for Markov decision processes'. In: *Mathematics of Operations Research* 22.1 (1997), pp. 222–255 (cited on pages 197, 216, 217, 235).

[80] Thomas Jaksch, Ronald Ortner, and Peter Auer. 'Near-optimal regret bounds for reinforcement learning'. In: *Journal of Machine Learning Research* 11.Apr (2010), pp. 1563–1600 (cited on pages 197, 235, 236, 252, 261).

[81] S. Filippi, O. Cappé, and A. Garivier. 'Optimism in Reinforcement Learning and Kullback-Leibler Divergence'. In: *Allerton*. 2010 (cited on pages 197, 235).

[82] Mohammad Sadegh Talebi and Odalric-Ambrym Maillard. 'Variance-Aware Regret Bounds for Undiscounted Reinforcement Learning in MDPs'. In: *Algorithmic Learning Theory*. 2018, pp. 770–805 (cited on pages 197, 198, 202).

[83] Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. 'Near Optimal Exploration-Exploitation in Non-Communicating Markov Decision Processes'. In: *arXiv preprint arXiv:1807.02373* (2018) (cited on page 197).

[84] Z. Zhang and X. Ji. 'Regret minimization for reinforcement learning by evaluating the optimal bias function'. In: *NeurIPS*. 2019 (cited on pages 197, 198).

[85] C.-Y. Wei et al. 'Model-free reinforcement learning in infinite-horizon average-reward Markov decision processes'. In: *ICML*. 2020 (cited on page 197).

[86] Hippolyte Bourel, Odalric Maillard, and Mohammad Sadegh Talebi. 'Tightening exploration in upper confidence reinforcement learning'. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1056–1066 (cited on pages 197, 221, 278).

[87] Aditya Gopalan and Shie Mannor. 'Thompson sampling for learning parameterized markov decision processes'. In: *Conference on Learning Theory*. PMLR. 2015, pp. 861–898 (cited on page 197).

[88] Ronald Ortner. 'Online Regret Bounds for Markov Decision Processes with Deterministic Transitions'. In: *Proceedings of the 20th international conference on Algorithmic Learning Theory*. Ed. by Ricard Gavaldà et al. Vol. 5809. ALT '09, Lecture Notes in Computer Science. Porto, Portugal: Springer, 2009, pp. 123–137 (cited on page 197).

[89] D. Tranos and A. Proutiere. 'Regret analysis in deterministic reinforcement learning'. In: *IEEE CDC*. 2021, pp. 2246–2251 (cited on page 197).

[90] Ian Osband, Daniel Russo, and Benjamin Van Roy. '(More) efficient reinforcement learning via posterior sampling'. In: *Advances in Neural Information Processing Systems* 26 (2013) (cited on pages 197, 198, 221, 278).

[91] M. Azar, I. Osband, and R. Munos. 'Minimax regret bounds for reinforcement learning'. In: *ICML*. 2017, pp. 263–272 (cited on page 197).

[92] Max Simchowitz and Kevin G Jamieson. 'Non-asymptotic gap-dependent regret bounds for tabular mdps'. In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on page 197).

[93] S. Filippi, O. Cappé, and A. Garivier. 'Optimism in reinforcement learning and Kullback-Leibler divergence'. In: *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing*. Monticello, US, 2010 (cited on page 197).

[94] R. Agrawal. 'Sample mean based index policies with O(log n) regret for the multi-armed bandit problem'. In: *Advances in Applied Probability* 27.4 (1995), pp. 1054–1078 (cited on page 197).

[95] P. Auer, N. Cesa-Bianchi, and P. Fischer. 'Finite-time analysis of the multiarmed bandit problem'. In: *Machine Learning* 47.2-3 (2002), pp. 235–256 (cited on page 197).

[96] Peter Auer and Ronald Ortner. 'Logarithmic online regret bounds for undiscounted reinforcement learning'. In: *Proceedings of the 20th conference on advances in Neural Information Processing Systems*. Ed. by Bernhard Schölkopf, John C. Platt, and Thomas Hoffman. NIPS '06. Vancouver, British Columbia, Canada: MIT Press, 2006, pp. 49–56 (cited on page 197).

[97] Thomas Jaksch, Ronald Ortner, and Peter Auer. 'Near-optimal Regret Bounds for Reinforcement Learning'. In: *Journal of Machine Learning Research* 99 (2010), pp. 1563–1600 (cited on pages 197, 203, 211).

[98] Christoph Dann, Tor Lattimore, and Emma Brunskill. 'Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning'. In: *Advances in Neural Information Processing Systems* 30 (2017) (cited on page 197).

[99] W. R. Thompson. 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples'. In: *Biometrika* 25.3/4 (1933), pp. 285–294 (cited on page 197).

[100] Peter L. Bartlett and Ambuj Tewari. 'REGAL: a regularization based algorithm for reinforcement learning in weakly communicating MDPs'. In: *Proceedings of the 25th conference on Uncertainty in Artificial Intelligence*. UAI '09. Montreal, Quebec, Canada: AUAI Press, 2009, pp. 35–42 (cited on page 198).

[101] Ronan Fruit et al. 'Efficient Bias-Span-Constrained Exploration-Exploitation in Reinforcement Learning'. In: *International Conference on Machine Learning*. 2018 (cited on page 198).

[102] J. Qian et al. 'Exploration bonus for regret minimization in discrete and continuous average reward MDPs'. In: *NeurIPS*. 2019 (cited on page 198).

[103] Shipra Agrawal and Randy Jia. 'Optimistic posterior sampling for reinforcement learning: worst-case regret bounds'. In: *Advances in Neural Information Processing Systems* 30 (2017) (cited on pages 198, 257).

[104] Shipra Agrawal and Randy Jia. 'Posterior sampling for reinforcement learning: worst-case regret bounds'. In: *arXiv preprint arXiv:1705.07041* (2017) (cited on page 198).

[105] Apostolos N. Burnetas and Michaël N. Katehakis. 'Optimal adaptive policies for Markov decision processes'. In: *Mathematics of Operations Research* 22 (1 1997), pp. 222–255 (cited on pages 201, 220).

[106] Omar Darwiche Domingues et al. 'Episodic Reinforcement Learning in Finite MDPs: Minimax Lower Bounds Revisited'. In: *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*. Ed. by Vitaly Feldman, Katrina Ligett, and Sivan Sabato. Vol. 132. Proceedings of Machine Learning Research. PMLR, 2021, pp. 578–598 (cited on page 202).

[107] Andrea Zanette and Emma Brunskill. 'Tighter Problem-Dependent Regret Bounds in Reinforcement Learning without Domain Knowledge using Value Function Bounds'. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 7304–7312 (cited on page 202).

[108] Chi Jin et al. 'Is Q-Learning Provably Efficient?' In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018 (cited on page 202).

[109] Ambuj Tewari and Peter L. Bartlett. 'Optimistic linear programming gives logarithmic regret for irreducible MDPs'. In: *Proceedings of the 21st conference on advances in Neural Information Processing Systems*. Ed. by John C. Platt et al. NIPS '07. Vancouver, British Columbia, Canada: MIT Press, 2007 (cited on page 202).

[110] Jungseul Ok, Alexandre Proutiere, and Damianos Tranos. 'Exploration in Structured Reinforcement Learning'. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018 (cited on page 202).

[111] Mengdi Wang. 'Primal-Dual π Learning: Sample Complexity and Sublinear Run Time for Ergodic Markov Decision Problems'. In: *ArXiv* abs/1710.06100 (2017) (cited on page 202).

[112] Yujia Jin and Aaron Sidford. 'Efficiently Solving MDPs with Stochastic Mirror Descent'. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4890–4900 (cited on page 202).

[113] Dimitri P. Bertsekas and Steven E. Shreve. *Stochastic Optimal Control (The Discrete Time Case)*. Academic Press, New York, 1978 (cited on page 209).

[114] O-A. Maillard, R. Munos, and G. Stoltz. 'A finite-time analysis of multi-armed bandits problems with Kullback-Leibler divergences'. In: *Proceedings of the 23rd Annual Conference on Learning Theory*. Budapest, Hungary, 2011 (cited on page 210).

[115] J.M. Borwein and A.S. Lewis. 'Duality relationships for entropy-like minimization problem'. In: *SIAM Journal on Computation and Optimization* 29(2) (1991), pp. 325–338 (cited on page 217).

[116] W. Chen, Y. Wang, and Y. Yuan. 'Combinatorial multi-armed bandit: General framework and applications'. In: *ICML*. 2013, pp. 151–159 (cited on page 238).

[117] R. Combes et al. 'Combinatorial bandits revisited'. In: *NIPS*. 2015 (cited on page 238).

[118] Hassan Saber, Pierre Ménard, and Odalric-Ambrym Maillard. 'Optimal Strategies for Graph-Structured Bandits'. In: *arXiv preprint arXiv:2007.03224* (2020) (cited on page 238).

[119] Winfried K. Grassmann, Michael I. Taksar, and Daniel P. Heyman. 'Regenerative Analysis and Steady State Distributions for Markov Chains'. In: *Oper. Res.* 33 (1985), pp. 1107–1116 (cited on page 277).