THÈSE

En vue de l'obtention du

DOCTORAT Informatique

Délivré par: L'Université de Lille 1 (ED SPI)
Cotutelle avec l'Institut Supérieur de Gestion de Tunis

A Fuzzy Framework for Multi-objective Optimization under Uncertainty

présentée et soutenue le 12 mai 2017 par

Oumayma BAHRI

Jury

M. El-Ghazali TALBI,	Professeur des Universités	Directeur de thèse
Mme Nahla BEN AMOR,	Professeur des Universités	Directrice de thèse
M. Frédéric SAUBION,	Professeur des Universités	Rapporteur
M. Lionel AMODEO,	Professeur des Universités	Rapporteur
Mme. Maria RIFQI,	Maître de Confrénces	Examinateur
Mme. Saoussen KRICHEN,	Professeur des Universités	Examinateur

THESE

Acknowledgments

I would like to thank all the people who have supported me in every step of my PhD study. Without their help and support, this dissertation would not have been achieved successfully.

First and foremost, I would like to single out my supervisors: I am deeply grateful to Pr. El-Ghazali Talbi, for his constant encouragement, valuable guidance and for all the opportunities he has provided me throughout my doctoral stages at INRIA laboratory and participation in many international conferences. I am very thankful to Pr. Nahla Ben Amor for her continuous supervision, motivation and the life lessons that she has taught me. It has been an honour and a privilege working with them both since my master degree.

I would also thank all members of my thesis committee: Pr. Frédéric Saubion, Pr. Lionel Amodeo, Dr. Maria Rifqi and Pr. Saoussen Krichen for their interest in my work, insightful comments and interesting questions. Additionally, I would like to thank my fellow lab-mates, colleagues and the staff at INRIA Lille for the friendly atmosphere, cooperation and feedback. Special thanks go to my close friends from Tunisia and Lille for their unconditional help, patience and for being there when I needed them the most.

Last but not the least, I record my deep and utmost gratitude to my family for their selfless support and encouragement during the entire period of my graduate studies. I am forever indebted to my mom and my grandmother whose unceasing prayers and sacrifices are behind every success of my life.

Finally, I would like to thank everybody who in one way or another contributed in the completion of this work, as well as expressing my apology that I could not mention personally one by one.

Contents

List of Figures

List of Tables

List of Algorithms

General Introduction

This thesis falls under the field of combinatorial multi-objective optimization, in particularly in the study of multi-objective problems under uncertainty. It is conducted within the LARODEC laboratory of ISG Tunis and the DOLPHIN research team of the CRISTAL laboratory of Lille1 University.

A Multi-objective Optimization Problem (MOP) is characterized by multiple conflicting objectives to be minimized or maximized simultaneously with respect to a set of constraints. For example, the decision of a new car purchase can be influenced by several incommensurable criteria such as the price, the safety options, the driving comforts, the fuel consumption and so forth. Usually, there is no unique solution that is optimal in terms of all these criteria at the same time, but rather many incomparable car models. Hence, the resolution of a MOP consists in finding a set of best compromise solutions between the different objectives. This set represents, in the objective space, the Pareto front from which the decision maker will subsequently choose one final alternative to realize. A wide range methods and techniques exist in the literature for solving combinatorial MOPs (Talbi, 2009 ; Gandibleux et al., 2004 ; Liefooghe, 2009). However, such problems are often NP-hard, large-scale and their resolution cannot be performed in an exact manner within a reasonable time. Thus, different approximate methods are extensively used to deal with them such as metaheuristics. In last years, these latters have gained great popularity due to their efficiency to achieve good feasible solutions in a reasonable time.

In addition, when dealing with practical applications, the massive amounts of data are generally associated with unavoidable imperfections. In other words, real-life problems are strongly connected to some uncertainties in inputs, parameters and environmental data. In fact, uncertain data may result from using unreliable information sources such as bad analysis or interpretation processes, faulty description, data incompleteness, ambiguity in perception and so on. Besides, it may be caused by poor decision maker opinions due to any lack of its background knowledge, absence of information or even difficulty of giving perfect qualification for some costly situations. Depending on the nature of uncertainty and the problem context, several tools have been proposed such as, probability theory (Steele, 1997), fuzzy set theory (Zadeh, 1965), evidence theory (Shafer et al., 1976) and possibility theory (Dubois & Prade, 1988). These theories have started to play an important role in treating uncertainty in the decision making problems. Indeed, the literature exposes many modeling approaches for reasoning under uncertainty in many single-objective optimization problems. Nevertheless, this aspect is, until today, not well considered in the multi-objective optimization context that reflects more reality in every domain of our lives.

In this view, our research works will be interested in handling combinatorial multi-objective problems with uncertain inputs data. However, propagation of inputs uncertainty through the optimization process rises as a major obstacle since it hampers the identification of efficient solutions. Most methods reported in this context simply transform such problems into crisp or single-objective equivalents. In consequence, such transformation may affect the problem results and decisions making. Therefore it is necessary to deal with the uncertain multi-objective problem in its original version without ignoring any of its characteristics.

In this thesis, we are focused on a specific type of uncertainty defined by vagueness and ambiguity, where fuzzy sets serve as modeling tool. More precisely, our aim is to address issues related to the effects of uncertainty propagation in multi-objective setting:

1. *Where does the effects of uncertainty propagation occur?*
2. *How to define new optimality concepts or to extend classical ones to our uncertain multi-objective context?*
3. *How to design optimization algorithms for solving such very complex problems?*
4. *What are the consequences in term of robustness?*

To deal with previous questions, we firstly present a global view the different contributions in this field. Then, we propose a new Pareto-based approach to solve any multi-objective problem under uncertainty (Bahri et al., 2014a).

The idea is to define new Pareto dominance relations for ranking the generated uncertain valued objectives. Thereafter, we suggest an extension of the most population-based metaheuristics by incorporating uncertainty into their search process (Bahri et al., 2014b). The second part of our work is devoted to the sensitivity analysis of results. Thus, we provide a robustness methodology for evaluating the performance of our extended algorithms in presence of uncertainty (Bahri et al., 2016).

The document is structured in two parts (cf. Figure 1). The first one (Chapter 1 and 2) covers the state of the art in multi-objective optimization in both deterministic and uncertain contexts. The second part (Chapter 3 to 5) presents our research works and main contributions for the case of fuzzy multi-objective problems. The detailed organization of chapters is described as follows:

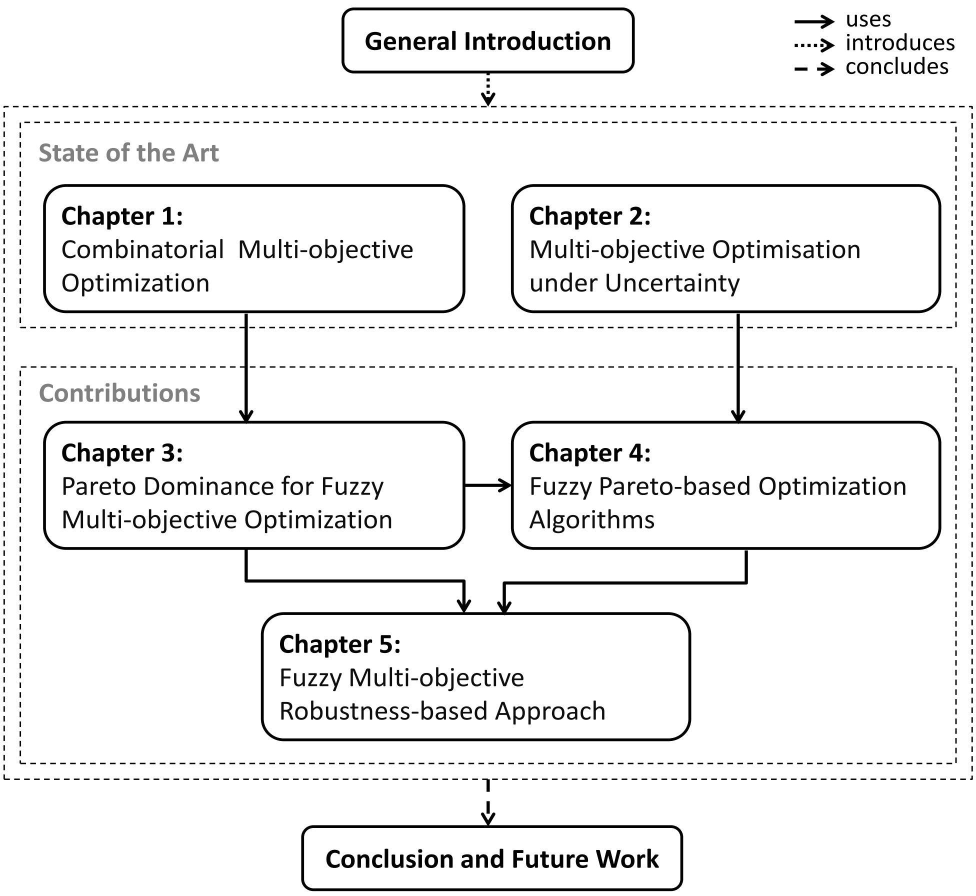

Figure 1 – Thesis structure

Chapter 1 presents a survey of the necessary background about multi-objective optimization. Initially, it gives an overview of basic concepts, notions and definitions essential for good comprehension of the global document. Then, it outlines the different multi-objective methods, namely metaheuristics in which we are interested. The chapter also presents the performance assessment and development tools. Finally, it reports some applications of MOPs, focusing on the vehicle routing problem as an example.

Chapter 2 addresses the uncertain context of multi-objective optimization. It introduces the concept of uncertainty and briefly reviews the different theories for modelling it. In addition, it gives a general description of uncertain MOPs and discusses existing approaches related to such problems. The chapter ends with a definition of an uncertain variant of the vehicle routing problem treated in our case.

Chapter 3 contains our main contribution in terms of Pareto optimality under uncertainty. It concentrates on the fuzzy modelling of uncertain MOPs and analyses the impacts of fuzziness on the outcomes. In particular, it examines the case of MOP with fuzzy valued objectives and provides a formulation of the problem. After that, it presents our proposal about how to define optimality in a fuzzy setting, namely the definition of new Pareto dominance relations between triangular fuzzy numbers. These relations are illustrated by some numerical examples and their relevance are then discussed.

Chapter 4 brings the algorithmic contributions. It focuses on the extension of multi-objective evolutionary algorithms, especially the Pareto based ones, to our fuzzy context. In fact, the chapter starts by recalling the different steps of classical algorithms, describing their generic extensions and presenting step by step the extension of two well known algorithms. Thereafter, it demonstrates their application on a vehicle routing problem. An experimental study is finally dressed for validating the performance of our proposals.

Chapter 5 covers the robustness analysis of our work and gives a survey of this concept. After briefly reviewing the existing robustness studies, the chapter gives rise to a new robust approach for fuzzy multi-objective problems. The effectiveness of the proposed approach is demonstrated by a set of test experiments conducted on the same vehicle routing problem.

We conclude this manuscript by different research perspectives that seem interesting to continue this work.

Part I

State of the Art

Chapter 1

Combinatorial Multi-objective Optimization

Contents

1.1 Introduction

Multi-objective optimization is a well-studied research field encountered in many academic and industrial applications such as in engineering, manufacturing and logistics. It rises as a salient paradigm of decision making in which the decision maker is always confronted with different conflicting objectives. For instance, a good purchase choice is associated with several factors like the price, the durability and the quality, etc. Hence, the most common purpose is to choose the best trade-off among all these factors. In that sense, it is practically impossible to find a single solution that optimize all predefined objectives at the same time but rather many efficient and incomparable solutions. Thus, the challenge of solving a combinatorial multi-objective problem lies in the difficulty to find a set of optimal solutions. Then if the number of multiple objectives and/or decision variables grows, the problem becomes much more complex. A wide variety of resolution methods and techniques have been designed according to the complexity and way of solving such a problems (Talbi, 2009 ; Liefooghe, 2009 ; Hwang & Mausud, 2012).

This chapter presents fundamental prerequisites necessary for understanding this research field as well as the multi-objective concepts, the resolution methods and the performance assessment. The chapter is organized as follows. In Section 1.2, we give a brief introduction of some basic multi-objective concepts. In Section 1.3, we highlight the main classes of multi-objective resolution methods and in Section 1.4, we recall the major domains of applications. Finally, Section 1.5 presents a combinatorial multi-objective routing problem that we will treat later.

1.2 Basics of multi-objective optimization

This section covers general concepts, background definitions and notations related to multi-objective optimization.

1.2.1 Terminology

An *optimization problem* involves making a best decision alternative between various possible ones. The chosen alternative should consider all available constraints and optimizes the problem objective. The person(s) responsible for taking such choices are called *decision maker(s)(DM)*. The process of identifying and deciding which alternative to choose at every step of the problem solving is known as *decision making process*. This process consists of several steps which are mainly: formulating and defining the problem, developing a quantitative model for it, optimizing the model, validating the obtained solutions and implementing one solution.

The term *decision space* represents the feasible region or search domain used by decision makers to compare and choose among a range of solutions. The *objective space* is defined by the set of outcomes called objective vectors. Every point or solution in the *decision space* can map to a point in the *objective space* which gives an evaluation of its quality. In the table 1.1, we outline the most common notations to unify modeling for all optimization problems.

Table 1.1 – Basic notations

Notation	**Explanation**
S	Decision space or feasible solution region
Y	Objective space
d	Number of decision variables
n	Number of objectives
$x = (x_1, .., x_d)$	Feasible solutions
$y = (y_1, .., y_n)$	Objective costs or fitness values
$f(x)$	Single Objective function
$F(x) = (f_1(x), \ldots, f_n(x))$	Objectives vector
$g_i(x) \leq b_i$	Set of inequality constraints
$h_i(x) = 0$	Set of equality constraints

Formally, an optimization problem is a mathematical program of the form:

$$Opt\ F(x) \quad s.t. \quad x \in S \tag{1.1}$$

where Opt indicates the optimization sense (i.e. maximization or minimization) of the objective vector $F(x)$. If $F(x)$ corresponds to a unique objective function ($F(x) = f(x)$), the problem is termed *single-objective optimization problem (SOP)*. Otherwise, if it encapsulates two or more objectives

$(F(x) = (f_1(x), \ldots, f_n(x)))$, the problem is called *multi-objective optimization problem (MOP)*. The decision variables x are numerical quantities from the feasible decision space S. These quantities can be either continuous or discrete variables depending on the problem structure. In consequence, optimization problems are continuous when all decision variables are real values, whereas they are called combinatorial when the variables take values from a finite discrete set. Notice that, combinatorial optimization problems tend to be harder to solve than continuous problems. In fact, a great number of standard benchmarks are available for the continuous case because of its simplicity and the availability of practical cases that can be formulated in linear forms. On the other hand, there is a lack of benchmarks for combinatorial optimization problems, which are usually difficult and require an exponential time to be solved.

1.2.2 Definitions

Multi-objective optimization is the process of optimizing systematically and simultaneously two or more conflicting objectives subject to certain constraints. Besides, contrary to the single-objective case, multi-objective optimization does not restrict to find a unique global solution but it aims to find a set of efficient solutions. The following definitions gives more details about multi-objective concepts. Without a loss of generality, we only consider, throughout this thesis, combinatorial *minimization problems* since the maximization can easily be deduced based on duality principle (i.e. maximizing an objective function (f) is equivalent to minimizing $(-f)$).

Definition 1.1. ***Multi-objective optimization problem***
A multi-objective optimization problem (MOP) *is defined as follows:*

$$\min \ F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \quad s.t. \quad x \in S \tag{1.2}$$

where $F(x)$ is the vector of n objectives to be minimized: $\forall i = \{1, \ldots, n\}, f_i : \mathbb{R}^n \to \mathbb{R}$, the number of these objectives must be equal to at least two ($n \geq 2$).

In a combinatorial MOP, the feasible region S becomes a discrete set of solutions. Besides, $F(x)$ maps the decision variables x from the decision space to the objective space. As shown in Figure 1.1, to each decision variable $x \in S$ is assigned a cost function $y \in Y$ that evaluates its quality:

$$F : S \to Y \subseteq \mathbb{R}^n, \quad F(x) = y = (y_1, \ldots, y_n) \tag{1.3}$$

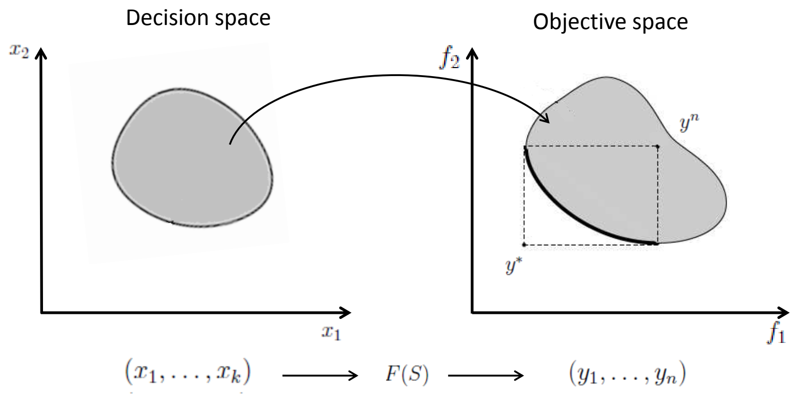

Figure 1.1 – Decision space vs Objective space

where $Y = F(S)$ represents the feasible points (solutions) in the objective space and $y_i = f_i(x)$ is a point of this space that represents the solution quality or fitness.

The objectives are often in conflict with each other (e.g., minimize cost and maximize profit), so that it is practically impossible to have a unique solution x^* optimal for all the objectives: $\forall i \in 1..n, \forall x^* \in S, f_i(x^*) \leq f_i(x)$). Therefore, a MOP may have a set of efficient solutions and other concepts of optimality should be applied to define a partial order relation between the solutions, known as Pareto optimality. The Pareto optimality definition comes from the dominance relations proposed initially by (Edgeworth, 1881) and extended by (Pareto, 1971).

Definition 1.2. ***Pareto dominance***
Let $y = \{y_i, \ldots, y_n\}, y' = \{y'_i, \ldots, y'_n\} \in Y$ be two different objective vectors. $\vec{y}$ dominates $\vec{y'}$, denoted by $y \prec y'$, iff:

$$\forall i \in \{1, \ldots, n\} : y_i \leq y'_i \wedge \exists j \in \{1, \ldots, n\} : y_j < y'_j \tag{1.4}$$

This means that no component of $\vec{y'}$ is smaller than the corresponding component of $\vec{y}$ and at least one component of $\vec{y}$ is strictly smaller.

Definition 1.3. ***Pareto optimal solution***
A solution x^* is said to be Pareto optimal (i.e. efficient, non-dominated and non-inferior solution) iff:

$$\forall x \in S, F(x) \nprec F(x^*) \tag{1.5}$$

This means that for every solution x, $F(x)$ does not dominate $F(x^*)$, so that it is not possible to find a solution that improves the performances of an objective without decreasing the quality of at least another objective.

Definition 1.4. ***Pareto optimal set***
A Pareto optimal set S^* is the set of Pareto optimal solutions defined by:

$$S^* = \{x \in S / \exists x' \in S, F(x') \prec F(x)\} \tag{1.6}$$

Definition 1.5. ***Pareto front***
The image of the Pareto optimal set S^* in the objective space is called Pareto front PF^* defined by:

$$PF^* = \{F(x), x \in S^*\} \tag{1.7}$$

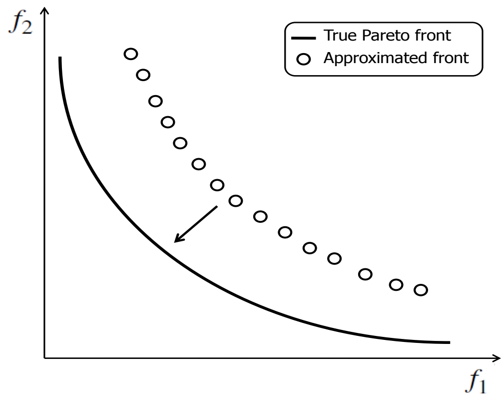

Figure 1.2 – Example of Pareto front

Finding the Pareto front is known to be a difficult task (Kao & Jacobson, 2008). In fact, identifying the Pareto optimal (non-dominated) solutions is generally an NP-hard problem. Thus, the main goal is to identify an approximation of the Pareto optimal set, from which the decision maker can choose a best solution based on the current situation. A good approximation of Pareto solutions should satisfy two properties:

1. Convergence and closeness to the Pareto optimal front.
2. Diversification of solutions around the Pareto front.

Figure 1.2 illustrates an example of approximated front having a very good spread of solutions (uniform diversity) but a bad convergence, since solutions are far from the true Pareto front.

1.2.3 Multi-objective optimization and decision making

In MOP, the decision making process involves finding a best alternative decision in the face of multiple conflicting objectives. Indeed, solving such a problem leads to generate a set of Pareto optimal solutions that help the decision maker (DM) to select his preferred solution. For this reason, different decision making support techniques are developed for introducing DM preferences in the optimization process (Coello, 2000 ; Rachmawati & Dipti, 2006 ; Hwang & Mausud, 2012). The existing techniques can be classified into three categories according to the moment when the DM is consulted: a priori, a posterior or interactively during the search:

In the category of **a-priori techniques**, preferences are provided by the DM before starting the optimization process. Indeed, the DM may give a rank order of the objectives based on their importance or some preference levels. Then, a Pareto optimal solution is generated by a single run to satisfy as much as possible the DM preferences. Despite the rapidity of these methods, it should be noticed that if the DM is not satisfied with the solution(s) found, the process is simply restarted with another compromise.

Interactive techniques require the presence of the DM throughout the search process. It is based on an iterative algorithm for the progressive articulation of preference information. More precisely, at each iteration of optimization process, the DM specifies his preferences to a particular solution generated. After a number of interactions, the optimization process converges to the final preferred solution.

For **a-posteriori techniques**, preference are not required before or during the optimization process. In fact, after the generation of Pareto optimal solutions, the DM implicitly chooses his adequate solution. Hence, it is recommended to enable the DM exploring the whole Pareto set (or a good approximation of this set) according to his preferences. This helps him to a have complete knowledge of the Pareto front and better apprehend the arbitration operated between the different criteria.

In our study, we focused on the category of a-posteriori techniques where the DM preferences are considered only after the overall optimization process. This latter must be powerful in order to obtain a well-distributed Pareto optimal set that guides the DM to make his best choice.

1.3 Multi-objective resolution methods

Several resolution methods have been proposed to deal with multi-objective problems depending on their complexity (Liefooghe, 2009 ; Talbi, 2009). Indeed, multi-objective problems are typically NP-hard and their difficulty increases with the growth of the number of variables and objectives (Curry & Dagli, 2014). Various resolution methods have been developed in order to tackle these problems. The existing methods can be classified into different classes according to their effectiveness, applicability and problems complexity (Liefooghe, 2009). Figure 1.3 describes a classification of the main multi-objective methods (Talbi, 2009).

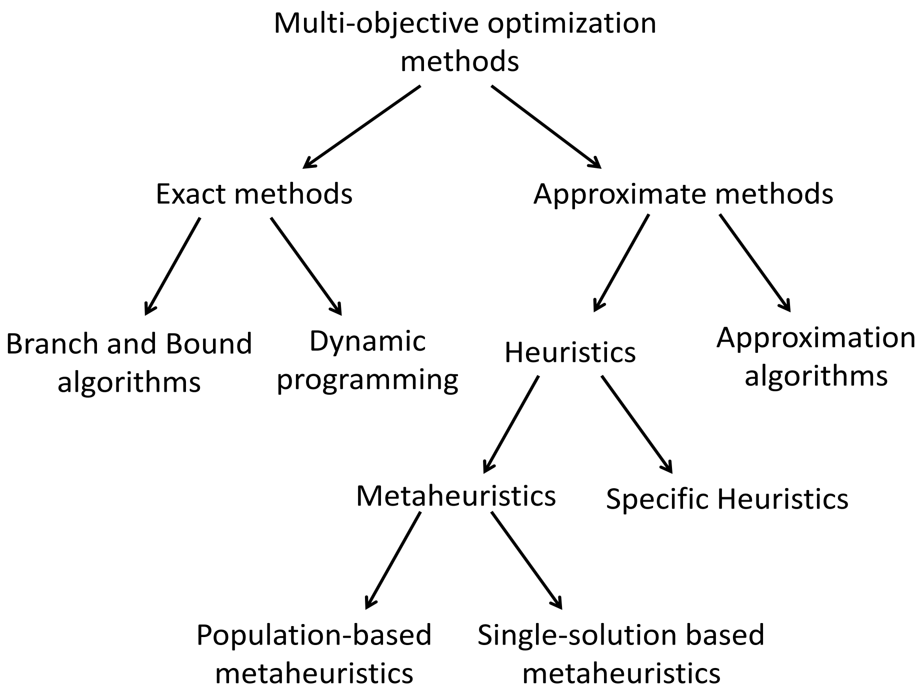

Figure 1.3 – Taxonomy of multi-objective methods

Exact methods such as *branch and bound algorithms* (Mavrotas & Diakoulaki, 1998), *dynamic programming* (Carraway et al., 1980), etc, have been widely used to solve bi-objective optimizations problems. In fact, they are known to be effective for small-sized problems. Their application leads often to generate solutions whose optimality is guaranteed but are time consuming. Hence, as soon as the number of objective functions or size of the problem increase, exact methods become ineffective.

Alternatively, **approximate methods** have been designed to solve large-size problems. The aim of these methods is to avoid complete enumeration of the solution space and to provide approximate solutions (i.e. near optimal solutions) in a reasonable computational time. In this case, there is no guarantee of reaching optimality. For instance, *approximation algorithms* (Segev, 2007) have been used to achieve provable quality of solutions with respect to the optimal, whereas *heuristics* have been intended to find promising solutions but do not have an optimality guarantee on the obtained solutions.

Metaheuristics (Gandibleux et al., 2004 ; Talbi, 2009) are the most practical approximate methods designed to solve almost all large and complex multi-objective problems. Unlike specific heuristics which are problem-dependent methods, metaheuristics are high-level problem-independent methods that can be used as a general methodology or guiding strategy in designing underlying heuristics. They have received more and more popularity over the years due to their efficiency and ability to offer a good trade-off between solutions quality and computing time. A common classification of metaheuristics is based on two design criteria: *Diversification (or exploration)* and *Intensification (or exploitation)*:

— *Single based metaheuristics* (S-META): are intensification-based methods which focus on the search in a local region by exploiting the best found solution. In this case, a single solution is often handled and transformed during the optimization process. Examples of S-META are the *local search* (Jaszkiewicz, 2002) and *tabu search* algorithms (Hansen, 1997).

— *Population based metaheuristics* (P-META): are diversification-based methods that allow to globally explore the search process and then to generate diverse solutions. They are referred to as nature-inspired metaheuristics, because they have been designed by drawing inspiration from nature. In this case, a set of solutions called population is evolved over many generations until reaching a good solution quality. The most popular P-METAs are *multi-objective evolutionary algorithms* (MOEAs) (Coello et al., 2005) that imitate some natural characteristics from biological evolution such as recombination and mutation. This type of algorithms is at the core of our work.

1.3.1 Common concepts in MOEAs

Multi-objective evolutionary computation becomes nowadays a very active research field reflected by a rapidly increasing number of publications (Veldhuizen & Lamont, 1998 ; Deb, 2001 ; Coello et al., 2005). In particular, a *multi-objective evolutionary algorithm* (MOEA) is one of the most powerful stochastic search methodology that may handle large-scale MOPs and generate multiple optimal solutions in one single optimization run. The optimization process of an MOEA is based on the iterative adaptation of a population until a pre-specified optimization goal as illustrated in Algorithm 1.1.

Algorithm 1.1 General MOEA template

Input : Initial population P_0
Output: Best population found
begin
  $Initialize(P_0)$;
  $Create(A)$;
  $t = 0$;
  while *Not-Termination-Criteria* (P_t, t_{max}) **do**
    $Evaluate(P_t, A)$;
    $Update(P_t, A)$;
    $P'_t := Select(P_t \cup A)$;
    $P'_t := Reproduction(P'_t)$;
    $P_{t+1} := Replace(P_t, P'_t)$; $t = t + 1$;
  end
end

The process starts with an initial population generated randomly. This is followed by the evaluation of candidate solutions in the population and the creation of an external population or archive to maintain only the non-dominated solutions. Thereafter, a selection process is performed. However, every solution is usually associated with a fitness value indicating its suitability to the problem. The solutions with better fitness are selected and then reproduced using variation operators (e.g., crossover and mutation) to generate new offsprings. At the last step, a replacement scheme is applied to determine which solutions in the population will survive from the offsprings and the parents. These steps are iterated until a stopping criteria hold. Besides, two major questions must be addressed when designing an MOEA:

1- *How to guide the search process towards the Pareto optimal set?*

2- *How to achieve a diverse and well distributed set of non-dominated solutions?*

Therefore, the aim is to obtain a set of efficient solutions that fulfills the requirements of convergence and uniform diversity. To address these criteria, all metaheuristics, especially MOEAs, are based on three main components (Talbi, 2009): (a) Fitness assignment, (b) Diversity preserving and (c) Elitism.

1.3.1.1 Fitness assignment

This component allows to guide the search algorithm toward Pareto optimal solutions for a better convergence. It associates a scalar-valued fitness to each objective vector which represents a solution of the search space. The fitness value measures the quality of solutions. According to the fitness assignment strategy, an interesting classification of metaheuristic approaches can be highlighted:

- **Scalar-based approaches**
  These approaches are based on the MOP transformation into one or many single-objective problems. Therefore, single-objective methods can simply be used to solve the problem and the optimization process leads naturally to find only one solution. Hence, there is a deterioration and loss of the diversity of obtained solutions. This class of approaches includes, for example, the aggregation methods that use a weighted sum function to combine all the objectives into a single objective function (Hwang & Mausud, 2012). In this case, the decision maker should have necessary knowledge of his problem.
- **Criterion-based approaches**
  The main idea of criterion-based approaches is to perform the search space by treating the various objectives separately. These approaches have often struggled to find compromise solutions since they focus on the extreme portions of the Pareto front. For example, we can mention the lexicographic methods that give a priority order on the objectives to be addressed (Fishburn, 1974) and the algorithm VEGA (Vector Evaluated Genetic Algorithm) based on a parallel selection process (Shaffer, 1985).

- **Indicator-based approaches**
  These approaches are based on the use of performance quality indicators to drive the search toward the optimal Pareto front. In particular, the optimization goal is defined in terms of a binary quality indicator I that can be regarded as an extension of the Pareto dominance. Formally, the main goal may be defined as $\min I(A, R)$, where the quantity $I(A, R)$ reflects the difference in quality between an approximation set A and a reference set R (which can be the Pareto optimal set). One of the major advantages of indicator-based approaches is that no diversity maintenance is required since it is implicitly considered in the performance definition. For example, we can cite the algorithm IBEA (Indicator-based Multi-objective Evolutionary Algorithm) (Zitzler & Künzli, 2004).
- **Dominance-based approaches**
  These approaches rely on the concept of dominance optimality to guide the search process. The objective vectors of solutions are scalarized using a dominance relation. The advantage of such approaches is that all the predefined objectives are treated equitably without any distinction or preference on one objective. This means that there is no transformation of the MOP into a single-objective problem. However, most of dominance-based approaches use the notion of Pareto optimality within multi-objective evolutionary algorithms (MOEAs), since these latter are able to provide a diverse set of optimal solutions in one single run. In these algorithms, dominance-based ranking strategies are usually applied to establish an order between the solutions. The most commonly used ones are:
  - *Dominance-rank:* In this strategy, each solution is associated with a rank representing its fitness/quality. This rank is the number of solutions in the population that dominates the concerned one. The dominance-rank strategy was firstly used in the MOGA algorithm (Multi-objective Genetic Algorithm) (Fonesca et al., 2003).
  - *Dominance-depth:* This strategy consists in dividing the population of solutions into different fronts. In fact, solutions in the first front f_1 belong to the best non-dominated set that receive rank 1, those of the second front f_2 (with rank 2) are non-dominated except by solutions of f_1, and so on. In a general way, the depth of a solution corresponds to the depth of the front to which it belongs.

This strategy is employed in the NSGA-II algorithm (Deb et al., 2002).

- *Dominance-count:* The solution fitness value in this strategy corresponds to its dominance-count. This quantity represents the number of solutions which are dominated by the concerned one. A combination with the aforementioned strategies can be applied in this case. For instance, in the SPEA2 algorithm (Zitzler et al., 2001), both strategies of dominance-rank and dominance-count are used.

1.3.1.2 Diversity preserving

This component is used to generate a diverse set of Pareto optimal solutions so as to explore the whole decision or objective space on a global scale. However, a loss of diversity is often observed in many metaheuristics, especially in the P-META. This may be related to a bad initial choice of population, biased sampling or stagnation during the search progress. To overcome this problem, diversity preservation techniques must be incorporated into the metaheuristics such as the kernel method, the nearest neighbor technique, etc. The basic idea of these techniques is to measure the dispersion in a given population and then to deteriorate solutions that have a high density in their neighborhood. The dispersion is often computed using distance measures like the crowding distance. More details are given in (Talbi, 2009).

1.3.1.3 Elitism

This concept plays an important role in the performance of a metaheuristic. It consists in the preservation and use of elite solutions in order to improve the search performance. In general terms, elitism allows the best solutions (e.g., Pareto optimal solutions) generated during the search to be stored into an elite population, called archive. The archived high-quality solutions can then be used to generate new solutions. Thus, elitism helps to achieve faster and robust convergence toward he Pareto front. Finally, the strategy applied in updating the archive may depend on a number of different criteria such as the size of archive, the number of Pareto solutions, etc.

According to the techniques used for fitness assignment, diversity preserving and elitism, the MOEAs can be classified into different families. For instance, depending on the manner in which the archiving process is performed, they can be divided into two groups, namely *Non-elitist* and *Elitist* MOEAs. Another type of classification is the one based on the optimality techniques that are used, namely *Pareto-based* and *Non-Pareto-based* MOEAs. In fact, the class of Pareto-based MOEAs (or PMOEAs) rely on the concept of dominance and Pareto optimality to deal with MOPs directly and without any transformation. This class concerns us in this thesis. The most popular examples of PMOEAs are the algorithms: SPEA2 (*Strength Pareto Evolutionary Approach 2)* (Zitzler et al., 2001) and NSGA-II *(Non-dominated Sorting Genetic Algorithm II)* (Deb et al., 2002). Finally, a discussion of the various families and techniques of MOEAs can be found in (Zitzler et al., 2000 ; Coello et al., 2007).

1.3.2 Software frameworks

Several frameworks dedicated to combinatorial multi-objective optimization have been proposed in the literature such as MOEA (K. Tan et al., 2000), jMetal (Durillo & Nebro, 2011), PISA (Bleuler et al., 2003), ParadisEO (Cahon et al., 2004), etc. These frameworks are distinguished according to the programming language, the availability of metaheuristics, metrics and parallel features. However, most of them are initially focused on single optimization, providing only extensions to the multi-objective case.

Thereby, the major metaheuristics, parallel models and well-known quality metrics are all provided at once only within the ParadisEO framework [1] (Cahon et al., 2004). Furthermore, ParadisEO is an open source and white-box object-oriented framework that offers flexible developments for almost all types of optimization problems and algorithms (e.g., mono-objective, multi-objective, parallel, hybrid, etc.). ParadisEO is represented as a generic software based on the C++ templates and designed to be portable across both Unix and Windows systems. This framework provides a rich set of template-based classes which can be used by both non-specialist and optimization experts. It can also be used by researchers to develop their own algorithms, taking the advantages of code-reusing and extensibility. As shown in Figure 1.4, it is mainly composed of four modules:

1- ParadisEO-EO (Evolving Object) dedicated to the development of

1. http://paradiseo.gforge.inria.fr

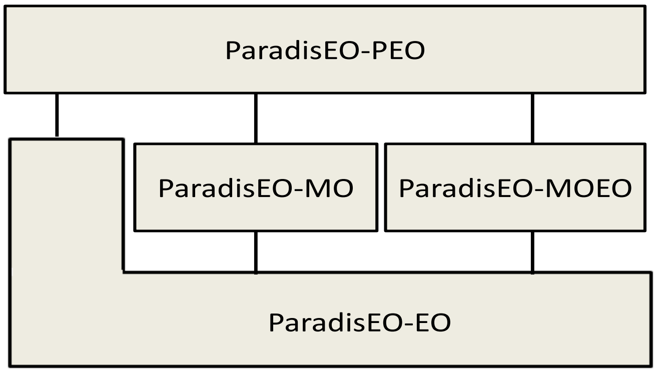

Figure 1.4 – ParadisEO modules

population-based metaheuristics (P-META) in a single-objective optimization.

2- ParadisEO-MO (Moving Object) dedicated to the development of single solution-based metaheuristics (S-META).

3- ParadisEO-MOEO (Multi-Objective EO) dedicated to the design of multi-objective metaheuristics, especially for the P-META. It provides the most common multi-objective techniques in order to facilitate the use or extension of popular MOEAs such as IBEA, SPEA and NSGA algorithms.

4- ParadisEO-PEO dedicated to the design of parallel and distributed metaheuristics.

All the contributions in this thesis have been implemented using the MOEO module of the ParadisEO framework.

1.4 Performance analysis

Performance analysis is an important and essential task when evaluating and validating any multi-objective optimization method. In fact, a rigorous manner to assess the performance of different metaheuristics consists of the following steps: First, it is necessary to define the experimentation goals such as examining the method results, comparing two methods, etc. Then, performance indicators or metrics must be selected and performed to statistically analyze the generated results. Finally, the performance results must be presented in a comprehensive way and an analysis is carried out considering the predefined goals. This section addresses the quality indicators and the statistical validation dedicated to the multi-objective optimization approaches.

1.4.1 Quality indicators

Several quality indicators have been proposed in the literature in order to assess the performance of two sets or approximations of Pareto solutions (Zitzler et al., 2003 ; Knowles et al., 2006). These latter can be classified depending on several features:

- *Arity (Unary/Binary)*: Unary indicators assign to each approximated Pareto front a scalar value that represents its quality but cannot determine whether a Pareto approximation is better that another. Whereas, the binary indicators allow to measure and compare directly the performance of two approximated Pareto fronts.
- *Performance goals*: Quality indicators can be distinguished according to their performance goals: convergence (or closeness) toward the optimal Pareto front or/and diversity of solutions along the front.
- *Required parameters*: Many quality indicators require the definition of some parameters. These indicators can be divided into those that need accurate knowledge (e.g., Optimal Pareto front, ideal point, ...) and those who require reference information provided by the user (e.g., reference set, reference point, etc.).

Table 2.2 summarizes the main features of some well studied quality indicators namely the arity type, the goal achieved by the indicator and the parameters required for its calculation. Usually, an analyzer can use a set of indicators to ensure the performance assessment. In our experimentation, we mainly use the two indicators: *Hypervolume* and *Epsilon* in order to evaluate the designed optimizers. In the following, A and B will denote two approximation sets found by multi-objective metaheuristics, $Z_n^\star$ denotes the reference set (the optimal front).

1.4.1.1 Epsilon indicator

The family of epsilon indicators has been introduced by (Zitzler et al., 2003) as a metric able to measure approximations quality in term of conver-

Table 1.2 – Overview of some quality indicators

Indicator	**Name / Reference**	**Arity**	**Perf. Goal**	**Parameters**
I_C	Contribution (Meunier et al., 2000)	Binary	Convergence	-
I_E	Entropy (Basseur et al., 2002)	Unary	Diversity	Niches number
I_{ER}	Error Ratio (Veldhuizen & Lamont, 1998)	Unary	Convergence/ Diversity	Exact Pareto front
I^1_ϵ, I_ϵ	Epsilon (Zitzler et al., 2003)	Unary/ Binary	Convergence	-
I_H, I^-_H	Hypervolume (Zitzler et al., 2003)	Unary/ Binary	Convergence/ Diversity	Reference point
I_{MS}	Max. Spread (Zitzler et al., 2003)	Unary	Diversity	-
I_S	Spacing (Knowles & Corne, 2002)	Unary	Diversity	-
R_2, R_3	R-metric (Hansen & Jaszkiewicz, 1998)	Binary	Convergence/ Diversity	Reference set, Ideal point

gence. It is based mainly on the notion of epsilon efficiency and comprises two versions in both unary and binary forms: *Multiplicative epsilon indicator* and *Additive epsilon indicator*. The additive version is that we will used. First, the binary additive ϵ-indicator, denoted $I_{\epsilon+}$, gives the minimum factor by which an approximation set A has to be translated in the criterion space to weakly dominate an approximation B. It is formally expressed by:

$$I_{\epsilon+}(A, B) = \min_{\epsilon}\{\forall x \in B, \exists x' \in A : x' \preccurlyeq_{\epsilon+} x\}. \tag{1.8}$$

As an extension of the $I_{\epsilon+}$, the unary additive epsilon indicator, denoted $I^1_{\epsilon+}$, may be defined as follows:

$$I^1_{\epsilon+}(A) = I_{\epsilon+}(A, Z^\star_n). \tag{1.9}$$

where $Z^\star_n$ is the reference set and n is the number of objectives. Otherwise, an $I^1_{\epsilon+}$ value less than or equal to 0 implies that the considered approximation A weakly dominates the reference set $Z^\star_n$. Then as mentioned above, the goal of this indicator is to evaluate the closeness of an approximation to the reference set or the true Pareto front. Therefore in the case of two approximations with similar quality in terms of convergence, the most diversified one is preferred. Finally, this indicator is sensitive with respect to the objective functions.

1.4.1.2 Hypervolume indicator

The hypervolume, proposed by (Zitzler et al., 2003), is considered as one of the few indicators that measure the approximation quality in terms of convergence and diversity at a same time. Typically, it belongs to the class of hybrid metrics and exists in both unary and binary forms.

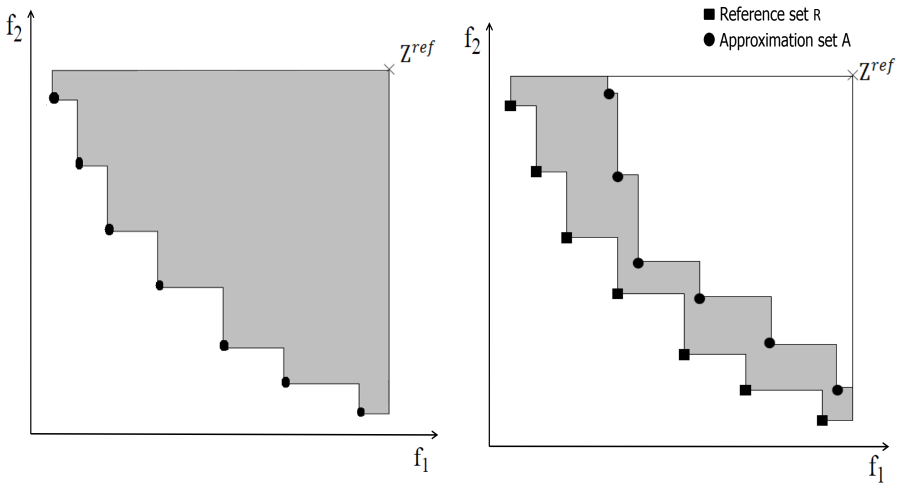

Figure 1.5 – Unary versus binary Hypervolume indicator

Figure 1.6 – Hypervolume indicator

The unary indicator, denoted I_H measures the volume (portion) of the objective space dominated by a given approximation. The higher this volume is, the better is the approximation. As shown in Figure 1.6, I_H requires the specification of a reference point z_{ref} which denotes an upper bound over all the objectives. Otherwise, z_{ref} must be at least weakly dominated by all the solutions of the considered approximation. As this point is usually not known, it is mostly estimated as the worst possible value in the objective space. Then, the volume I_H represents the union of the hypercubes (bounded by z_{ref}).

The binary variant of this indicator, the so-called hypervolume difference I_H^-, measures the quality of a given approximation set A in comparison to a reference set $Z_n^\star$. It computes the difference, in terms of the hypervolume, between these two sets by measuring the portion of objective space weakly dominated by $Z_n^\star$ and not by A:

$$I_H^-(A) = I_H(Z_n^\star) - I_H(A) \tag{1.10}$$

where smaller values correspond to higher quality in contrast to the unary hypervolume I_H (i.e., the closer I_H^- to 0, the better is the approximation).

1.4.2 Statistical validation

In recent years, statistical analysis has become a widespread and indispensable phase for drawing reliable conclusions about the performance of multi-objective optimizers. Otherwise, the goal of any statistical analysis is to increase the clarity and objectivity of the interpretation and validation of results. Generally, several runs of the same algorithm and the same problem instance are needed for a good performance analysis. Then, non-parametric statistical tests can be applied depending on the objective of analysis (Sheskin, 2003):

Comparison of the dominance rank of approximation sets: In this case, classical tests can be simply used such as the Mann-Whitney rank sum test for comparing two approximations or groups of data or the Kruskal-Wallis test for comparing more than two approximations.

Comparison of the indicators results: For this type of comparison, some protocol steps can be followed: First, consider a set of runs per problem instance of designed algorithms, quality indicators are performed to empirically evaluate them. Once indicator results are reported, statistical tests are then applied on the obtained $I-$values scalars. Two possible scenarios can occur when comparing algorithms:

- Independent samples: A run of each algorithm is a completely independent random sample. This means that the influence of one or more random variable is not taking into consideration. The Mann-Whitney or Kruskal-Wallis tests can be adopted.
- Matched samples: The parameters used for the runs of algorithms are the same and then the generated samples are matched or paired. In

this case, the Fisher permutation test or Wilcoxon signed-rank test can be used.

1.5 Multi-objective optimization applications

Multi-objective optimization has a huge number of applications in various domains such as telecommunication (Meunier et al., 2000), software engineering (Marler & Arora, 2004), bio-informatics (Handl et al., 2007), logistics and transportation (Jozefowiez et al., 2008), scheduling (Basseur et al., 2002), etc. As part of this thesis, we deal with routing problems belonging to the class of NP-hard combinatorial optimization. In the following, we give an overview of the problem that will be treated in next chapters.

The *Vehicle Routing Problem* (VRP), introduced more than four decades ago by (Dantzig & Ramser, 1959), is one of the well-known combinatorial optimization problems. It has received great attention over the last years due to its considerable difficulty (NP-hard problem) and potential real-world applicability in the area of distribution logistics and transportation systems. The problem aims to find best routes of a fleet of vehicles in order to distribute customers' demands with overall minimum travel cost. In general, the VRP consists in giving a set of vehicles with limited capacities, one or more common depots and several demands (or goods) for delivery to a set of geographically distributed customers. It is often subject to several side constraints related to customers, vehicles and routes. For instance:

- Customer constraints:
  - Customer demands can be delivered and/or collected,
  - Each customer is visited exactly once by only one vehicle,
  - A customer can be served during a time window.
- Vehicle constraints:
  - Vehicles can be homogenous or heterogeneous depending on their identical or variable capacity,
  - A vehicle starts from one or multiple depot(s) and terminates by returning to the original depot(s).
  - A required time for vehicles to deliver and/or collect goods.
- Routes constraints:
  - The vehicle capacity should not be exceeded by the total demands of customers during any route,

- The service time plus the travel time should be smaller than a preset constant,
- The total time of a feasible route must not exceed the maximum time fixed a priori.

Different possible objectives may also be defined on this problem, for example minimizing the total transportation cost, maximizing the variability of routes, minimizing the number of vehicles required, etc. Depending on the type of constraints and objectives, different variants of VRP exist in the literature . We present here the classical and basic variant called *Capacitated Vehicle Routing Problem* and referred to as VRP. The problem consists in designing routes for a set of homogeneous vehicles (having the same capacity restriction) to service the customers at the least cost. This cost can be a weighted function of the total traveled distance or total travel time. In the CVRP, all the vehicles start from and end to a single central depot, all the customers have deterministic demand quantities which are known in advance and the total demands for every route must not exceed the vehicle capacity.

Formally, the VRP can be defined as an asymmetric graph $G = (V, A)$ where A denotes the arcs set and $V = \{0, 1, \ldots, n\}$ denotes the set of vertices or nodes. The vertex 0 represents the central depot with vehicles $K = \{1, \ldots, m\}$ having the same capacity Q. Vertices $C = V \neq 0$ enumerate the customers with their crisp demands. For each arc $a_{ij} \in A$ is associated to a transportation cost C_{ij} that can be interpreted as the total traveled distance or travel time spent to go from customer i to customer j. The classical mathematical model is as follows:

$$\min \sum_{i,j}^{n} \sum_{k=1}^{m} C_{ij} x_{ij}^{k} \tag{1.11}$$

$$\text{s.t. } \textstyle\sum_{i,j}^{n} dm_i \; x_{ij}^{k} \leq Q, \; k \in K$$

where C_{ij} is the travel cost from i to j, x_{ij}^{k} is the decision variable that is equal to 1 if a vehicle k travels directly from customer i to customer j, 0 otherwise, dm_i represents the demand of the customer i and Q is the vehicle capacity.

Example 1.1. *Figure 1.7 illustrates an example of VRP with a central depot, a set of 14 customers represented by nodes and a set of 3 vehicles having an identical maximum capacity $Q = 15$. Each customer has a value associated with it which represents the quantity supplied. Clearly, in this example, the*

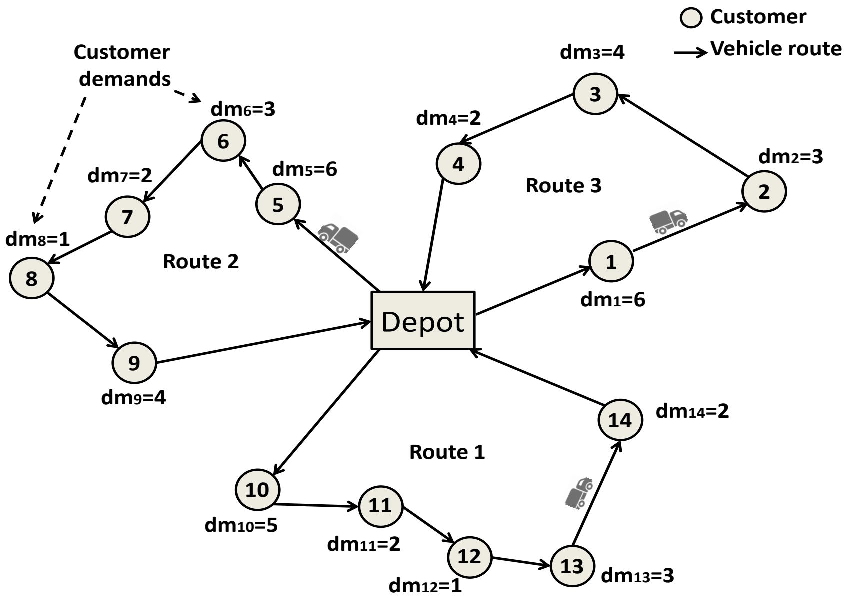

Figure 1.7 – CVRP problem

vehicles perform three different routes that start and end at the same depot. It is also shown that the total demands of visited customers by a route does not exceed the limited vehicle capacity. For instance, consider the route 1, the sum of supplied demands $5+2+1+3+2=13$ *is less than the maximum capacity* $Q=15$.

However, practical routing problems are often much more complex than the basic VRP. In the last years, several studies have investigated many variants of this problem (Golden et al., 2008 ; Ismail et al., 2011 ; Toth & Vigo, 2014). Figure 1.8 represents some VRP variants and their interconnections. Every variant is an extension of the basic problem with some additional constraints (i.e. an arrow moving from variant A to variant B means that this latter is an extension of A). Besides, a combination of two or more variants can be treated as a VRP with further specific constraints.

For instance, the *VRP with Time Windows* (VRPTW) extends the VRP by specifying a constraint of time windows in which the customers have to be served. Otherwise, each customer has a given time interval within which the delivery (or visit) must be made. The VRPTW has been extensively studied and various solution techniques for such a problem have been proposed. An

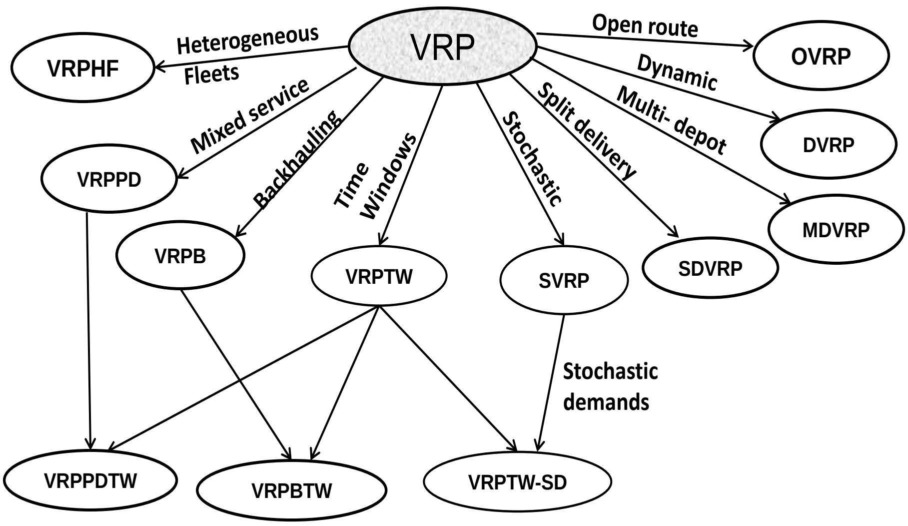

Figure 1.8 – Variants of VRP

overview of existing techniques is given by (Solomon, 1987).

The *Stochastic VRP* (SVRP) is another well-studied variant that has great applicability to real-life situations. It covers all the VRPs in which some inputs elements are stochastic such as the set of customer visited, the demands at the vertices and/or the travel times. This problem has different formulations: *VRP with Stochastic Demands* (VRP-SD), *VRP with Stochastic Customers* (VRP-SC), *VRP with Stochastic Customers and Demands* (VRP-SCD) and so on. A survey of methods and techniques used for SVRP problems can be found in (Gendreau, Laporte, & Séguin, 1996).

Most of these variants are often limited to optimize one single objective (e.g., the total transportation cost). Although, the majority of real-life routing applications are multi-objective by nature and require the satisfaction of many conflicting objectives simultaneously (e.g., minimizing the total vehicle travel distance and maximizing the expectation of customers' degree satisfaction). The different objectives in such problems may be classified according to different factors, that are, the tour, the resources, and the node activity as following:

Objectives on tour: Regarding the tour, the most common objectives are to reduce the tour costs in terms of travelled distance, required time (Jozefowiez et al., 2008) or duration of the longest tour (makespan) (Lacomme et al., 2006). These objective are important economic parameters associated to fuel consumption. Another type of objective consists of minimizing the disparity (imbalance) in the tour's workloads of the vehicles (Baños et al., 2013). The aim is so to induce an equilibration between tours (i.e. tour balancing) by defining workloads as the number of visited customers, the quantity of delivered goods, the time required, or the tour length, etc.

Objectives on resources: The resources are mainly the vehicles and the goods. For instance, minimizing the number of vehicles is one of the most studied resources objectives (Ghoseiri & Ghannadpour, 2010). This objective has often an economic significance such that fewer is the number of vehicles, fewer is the investment. However, sometimes in real life applications, it is impossible to reduce the cost by reducing the number of vehicles employed by the company. Another objective related to vehicles consists of maximizing their profit in terms of capacity or required time (Cordeau & Laporte, 2003). On goods, the common objectives are minimizing the damage of perishable goods or the risk related to the transportation of dangerous goods (Y. B. Park & Koelling, 1989).

Objectives on node/arc activities: Most of studies including objectives on nodes or arcs treat routing problems with time windows. In this case, the time windows constraint can be replaced by an objective that minimize the number of violated constraints or the total wait time caused by earliness or lateness (Gupta et al., 2010). Another objective consists of optimizing the access to the visited nodes by a set of unvisited nodes (Prins & Bouchenoua, 2005).

Additional types of objectives were thereby added to the above classification in order to cover the practical applications such as objectives on customer satisfaction (Ribeiro & Lourenço, 2001), objectives on transportation of goods (Tavakkoli et al., 2012), objectives on driver or arrival time consistency (Kovacs et al., 2014), etc. Moreover, a *Multi-Objective VRP* (MO-VRP) arises in many real-life applications (Golden et al., 2008 ; Toth & Vigo, 2014 ; Ghannadpour et al., 2014) such as commercial distribution or collection of goods like restaurant delivery services, newspapers distribution, bank or postal deliveries, industrial waste management, and so on. Other practical applications (Bowerman et al., 1995 ; J. Park & Kim, 2010) involve school bus routing, transportation of handicapped persons, routing of salespeople, security patrol services.

1.6 Conclusion

In this chapter, we have presented the fundamental background of deterministic multi-objective optimization, starting with classical definitions, concepts and formulations to a literature review about existing resolution methods and performance indicators. We have then focused some attention on the most common multi-objective methods, namely metaheuristics by giving an overview of their principle, main components and the used frameworks to develop them. Finally, we have briefly described the VRP variants as popular examples of combinatorial optimization problems. Next chapter discusses the case of optimization under uncertainty in both mono-objective and multi-objective contexts.

Chapter 2

Multi-objective Optimisation under Uncertainty

Contents

2.1 Introduction

Uncertainty rises as one of the major issues in decision making since it characterises almost all real-life problems and practical applications in which the big amount of data provides certainly some inevitable imperfections. These imperfections may occur from imprecision in information sources, incompleteness of some training data, ambiguity or difficulty of giving sharp observations and also from poor decision maker opinions. Then depending on the nature of imperfection, two types of uncertainty can be distinguished: *Aleatory uncertainty* (also called irreducible, objective or systemic uncertainty) which is characterized by natural randomness and variability that cannot be reduced by further efforts and *Epistemic uncertainty* (also called reducible, subjective or cognitive uncertainty) associated with ambiguity, fuzziness or any lack of knowledge and information. However, the uncertainty aspects have been intensively studied in the context of single-objective optimization, whereas their combination with multi-objectivity aspects has not been deeply studied so far. Moreover, almost all existing approaches for uncertain multi-objective optimization have been often limited to transform the problem into a crisp version or to reduce it into one or more mono-objective problems (Teich, 2001 ; Paquete & Stützle, 2007 ; Liefooghe, 2009). In this thesis, we focus on the case of multi-objective problems with epistemic uncertainty, especially in a fuzzy setting.

The chapter is organized as follows. Section 2.2 gives some basics of fuzzy theory and possibility theory. Section 2.3 briefly discusses the specific case of multi-objective optimization under uncertainty by formulating the problem and reviewing some existing approaches in this field. Finally, section 2.4 describes an uncertain variant of multi-objective vehicle routing problems to which we are interested.

2.2 Fuzzy and possibilistic frameworks

This section presents fundamental concepts of the two uncertainty frameworks used in this work, namely fuzzy sets theory (Zadeh, 1965) and its extension possibility theory (Dubois & Prade, 1988). Before introducing these frameworks, let us first give some useful notations:

- $\Omega = \{\omega_1, \ldots, \omega_n\}$ denotes the universe of discourse of a given problem,

- Each element $\omega \in \Omega$ is called an interpretation, a possible realization or a state of Ω,
- $A \subseteq \Omega$ denotes a subset of variables from Ω,
- a_i denotes any instance of A_i, means that if $A = \{A_1, \ldots, A_n\}$ then $a = (a_1, \ldots, a_n)$.
- $R \subset \Omega$ denotes the universal set of real numbers.

2.2.1 Fuzzy sets theory

The theory of fuzzy sets was introduced by (Zadeh, 1965) as an extension of dual logic and/or classical set theory. Over the years, it has been thoroughly studied through a large amount of scientific research and addressed by various real-world applications. This theory provides a powerful natural framework for handling ambiguity and vagueness arising from human linguistic labels. Its main idea is that the classical bivalent reasoning (i.e., in a crisp manner such as yes or no, 0 or 1) is unable to represent many situations that involve natural language uncertainty (as induced by human think). For instance, the linguistic qualifiers like "expensive", "appropriate" or "cheap" have different perceptions which vary from one person to another. Hence, the fuzzy logic proposes a flexible and suitable way to model any uncertain vague information using an approximate reasoning. In this reasoning, everything is interpreted as a matter of degree.

Example 2.1. *Consider the example of vehicle speeds on expressways and trunk roads. Suppose that a normal speed is about 100 km/h. In the crisp logic, a maximum speed limit (for example S = 130 km/h) is often required to decide whether it is high or low. Then, a speed of 131 km/h will be seen as high (131 > 130), while a speed of 129 km/h will be seen as low (129 < 130). On the other hand, fuzzy logic associates a set of degrees* $\in [0, 1]$ *to qualify each speed (see Figure 2.1). In this case, the speed is seen as high with a degree 1 above 130 km/h, with a degree 0 under 50 km/h and with a degree 0.5 for 100 km/h. Besides, one can say that the speed of 129 km/h is high with a degree 0.9 and low with a degree 0.1. We remark that the same speed can have different degrees according to the given qualifier meaning (e.g., high or low).*

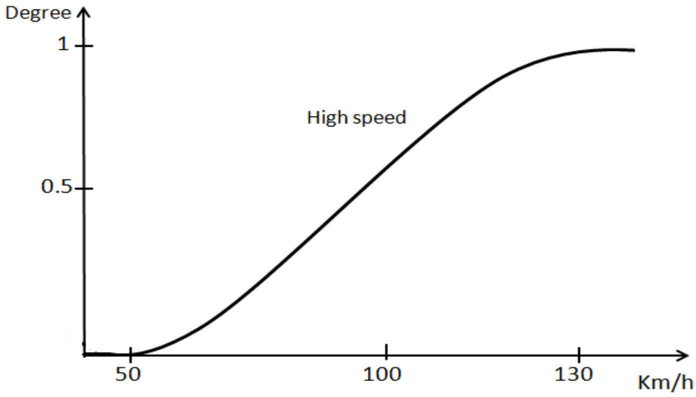

Figure 2.1 – Fuzzy sets representing high vehicle speeds

2.2.1.1 Definitions

In the following, we present some basic definitions and notions in fuzzy sets theory (Ayyub & Klir, 2006). Indeed, this theory is based on a gradual aspect represented by a membership function (also called grade of membership or degree of compatibility). This function indicates the degree that an element ω, of the universal set Ω belongs to a given fuzzy set by ranging it from 0 (perfect exclusion) to 1 (perfect inclusion).

Definition 2.1. ***Fuzzy set***
A fuzzy set A in Ω is a non-empty set of ordered pairs that admits degrees in the real interval $[0,1]$. Formally, it is defined by means of a membership function $\mu_A(\omega) : \Omega \rightarrow [0,1]$ that has the following mathematical form:

$$\mu_A(\omega) = \begin{cases} f_A(\omega) & \forall \omega \in [a,b) \\ 1 & \forall \omega \in [b,c] \\ g_A(\omega) & \forall \omega \in (c,d] \\ 0 & otherwise. \end{cases} \tag{2.1}$$

where $a \leq b \leq c \leq d \in A$ and $f_A(\omega) : [a,b) \rightarrow [0,1]$ and $g_A(\omega) : (c,d] \rightarrow [0,1]$ are real-valued increasing and decreasing functions respectively.

Definition 2.2. ***Support***
The crisp subset of fuzzy set A whose elements all have non-zero membership degrees is called the support of A and defined as:

$$Supp(A) : \{\omega \in \Omega | \mu_A(\omega) > 0\} \tag{2.2}$$

Definition 2.3. *Core*
The core of fuzzy set A is the crisp subset of elements having membership grade equal to 1 *such that:*

$$Core(A) : \{\omega \in \Omega | \mu_A(\omega) = 1\} \tag{2.3}$$

Definition 2.4. α*-cut*
An α-cut or α-level of fuzzy set A (denoted $[A]_\alpha$) is the crisp subset of elements of the universe Ω whose membership values belong at least to the degree α. That is:

$$[A]_\alpha = \{\omega \in \Omega \mid \mu_A(\omega) \geq \alpha\} \tag{2.4}$$

The α-cut is defined by the set of all ω such that $A(\omega)$ is greater than or equal to α. It is generally adopted as an effective tool for defining fuzzy measures and performing different arithmetic operations on fuzzy sets.

Definition 2.5. *Normal fuzzy set*
A fuzzy set A is called normal or normalized if there exist an element $\omega \in R$ such that:

$$\sup_{\omega \in \Omega} \mu_A(\omega) = 1 \tag{2.5}$$

where sup denotes the high of A (largest membership grade). Otherwise, A is considered as sub-normalized.

Definition 2.6. *Convex fuzzy set*
A fuzzy set A is said to be a convex set if there exist elements $\omega_1, \omega_2 \in \Omega$ such that:

$$\mu_A(\lambda\omega_1 + (1 - \lambda)\omega_2) \geq \min(\mu_A(\omega_1), \mu_A(\omega_2)), \lambda \in [0, 1] \tag{2.6}$$

Definition 2.7. *Fuzzy number*
A fuzzy set A is called a fuzzy number in the universal set R if:
(i) it is a normalized and convex fuzzy set,
(ii) has a non-empty and bounded support,
(iii) its membership function is piecewise continuous.

Example 2.2. *Figure 2.2 illustrates examples of three linear shapes of fuzzy numbers having a membership function "around or close to 130" (Ex. the vehicle speed limit). The shape in the left illustrates a triangular fuzzy number, the shape in the middle represents a trapezoidal fuzzy number and the curve at the right shows a bell-shape fuzzy number.*

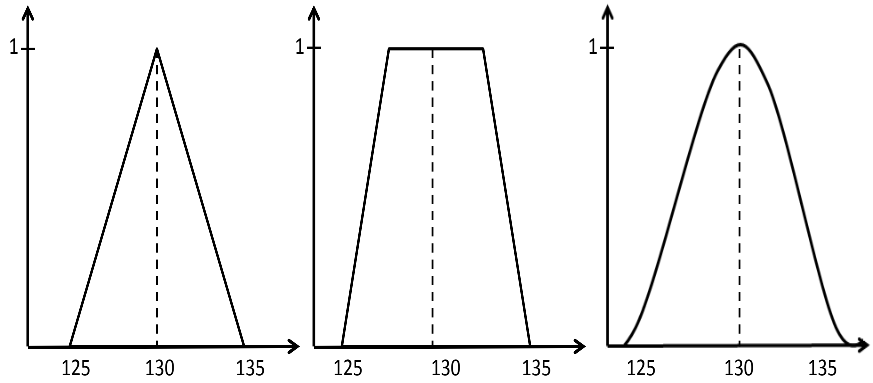

Figure 2.2 – Examples of fuzzy shapes

The selection of an appropriate shape depends on the type of fuzzy data to be represented. The most common and popular shape of membership functions is triangular.

Definition 2.8. ***Triangular Fuzzy Number (TFN)***
It is a fuzzy number represented with a triplet of values $A = [\underline{a}, \widehat{a}, \overline{a}]$, where $[\underline{a}, \overline{a}]$ is the interval of possible valued called its support and $\widehat{a}$ denotes its modal or kernel value (i.e. the most plausible) as shown in Figure 2.3. This representation is interpreted as a linear membership function $\mu_A(x)$ that holds the following mathematical definition:

$$\mu_A(x) = \begin{cases} \frac{x-\underline{a}}{\widehat{a}-\underline{a}}, & \underline{a} \leq x \leq \widehat{a} \\ 1, & x = \widehat{a} \\ \frac{\overline{a}-x}{\overline{a}-\widehat{a}}, & \widehat{a} \leq x \leq \overline{a} \\ 0, & otherwise. \end{cases} \tag{2.7}$$

Notice that, a triangular fuzzy number can be deduced from transformations of other shapes (like trapezoidal or rectangular fuzzy numbers) by inducing linguistic modifiers, compositions, projections and other operations (Zadeh, 1965). For these reasons, TFNs are simple to implement, fast for computation and usually employed as start functions for every problem (Pedrycz, 1994).

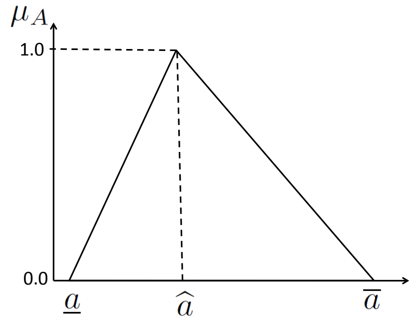

Figure 2.3 – Triangular Fuzzy Number

2.2.1.2 Fuzzy set properties and operations

Fuzzy subsets are often used with some mathematical properties in order to describe vague/uncertain concepts. Therefore, various standard properties (such as equality, inclusion, union, etc.) have been extended in the context of fuzziness. Let A and B be two fuzzy sets:

1. *Fuzzy Equality:* $A = B$ if $\forall \omega \in \Omega : \mu_A(\omega) = \mu_B(\omega)$.
2. *Fuzzy Inclusion:* $A \subset B$ if $\forall \omega \in \Omega : \mu_A(\omega) \leq \mu_B(\omega)$.
3. *Fuzzy Intersection:* $\forall \omega \in \Omega : (A \cap B)(\omega) = \min(\mu_A(\omega), \mu_B(\omega))$.
4. *Fuzzy Union:* $\forall \omega \in \Omega : (A \cup B)(\omega) = \max(\mu_A(\omega), \mu_B(\omega))$.
5. *Fuzzy Complement:* $\mu_{A^c}(\omega) = 1 - \mu_A(\omega)$.
6. *Fuzzy Cardinality:* $|A| = \sum_{\omega \in \Omega} \mu_A(\omega)$.

In the literature, the operators of fuzzy intersection and fuzzy union are respectively referred as "t-norms" and "t-conorms" (Klir & Yuan, 1995). Notice that, all previous properties are also applicable on fuzzy numbers in the same manner.

Another important issue in fuzzy sets theory is the extension principle described by (Zadeh, 1965) and developed later by (Yager, 1986). This principle provides a mathematical way for extending the classical domain of a function (arithmetic, relation...) to fuzzy sets. More formally, let $F : \Omega \longrightarrow R$ be a real function with $F(\omega) = u$, A be a fuzzy set in Ω and let $\mu_A(\omega)$ be the membership function for ω. Using the extension principle, the membership for u is defined as:

$$\mu_B(u) = \sup\{\mu_A(\omega) | F(\omega) = u\} \tag{2.8}$$

where B is the direct image of A mapped under of $F(.)$ (i.e. $B = F(A)$).
Furthermore, the extension principle is particularly useful in the analysis and arithmetics of fuzzy numbers. Based on this principle, the classical arithmetic operations namely operation of addition, substraction, division and multiplication are generalized for fuzzy numbers. Examples of operations that can be performed on two triangular fuzzy numbers $A = [\underline{a}, \widehat{a}, \overline{a}]$ and $B = [\underline{b}, \widehat{b}, \overline{b}]$ are:
(i) *Addition:* $A + B = (\underline{a} + \underline{b}, \widehat{a} + \widehat{b}, \overline{a} + \overline{b})$
(ii) *Substraction:* $A - B = (\underline{a} - \underline{b}, \widehat{a} - \widehat{b}, \overline{a} - \overline{b})$
(iii) *Symmetric image:* $-A = [-\overline{a}, -\widehat{a}, -\underline{a}]$

In this scope, the α-cut can be used to state a fuzzy number under a crisp interval of α-boundaries. Then, the arithmetic operation can be directly applied on the interval of α-cuts. For instance, consider two fuzzy numbers A and B and their α-cuts $[A]_\alpha = [a^L_\alpha, a^R_\alpha]$ and $[B]_\alpha = [b^L_\alpha, b^R_\alpha]$ respectively, a standard addition operation is given by:

$$A + B = [A + B]_\alpha = [a^L_\alpha + b^L_\alpha, a^R_\alpha + b^R_\alpha] \tag{2.9}$$

2.2.2 Possibility theory

Possibility theory, issued from fuzzy theory, was initially introduced by (Zadeh, 1978) and further developed by (Dubois & Prade, 1988). This theory allows to handle uncertainty in a flexible and simple way as expressed by humans. Indeed, it represents an appropriate framework for experts to express their partial belief numerically or qualitatively. In other words, possibility theory deals with epistemic uncertainty in two ways, namely the quantitative and qualitative settings. Before presenting the possibilistic aspects, let us first give some meanings of the notion of possibility. Typically, the term "it is possible that" can have several interpretations such as:

- It is feasible (or realizable) to do something.
- It is consistent (not contradictory or reasonable) to achieve an action or event.
- It is plausible (believable or realistic) to be true.

According to these semantics, different levels of possibility could be expressed either using possibility degrees (in quantitative setting) or using orderings on the possible events (in qualitative setting). In this thesis, we focus on the quantitative facet of possibility theory.

Additionally, this theory is in complete accordance with the basic principle of fuzzy logic where gradual membership properties are present. In fact, starting from the idea of quantifying the membership of an element, possibility theory is meant to provide a gradual possibilistic semantic to a set of elements. Moreover, it can be used to quantify the possibility of an element regarding a set by means of a possibility measure. For instance, the definition of linguistic expressions such as "high" or "low" may refer to a set of possible values in a specific context. Then, each value will be associated with a possibility degree which quantifies how much this value is typical with respect to the qualifier meaning.

2.2.2.1 Possibility distributions

The basic concept of possibility theory is the notion of *possibility distributions* denoted by π and corresponding to a mapping from the universe of discourse Ω to a given possibilistic scale. More precisely, a possibility distribution is a function which associates to each element ω_i of Ω, a value $\pi(\omega)$ in a bounded and linearly valuation scale. This value is called a *possibility degree*. In the quantitative sense, the scale is taken as a unit interval $[0, 1]$ encoding our knowledge on the real world which is generally ill known.

Moreover, possibility distributions are used as flexible constraints restricting the more or less possible values of a single-valued variable. In fact, a possibility distribution π_v represents a state of knowledge about the unknown values of variable v (ranging on Ω) distinguishing what is plausible from what is less plausible. The quantity $\pi_v(\omega)$ is the degree of possibility that $v = \omega$. It provides a restriction of the values of v with respect to some conventions as described in Table 2.1. However, distinct values may have simultaneously

Table 2.1 – Possibility distribution π_v

$\pi_v(\omega) = 0$	$\omega = v$ is impossible
$\pi_v(\omega) = 1$	$\omega = v$ is totally possible
$\pi_v(\omega) > \pi_v(\omega')$	$\omega = v$ is preferred to $\omega' = v$ (or more plausible)

a degree of possibility equal to 1. Then, flexibility is determined by ranging some values of $\pi_v(\omega)$ between 0 and 1. Besides, the use of intermediary degrees of the possibility enable us to acknowledge that some values are more possible than others. It should also be noticed that a possibility distribution $\pi_v(\omega)$ is said to be *normalized* if there exist at least one state ω of Ω which is totally possible as a value of v. In this case, Ω is assumed as the complete range of v and thus $\max_{\omega \in \Omega}\{\pi(\omega) = 1\}$. Otherwise, $\pi_v(\omega)$ is considered as sub-normalized. In what follows, we give some particular or extreme cases of possibility distributions:

— *Complete knowledge*: $\exists \omega_i \in \Omega, \pi(\omega_i) = 1$ and $\forall \omega_i \neq \omega_j, \pi(\omega_j) = 0$.
— *Partial knowledge*: $\forall \omega_i \in A \subseteq \Omega, \pi(\omega_i) = 1$. If A is not a singleton $\forall \omega_i \notin A, \pi(\omega_i) = 0$.
— *Total ignorance*: $\forall \omega_i \in \Omega, \pi(\omega_i) = 1$ (all values in Ω are possible).

2.2.2.2 Possibility and Necessity measures

A possibility distribution π on Ω enables events to be qualified in terms of their plausibility and their certainty, by means of two dual measures: the possibility Π and the necessity N. In fact, the expression "it is not possible that A is true" does not only mean that "not A is possible", but it also leads to a stronger conclusion: "it is necessary that not A". Conversely, the expression "it is possible that A is true" does not entail anything about the possibility nor the impossibility of not A. Hence, a distinction between the concepts of possibility (plausibility) and necessity (certainty) of an event is defined using the two dual measures Π and N.

Possibility measure Given a possibility distribution π, the possibility measure of any subset $A \subseteq \Omega$ may be defined by:

$$\Pi(A) = \max_{\omega \in A} \pi_v(\omega) \tag{2.10}$$

$\Pi(A)$ is called the possibility degree of A and it evaluates to what extent it is possible that the actual value of v belongs to A. In classical way, this measure evaluates at which level A is consistent (i.e. not contradictory) with the knowledge represented by π (i.e. ω is a unique value). Table 2.2 gives main properties of possibility measure in the case of classical normalized distribution π.

On the other hand, if a fuzzy logic is adopted, a membership function μ will be used instead of π and then $\Pi(A)$ will be interpreted as a graded fuzzy measure.

$$\Pi(A) = \sup_{\omega \in A} \mu_v(\omega) \tag{2.11}$$

The evaluation provided by $\Pi(A)$ corresponds to a degree of non-emptiness of the intersection of subset A with the given fuzzy set.

Table 2.2 – Possibility measure Π

$\Pi(A) = 1$ and $\Pi(\neg A) = 0$	A is certainly true
$\Pi(A) = 1$ and $\Pi(\neg A) \in [0, 1]$	A is somewhat certain
$\Pi(A) = 1$ and $\Pi(\neg A) = 1$	total ignorance (A is unknown)
$\Pi(A) > \Pi(B)$	A is a-priori more plausible than B
$\max(\Pi(A), \Pi(B)) = 1$	A and $\neg A$ cannot be both impossible

Necessity measure The necessity is a dual measure of the possibility Π. Thereby, $N(A)$ is defined with reference to the complementary set $\neg A$ of that under study as follows:

$$N(A) = 1 - \Pi(\neg A) = \min_{\omega \notin A}(1 - \pi(\omega)) \tag{2.12}$$

$N(A)$ is called the necessity degree of A. It evaluates to extent we are certain that the actual value of ω belongs to A. This measure corresponds at which level A is certainly implied by the knowledge expressed by π. It represents the certainty degree of A. Main properties of necessity measure are summarized in Table 2.3. Similarly in fuzzy logic, the grade of membership μ will be used as the possibility degree:

$$N(A) = 1 - \sup_{\omega \notin A} \mu_v(\omega) \tag{2.13}$$

Table 2.3 – Necessity measure N

$N(A) = 1$ and $N(\neg A) = 0$	A is certainly true
$N(A) \in [0, 1]$ and $N(\neg A) = 0$	A is somewhat certain
$N(A) = 0$ and $N(\neg A) = 0$	total ignorance
$\min(N(A), N(\neg A)) = 0$	unique link existing between $N(A)$ and $N(\neg A)$

Moreover, the duality of both measures implies the following relation: $\Pi(A) \geq N(A)$. This relation is translated by two interpretations: The first one means that any subset about which we are certain, at least a little, is completely possible (i.e. if $N(A) \neq 0$, then $\Pi(A) = 1$)). The second interpretation means that we have no certainty about an event which is only relatively possible (i.e. if $\Pi(A) \neq 1$, then $N(A) = 0$).

Two key axioms of 'maxitivity' and 'minitivity' complete the basics of possibility theorty. Indeed, the possibility measures satisfy the basic 'maxitivity' property, while the necessity satisfy the dual property of 'minitivity':

$$\Pi(A \cup B) = \max(\Pi(A), \Pi(B)) \tag{2.14}$$

$$N(A \cap B) = \min(N(A), N(B)) \tag{2.15}$$

These axioms represent a generalization of the basic disjunctive (or union $A \cup B$) and conjunctive (or intersection $A \cap B$) fusion operators. They allow the fusion of possibility and necessity degrees using t-norm and t-conorm rules described below.

Example 2.3. *Let π be a possibility distribution defined on $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ such that $\pi(\omega_1) = 0.4$, $\pi(\omega_2) = 0.7$, $\pi(\omega_3) = 1$, $\pi(\omega_4) = 0.1$ and $A = \{\omega_1, \omega_2\}$, the possibility and necessity degrees of A are:*

$$\Pi(A) = \max(0.4, 0.7) = 0.7 \textit{ and } N(A) = 1 - \max(1, 0.1) = 0.$$

2.2.2.3 Fusion rules in possibility theory

The fusion of pieces of information from different sources is an important aspect having a potential application in many areas such as expert opinions fusion, functions fusion, multiple results combination, and so on. A typology of fusion (or merging) rules has been defined in the setting of possibility theory, distinguishing rules that use the intersection between dependent pieces of information, from rules that use the union operator between independent information. In particular, the choice of a suitable merging rule is too related to the problem under study, the reliability of information sources and the dependence between the different pieces of information. Thereby, two well-known concepts can be captured: t-norms and t-conorms.

t-norms (or triangular norms) A t-norm is defined as a fusion function of two variables from $[0, 1] \times [0, 1]$ to $[0, 1]$. Such function is associative, monotonically increasing, and admits 1 as unit element (Dubois & Prade, 2000).

Besides, t-norm has a context dependent behavior and it is mainly used for a conjunctive fusion. Otherwise, the conjunctive fusion is commonly applied to a set of reliable information sources which agree with each other. This combination mode uses intersection between pieces of information to provide the resulting fusion. Notice that it is also generalized to fuzzy sets. Examples of t-norms are minimum-based rule and product-based rule. More formally, given different pieces of information from n sources: $\pi_i, i = \{1, \ldots, n\}$, the t-norm function used to merge them can be defined by:

$$\pi_{tnorm}(\omega) = \otimes_{i=\{1,\ldots,n\}}\pi_i(\omega), \quad \forall\omega \in \Omega \tag{2.16}$$

where $\otimes$ is a t-norm operator which could be the min or the product. Thus, the minimum-based rule is given by: $a \otimes b = \min(a, b)$

t-conorms (or triangular conorms) A t-conorm is defined as an operation from $[0, 1] \times [0, 1]$ to $[0, 1]$, satisfying the following properties: commutativity, associativity, monotonicity and 0 is its unit element. It is mainly used for a disjunctive fusion, especially when information sources disagree and when reliability condition could not totally be checked (means that we are enable to determine which source of information is the reliable). Otherwise, the disjunctive mode assumes that if propositions are contradictory, then it is better to consider the maximum consistent subsets of propositions by assuming that the real proposition is one of them. This logic comes from the fact that if propositions are very discarded (intersection gives the empty set with t-norms very close to 0), it is more natural to say that one of them may be reliable rather than to say that simply we are in a total conflict and no proposition could be assumed. Formally, the t-conorm function is defined by:

$$\pi_{tconorm}(\omega) = \oplus_{i=\{1,\ldots,n\}}\pi_i(\omega), \quad \forall\omega \in \Omega \tag{2.17}$$

where $\oplus$ is a t-conorm operator such that: $a \oplus b = \max(a, b)$. Other refining rules, combining t-norm and t-conorm concepts have been proposed for situations where sources may partially be in agreement and/or only some sources are reliable.

Example 2.4. *Let us reconsider the possibility distribution π in Example 2.3, suppose that another information source provides us the possibility distribution π' such that $\pi'(\omega_1) = 1$, $\pi'(\omega_2) = 0.4$, $\pi'(\omega_3) = 1$ and $\pi'(\omega_4) = 0.8$. If we use the t-norm to merge π and π', for instance by taking $\otimes = \min$ in*

Equation 2.16, we obtain:
$\pi_{tnorm}(\omega_1) = 0.4$,
$\pi_{tnorm}(\omega_2) = 0.4$,
$\pi_{tnorm}(\omega_3) = 1$,
$\pi_{tnorm}(\omega_5) = 0.1$.
Then, if we use the t-conorm to merge them, for instance by taking $\oplus = \max$ *in Equation 2.17, we obtain:*
$\pi_{tconorm}(\omega_1) = 1$,
$\pi_{tconorm}(\omega_2) = 0.7$,
$\pi_{tconorm}(\omega_3) = 1$,
$\pi_{tconorm}(\omega_5) = 0.8$.

The remaining of this chapter is devoted to the analysis of uncertainty in multi-objective optimization problems.

2.3 Existing approaches for optimization problems under uncertainty

The aim of optimization problems under uncertainty is to optimize predefined objective(s) while considering that some information are uncertain and without knowing what their full effects will be. A literature review on optimization under uncertainty can be found in (Rockafellar, s. d. ; Sahinidis, 2004 ; Diwekar, 2008 ; Petrone, 2011). Furthermore, this field reflects reality in many areas of application and presents many issues that should be taken into account such as:

- *What are the sources of uncertainty ?*
- *How modeling and quantifying such uncertainty ?*
- *How propagating uncertainty through the optimization process ?*
- *What are their effects and consequences ?*
- *How to develop and perform a resolution method ?*
- *How to analyse the efficiency and performance of outcomes ?*

These issues will be addressed one by one in this thesis. Firstly, we must be aware of the different sources of uncertainty and their implications in the process. Among others, some of these sources are the problem it-self, the input variables, the parameters of the problem, the modeling assumptions and the procedures for analysing the model. Then, once the uncertainties are quantified, it is necessary to analyze their effects before optimization. Yet, the effects of uncertainty are generally related to the problem context, the environmental factors and are in close connection with the uncertainty sources. In fact, the global effect is that disturbances of the input data may be propagated through the model to the quantities of interest. This effect is usually the most critical and complex for realistic simulations since it may hamper the identification of efficient outcomes. Otherwise, propagating uncertainty may mislead the analyst into determining the optimal alternatives, leading to a final bad choice. In that sense, the most interesting and challenging issue is how uncertainty propagation through the optimization process can properly be taken into account.

Another important issue is the uncertainty inherent to the resolution stage. The question is how to explore an uncertainty design space that leads often to very large-scale and complex optimization models. Evidently, as a deterministic optimization problem (i.e. the objective(s) are deterministic) is already NP-hard and time-consuming, the problems of optimization under uncertainty may lead to prohibitive computation. Therefore, their increasing costs and complexity motivates more and more the scientific research efforts to develop efficient resolution methods. In this context, a resolution method may still contain some uncertainties due to its inability to provide exact results or to the lack of optimality proof. Some of research works for coping with uncertainty in the context of single-objective optimization are shown in table 2.4.

Table 2.4 – Uncertain single-objective design optimization

Method	Authors and works
Mean-penalty model	(F. Li et al., 2010)
Chance constrained model	(P. Li et al., 2008)
Recourse Model	(Dantzig, 1955)
Minimax regret model	(Averbakh, 2000 ; Aissi et al., 2009)
Robustness/ Reliability	(Agarwal, 2004 ; Beyer & Sendhoff, 2007)
based models	(Gabrel et al., 2014)

However, despite the large number of contributions in this field, only few studies have been made during the last years regarding the combination of uncertainty and multi-objectivity. In what follows, we review some existing approaches in the field of multi-objective optimization under uncertainty. Nowadays, this field has attracted increasing attention since it appears in many real-life applications and poses several interesting challenges. Surveys of related studies are given in (Klein et al., 1990 ; Fieldsend & Everson, 2005 ; Goh & Tan, 2009 ; Petrone, 2011 ; Zhou et al., 2014).

In general, a MOP under uncertainty is characterized by the necessity of optimizing simultaneously several conflicting objectives in presence of some uncertain input data. However, taking into account uncertainty adds important challenges to the optimization of multiple objectives. One of these challenges is how to identify the type of inevitable uncertainties and their impacts on the results and optimal decision making. The purpose at this level is to analyse the manner in which such uncertainties are propagated through the optimization process. Usually, the quantum of propagation depends on the nature of inputs uncertainty, their distributions and their transfer to the outputs through the functional relationship. Obviously, uncertainty propagation leads to an excessive increase in problem's complexity and difficulty for finding optimal solutions. In particular, the resolution stage will be much more complicated since propagating uncertainties may affect the optimization process and even the key elements of decision making such as preference parameters, decision variables, constraints and/or objectives.

Many studies addressing the effects of uncertainty propagation in multi-objective setting have focused on the case where uncertainty is assumed to occur in the objective functions. Yet, uncertainty in the objectives presents a critical and sensitive obstacle because it may influence the search process and consequently hamper the identification of efficient solutions. A minimization MOP with uncertain objectives may be defined as:

$$\min\ F(x,\xi) = \min[f_1(x,\xi), f_2(x,\xi), \ldots, f_n(x,\xi)]\ s.t.\ \ x \in X, \xi \in U \quad (2.18)$$

where F is the set of objective functions that may depend on uncertainty scenarios U, x is a decision variable vector from its admissible region $X \subseteq \mathbb{R}^n$ and $\xi = (\xi_1, \xi_2, \ldots, \xi_q)$ is a vector of independent uncertain variables. Clearly, the problem here is that each $f_i(x,\xi)$ is an uncertain quantity induced by ξ.

Once the uncertain scenarios and their effects are identified, the second relevant challenge consists to find a suitable way for handling uncertainty in multi-objective resolution methods. Nonetheless, very little research works have been proposed so far to deal with uncertain multi-objective optimization. Besides, although uncertainty in the objective functions has gained attention in recent years, the efforts devoted to this problem are still limited. In Figure 2.4, we propose to classify the existing approaches according to how the uncertain multiple objectives are managed.

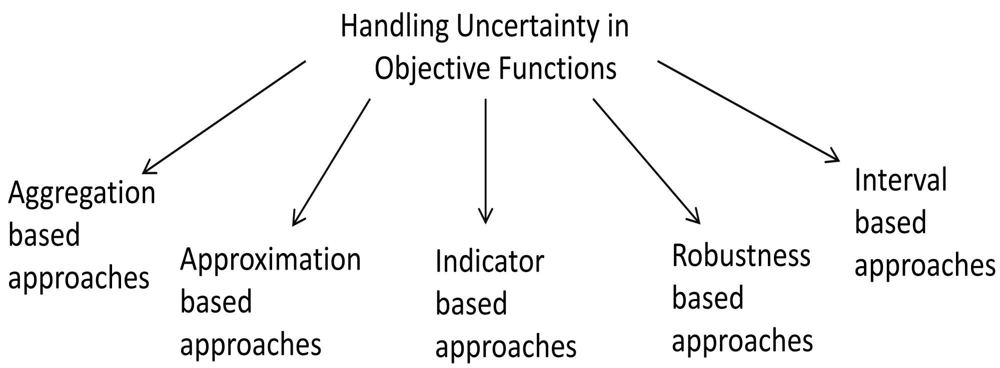

Figure 2.4 – Taxonomy of approaches for MOPs with uncertain objectives

The first attempts to cope with uncertainty in objectives belong to the category of aggregation-based approaches (Goncalves et al., 2009 ; Paquete & Stützle, 2007). The basic idea of these traditional approaches is to combine the multiple objectives into a single uncertain one. In other words, they convert the MOP into a one or a set of single-objective problems. Furthermore, the different objectives can be rewritten into an aggregate objective f_A by applying a weighted sum function as follows:

$$f_A(x, \xi) = \sum [f_1(x, \xi), f_2(x, \xi), \ldots, f_n(x, \xi)] \tag{2.19}$$

In this case, the existing approaches designed for single-objective optimization problems under uncertainty can simply be applied. For example, the approaches cited previously in the table 2.4 can be applied to solve this problem. Clearly, aggregation-based approaches have the advantage of simplicity because they do not require a particular development for uncertain multi-objective optimization. Yet, they still not efficient since they limit the objective space, ignore the significant role of multi-objectivity and also relationship between the conflicting objectives. In consequence, the obtained results are very often useless and far from reality.

The second category encloses approximation-based approaches that use statistical functions to convert the uncertain objectives into their crisp equivalents (Hughes, 2001 ; Teich, 2001). Otherwise, these approaches still abide to the certainty of objectives and usually allow to carry out an approximation of observed uncertainty. In this case, a statistical function may be applied to approximate each objective function as follows:

$$\Phi(f_1(x,\xi)), \Phi(f_2(x,\xi)), \ldots, \Phi(f_n(x,\xi)) \tag{2.20}$$

where $\Phi(.)$ denotes the statistical operator which can be the expected function $E[.]$ with respect to ξ. This category includes also mean-value and mean-penalty approaches (Kim & hyun Ryu, 2011 ; Meng et al., 2011). Commonly, each objective is approximated by estimating the mean value of each random sample. This allows to transform the uncertain MOP into a crisp problem that can be resolved using standard deterministic multi-objective optimizers. A major limit of approximation-based approaches is that the propagation and effects of uncertainty are neglected. Yet, ignoring the uncertainty propagation in the optimization process can lead to very poor decisions with often misleading simulation results. It is therefore necessary to account for the relationship between uncertain inputs and generated solutions, because if the input data or parameters are highly uncertain, how can the optimizer simply state that the outputs are exact values? It may be feasible only for simplicity or other practical reasons as long as the optimization performance will not be affected.

The third category includes different approaches (Basseur & Zitzler, 2006 ; Liefooghe et al., 2010) that combine uncertainty of objectives and quality indicators (i.e., real-valued functions which allow assessment of Pareto approximations). This combination is done by estimating indicator evaluations for the uncertain objective vectors as:

$$I(f_1(x,\xi), X^*), I(f_2(x,\xi), X^*), \ldots, I(f_n(x,\xi), X^*) \tag{2.21}$$

where $X^* = \{x_1^*, \ldots, x_r^*\}$ is a variable reference set and $I(.)$ stands for the vector of indicator values that can be minimized or maximized depending on the quality goal. For instance, in (Basseur & Zitzler, 2006), authors proposed an indicator-based model to reflect the uncertainty of objectives. More precisely, the objective vector is associated with uncertain distributions, where the optimization goal is defined in terms of the $\epsilon-$indicator values.

Another category of approaches refers to the robustness aspect that will be presented in Chapter5 (Deb & Gupta, 2005, 2006 ; Barrico & Antunes, 2006 ; Ehrgott et al., 2014). This aspect is connected to the idea that in presence of uncertain inputs, the outputs should be relatively insensitive (small uncertainty outputs). The robustness in objective functions can be modeled as:

$$(f_1(x,\xi), R_1), (f_2(x,\xi), R_2), \ldots, (f_n(x,\xi), R_n) \tag{2.22}$$

where R_i is the robustness criterion that should be maximized. It is defined in terms of the variation of $f_i(x)$ regarding the uncertainty associated with x. For instance, (Deb & Gupta, 2005) proposed to estimate the expected uncertainty using Monte Carlo simulations based on effective objective function that takes into account robustness. (Barrico & Antunes, 2006), a propagating approach based on the concept of robustness degrees of uncertain objectives is introduced. Similarly, reliability criterion R_i^γ can be used to optimize objectives which are uncertain with small chance of failure under predefined acceptable level γ. For instance, (Coelho & Bouillard, 2011) suggested a reliability based formulation within a multi-objective context.

However, the main drawback of robustness-based and indicator-based approaches is that they rely on the assumption of a priori knowledge about decisive information such as the reference set of solutions or the robustness confidence level. Evidently, if such information is inappropriate or incorrect, the outputs of theses approaches can be misleading.

Further studies assume to display uncertainty of objectives through intervals and thereby to perform the multi-objective optimization based on this uniform distribution. These studies fall under the category of interval-based approaches (Limbourg, 2005 ; Limbourg & Aponte, 2005). In this case, the cost of evaluating $f(x,\xi)$, namely Y is represented as intervals as:

$$F(x,\xi) = Y = ([\underline{y_1}, \overline{y_1}], \ldots, [\underline{y_n}, \overline{y_n}]) \tag{2.23}$$

where $\underline{y_i}$ and $\overline{y_i}$ are respectively lower and upper bounds of the corresponding interval-valued function i. For instance, in (Limbourg & Aponte, 2005) authors defined the uncertainty via intervals and then introduced an extension of Pareto dominance for ranking the generated interval-valued solutions.

2.4 Conclusion

In this chapter, we have surveyed the state of the art relative to combinatorial optimization under uncertainty. In particular, we have focused on fuzzy and possibilistic frameworks for representing the uncertainty aspect. Then, we have discussed the major impacts of propagating uncertainty through the optimization process.

In the second part of this chapter, we have introduced a classification of the different existing approaches to handle an uncertain multi-objective problem. This classification shows that a multitude of researchers have addressed this problem by either transforming it into a crisp problem or by reducing it into a mono-objective one. Only few of them have developed interval models to treat the problem without neglecting the uncertainty propagation.

Next chapters propose a new framework for dealing with the specific case of MOPs with fuzzy-valued objectives. The proposed framework will tackle the issue of extending classical multi-objective concepts, techniques and resolution methods to such fuzzy setting.

Part II

Contributions

Chapter 3

Pareto Dominance for Fuzzy Multi-objective Optimization

Contents

Main publications related to this chapter

Bahri, O., Ben Amor, N., and Talbi, E.-G.: New Pareto approach for ranking triangular fuzzy numbers, 15th International Conference on Information Processing and Management of Uncertainty (IPMU'14), Montpellier, France, July 2014.

Bahri, O., Ben Amor, N., and Talbi, E.-G.: A Possibilistic framework for solving multi-objective problems under uncertainty : Definition of new Pareto optimality, 16th International Workshop on Nature Inspired Distributed Computing (NIDISC'13), Boston, Massachusetts (USA), May 2013.

3.1 Introduction

This chapter mainly focuses on the concept of Pareto dominance which is crucial for comparing any two solutions in multi-objective optimization, especially where the objectives are typically conflicting. In uncertain multi-objective optimization, these objectives are often affected by uncertainty preventing the use of deterministic Pareto dominance. However, propagating uncertainty to the objectives is of extreme importance because it may affect the whole dominance process and consequently the optimality of solutions. As described in the previous chapter, several methods were proposed in the literature in order to handle such a problem. Our interest concentrates on the specific case of multi-objective optimization with fuzzy-valued objectives. To this end, we propose a novel Pareto approach for classifying the generated fuzzy solutions. Our approach is inspired by the paradigm of ranking fuzzy numbers and uses the concepts of deterministic Pareto dominance.

This chapter is structured as follows: Section 3.2 attempts to explain our motivations and gives a formal definition of the treated problem. Section 3.3 presents our proposal which is composed of two dominance stages. In the first stage, mono-objective dominance relations are defined between two triangular fuzzy numbers. In the second stage, a Pareto dominance is defined for classifying vectors of fuzzy solutions. Finally, Section 3.4 discusses the advantages of the proposed approach with some numerical examples.

3.2 Problem description

As mentioned in Chapter 1, the Pareto dominance concepts (Pareto, 1971) are crucial when dealing with multi-objective optimization because we need to maintain a set of non dominated solutions rather than only one. These solutions often called *Pareto optimal set* correspond to a group of solutions with compromise between different objectives (i.e. achieving the optimal value for one objective requires some sacrifice of quality on at least one other objective). This notion of optimality is based on intuitive discrimination of what are the most good or desired alternatives among many others.

For multi-objective optimization under uncertainty, the purpose becomes to find out best solutions within an uncertain context. Indeed, as explained before, propagating uncertainty to the objectives should not be ignored or approximated. In this case, the generated solutions are disrupted by the uncertainty type of objectives. In consequence, the classic Pareto concepts cannot be used for comparing the uncertain outcomes. To this end, extensions of the classic Pareto optimality have been discussed and addressed in recent literature (Teich, 2001 ; Haubelt & Teich, 2003 ; Limbourg, 2005 ; Silva & Yamakami, 2009 ; Hendriks et al., 2011). For instance, in (Teich, 2001 ; Haubelt & Teich, 2003), a probabilistic dominance based on intervals is used to guide the selection process of Pareto-set. In (Limbourg, 2005 ; Limbourg & Aponte, 2005), intervals of belief functions are used to represent the uncertain Pareto optimal solutions. Moreover, authors in (Silva & Yamakami, 2009) involve uncertainty as fuzzy coefficients in the objective functions. Then, an interval-based abstraction is introduced to generate the Pareto optimality of candidate solutions. However, all of these works consider intervals on objectives and use Pareto analysis between interval-valued outcomes. Consequently, the solutions are represented by finite bounding-boxes in the objective space as shown in Figure 3.1, where each rectangular represents one solution.

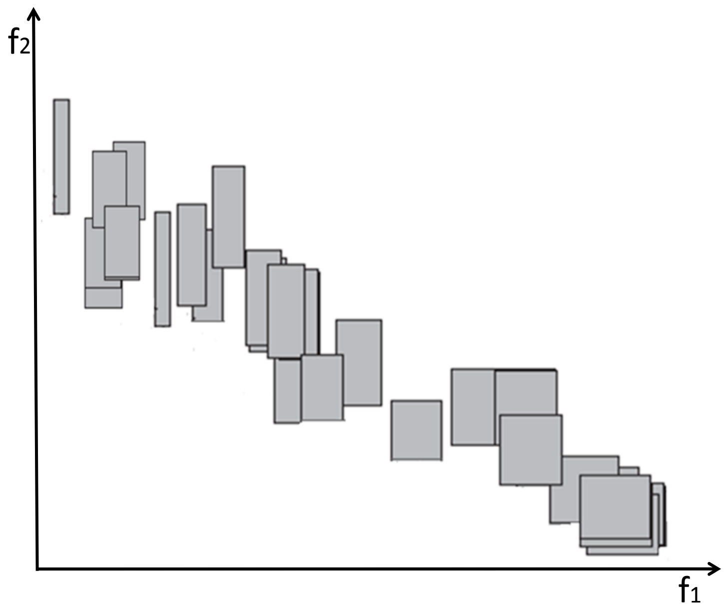

Figure 3.1 – Examples of interval-based solutions

One of the most interesting questions is *how to analyse the Pareto optimality when uncertainty is modeled by non-crisp intervals.* In fact, a representation of uncertain quantities can be defined by means of possibility distributions or fuzzy sets. Unfortunately, it is not easy to compare among two or more non-crisp intervals, possible sets and/or fuzzy numbers. In the following subsection, we give more details about the problem of our interest, especially the choice of using fuzzy numbers and their impact on the optimality process.

3.2.1 Fuzzy MOP formulation

We address here multi-objective optimization problems with fuzzy data in which fuzziness can be associated with the linguistic vagueness, polysemy or ambiguity of information due to limited knowledge. In particular, we focus our attention on fuzzy numbers which are frequently used to represent the approximate reasoning of linguistic values in many real-world applications (Heilpern, 1997). Nevertheless, as fuzziness can have infinite interpretations, there are different shapes of fuzzy numbers to model this fuzziness like triangular, trapezoidal or rectangular. The choice of an appropriate fuzzy shape depends entirely on the type of imperfect data and the problem size. This choice plays substantial role in the design and the overall optimization process of any multi-objective problem.

The most common and popular shape is a Triangular Fuzzy Number (or TFN) depicted using the straight line membership functions and denoted as $A = [\underline{a}, \widehat{a}, \overline{a}]$. As defined in Chapter 2, this linear shape has the advantages of simplicity, smoothness and concise notation (Pedrycz, 1994). Moreover, it can be deduced from transformation of other fuzzy shapes such as trapezoidal, rectangular, etc. For instance, a trapezoidal fuzzy number $[a_1, a_2, a_3, a_4]$ is called a TFN if $a_2 = a_3$. Therefore, the practical aspect of the handling of TFNs encourages us to consider them when dealing with fuzzy multi-objective problems.

In particular, we suppose that fuzzy inputs data in our problems are modeled with the triangular shape. Then as explained before, propagating fuzziness through the optimization process should not be ignored because this may distort the results. Otherwise, fuzziness in inputs data will clearly has great influence over the way a MOP is designed and optimized. Thus we should first consider this fuzziness when designing the multi-objective problem and then predict their unavoidable impacts on the search process. In that sense, we should analyse the effects of the triangular fuzzy shape on

MOPs, especially on the problem outcomes and their optimality. Hence, the objective functions in such problems will be disrupted by the triangular fuzzy shape. Let us assume that a minimization MOP with triangular fuzzy-valued objectives is defined as follows:

Definition 3.1. *Triangular fuzzy MOP*

$$\min \ F(x^{\tau}) = (f_1(x^{\tau}), f_2(x^{\tau}), \ldots, f_n(x^{\tau})) \ s.t. \ \ x \in X, \ \tau \in R \tag{3.1}$$

Where $F(x^{\tau})$ *is the vector objective functions to be minimized, which are disrupted by the triangular form* τ *from the universal set* R *of fuzzy numbers. In the objective space, the vector* F *can be defined as a fuzzy cost function that represents the fitness of solutions by assigning a triangular-valued objective vector* Y^{τ}*:*

$$F : X \rightarrow Y \subseteq (\mathbb{R} \times \mathbb{R} \times \mathbb{R})^n,$$

$$F(x^{\tau}) = Y^{\tau} = \begin{pmatrix} y_1 = [\underline{y_1}, \widehat{y_1}, \overline{y_1}] \\ y_2 = [\underline{y_2}, \widehat{y_2}, \overline{y_2}] \\ \cdots \\ y_n = [\underline{y_n}, \widehat{y_n}, \overline{y_n}] \end{pmatrix} \tag{3.2}$$

It is clear that this formulation is a fuzzy counterpart of the classical MOP definition given in Chapter 1 (Definition 1.1). Figure 3.2 shows an example of triangular solutions in a bi-objective space.

In this case, the solutions are modeled by a set of triangles (i.e. vectors of triangular fuzzy numbers), where each triangle represents one fuzzy solution. Subsequently, once the form of problem solutions is predicted, the issue now is *how to explore the optimality process between them.* Yet, the classical Pareto concepts cannot be used in this case since they are only meant for deterministic case (i.e. when the solutions are exact values).

To this end, a need for special optimality aspects capable to handle the generated solutions of triangular fuzzy values is evident. Before describing our proposal, it is necessary to present another important aspect of fuzzy ranking.

3.2.2 Ranking Fuzzy numbers

In practical use of fuzzy numbers, a comparison procedure is frequently required to check and analyse the relationship between them. Usually, one

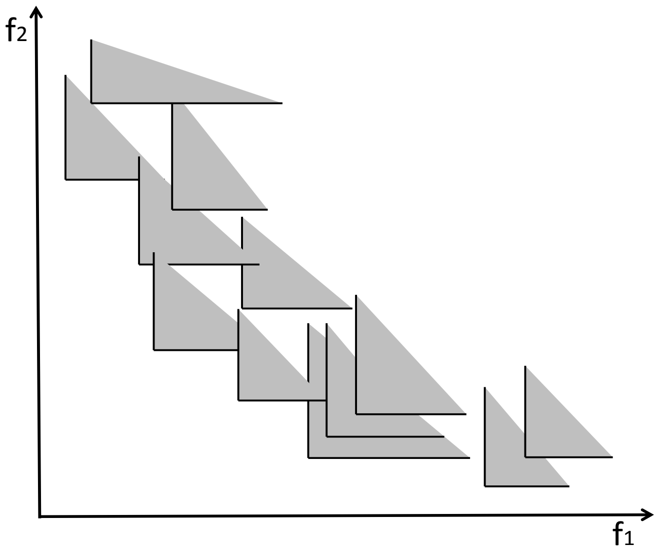

Figure 3.2 – Examples of triangular solutions

fuzzy number needs to be evaluated and compared with the others in order to make a classification of different values. This aspect of fuzzy ranking has been widely discussed by many researchers (Bortolan & Degani, 1985 ; Chen, 1985 ; Choobineh & Li, 1993 ; Cheng, 1998 ; Yao & Wu, 2000) and still receives great attention in recent years (Chu & Tsao, 2002 ; Abbasbandy & Asady, 2006 ; Y.-M. Wang, 2009 ; Ezzati et al., 2012 ; Boulmakoul et al., 2013).

Table 3.1 lists some of existing methods by specifying the used ranking concept and the type of fuzzy shapes to be compared or sorted. For instance, the method of (Chen, 1985) is based on the concept of Max/Min functions to determine the order of triangular or trapezoidal fuzzy numbers, the ordering index of (Choobineh & Li, 1993) is related to area between the left and right barriers of any fuzzy number, the centroid-index method proposed by (Cheng, 1998) consists on calculating the distance of centroid points as order quantities, (Abbasbandy & Asady, 2006) used sign distance to rank triangular or trapezoidal shapes and (Y.-M. Wang, 2009) pointed out the notion of α level set to define ordering between any type of fuzzy numbers.

However, almost each method may contain some shortcomings such as inconsistency with human intuition, requirement of complicated calculations, difficulty of interpretation or indiscrimination in many situations. It is so

Table 3.1 – Fuzzy ranking methods

Concept	**Fuzzy numbers**	**Authors/Refs**
Max/Min sets	Triangular or Trapezoidal	(Chen, 1985)
Ordering index	All types	(Choobineh & Li, 1993)
Centroid index	Triangular or Trapezoidal	(Cheng, 1998)
Sign distance	Triangular or Trapezoidal	(Abbasbandy & Asady, 2006)
Ranking based on α-level sets	All types	(Y.-M. Wang, 2009)

obvious that there is no single best method which may recover all these limits. Moreover, although the existing methods have been successfully applied for ranking fuzzy numbers, they are not powerful when we need to compare at least two vectors of fuzzy numbers. In other words, they are not enough to compare fuzzy-valued solutions in the multi-objective setting.

After presenting the problem details, we will now describe our proposal inspired from the classical Pareto concepts and the fuzzy ranking aspect.

3.3 Pareto optimality for fuzzy MOPs

In this section, we propose a new Pareto dominance for handling optimality in any MOP with fuzzy data, especially with triangular-valued objectives. Then as multi-objectivity usually involves problems with only two objectives, each solution here is a vector of two triangular fuzzy numbers (TFNs).

Hence, our proposal, called *fuzzy Pareto dominance*, is composed of two main phases which are: (i) the definition of mono-objective dominance relations between two TFNs; (ii) the determination of Pareto optimality conditions based on the types of mono-objective dominance found for both objectives.

3.3.1 Mono-objective dominance between two TFNs

At this stage, our aim is to define a dominance ordering between two triangular fuzzy numbers (TFNs). First, it is important to note that all possible topological relationships between two TFNs $A = [\underline{a}, \hat{a}, \overline{a}]$ and $B = [\underline{b}, \widehat{b}, \overline{b}]$ may be covered by only four different situations illustrated in Figure 3.3, namely: *Fuzzy disjoint*, *Fuzzy inclusion*, *Fuzzy weak overlapping* and *Fuzzy overlapping*.

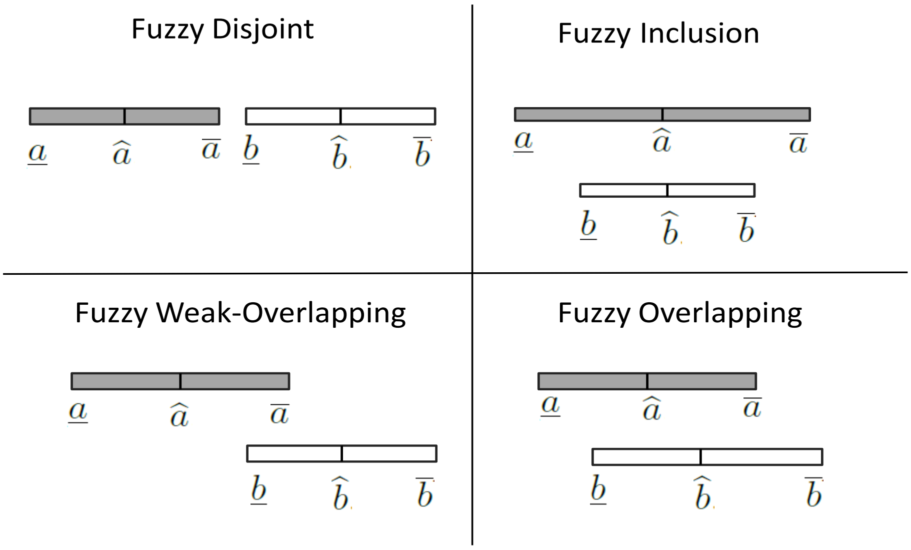

Figure 3.3 – Possible topological situations for two TFNs

Taking these situations into account, we propose three mono-objective dominance relations which are: *Total dominance* ($\prec_t$), *Partial strong-dominance* ($\prec_s$) and *Partial weak-dominance* ($\prec_w$).

Definition 3.2. *Total dominance*
Let $y = [\underline{y}, \widehat{y}, \overline{y}] \subseteq \mathbb{R}$ and $y' = [\underline{y}', \widehat{y}', \overline{y}'] \subseteq \mathbb{R}$ be two triangular fuzzy numbers. y dominates y' totally or certainly, denoted by $y \prec_t y'$, iff:

$$\overline{y} < \underline{y}' \tag{3.3}$$

This dominance relation represents the fuzzy disjoint situation between two triangular fuzzy numbers and it imposes that the upper bound of y is strictly inferior than the lower bound of y' as shown in Figure 3.4.

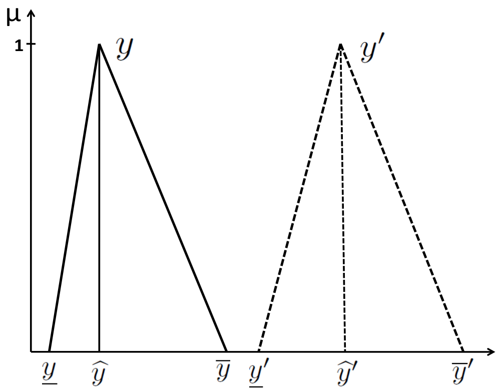

Figure 3.4 – Total dominance

Definition 3.3. *Partial strong-dominance*
Let $y = [\underline{y}, \widehat{y}, \overline{y}] \subseteq \mathbb{R}$ *and* $y' = [\underline{y}', \widehat{y}', \overline{y}'] \subseteq \mathbb{R}$ *be two triangular fuzzy numbers.* y *strong dominates* y' *partially or uncertainly, denoted by* $y \prec_s y'$*, iff:*

$$(\overline{y} \geq \underline{y}') \wedge (\widehat{y} \leq \underline{y}') \wedge (\overline{y} \leq \widehat{y}') \tag{3.4}$$

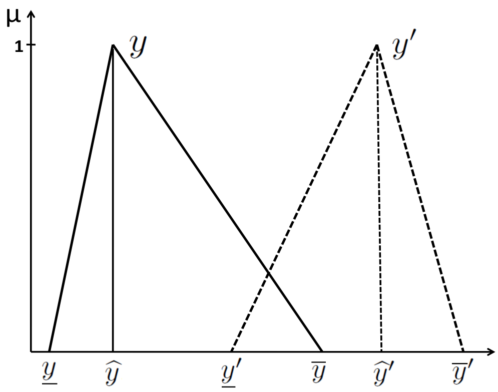

Figure 3.5 – Partial strong-dominance

This dominance relation appears when there is a fuzzy weak-overlapping between both triangles and it imposes that firstly there is at most one intersection between them and secondly this intersection should not exceed the interval of their kernel values $[\widehat{y}, \widehat{y}']$ as shown in Figure 3.5.

Definition 3.4. *Partial weak-dominance*
Let $y = [\underline{y}, \widehat{y}, \overline{y}] \subseteq \mathbb{R}$ *and* $y' = [\underline{y}', \widehat{y}', \overline{y}'] \subseteq \mathbb{R}$ *be two triangular fuzzy numbers.*

y weak dominates y' partially or uncertainly, denoted by $y \prec_w y'$, iff we have:
1. Fuzzy overlapping

$$\begin{gathered}[\underline{y} < \underline{y}' \wedge \overline{y} < \overline{y}'] \;\; \wedge \\ [(\widehat{y} \leq \underline{y}' \wedge \overline{y} > \widehat{y}') \vee (\widehat{y} > \underline{y}' \wedge \overline{y} \leq \widehat{y}') \vee (\widehat{y} > \underline{y}' \wedge \overline{y} > \widehat{y}')]\end{gathered} \tag{3.5}$$

2. Fuzzy Inclusion

$$(\underline{y} < \underline{y}') \wedge (\overline{y} \geq \overline{y}') \tag{3.6}$$

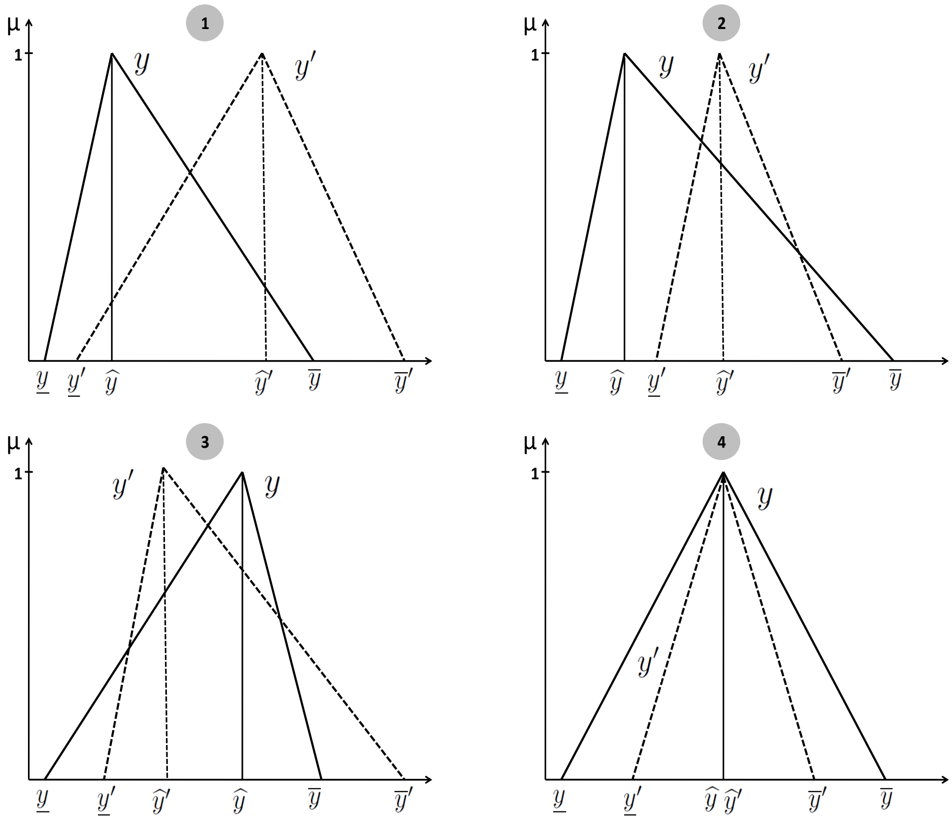

Figure 3.6 – Partial weak-dominance

In this dominance relation, the two situations of fuzzy overlapping and inclusion may occur. Figure 3.6 presents four examples of possible situation. For cases (1) and (2) where both numbers are overlapped, we may conclude that y partially weak dominates y' using Equation 3.5. For cases (3) and (4), where y' is included into y, a situation of incomparability is identified. In fact, the partial weak-dominance relation cannot discriminate all possible cases and leads often to some incomparable situations. In order to discriminate these cases, we suggest to use the middle value positions (the kernel or most plausible value) as an additional criterion of comparison. This may be formally defined by:

$$\widehat{y} - \widehat{y}' = \begin{cases} < 0, & y \prec_w y' \\ \geq 0, & y \text{ and } y' \text{ can be incomparable.} \end{cases}$$

Clearly, we remark that an incomparable situation is identified if we have $\widehat{y} - \widehat{y}' \geq 0$. At this level, the kernel criterion which consists in comparing the discard between both fuzzy triangles will be applied as follows:

$$y \prec_w y' \longleftrightarrow (\underline{y}' - \underline{y}) \leq (\overline{y}' - \overline{y})$$

Similarly, it is obvious that:

$$y' \prec_w y \longleftrightarrow (\underline{y}' - \underline{y}) > (\overline{y}' - \overline{y}).$$

It is easy to check that in the mono-objective case, we obtain a total pre-order between two triangular fuzzy numbers, contrarily to the multi-objective case, where the situation is more complex and it is common to have some cases of indifference.

3.3.2 Fuzzy Pareto dominance

Our goal now is to determine an optimal ordering on the set of fuzzy solutions, where each solution is represented by a vector of triangular fuzzy numbers. Thus, we propose to use the mono-objective dominance relations, defined previously, in order to rank separately the triangular fuzzy values of each objective function. Then depending to the dominance types found for all objectives, we define the Pareto optimality between the triangular fuzzy solutions, which are represented by vectors of triangular fuzzy numbers. Hence, three Pareto relationships are introduced: *Strong Pareto dominance* ($\prec_{SP}$), *Weak Pareto dominance* ($\prec_{WP}$) and *Case of indifference* ($\|$).

Definition 3.5. *Strong Pareto dominance*
Let Y and Y' be two triangular fuzzy solutions. Y strong Pareto dominates Y', denoted by $Y \prec_{SP} Y'$ iff:

$$\begin{array}{r} \forall i \in 1, \ldots, n : [y_i \prec_t y'_i \vee y_i \prec_s y'_i] \quad \vee \\ \exists i \in 1, \ldots, n : [y_i \prec_t y'_i \vee y_i \prec_s y'_i] \wedge \forall j \neq i : [y_j \prec_s y'_j \vee y_j \prec_w y'_j] \end{array} \tag{3.7}$$

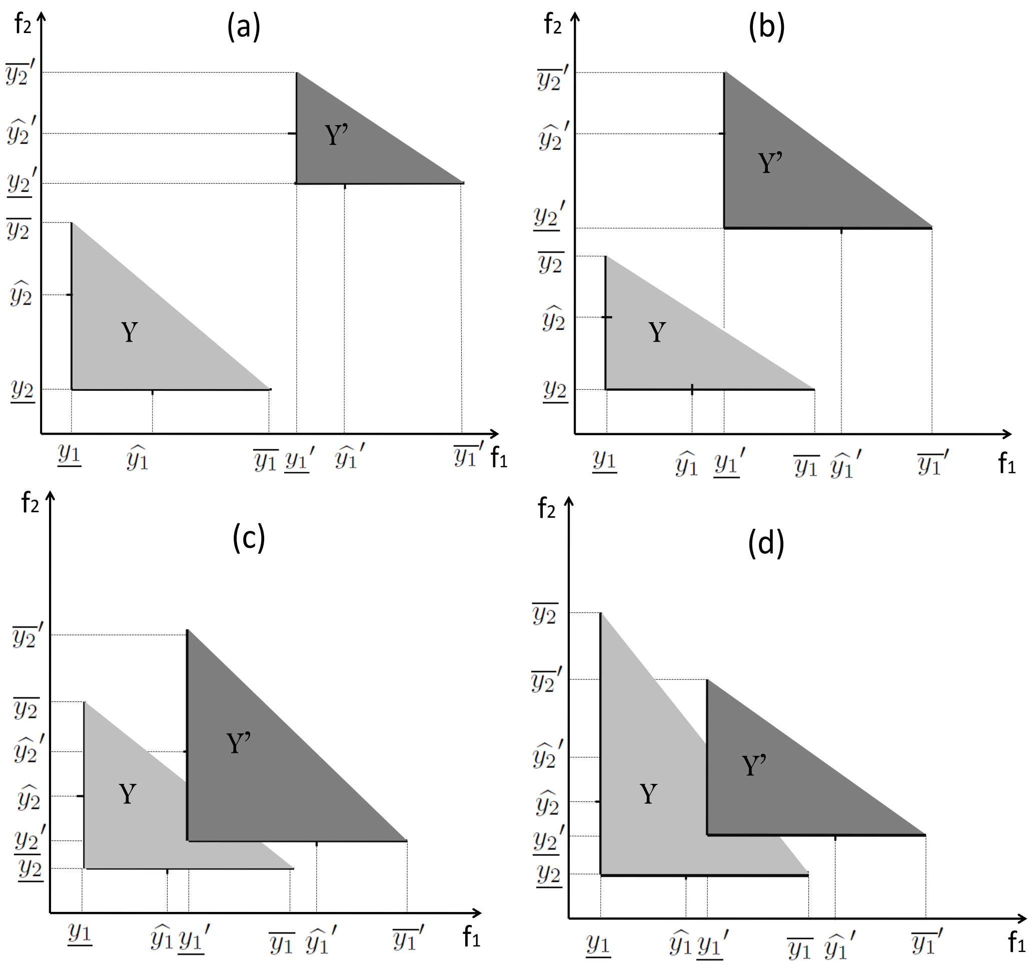

Figure 3.7 – Strong Pareto dominance

The strong Pareto dominance holds if either y_i total dominates or partial strong dominates y'_i in all the objectives (see Figure 3.7-(a): $y_1 \prec_t y'_1$ *and* $y_2 \prec_t y'_2$), either y_i total dominates y'_i in one objective and partial strong dominates it in another (Fig.-(b):$y_1 \prec_s y'_1$ *and* $y_2 \prec_t y'_2$), or at least y_i partial strong dominates y'_i in one objective and weak dominates it in another (Figure 3.7-(c),(d): $y_1 \prec_s y'_1$ *and* $y_2 \prec_w y'_2$).

Definition 3.6. *Weak Pareto dominance*
Let $\overrightarrow{y}$ and $\overrightarrow{y}'$ be two triangular fuzzy solutions. $\overrightarrow{y}$ weak Pareto dominates $\overrightarrow{y}'$, denoted by $\overrightarrow{y} \prec_{WP} \overrightarrow{y}'$, iff:

$$\forall i \in 1, \dots, n : y_i \prec_w y'_i \tag{3.8}$$

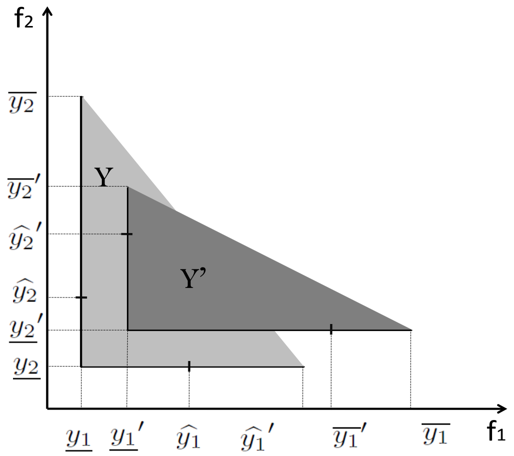

Figure 3.8 – Weak Pareto dominance

The weak Pareto dominance holds if y_i weak dominates y_i' in all the objectives (see Figure 3.8). Yet, a case of indifference (defined below) can occur if there is a weak dominance with inclusion type in all the objectives (see Figure 3.9).

Definition 3.7. *Case of indifference*
Two triangular fuzzy solutions are indifferent or incomparable, denoted by $\overrightarrow{y} \| \overrightarrow{y}'$, iff:

$$\forall i \in 1, \ldots, n : y_i \subseteq y_i' \tag{3.9}$$

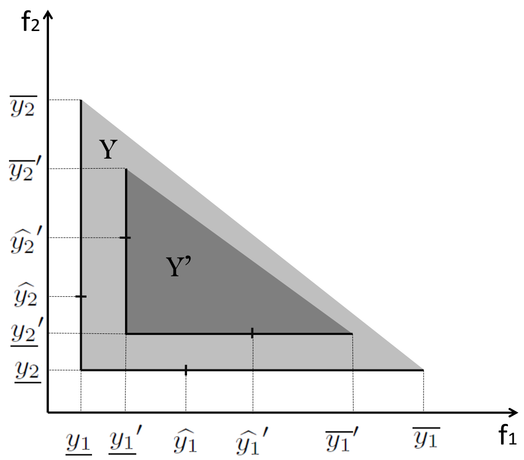

Figure 3.9 – Case of indifference

3.4 Numerical examples

We present in Figure 3.10 some examples to illustrate the advantages of our mono-objective dominance for ranking triangular fuzzy numbers by comparing our results with some other ranking methods.

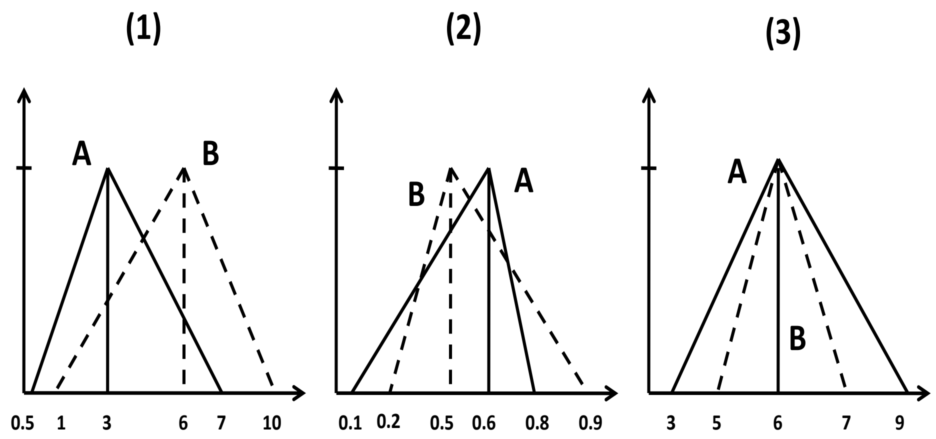

Figure 3.10 – Mono-objective dominance examples

Example 3.1. *Consider the two triangular fuzzy numbers $A = [0.5, 3, 7]$ and $B = [1, 6, 10]$ in Figure 3.10-(1). The ranking order found by most of methods like Cheng's distance (Cheng, 1998), Chu's index (Chu & Tsao, 2002), Wang's centroid index (Y.-M. Wang, 2009) and kaufman's left and right scores (Kaufmann & Gupta, 1988), is $A \prec B$. By using our dominance method (Definition 3), it is easy to check that A weak dominates B partially ($A \prec_w B$). Therefore, the ranking order in our case is the same as other tested methods ($A \prec B$).*

Example 3.2. *Consider the two triangular fuzzy numbers $A = [0.1, 0.6, 0.8]$ and $B = [0.2, 0.5, 0.9]$ (see Figure 3.10-(2)). By using some ranking methods such as Yao's signed distance (Yao & Wu, 2000), Chu's index (Chu & Tsao, 2002) and Abbas's sign distance (Abbasbandy & Asady, 2006), the ranking order is $A \approx B$. This is the shortcoming of previous methods that rank two different fuzzy numbers equally. However, by applying our dominance method, we observe at the first step, that the discrimination between A and B is not possible using Definition 3, since the kernel condition gives $0.6 - 0.5 \geq 0$. At this level, we use the discard criterion ($0.2 - 0.1 = 0.9 - 0.8$) which leads to conclude that A partial weak dominates B, and consequently $A \prec B$.*

Example 3.3. *Consider the two triangular fuzzy numbers $A = [3, 6, 9]$ and $B = [5, 6, 7]$ (see Figure 3.10-(3)). Almost the majority of ranking methods such as (Yao & Wu, 2000 ; Chu & Tsao, 2002 ; Abbasbandy & Asady, 2006 ; Y.-M. Wang, 2009) failed to discriminate two fuzzy numbers having the same symmetrical spread, as for this example $A \approx B$, whereas (Ezzati et al., 2012) prefer the ranking order $B \prec A$ and consider this choice as reasonable result, since it agrees with human intuition. By using our dominance method, we conclude that B partial weak dominates A ($B \prec_w A$), since the discard criterion gives: $5 - 3 > 7 - 9$. Thus, we obtain the same rational result $B \prec A$.*

From these examples, we conclude that our mono-objective dominance method can effectively rank two triangular fuzzy numbers and produces reasonable and intuitive results to the well-defined problems of indiscrimination, that have failed to be ranked by some previous ranking methods. The next example presents a comparison of our Pareto dominance relations with the interval-based optimality proposed by (Limbourg, 2005 ; Limbourg & Aponte, 2005).

Example 3.4. *Figure 3.11 illustrates 4 possible cases between a pair of solutions S_1 and S_2 in a two-dimensional objective space. For each case, we intend to determine the type of dominance relation between both solutions by using:*

- *The Limbourg's interval-based Pareto optimality, denoted by $<_{IP}$,*
- *Our fuzzy Pareto dominance relations, denoted by $\prec_{SP}$ for strong Pareto dominance, $\prec_{WP}$ for weak Pareto dominance and $\|$ for the case of indifference.*

In that sense, every solution in our case is represented by a triangular fuzzy shape (colore in light and dark gray), that is a vector of two TFNs (respectively for objectives f_1 and f_2). For instance, in Figure 3.11-(c1), $S_1 = ([2, 4, 9][1, 3, 6])$ and $S_2 = ([8, 10, 13][5, 9, 12])$. On the other hand, by applying the interval-based approach, every solution will be represented by a rectangular shape (with dotted lines), that is a vector of intervals as for example in (c1): $S_1 = ([2, 9][1, 6])$ and $S_2 = ([8, 13][5, 12])$.

(c1) is the case where S_1 is lower than S_2 in both f_1 and f_2. By using the interval-based optimality (Limbourg & Aponte, 2005), the lowest solution is preferred and thus $S_1 <_{IP} S_2$. In our fuzzy context, we can clearly deduce that $S_1 \prec_{SP}$, because S_1 strong dominates S_2 in all the objectives.

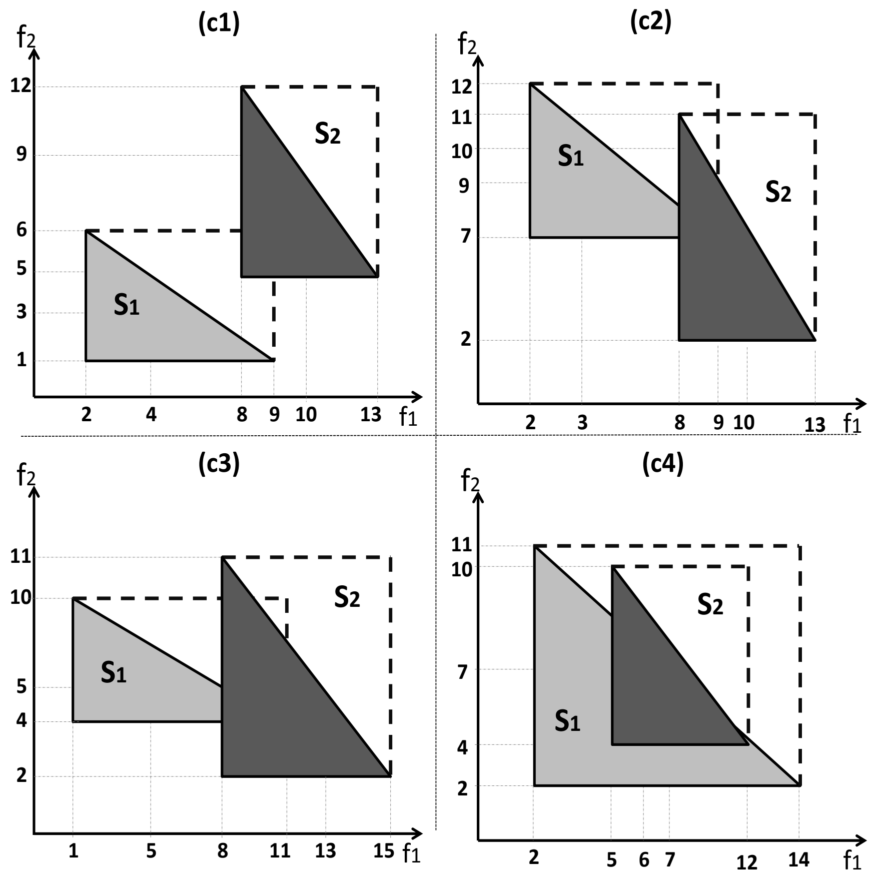

Figure 3.11 – Pareto dominance examples

In cases (c2) and c3, S_1 is better than S_2 in one objective but greater than it or incomparable in the other. In (Limbourg & Aponte, 2005), authors state that these cases depend heavily on the decision maker choice. Otherwise, this latter can interpret them as situations of indifference or use a preference-based decision.

Yet, by applying our fuzzy Pareto optimality, we conclude that $S_1 \prec_{WP} S_2$ in both cases, because S_1 strong dominates S_2 in f_1 and weak dominates it in f_2. For instance in (c2), we have $S_{11} = [2, 3, 9] \leq_s S_{21} = [8, 10, 13]$ and $S_{12} = [7, 9, 12] \leq_w S_{22} = [2, 10, 11]$ with respect to their kernel values comparison, i.e., $(9 - 10) \leq 0$.

Likewise, (c4) represents always a case of incomparability because one solution encloses the other. Thus, we have $S_1 \| S_2$.

From these examples, we remark that our fuzzy Pareto dominance can successfully discriminate incomparable solutions in some critical decision cases. This is mainly due to the flexibility that our approach offers by giving us the choice between weak and strong dominance relation. This may also be explained by the use of the kernel values of our triangular fuzzy distributions as comparison criteria in the cases of indifference. In general, we may conclude that our fuzzy Pareto approach offers a better classification and more accurate knowledge comparing with the interval-based approach.

3.5 Conclusion

In this chapter, we have contributed to the search for Pareto optimal solutions in presence of uncertain objective functions. First, as we have focused on the specific case of MOPs with fuzzy-valued objectives, a survey of some fuzzy ranking methods was conducted. Thereafter, we have suggested new mono-objective dominance relations between a pair of TFNs inspired from the fuzzy ranking reasoning. Through the use of these new dominance relations, a Pareto optimality between fuzzy solutions (i.e. vectors of TFNs) was proposed. Finally, some numerical examples have shown the main advantages of our proposal. The next chapter details our second contribution that aims mainly to use our fuzzy Pareto optimality for extending multi-objective optimization algorithms.

Chapter 4

Fuzzy Pareto-based Optimization Algorithms

Contents

Main publications related to this chapter

Bahri, O., Ben Amor, N., and Talbi, E.-G.: Optimization Algorithms for Multi-objective Problems with Fuzzy Data, 15th International IEEE Symposium Series on Computational Intelligence (SSCI'14), Orlando-Florida USA, December 2014.

Bahri, O., Ben Amor, N., and Talbi, E.-G.:Optimization Algorithms for Multi-objective Combinatorial Problems under Uncertainty, 5th International Conference on Metaheuristics and Nature Inspired Computing (META'14), Marrakech, October 2014.

Bahri, O., Ben Amor, N., and Talbi, E.-G.: Optimization algorithms for solving multi-objective vehicle routing problem with fuzzy demands, 4th International Conference on Complex Systems and Applications, Normandie University (ICCSA'14), Le Havre, France, June 2014.

Bahri, O., Ben Amor, N., and Talbi, E.-G.: Multi-objective Optimization of Combinatorial Problems with Fuzzy Data, 22nd International conference on multiple criteria decision making (MCDM'13), Málaga Spain, June 2013.

4.1 Introduction

As noted in the first chapter, multi-objective evolutionary algorithms (MOEAs) have proved to be as the most popular and powerful methods for solving all combinatorial MOPs. However, in EMO (Evolutionary Multi-objective Optimization) community, the problematic of including uncertainty in resolution methods is often ignored by most of researchers. Usually, they are limited to transform the uncertain MOP into a mono-objective or deterministic problem and then to simply resolve it using the classical deterministic algorithms. To this end, our interest consists in providing a new evolutionary approach able to solve any fuzzy MOP with fuzzy-valued objectives. Especially, we intend to incorporate fuzziness in the search process of Pareto-based MOEAs. In this setting, there are many aspects to pursue such as the Pareto optimality, the diversity and convergence criteria, the solutions distribution, the computational costs, etc. This work is an attempt to tackle the evolutionary computation domain in fuzzy environment.

The rest of this chapter is structured as follows. Section 4.2 describes the main components of PMOEAs and presents in detail the well-studied SPEA2 and NSGAII algorithms. Section 4.3 presents the techniques adopted for extending these algorithms to our fuzzy context. Section 4.4 demonstrates the usefulness of our approach through the resolution of a practical *Multi-objective Vehicle Routing Problem with Time Windows and Fuzzy Demands* and Section 4.5 shows the experimental analysis and gives on overview of achieved results.

4.2 Pareto-based MOEAs

This section provides a unified view of PMOEAs and then focuses on two algorithms that we are interested in.

4.2.1 Components description

Almost all PMOEAs follow the main steps described in the flowchart given in Figure 4.1. This description is inspired from the MOEA process flowchart proposed in (Liefooghe, 2009).

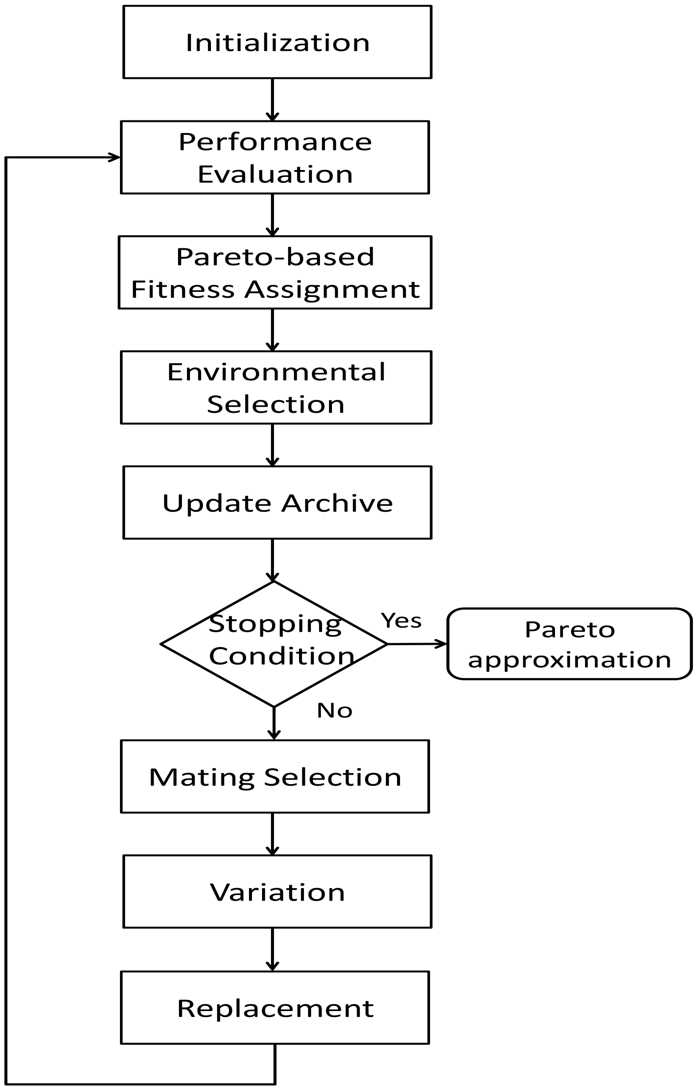

Figure 4.1 – General scheme of PMOEAs

- **Step 1: Initialisation**
  The first and important step performed by an PMOEA is the initialization which consists in generating the initial population. This phase affects the outcomes quality in terms of diversity and convergence. In fact, the initialization is done by taking randomly or according to given diversity functions a sample of individuals (i.e. solutions) to fill up the population.

- **Step 2: Performance Evaluation**
  This step is the first search direction that determines the overall performance of algorithm. Otherwise, the search procedure will be misguided if the performance evaluation is inaccurate. The aim is to evaluate solutions in the objective space using a performance measure, so called fitness or dummy function. At each generation, a fitness value is so assigned to each solution in order to ensure its quality.
- **Step 3: Pareto-based fitness assignment**
  This step is the guiding mechanism of the algorithm. It is based on the use of Pareto dominance as fitness assignment strategy. At every generation of PMOEA, solutions are ranked in the search process using the dominance relationship. This rank represents the quality of solutions in terms of convergence and guides the population toward non-dominated set.
- **Step 4: Environmental selection/Diversity**
  In order to prevent premature convergence provided by privileging non-dominated solutions, an environmental selection phase is usually applied. This step aims to guarantee a uniformly distributed approximation set. Otherwise, the best dispersed solutions over the objective space are selected at each generation. The selection is invoked by applying a diversity preserving technique. Such technique is often based on a given distance measure for estimating the density between solutions.
- **Step 5: Archiving/Elitism**
  This step consists in storing and/or updating best non-dominated individuals found during the search process in an external population so-called archive. This latter prevents the loss of solutions during the optimization process. Depending on the PMOEA parameters (i.e. elitist or non elitist algorithms), there are many types of archiving such as: no archive, an unbounded archive, a bounded archive or a fixed-size archive.
- **Step 6: Stopping condition**
  Defining one or more stopping conditions is not trivial because this has a significant impact on the computational algorithm costs. In PMOEAs, stopping criteria may be simply related to a certain number of iterations or evaluations. For instance, a criterion may be defined to analyze the Pareto approximations yielded by different iterations

and consequently to check the progress of optimization process.

- **Step 7: Mating Selection**
  In this step, pairs of solution sets are selected for the mating pool. This means that the mutated sets will be used to form the parents of recombination operations. Usually, the mating selection strategy is a deterministic binary tournament between two random solutions.
- **Step 8: Variation**
  Variation operators are used to progress the search space by manipulating the solutions encoding. In particular, these operators are applied to some parent solutions in order to recombine and/or change them and thereafter to produce new solutions called offsprings population. PMOEAs are mainly based on two genetic operators: crossover (or recombination) and mutation.
- **Step 9: Replacement**
  In the replacement step, survivors are selected from both current and offspring populations. For instance, an elitist replacement strategy consists of preserving only the best solutions with respect to a predefined population size. For that, the worst solution is iteratively deleted until reaching the required group size of solutions.

Depending on the problem design and properties, simple modifications on this unified scheme are introduced to model the different existing PMOEAs. Such modifications basically appear at the fitness assignment step, the diversity preservation technique and the elitism strategy which are the three key components for every MOEA. Table 4.1 summarizes the components of two well-known algorithms, namely NSGAII (Deb et al., 2002) and SPEA2 (Zitzler et al., 2001) as instances of the general PMOEA scheme. In the following, we present in more detail the principle of each algorithm.

4.2.2 NSGA-II

The *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) (Deb et al., 2002) is one of the most popular PMOEAs for solving MOPs. Its popularity is often associated to its low computational complexity, explicit technique for diversity preserving and the no-archive strategy. Contrary to its original version NSGA (Srinivas & Deb, 1994), it does not use a fitness sharing mechanism for diversity preserving, but rely on a crowded comparison procedure.

Table 4.1 – NSGAII vs. SPEA2 components

	NSGAII	SPEA2
Step 1	random initial population	random initial population
Step 2	dominance-depth strategy	dominance-rank/count strategy
Step 3	Pareto-based fitness	Pareto-based fitness
Step 4	crowding-diversity technique	nearest neighbor method
Step 5	no archive	fixed size archive
Step 6	number of generations	number of generations
Step 7	binary tournament	binary tournament
Step 8	crossover and mutation	crossover and mutation
Step 9	elitist replacement	generational replacement

The principle of NSGAII is as follows. At each generation, solutions from the current population are ranked in terms of two criteria (i.e. convergence and diversity). In this setting, two ordering operations are performed. The first ordering represents the quality of solutions in terms of convergence and consists in consists in assigning a fitness value to every solution. For that, a fitness assignment procedure based on the dominance-depth and Pareto relationship is applied. More precisely, all population individuals are subdivided into several fronts using the Pareto dominance. The non-dominated solutions belonging to the first front are the best efficient set and assigned a large fitness value. Then, those of the second front are assigned a smaller fitness value, and so on. This process is repeated until classifying the whole population.

The second ordering represents the density estimation or local quality of solutions surrounding a particular point in the i-th front they belong to. At this stage, NSGAII uses a diversity preserving technique based on a *Crowding distance* CD in order to evaluate solutions belonging to the same ranking level (i.e. with identical fitness values). Formally, the normalized CD of a solution is the sum of its individual objectives' distances, that in turn are the differences between the solution and its closest neighbors.

$$CD(i) = \sum_{i=1..n} (f_i(i+1) - f_i(i-1))/(f_i^{max} - f_i^{min}) \ \ s.t. \ \ i \in F \tag{4.1}$$

where $f_i(i+1)$ and $f_i(i-1)$ are the neighbor objective values of $i-$th objective, n is the number of objectives ,f_i^{max} and f_i^{min} are respectively the population maximum and minimum objective values and F is the $i-$th front to which solutions are associated. Notice that, the first and last solutions in the rank (i.e. those with smallest and largest objective values) have a CD equal to infinity. By considering both ordering criteria, a solution is said to be better than another one if it has the best fitness value (i.e. the better rank), or in case of equality (i.e. belong to the same front) if it is located in the least crowded region (i.e. lower CD value).

Subsequently, the mating selection step is a deterministic tournament strategy between two random solutions. At the replacement step, only the best solutions survive with respect to the appropriate population size. It should be noted here that the distinctive feature of NSGAII lies in using the crowded-comparison procedure as truncation operator to reduce the population in the environmental selection step and in considering it as a second selection criteria when two solutions have the same rank in the tournament selection step. Another important feature is that this algorithm does not use an explicit archive for the archiving operation, but it only considers an intermediate population to store efficient solutions found during the search.

4.2.3 SPEA2

The *Strength Pareto Evolutionary Algorithm* (SPEA2) is an improved version of SPEA algorithm (Zitzler et al., 2000), where a mixed strategy of fitness ranking is adopted (Zitzler et al., 2001). Otherwise, SPEA2 uses a Pareto-based fitness assignment which incorporates both rank and count dominance strategies. In this way, at each iteration, every individual is assigned a strength value which is proportional to the number of solutions dominated by it according to the Pareto approach. Then, the fitness of an individual F^t is computed as the sum of its strength dominators. It is important to note that fitness is to be minimized, i.e. a non-dominated solutions yields a zero fitness value $R_i = 0$. Then, in the case of identical solutions (i.e. having the same fitness values), a density factor is added in SPEA2 to discriminate between them and to preserve their diversity. This density is estimated by means of a nearest neighbor method. This method consists in calculating for each solution the Euclidean distance to its k^{th} nearest neighbor and then in adding the reciprocal value to the fitness vector. This means that the neighborhood density corresponds to the inverse of distance to the k^{th} nearest solutions. Formally, the *Euclidean distance EU* between two solutions (i.e. deterministic objective vectors) y_1 and y_2 is given by:

$$EU(y_1, y_2) = \sqrt{\sum_{i=1}^{n} (y_{1i} - y_{2i})^2} \tag{4.2}$$

Once all solutions have their fitness values, a binary tournament selection with generational replacement is applied to create the mating pool. During the selection process, SPEA2 uses an external fixed size archive to store the offspring non-dominated solutions. This means that the number of stored external solutions is constant over time and so if the predefined archive size is higher than the number of non-dominated solutions, the archive will be filled up by dominated solutions. SPEA2 has potential features that makes it different from other algorithms. This uniqueness lies mainly in its mixed fitness assignment strategy that reduces the number of selected non-dominated solutions without destroying the efficiency of the Pareto-optimal front, its density estimation technique based on neighborhood strategy and the use of both intermediate and external populations to store elite/non-elite solutions.

After presenting the algorithms details that are of our interest, the issue now is how to extend them to uncertain data environments. In fact, all existing PMOEAs assume that all inputs and outputs are deterministic values. Then as we focus our study on multi-objective problems (MOP) with fuzzy data, the goal becomes to propose optimization algorithms able to cope with fuzziness. The next section presents our proposal.

4.3 Extended PMOEA for fuzzy MOPs

Our contribution here is to design optimization algorithms for handling any MOP with fuzzy-valued objectives. The basic idea is to exploit the common steps from the general PMOEA scheme and then adapt them to the fuzzy context. As mentioned before, the overall PMOEA process is most strongly influenced by three main components: Pareto dominance relationship, diversity preserving strategy and archiving/elitism technique. It is therefore necessary to adapt each of theses components in order to enable such algorithms working in a fuzzy context.

First, knowing that in a MOP with fuzzy-valued objectives, solutions are usually affected by fuzziness. In addition, we remember that in our case the encoding type of objective functions is a triangular fuzzy shape (see

Equation 3.2 in chapter 3). Consequently, every solution is composed of a vector of triangular fuzzy numbers (TFNs). This kind of fuzziness will be propagated step by step in the PMOEA optimization proces. In the initial step, a population with a set of triangular fuzzy solutions is randomly sampled. These solutions should then be evaluated and ranked based on Pareto dominance concepts. However as the classical Pareto concepts can only be applied between exact (or crisp) solutions, we propose to use our new fuzzy Pareto dominance relations introduced in the previous chapter.

Next, once all fuzzy solutions are ranked, the local density will be estimated in the selection step by measuring the distance between neighbor solutions. Such a distance measure is typically applied between exact values. At this level, we need clearly a specific measure for computing distance between fuzzy values. Thereafter, the archiving tool must also be able to store the selected fuzzy solutions. The remaining steps still unchanged because they are independent to the type of solutions.

Following these remarks, any PMOEA can be extended to our fuzzy context by integrating these modifications into the search process. In this work, we illustrate the fuzzy extension on the two popular algorithms: NSGAII (Deb et al., 2002) and SPEA2 (Zitzler et al., 2001). These two algorithms have proved to be very powerful tools for multi-objective optimization. Due to their population-based nature, they are able to generate multiple optimal solutions in a single run with respect to the good convergence and diversification of obtained solutions.

4.3.1 E-NSGAII

To extend NSGAII, we propose at the first stage to replace the standard Pareto by our new fuzzy Pareto dominance defined in the previous chapter. This modification allows to ensure the fitness assignment ranking in a fuzzy setting. At the second stage, we provide an adaptation of the diversity preserving technique where a crowded-comparison procedure is applied. This procedure is based on a *Crowding distance* CD that serves to get a discrimination of solutions having the same rank level (see Equation 4.1).

However, this distance cannot remain unchanged in our fuzzy context since it depends directly on the differences between objective values. Then as our objective functions are vectors of triangular fuzzy numbers (or TFNs), the distance measure must be adapted to fuzziness. Thus, we simply propose to approximate these objectives by computing their expected values before applying the *Crowding distance*. Indeed, the distance between each pair of objective values will be then substituted by the differences between the corresponding expectations. Formally, the *Expected value* E of a given triangular fuzzy number $y_i = [\underline{y_i}, \widehat{y_i}, \overline{y_i}]$ is calculated using the following formula (Z. Wang & Tian, 2010):

$$E(y_i) = (\underline{y_i} + 2 \times \widehat{y_i} + \overline{y_i})/4 \tag{4.3}$$

Notice that, we do not need a fuzzy extension of the elitism technique because there is no archive. Finally, the two simple refinements at fitness assignment and diversity preserving steps are incorporated into the search process of NSGAII. We denote by E-NSGAII the new algorithm detailed in Algorithm 4.1.

4.3.2 E-SPEA2

Similarly to the previous algorithm, we first suggest to integrate our fuzzy Pareto dominance relations in the SPEA2 fitness assignment strategy. Then as detailed before, SPEA2 uses a nearest neighbor density estimation technique which allows a more precise guidance of the search process. This technique requires an *Euclidean distance* to preserve diversity in the population. However knowing that *Euclidean distance* should be applied only between two exact vectors (see Equation 4.2) and as solutions in our case are vectors of TFNs, we should replace this latter by a specific fuzzy distance.

Therefore, we choose to use the so-called *Bertoluzza metric* (Bertoluzza et al., 1995) in order to compute the distance between two fuzzy solutions based on α-cut principle. More precisely, given two vectors of TFNs $y = (y_1, ..., y_n)$ and $y' = (y'_1, ..., y'_n)$ such that $y_i = [\underline{y_i}, \widehat{y_i}, \overline{y_i}]$ and $y'_i = [\underline{y_i}', \widehat{y_i}', \overline{y_i}']$, the *Bertoluzza metric* is first applied to compute distances between every pair of fuzzy numbers y_i and $y - i'$. Then, we propose to compute the weighted mean of the overall distances $d(y_i, y'_i)$ in order to estimate the final distance between both vectors $D(y, y')$. Formally, it is given by:

$$d_{\theta}(y_i, y'_i) = \sqrt{\int_0^1 (mid(y_{i_\alpha}) - mid(y'_{i_\alpha}))^2 + \theta\ (spr(y_{i_\alpha}) - spr(y'_{i_\alpha}))^2 d_\alpha} \tag{4.4}$$

where y_{i_α} denotes the α-cut of y_i defined as as an α level set (or bijection) associating for any $\alpha \in [0, 1]$ a bounded interval $[\underline{y_{i_\alpha}}, \overline{y_{i_\alpha}}]$, $mid(y_{i_\alpha}) = \frac{1}{2}(\underline{y_{i_\alpha}} +$

$\overline{y_{i_\alpha}}$) denotes the midpoint of y_{i_α}, $spr(y_{i_\alpha}) = \frac{1}{2}(\overline{y_{i_\alpha}} - \underline{y_{i_\alpha}})$ is the spread (or radius) of y_{i_α} and $\theta \in [0, 1]$ is a parameter that allows us to weight the effect of the deviation between spreads. For a sake of simplicity, we will consider in our case the 0-cut level ($\alpha = 0$) where $y_{i_0} = [\underline{y_{i_0}}, \overline{y_{i_0}}] = [\underline{y_i}, \overline{y_i}]$ is the topological support of y_i and we choose a parametrization with the value $\theta = \frac{1}{2}$.

By using this fuzzy distance, the distance to each k-nearest fuzzy solution can be estimated in order to select the well-distributed solutions. Thereafter, unlike the NSGAII algorithm, SPEA2 uses a regular population and also an external storage, the so-called archive. Thus, we need to extend this archive to the triangular fuzzy space in order to enable it keeping the best triangular solutions during the optimization process.

In next steps, namely *mating selection*, *variation* and *replacement* remain unchanged in our fuzzy context. The above extensions are integrated into the search process of SPEA2 and leads to a new version called E-SPEA2 and described in Algorithm 4.2.

Algorithm 4.1 E-NSGAII

Input : Initial population P
Maximum number of generations T
Output: Best non-dominated set
begin
Initialization. create a random population P of N fuzzy solutions;
repeat
Evaluation. calculate fitness values of solutions in P using dominance-depth strategy;
Fitness Assignment. rank all solutions in P using the fuzzy Pareto dominance;
Environmental Selection. select the non-dominated fuzzy solutions based on their expected crowding values and copy them in an external population P';
if *size of* P' *exceeds* N **then**
add the least crowded solutions to P';
else if *size of* P' *is less than* N **then**
set P' with dominated solutions;
else
the environmental selection is completed;
end
Elitism. update $P \cup P'$
until *Stopping condition. Number of generations* $> T$ *or another criterion is satisfied*
Mating Selection. perform a binary crowded tournament selection to select parents from P';
Variation. apply job order crossover (JOX) and simple swap mutation operators;
Replacement. replace old population by the resulting offspring population members.
end

Algorithm 4.2 E-SPEA2

Input : Initial population P
Archive A
Maximum number of generations T
Output: Best non-dominated set
begin
Initialization. create a random population P of N fuzzy solutions and create an empty triangular archive of fixed size M;
repeat
Evaluation. calculate fitness values of solutions in P and A based on dominance-count and rank strategy;
Fitness Assignment. rank solutions in population and archive using the fuzzy Pareto dominance;
Environmental Selection. copy all non-dominated solutions from P to the triangular archive A;
if *size of A exceeds M* **then**
A is pruned by means of a clustering procedure;
else if *size of A is less than M* **then**
fill A with best dominated solutions;
else
the environmental selection is completed;
end
Elitism. update A;
until *Stopping condition. Number of generations $> T$ or another criterion is satisfied;*
Mating Selection. perform a binary tournament selection with replacement on A to fill the mating pool;
Variation. apply job order crossover (JOX) and simple swap mutation operators;
Replacement. replace old population by the resulting offspring population members.
end

4.4 Application on a multi-objective VRPTW with fuzzy demands

This section provides a formal definition and mathematical formulation of the problem we seek to solve, namely the *Multi-objective Vehicle Routing Problem with Time Windows and Fuzzy Demands* (MO-VRPTW-FD).

4.4.1 Problem definition

MO-VRPTW-FD is an NP-hard and well-studied problem that can be defined as a combination of two variants namely, *VRP with Time Windows* (VRPTW) and *Stochastic VRP* (see Figure 1.8), where the customer demands are not stochastic but fuzzy and many conflicting objectives are to be optimized.

The problem can be modeled with a weighted graph $G(V, A)$ with an arc set A and a set of vertices $V = \{0, 1, \ldots, n\}$, where the vertex 0 represents the central depot. Let $V - \{0\}$ be the set of customer vertices and let $K = \{1, \ldots, m\}$ be the set of homogenous vehicles having a limited capacity Q which must not be exceeded. Note that, each vehicle $k \in K$ has a symmetric distance $D_{i,j}$ to travel from a customer i to customer j. A feasible route is usually defined by the set of served customers starting and ending at the depot vertex 0, i.e., that is $r = \{0, 1, \ldots, n, 0\}$. In our case, every customer has a fuzzy demand value dm represented with a triangular fuzzy number. In that sense, the exact demand is only known when the vehicle arrives at the customer location. Additionally, customers should be served within a fixed interval of time $[e_i, l_i]$, the so-called time window. For sake of simplicity, there is no demands and service times for the depot.

For this problem, two objective functions have to be minimized which are respectively, the total traveled distance D and total tardiness time T. Indeed, the tardiness time objective is associated to the failure to respect the *time windows constraint*, in particular when the service time exceeds its upper bound. On the other hand, the traveled distance objective depends mainly on the *vehicle capacity constraint*. This latter imposes that the total of customer demands is less than or equal to the vehicle capacity. Then if the constraint is not satisfied, the delivery fails and causes wasted costs in terms of distance.

Figure 4.2 illustrates an example with a central depot, 3 vehicles having a maximum capacity $Q = 10$ and a set of 8 customers represented by nodes. Each customer $i = 1 \dots 8$ has a demand expressed by a triangular fuzzy number $dm = [\underline{dm}, \widehat{dm}, \overline{dm}]$ (e.g. fuzzy demand of customer 1 is $dm_1 = [2, 7, 11]$). However, in this case, we cannot directly determine if the *capacity constraint* is satisfied or not, since the customer' demands are fuzzy values. For instance, consider the customer 7 with fuzzy demand $dm_7 = [8, 10, 13]$, we cannot check if dm_7 is lower, equal or higher than $Q = 10$ in order to estimate the traveled distance. This figure illustrates also a case of routes failure (represented with dotted lines) when we suppose that vehicle 1 cannot serve its second customer with its remaining capacity. Thus, the vehicle must return to the depot to load and then goes back to serve its last customer.

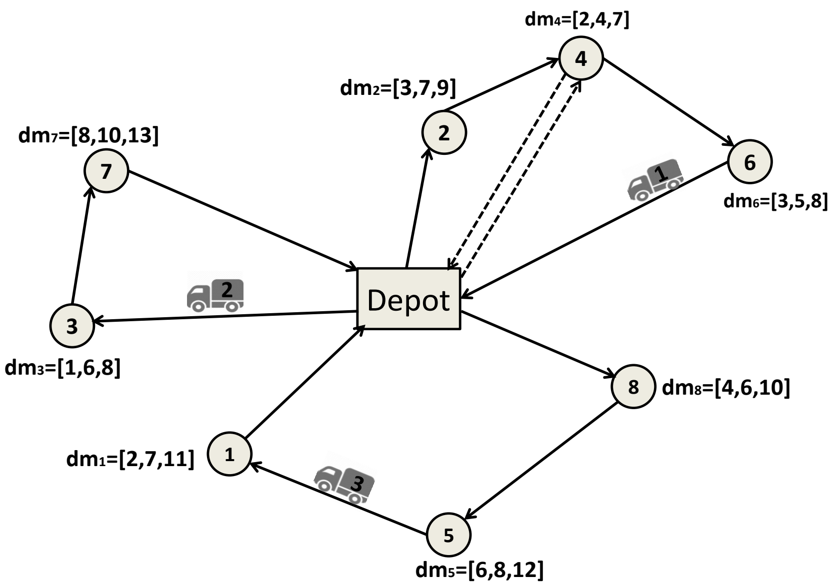

Figure 4.2 – MO-VRPTW-FD problem

In general, there are three possible situations in order to avoid a route failure:

- *Customer demand is lower than the vehicle capacity, i.e.,* $dm_i < Q$: The vehicle serves the current customer c and then moves to the next one $c + 1$.
- *Customer demand is equal to the vehicle capacity, i.e.,* $dm_i = Q$: The

vehicle leaves the depot 0 to serve the first customer c with its total capacity, it returns to the depot to load and then serves the next customer $c+1$. This is so-called a priori optimization strategy. In this situation, the traveled distance is given by:

$$D(r) = D_{0,1} + \sum_{i=1}^{c-1} D_{i,i+1} + D_{c,0} + D_{0,c+1} + \sum_{i=c+1}^{c-1} D_{i,i+1} + D_{n,0}$$

- *Customer demand is higher than the vehicle capacity, i.e.,* $dm_i > Q$: The vehicle serves the customer c with its remaining capacity, goes to the depot to load, returns back to the same customer c to deliver the remaining quantity and then moves to the next customer $c+1$. The traveled distance is given by:

$$D(r) = D_{0,1} + \sum_{i=1}^{c-1} D_{i,i+1} + D_{c,0} + D_{0,c} + D_{n,0}$$

Yet, as the demand of each customer is a triangular fuzzy number $dm_i = [\underline{dm_i}, \widehat{dm_i}, \overline{dm_i}]$, we propose to verify separately the capacity constraint satisfaction for the triplet of its demand values. Hence, the traveled distance will be computed three times for the lower, middle and upper demand values and consequently obtained as a triangular variable $D = [\underline{D}, \widehat{D}, \overline{D}]$. Typically, it should be noticed that the travel time depends on the corresponding traveled distance to serve customers. Then as distance in our context is obtained as a triangular fuzzy variable, the total tardiness time will be also disrupted by this fuzzy form $T = [\underline{T}, \widehat{T}, \overline{T}]$.

4.4.2 Mathematical formulation

The MO-VRPTW-FD can be formulated as follows:

Notations:

- dm_i denotes the fuzzy demand of a customer $i \in V - \{0\}$,
- Q denotes the loading capacity of each vehicle $k \in K$,
- e_i and l_i are respectively the lower and upper bounds of the time window of vertex i (i.e., depot or customers),
- a_i and b_i refers respectively to the arrival and departure times to vertex i,
- t_{ij} refers to the travel time from vertex i to j,
- s_i is the customer service time which corresponds to the time of goods loading/unloading,
- w_i is the waiting arrival time at customer i,
- $D_{i,j}$ is the traveled distance between vertices i and j,
- T_i is the delay or tardiness time at vertex i,

Decision variable:

$$x_{ij}^k = \begin{cases} 1 & \text{if the } arc(i,j) \in A \text{ is used by a vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

Objective functions:

$$\min \ (f_1(x) = \sum_{(i,j)\in A} D_{ij} \ x_{ij}^k \ , \ f_2(x) = \sum_{i\in V} T_i) \tag{4.5}$$

s.t.

$$\sum_{k\in K}\sum_{j\in V} x_{ij}^k = 1 \qquad \forall i \in V - \{0\} \tag{4.6}$$

$$\sum_{i\in V-\{0\}} dm_i \sum_{j\in V} x_{ij}^k \ \leq \ Q \qquad \forall k \in K \tag{4.7}$$

$$\sum_{j\in V} x_{0j}^k = 1 \qquad \forall k \in K \tag{4.8}$$

$$\sum_{i\in V} x_{iu}^k - \sum_{j\in V} x_{uj}^k = 0 \quad \forall u \in V - \{0\}, \ \forall k \in K \tag{4.9}$$

$$x_{ij}^k(b_i + s_i + t_{ij} - b_j) \leq 0 \quad \forall (i,j) \in A, \ \forall k \in K \tag{4.10}$$

$$w_i = \begin{cases} 0 & \text{if } a_i \geq e_i \\ e_i - a_i & \text{otherwise.} \end{cases} \tag{4.11}$$

$$T_i = \begin{cases} 0 & \text{if } b_i \leq l_i \\ b_i - l_i & \text{otherwise.} \end{cases} \tag{4.12}$$

The following equations represent the problem constraints to be satisfied:

— Equation (4.5) gives the objective functions which are respectively the minimal total traveled distance and the minimal total tardiness time.

— Equation (4.6) states that every customer is visited just once and by one vehicle.

— Equation (4.7) is the *vehicle capacity constraint* which imposes that the total customer demands cannot exceed the vehicle capacity Q.

— Equations (4.8) and (4.9) require that each vehicle leaves depot once, leaves a customer u after completing its service, and returns next to the depot.

— Equation (4.10) states that vehicle k cannot arrive at customer j before $b_i + s_i + t_{ij}$ if it travels from i to j.

— Equations (4.11) and (4.12) assumes that the time window $[e_i, l_i]$ of vertex i can be not respected. Then if a vehicle arrives before the lower bound e_i, it must wait a while waiting time w_i. Otherwise, if it leaves after the upper bound of its time window l_i, a delay or tardiness time T_i is resulted.

4.5 Experimental design

The extended E-SPEA2 and E-NSGAII algorithms were applied on the MO-VRPTW-FD problem and implemented in C++ using the ParadisEO framework, especially with the MOEO module under Linux (Liefooghe et al., 2007). For a fair comparison and evaluation, all tested algorithms share the same base parameters such as the chosen variation operators, the initial population, the random seed, etc.

4.5.1 Benchmarks

To the best of our knowledge, there is no common benchmark available in the literature for stochastic VRPs (K. C. Tan et al., 2007). Hence, to evaluate our model with fuzzy customer demands, we create a new benchmark for MO-VRPTW-FD by adapting the problem instances provided by (Solomon, 1987), which is a popular reference for evaluating most methods designed for VRPTW. Our interest is focused on the larger benchmark of 100-customers that includes a total of 56 different problems. Each problem is composed of 100 geographically distributed customers, a unique depot and a fleet of homogenous vehicles having a same capacity Q=60 and a constant speed fixed to 1 (distance unit/time unit). The travel times between customers is proportional to the corresponding *Euclidean distances* between them.

According to the customer distribution and time-windows size, these problems are classified into 6 categories, namely C1, C2, R1, R2, RC1, RC2. For instance, the category C corresponds to a uniform geographic distribution of customers in clusters, the category R corresponds to a random generation of customers and the category RC represent a mix of random and clustered customers. Besides, the notation 1 or 2 associated with the name letters indicate the size of time windows. Problems belonging to categories C1, R1 and RC1 have a very tight time windows and allow a short service horizon (i.e. a low capacity of vehicles) with few customers per route (approximately 5 to 10). In contrast, problems in R2, C2 and RC2 have large time windows which are often hardly constraining and permit many customers (more than 30) to be visited per route. Each problem differs also with respect to the percentage of customers with time windows (density of 25, 50, 75 and 100 %), their positions and restrictions. For example, in R104 (belonging to the category-type R1), the customers are uniformly distributed in the space with a time density of 25 %. In RC205, the customers are dispersed in a mixed way (uniform and clusters) and every customer has a time window (time density is 100 %).

Thereafter, we propose to adapt the 56 Solomon's problem to our fuzzy context by applying the following methodology: The basic idea is to generate for each deterministic instance its fuzzy sampled version, in which each crisp demand values is replaced by its corresponding fuzzy value. As shown in Figure 4.3, the kernel value ($\widehat{dm}$) for each triangular fuzzy demand dm is firstly kept the same as the crisp demand value dm_i of the current instance. Then, the lower ($\underline{dm}$) and upper ($\overline{dm}$) bounds of this triangular fuzzy demand are uniformly sampled at random in the intervals $[50\%dm, 95\%dm]$ and $[105\%dm, 150\%dm]$, respectively. The new fuzzy instances are labeled with names preceded by the word "Fuzz" like Fuzz-C101, Fuzz-R101, Fuzz-RC101, etc

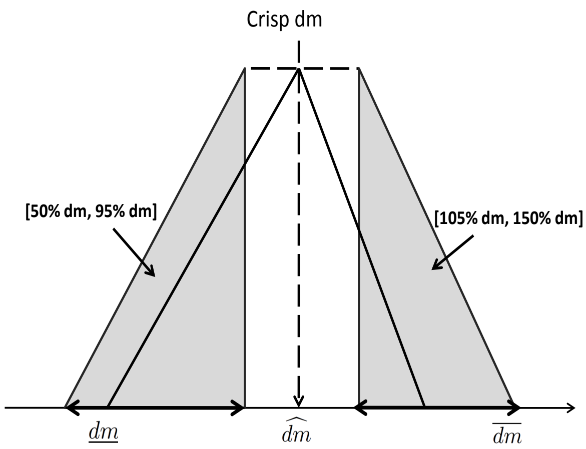

Figure 4.3 – Fuzzy sampled demand

4.5.2 Experimental protocol

To assess the performance of both E-SPEA2 and E-NSGAII algorithms, we have conducted a set of experiments that can be divided into 2 tests:

(i) First, we aim to examine the ability of our extended algorithms to tolerate fuzziness versus their crisp versions.

(ii) Second, we aim to compare the quality of generated front approximations of both proposed algorithms.

In these experiments, we have used the 56 fuzzy instances sampled uniformly at random from the classical Solomon's instances. Each fuzzy instances is tested on the following four algorithms executed 30 times: C-SPEA2 and C-NSGAII are respectively the crisp SPEA2 and NSGAII algorithms considering only the core (i.e. the most plausible value) of fuzzy demands, E-SPEA2 and E-NSGAII are the extended algorithms considering the triangular fuzzy representation of demands (i.e. the triplet of values) and propagating fuzziness to the objective functions. For each algorithm, a set of 30 runs per instance was performed with random initial populations of size=100 evolving across 1000 generations; crossover rate of 0.8 and mutation rate of 0.1. Thereafter, with 4 algorithms tested on 56 instances and repeated 30 runs, we have done $4 \times 56 \times 30 = 6720$ runs. Hence, we obtained, for every test instance, 30 sets of optimal solutions that represent Pareto front approximations. Each solution shows the minimum total traveled distance and total tardiness time for an efficient vehicles route. However, the solutions obtained for the C-SPEA2 and C-NSGAII algorithms are represented by a set of exact numbers, while for the E-SPEA2 and E-NSGAII algorithms the solutions returned are vectors of triangular fuzzy numbers. Examples of front approximations found for the instance Fuzz-C101 using each of algorithms E-SPEA2 and E-NSGAII are shown respectively in Figures 4.4 and 4.5. The illustrated fronts represent a set of triangular fuzzy solutions. For instance, the bold triangle in Figure 4.4 represents a solution with minimum total distance (the green side) equal to $[2413, 2515, 2623]$ and total tardiness time (the red side) equal to $[284312, 295280, 315322]$.

In order to evaluate and compare the quality of the generated front approximations for every test instance, we use the following multi-objective quality indicators (detailed in Chapter 1):

- *Unary hypervolume indicator* I_H to measure the approximations quality of each of C-SPEA2, C-NSGAII, E-SPEA2 and E-NSGAII algorithms. This indicator is to be maximized.

- *Binary hypervolume difference* I_H^- *and additive* ϵ*-indicator* $I_{\epsilon+}$ to compare the performance of implemented algorithms. Both indicators are to be minimized.

Yet, as these indicators are performed only on exact approximation samples, we propose to defuzzify the triangular fuzzy solutions of E-SPEA2 and E-NSGAII algorithms by computing their expected values (see Equation 4.3). As the computation of hypervolume and epsilon indicators usually require a reference set $Z_n^\star$ (or a reference point z_{ref} for the unary case), we propose to follow the experimental steps given in (Knowles et al., 2006): We first consider $Z_n^\star$ as the set of non-dominated solutions extracted from the union of all front approximations. Then, we compute $z^{\max} = (z_1^{\max}, z_2^{\max})$, where $z_1^{\max}$ and $z_2^{\max}$ denote the upper bounds of both objective functions in the whole non-dominated fronts. The reference point used for the unary hypervolume I_H is fixed to $z_{ref} = (1.05 \times z_1^{\max}, 1.05 \times z_2^{\max})$.

Subsequently, by applying the quality indicators with respect to the reference set (or reference point), we transform our solutions to samples with I-values scalars. In this way, we reach a single I-value for each test run per algorithm. For instance, according to the maximum hypervolume value found with the unary hypervolume metric, we can evaluate the quality of every algorithm outputs (i.e. higher I_H value indicates a better front approximation). Thereby, by analyzing the change in the set of I_H values provided for 30 runs per test instance, we can realize whether the extended algorithm (E-SPEA2 and E-NSGAII) are capable to tolerate the fuzziness comparing with their crisp versions (C-SPEA2 and C-NSGAII).

Afterwards, the binary I_H^- and $I_{\epsilon+}$ indicators are applied to compare the performance for each pair of algorithms (w.r.t the reference set). Similarly, since 30 runs per algorithm have been performed, we obtain 30 hypervolume differences and 30 epsilon measures for each test instance. Once all these I-values are computed, we need to use a statistical analysis in order to compare them and so obtain valid statements about their quality. Since both algorithms share the same parameters for all the runs such as the initial population, the random seed, the variation operators, etc, the resulted approximations can be considered as *matched samples*. To this end, we use the Wilcoxon-signed rank test with a P-value=0.5% for a pairwise comparison between them. Consequently, for every test instance and according to the indicator under consideration (i.e. I_H^- or $I_{\epsilon+}$), this statistical test indicates if the approximation samples obtained by a given algorithm are significantly better than the ones of another algorithm, or if there is no significant difference between both.

Notice that, all the experimental tests have been conducted using the PISA [1] performance assessment tool suite (Bleuler et al., 2003).

4.5.3 Computational results

Figures 4.6 and 4.7 summarize the unary hypervolume results for the four implemented algorithms by using box-plots, such that each box presents 30 hypervolume values from the 30 runs of a corresponding algorithm on one test instance. Then as we have a total of 56 tested fuzzy instances, we present in these figures only the box-plots for 12 sampled fuzzy instances such as: *Fuzz-C101*, *Fuzz-C104*, *Fuzz-C201*, etc. The box-plots for other instances are completely similar and exhibit the same trend.

More precisely, by examining the behaviour of plotted boxes between the accumulated hypervolume values, we can assert the quality of our extended algorithms E-SPEA2 and E-NSGAII and compare them with the crisp algorithms C-SPEA2 and C-NSGAII. Taking a look at the Figure 4.6, we can intuitively compare the C-SPEA2 and E-SPEA2 algorithms based on their boxes of hypervolume values. In fact, it is not difficult to realize that the boxes of crisp C-SPEA2 are very large, in the sense that their hypervolume values vary from instance to instance (approximately from 0.3 to 1.1). Also, we can remark that the E-SPEA2 boxes are better (higher) than those of C-SPEA2, because they are less variable (i.e. vary slightly from 0.6 to 1.2) and look identical for all the illustrated instances.

In the same way, a comparison between the C-NSGAII and E-NSGAII algorithms is shown in Figure 4.7. Indeed, from the illustrated box-plots, we can easily observe that the boxes of the crisp algorithm are larger than those of the extended one. Otherwise, the E-SPEA2 has high and less dispersed hypervolume values varying from 0.5 to 1.2, whereas the C-NSGAII has hypervolume values varying from 0.2 to 1.1. These remarks leads us to conclude that for all the sampled fuzzy instances, E-SPEA2 and E-NSGAII are less sensitive to fuzziness and so converge better than C-SPEA2 and C-NSGAII.

1. http://www.tik.ee.ethz.ch/pisa/assessment.html

This may be explained by the fact that taking into account all the three vertices of a triangular fuzzy demand instead of only consider the most plausible demand in the crisp algorithms, provide more accurate approximations and consequently a better estimate of the algorithm quality.

Table 4.2 and 4.3 present the performance comparison of four algorithms, namely C-SPEA2, C-NSGAII, E-SPEA2 and E-NSGAII, in solving the MO-VRPTW-FD. These algorithms are compared on 18 random selected fuzzy instances with respect to both binary I_H^- and $I_{\epsilon+}$ indicators and using the Wilcoxon-signed rank test (with a P-value less or equal to 0.05). For each test instance, either the algorithm located at a specific row significantly dominates the algorithm located at a specific column ($\prec$), either it is significantly dominated ($\succ$) or there is no significant difference between both ($\equiv$). Observing the results based on both indicators, E-SPEA2 statistically outperforms the three other algorithms with respect to both I_H^- and $I_{\epsilon+}$, excepting for the *Fuzz-RC204*, *Fuzz-R110* and *Fuzz-RC208* instances where there is no significant difference between both algorithms E-SPEA2 and E-NSGAII algorithms. Another exception is for the *Fuzz-C103* and *Fuzz-RC106* instances, where the E-NSGAII obtained better results than E-SPEA2 with respect to the $I_{\epsilon+}$ indicator.

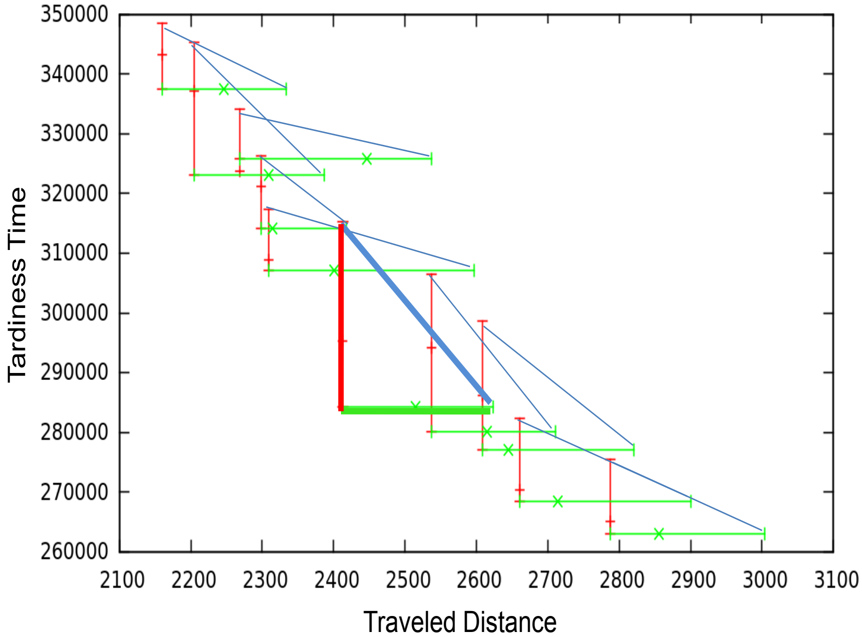

Figure 4.4 – Example of E-SPEA2 solutions for Fuzz-C101

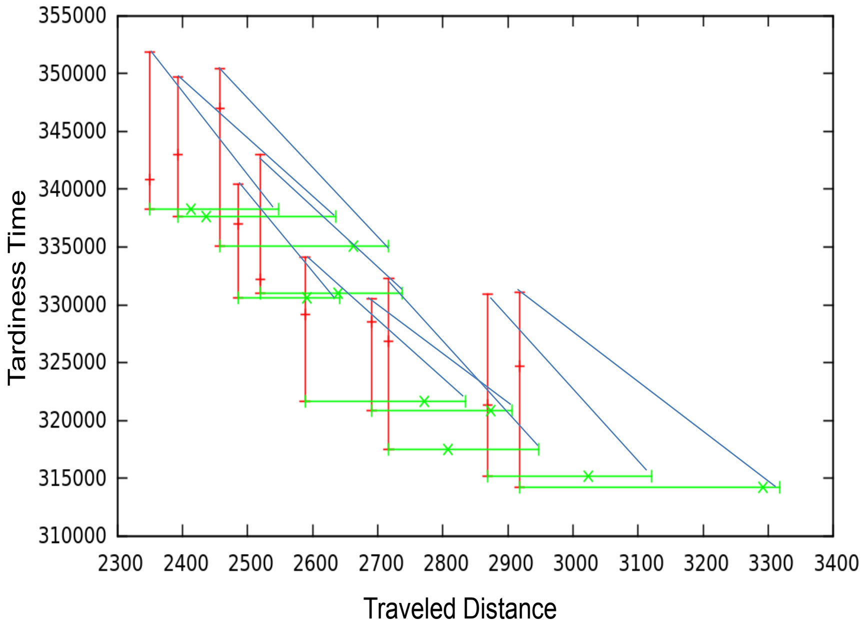

Figure 4.5 – Example of E-NSGAII solutions for Fuzz-C101

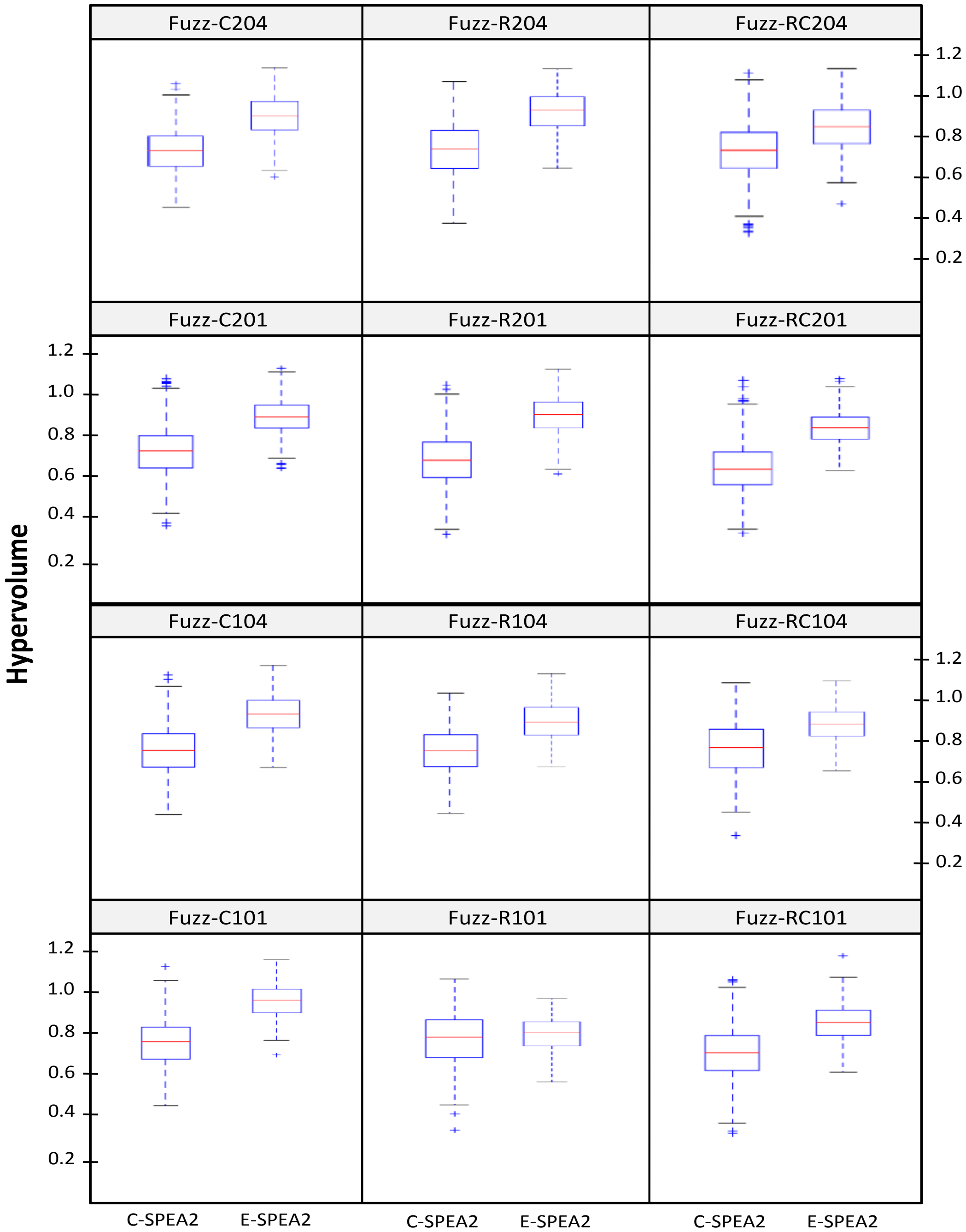

Figure 4.6 – Hypervolume results of C-SPEA2 and E-SPEA2

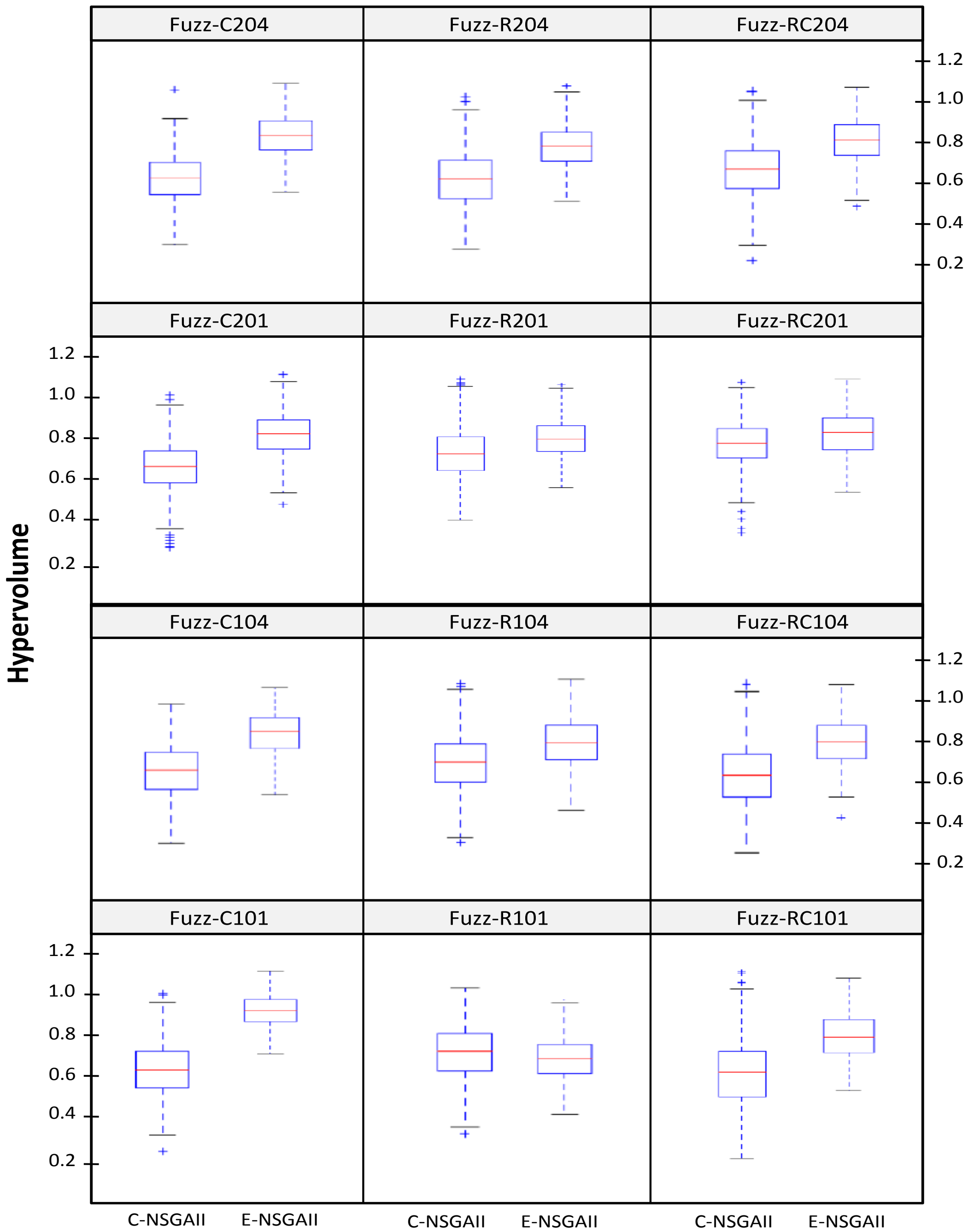

Figure 4.7 – Hypervolume results of C-NSGAII and E-NSGAII

Table 4.2 – Algorithms comparison according to the I_H^- indicator

Instances	Algorithms	C-SPEA2	E-SPEA2	C-NSGAII	E-NSGAII
Fuzz-C101	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-C201	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R101	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R201	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-
Fuzz-RC101	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-RC201	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-
Fuzz-C104	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-C204	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R104	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R204	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-
Fuzz-RC104	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-RC204	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-
Fuzz-C103	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-C207	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R110	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-
Fuzz-R208	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-RC106	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-RC208	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-

Table 4.3 – Algorithms comparison according to the $I_{\epsilon+}$ indicator

Instances	Algorithms	C-SPEA2	E-SPEA2	C-NSGAII	E-NSGAII
Fuzz-C101	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-C201	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R101	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R201	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-RC101	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-RC201	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-C104	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-C204	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\prec$	$\prec$	-
Fuzz-R104	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R204	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\prec$	$\prec$	-
Fuzz-RC104	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-RC204	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-
Fuzz-C103	E-SPEA2	$\prec$	-	$\prec$	$\succ$
	E-NSGAII	$\prec$	$\prec$	$\prec$	-
Fuzz-C207	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-R110	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-
Fuzz-R208	E-SPEA2	$\prec$	-	$\prec$	$\prec$
	E-NSGAII	$\prec$	$\succ$	$\prec$	-
Fuzz-RC106	E-SPEA2	$\prec$	-	$\prec$	$\succ$
	E-NSGAII	$\prec$	$\prec$	$\prec$	-
Fuzz-RC208	E-SPEA2	$\prec$	-	$\prec$	$\equiv$
	E-NSGAII	$\prec$	$\equiv$	$\prec$	-

4.6 Conclusion

This chapter was devoted to present our algorithmic contribution, especially in the field of multi-objective evolutionary computation. In fact, an extension of the two most popular algorithms, namely SPEA2 and NSGAII, was proposed in order to enable them handling any MOP with fuzzy-valued objectives. The usefulness of extended algorithms was illustrated through the resolution of a practical VRP problem and their performance assessment was validated by means of some experimental tests. The computational results were straightforward and encouraging for multi-objective problems confronted with fuzziness. Yet, the robustness of obtained results cannot be evaluated using standard performance indicators. So in the next chapter, we will introduce a new robustness approach for validating our proposals.

Chapter 5

Fuzzy Multi-objective Robustness-based Approach

Contents

Main publications related to this chapter

Bahri, O., Ben Amor, N., and Talbi, E.-G.: B-Robustness Approach for Fuzzy Multi-objective Problems: in the 15th International Conference on IPMU'16, Springer, Eindhoven, the Netherlands, June 2016.

Bahri, O., Ben Amor, N., and Talbi, E.-G.: Robust Routes for the Fuzzy Multi-objective Vehicle Routing Problem. in the 8th IFAC Conference on MIM'16, Troyes, France, June 2016.

Bahri, O., Ben Amor, N., and Talbi, E.-G.: Robustness Approach for Multi-objective Optimization under Uncertainty, 11th Metaheuristics International Conference (MIC'15), Agadir, Morocco, June 2015.

5.1 Introduction

Facing increased competition in designing uncertain optimization methods, robustness aspect becomes necessary for sensitivity analysis and performance validation. This aspect can be defined as the ability of a resolution method to remain unaffected despite potential perturbations due to uncertain parameters. Various concepts and techniques associated with robustness have been more and more discussed in the literature during the last few years. In most of existing approaches (Aissi et al., 2009 ; Kasperski & Kulej, 2009), robustness is evaluated based on the most pessimistic (i.e. worst case) scenario. Some other approaches such as the β-robustness approach (Palacios et al., 2014) consider that the pessimistic way of performance analysis is limited in the sense that it may be extremely conservative when the worst case is not crucial and so an overall acceptable performance is preferred. Unfortunately, almost all these approaches are restricted to a single-objective context and often fail to consider the robustness requirement for real-world applications. Otherwise, there are very few approaches regarding the robustness in multi-objective setting (Deb & Gupta, 2006 ; Bader & Zitzler, 2010), where the goal becomes to achieve a set of solutions that are not only optimal but also safe, reliable and robust. The purpose of this chapter is to contribute to the search of robust optimal solutions in fuzzy multi-objective optimization. This is achieved through three main phases: the generalization of standard β-robustness concepts to the fuzzy multi-objective context, the extension of our fuzzy Pareto dominance for integrating robustness and the refinements of our previously proposed algorithms in order to enable them converging towards robust solutions.

The chapter is structured as follows. Section 5.2 explains our motivation behind this contribution and gives some definitions related to the robustness aspect. Section 5.3 provides a survey and discussion of the most prominent robustness approaches. Section 5.4 presents our novel approach and finally, Section 5.5 describes the experimental study in the case of the MO-VRP-FD problem.

5.2 Motivation

Most of studies in the field of non-deterministic optimization tackle the issue of designing efficient methods able to cope with uncertainty propagation. These methods are of great importance as they consider the not-perfect reality of practical applications. Their performance is usually examined in terms of solution costs or optimality, but their robustness has long been neglected by researchers. Remember that in the previous chapters, our main task was to find optimal solutions for multi-objective problems that are subject to fuzziness. To this end, fuzzy optimization algorithms were developed to explicitly consider this fuzziness within the search process. Afterwards, in order to evaluate the generated fuzzy solutions, we have firstly defuzzified them into crisp solutions and then simply applied the classical multi-objective quality indicators. Yet, these latter are often not sufficient to analyse unexpected variations in the obtained solution(s) or to detect the potentially responsible factor. Although the classical indicators should not be used to evaluate such problems, it should also not defuzzify the generated solutions since this can lead to lose a part of information and consequently to decrease their quality. Otherwise, the decision maker is not able to judge the robustness of defuzzified or approximated outputs. Therefore, the assessment of robustness is required in our study to conduct a complete sensitivity analysis and draw further conclusions about our methods.

Robustness plays a central role in the field of optimization and decision making under uncertainty. In its general form, this notion refers to the ability to favor flexibility in any unexpected situation or unpredictable changes in problem data, components or environments. It is intuitively connected to the idea that in presence of potential uncertainties, the decision making process should remain steady (i.e. with the minimal damage or loss of efficiency). Several interpretations of robustness are made in theoretical and algorithmic contexts such as a robust decision, robust solution or robust algorithm. In addition, two cases of robustness evaluation can be distinguished depending on the moment and way in which it is performed (Sevaux & Sörensen, 2004):

- At the end of development phase: It may be interpreted as a sensitivity test or conclusion of method validation. Then, if the method is concluded not to be robust, further efforts should be made for re-optimizing or redeveloping it.
- During the optimization phase: It can be an analytical procedure performed during the optimization phase or development of a method. In this case the robustness of solution(s) is guaranteed, in the sense that remain optimal or feasible under all variations.

For our context, robustness refers to the second case, where it is a part of the optimization method intended to be robust. This combination of optimization and robustness leads to the so-called *robust optimization* (Ben-Tal et al., 2009). In general, robust optimization may be defined as an extension of classical optimization methods in which a robustness procedure is incorporated. It can also be defined as an optimal design proved relatively insensitive to any perturbations of inputs data and where the most robust solution is sought.

In multi-objective setting, the goal of robust optimization becomes to find a set of solutions that are optimal, flexible and reliable against uncertainty (Soares, Adriano, et al., 2009). Afterwards, most formulations of robustness in multi-objective context cover the following situations:

- modification of the objective values or constraints to account for robustness;
- definition of robustness as an objective function to be maximized;
- use of additional robustness constraints to restrict the search process.

The next section gives an overview of the state of the art relative to robustness-based optimization approaches.

5.3 Robust optimization approaches

The approaches to cope with robustness in the literature are multiple and varied, notably those related to the issue of finding robust solution(s) for practical optimization problems. The first study was published by (Soyster, 1973) which assumes that robustness is evaluated according to the most pessimistic scenario. Afterwards, (Kouvelis & Yu, 1997) suggested a robust deviation approach that uses the *notion of regret*[1]. Moreover, in (Aissi et al., 2009 ; Kasperski & Kulej, 2009), a *minimax criteria* is used to seek robust solutions having the best possible performance in the worst case (by minimizing the maximum possible regrets of solutions).

The pessimistic way of analysis in these classical approaches leads often to a rapid and systematic convergence toward robust solutions. Yet, it may be deemed as extremely conservative when the worst case is not crucial, in the sense that it is not possible to configure the amount of targeted robustness against uncertainty (known as level of conservatism). In (Bertsimas &

1. defined as the difference between the best minimum cost and the second minimum for every problem solution

Sim, 2004), the authors proposed the use of a *conservativeness degree* when looking for robust solutions. This degree, specified by the decision maker, imposes that the total amount of perturbations are less than it. Thereafter, (Bertsimas & Nohadani, 2010) defined a *target robustness* according to the minimal cost of the worst outcomes and then used it to adjust the conservativeness of his optimization method.

Another important and thoroughly studied concept is the subjectivity of robustness. In fact, robustness is strongly subjective and so it is essential to explicitly define the tradeoff between satisfactory performance and regrettable impacts. Therefore, several researches such as (Daniels & Carrillo, 1997 ; Maeda & Kawachi, 2001 ; Palacios et al., 2014) proposed to take into account the subjective aspect of robustness through a target level specified by the decision maker. Their main idea is that rather than gambling on the possible performance, it may be more efficient to make a level on the overall acceptable performance and then to state the confidence in being able to achieve that level. Among the frequently used approaches for this kind of subjective robustness, we focus in this thesis on the β-robustness approach (Daniels & Carrillo, 1997) which consists in maximising the likelihood that a solutions's actual performance is not worse than a given threshold. Most of these approaches can only deal with specific single-objective problems for which they have been developed.

Only some of them were recently extended to the multi-objective setting such as the minimax robustness approach (Ehrgott et al., 2014) and the concepts of robustness degrees (Barrico & Antunes, 2006). For example, in (Soares, Guimarães, et al., 2009 ; Rivaz & Yaghoobi, 2013), a minimax-based robust procedure is employed for checking Pareto optimal solutions in the context of interval-valued objectives. On the other hand, a number of approaches are designed to cope with robustness in multi-objective optimization. As mentioned before, the robustness evaluation is often incorporated in the optimization method by modifying the objective values and/or constraints. For instance, in (Deb & Gupta, 2005 ; Gunawan & Azarm, 2005, 2005), the multi-objective search process is restricted with respect to predefined constraints of robustness. In (Jin & Sendhoff, 2003 ; M. Li et al., 2005), the authors tried to design robust multi-objective evolutionary algorithms by treating robustness as an additional objective. In (Deb & Gupta, 2006), the authors considered the robustness in both constraints and objectives by fixing a predefined limit of variation and using the mean objective values. Thereby, the incorporation of robustness as additional constraints and/or objectives is also suggested in (Bader & Zitzler, 2010) through an hypervolume-based evolutionary algorithm. Notice that, the main advantage of using evolutionary algorithms in most of these approaches is that they provide a trade-off between robustness and optimality in a computational run-time.

Unlike the existing approaches, the approach we propose is a combination of β-robustness concepts, multi-objective evolutionary optimization and fuzzy aspect. To the best of our knowledge, there was no similar work done in the robust optimization field that brings these aspects together. Note that, in the light of our previous achievement, the following issues concerning robustness arise:

(i) *How defining robustness in a fuzzy multi-objective context?*

(ii) *How integrating robustness into the previously proposed algorithms?*

(iii) *How checking if the obtained solutions are sufficiently robust?*

5.4 Robustness approach for fuzzy MOPs

In this section, we present a new generic approach able to determine robust optimal solutions for any multi-objective problem with fuzzy-valued objectives. The proposed approach is comprised of three main steps:

1. defining new β-robustness concepts in a fuzzy multi-objective context;
2. extending our fuzzy Pareto dominance to consider robustness criteria;
3. refining our previous algorithms to seek robust optimal solutions.

Each of these steps is described in detail in the following sub-sections.

5.4.1 New β-robustness concepts

Our proposal retains the basic idea of $\beta-$robustness approach (Daniels & Carrillo, 1997) and tries to generalize it for dealing with fuzzy multi-objective optimization. In fact, β-robustness consists in maximizing the confidence that a solution's actual quality is not worse than the overall acceptable level of performance. This very general definition was often reformulated to emphasize single-objective problems in specific areas such transportation and scheduling (Maeda & Kawachi, 2001 ; Palacios et al., 2014 ; Pishevar & Tavakkoi-Moghaddam, 2014). Besides, it includes two important components which are the quality (or cost) of solution related to the objective values and the performance level (or threshold) usually provided by an expert opinion.

In the following, we present our proposal to refine the usual definitions and concepts of β-robustness to our fuzzy multi-objective context. In particular, our aim is to achieve a set of β-robust solutions with respect to the fuzziness of multiple objectives. For the sake of simplicity and without loss in generality, we suppose that we have any MOP with two fuzzy-valued objectives to be minimized. Then, β-robust solutions will be redefined as follows:

Definition 5.1. *Let $f_1(x)$ and $f_2(x)$ be two objective functions expressed with triangular fuzzy numbers, then: β-robust solution is a solution with certain confidence that the cost in terms of both objective values f_1 and f_2 will be less than a given threshold level.*

A question then arises; *how choosing a confidence threshold in such multi-objective context ?* In real-life problems, this threshold is often provided by expert(s). Yet as in our case we are treating only synthetic problems, such threshold is not always available. Thereby, knowing that the objectives in a MOP are usually considered to be independent from each other (i.e., they depend only on the decision variable), we suggest to define a threshold to each objective as follows:

Definition 5.2. *Let $A = [\underline{a}, \widehat{a}, \overline{a}]$ and $B = [\underline{b}, \widehat{b}, \overline{b}]$ be the best known objective values for $f_1(x)$ and $f_2(x)$ respectively and let f_1^* and f_2^* be their predefined thresholds, then the confidence threshold or level for each of objectives is estimated separately by:*

$$f_1^* = \widehat{a} + TF \times (\overline{a} - \widehat{a}), \quad f_2^* = \widehat{b} + TF \times (\overline{b} - \widehat{b}) \tag{5.1}$$

where TF is a given tightness factor of best possible performance.

A parametric analysis is performed to select TF which consists of examining the variability and closeness of triangular fuzzy values to the peaked ones (i.e. the most plausible values). An approximation is said "tight" within factor $TF < 1$ if the level of conservatism against fuzziness is computationally safe with respect to the crisp situation.

Then our goal is to maximise the confidence that the objective costs f_1 and f_2 of each solution will certainly or necessarily be less than their fixed thresholds f_1^* and f_2^*, respectively. In other words, we intend to maximize the necessity that each objective value lies within its confidence level. This β-robustness reasoning can be formulated in the possibilistic setting (Dubois & Prade, 1998) as the necessity measure (Equations 2.12 and 2.13 in Chapter 2). In fact, the necessity measure N corresponds to the certainty degree of any subset of knowledge and gives a pessimistic view in decision making. Hence, we propose to use this measure for estimating the necessary robustness. More precisely, we propose to compute the β-robust degrees, denoted β_N^i, which are equivalent to the necessity measure that every objective value will be lower than its given threshold. Notice that, these degrees must be computed with respect to the triangular fuzzy shape of different objectives. Formally, the necessity degree N that a triangular objective value $x = [\underline{x}, \widehat{x}, \overline{x}]$ is less than a real number r is given by:

$$N(x \leq r) = \begin{cases} 1 & \text{if } \overline{x} < r \\ \frac{r-\widehat{x}}{\overline{x}-\widehat{x}} & \text{if } \widehat{x} \leq r \leq \overline{x} \\ 0 & \text{if } \widehat{x} > r. \end{cases} \tag{5.2}$$

Then suppose we have computed the β_N degrees for each solution w.r.t both objectives f_1 and f_2, i.e. $\beta_N^1 = N(f_1 \leq f_1^*)$ and $\beta_N^2 = N(f_2 \leq f_2^*)$, we need to check if a solution has a good necessary robustness based on its β_N values. Remember that an important property of necessity measure is the 'minitivity' w.r.t. conjunction $N(A \cap B) = \min(N(A), N(B))$. Thus, we propose to aggregate the set of β_N degrees of each solution as follows:

$$(\beta_N^1 \cap \beta_N^2) = \min\ (\beta_N^1, \beta_N^2) \tag{5.3}$$

This aggregation allows us to make a decision based on the most pessimistic value given by the minimum necessary robustness. Thereafter, to avoid achieving a solution with low β_N values, we propose to enhance them with an interval of desired robustness level $[R, 1]$ where R expresses the decision maker's attitude. Therefore, we define the necessary robustness as follows:

Definition 5.3. *A solution s with both objective values f_1 and f_2 is said to be necessarily β_N-robust w.r.t. thresholds f_1^* and f_2^* respectively, iff:*

$$\beta_N^1 = N(f_1 \leq f_1^*) \wedge \beta_N^2 = N(f_2 \leq f_2^*) \wedge \beta_N(s) = \min\ (\beta_N^1, \beta_N^2) \in [R, 1] \tag{5.4}$$

Example 5.1. *Let us consider an example of β_N-robustness evaluation with a set of 4 solutions $\{s_1, s_2, s_3, s_4\}$ and a confidence parameter $R = 0.5$ to* *suppose that the attitude of decision maker is neither pessimistic nor optimistic. Two degrees* (β_N^1, β_N^2) *are assigned to each solution as follows:* $s_1 : (0.30, 0.60)$, $s_2 : (0.55, 0.80)$, $s_3 : (0.95, 0.15)$ *and* $s_4 : (0.75, 1.00)$. *Then in order to find the necessarily robust solutions, we need to check for every one if the minimum of its both* β_N *degrees is within the interval* $[0.5, 1]$. *Thus, we clearly remark that solutions* s_1 *and* s_3 *are not sufficiently* β_N*-robust because* $\beta_N(s_1) = \min(0.3, 0.6) = 0.3 \notin [0.5, 1]$ *and* $\beta_N(s3) = 0.15 \notin [0.5, 1]$. *On the other hand, solutions* s_2 *and* s_4 *are judged as necessary robust since they reach the desired level of robustness (i.e. the minimum value of their* β_N *degrees is higher than* $R = 0.5$*).*

Subsequently, as the necessity and possibility are inter-definable (i.e. in the sense that if $N \neq 0$, then $\Pi = 1$), we may conclude that by maximising the necessary robustness of any solution, we are also maximising its possible robustness denoted $\beta_\Pi(s)$. Obviously, we have:

Definition 5.4. *A solution* s *is said to be possibly* β*-robust, i.e.* $\beta_\Pi(s) = 1$, *if its necessary robustness* $\beta_N(s) \in [R, 1]$ *is satisfied.*

Finally, the β_N and β_Π can be seen as lower and upper bounds of the degree that s is β-robust and so we may deduce that:

Definition 5.5. *If* s *is necessarily and possibly* β*-robust w.r.t. the predefined thresholds, then we have:*

$$\forall i = \{1, 2\}, \beta_N^i(s) \leq \beta^i(s) \leq \beta_\Pi^i(s) \tag{5.5}$$

It is important to note that the proposed β-robustness concepts can be used not only to evaluate the optimal solutions in case of fuzzy multi-objective optimization, but also solutions that are not optimal. In other words, a not optimal solution can be robust if its β_N degrees are within the desired level of robustness. Yet, the major difficulty of multi-objective optimization lies in finding robust optimal solutions, especially the robust Pareto set. In the following, we propose to combine both aspects of robustness and Pareto optimality.

5.4.2 Robust Pareto optimality

We shall begin here by defining the aspect of robust Pareto optimality and its related properties. In general, Pareto dominance is devoted to rank solutions according to their quality (objective values), but it cannot discriminate between the robust ones. Thus, a major question is, *whether to deploy robustness concepts for achieving a robust Pareto optimal set ?*

Usually, a Pareto optimal solution is called robust if it remains optimal against any uncertainty and there is no better solution dominated it. In our case, a solution is robustly Pareto optimal if and only if it is feasible, non-dominated and the sensitivity of its objective values to fuzziness is minimal. Intuitively, we may conclude that our previously proposed Pareto dominance relations are not enough to evaluate the robustness of fuzzy solutions. However, as we have proposed above new concepts to analyse their robustness, the issue now is how to combine them with the Pareto optimality aspect. At this stage, we have to check that for any Pareto optimal solution there is no negative impacts on its feasibility, optimality and reliability in the face of fuzziness. The idea is so to consider the robustness as an additional constraint to improve the Pareto ranking. More precisely, we suggest to extend our fuzzy Pareto dominance for integrating the β-robustness concepts. Remember that we have three relations detailed in Chapter 3: *Strong Pareto dominance* ($\prec_{SP}$), *Weak Pareto dominance* ($\prec_{WP}$) and *Case of indifference* ($\|$). Their extension leads to the following new definitions:

Definition 5.6. *Robust strong Pareto dominance*
Let Y and Y' be two triangular fuzzy solutions, $\beta_N(Y)$ and $\beta_N(Y')$ be their necessary robustness degrees. Y strongly and robustly Pareto dominates Y', denoted by $Y \prec_{Srob} Y'$ iff:

$$Y \prec_{SP} Y' \;\wedge\; \beta_N(Y) \geq \beta_N(Y') \tag{5.6}$$

This means that if a solution Y is preferred over Y' according to the dominance relation $\prec_{SP}$, and its necessary robustness degree is higher than or at least equal to that of Y', then $Y \prec_{Srob} Y'$ must hold.

Definition 5.7. *Robust weak Pareto dominance*
Let Y and Y' be two triangular fuzzy solutions, $\beta_N(Y)$ and $\beta_N(Y')$ be their necessary robustness degrees. Y weakly and robustly Pareto dominates Y', denoted by $Y \prec_{Wrob} Y'$ iff:

$$[Y \prec_{WP} Y' \;\wedge\; \beta_N(Y) \geq \beta_N(Y')] \;\vee\; [Y \| Y' \;\wedge\; \beta_N(Y) > \beta_N(Y')] \tag{5.7}$$

In this case, we propose to consider in addition to the relation $\prec_{WP}$, the case of indifference between two solutions. Indeed, if the incomparable solutions can be discriminated by their β_N degrees, then there exists a relation of $\prec_{Wrob}$. On the other hand, the solutions remain incomparable if their β_N degrees are equal.

Definition 5.8. *Case of indifference*
Two triangular fuzzy solutions are indifferent or incomparable, denoted by $\overrightarrow{Y} \|_{rob} \overrightarrow{Y}'$, *iff:*

$$Y \| Y' \ \wedge \ \beta_N(Y) = \beta_N(Y') \tag{5.8}$$

The issue now is how to use and incorporate these robust Pareto dominance into the optimization process.

5.4.3 Algorithmic refinements

This section describes our idea to extend the optimization algorithms proposed in Chapter 4, namely E-SPEA2 and E-NSGAII, by the consideration of robustness. In particular, we shall refine some features for integrating our methodology of robustness evaluation, as well as the new robust Pareto dominance into the search process of our algorithms. The aim behind these refinements is to develop robust optimization algorithms, in the sense that they are able to find robust optimal solutions for any multi-objective problem with fuzzy-valued objectives. It should be specified that the discussion to follow focuses only on the necessary robustness β_N since the possible robustness β_Π can always be deduced from the dual relationship.

First, we suggest to consider the new $\beta-$robustness concepts at the evaluation step of both algorithms (see Algorithms 4.1 and 4.2). Conventionally, this step requires the use of fitness measures to evaluate the solutions at every generation or iteration. At this level, we propose to replace the fitness value assigned to each of solutions by its degree of necessary robustness, specifically with the minimum of its β_N degrees (Equation 5.3). This helps to improve the fitness-based process and guide the evaluation of solutions according to their robustness. In general, at the first iteration, a set of solutions are randomly generated to initialize a population and then evolved until reaching a stopping criterion. Yet, these initial solutions have usually a poorer quality, with relatively high objective values. Thus, they probably will yield very bad β_N-based fitness values, in the sense that their corresponding robustness degrees are closer or equal to zero for any reasonable threshold. As a consequence, this will prevent the algorithm from converging more quickly to robust solutions.

Taking into account these considerations, we propose to use the methodology of "adaptive" thresholds proposed by (Palacios et al., 2014). This methodology consists in progressively providing a set of successive smaller thresholds with linearly decreasing approximations. More precisely, in order to avoid returning zero-β_N values, we suggest to begin the evaluation of our initial population with two first thresholds f_1^0 and f_2^0 taken as most pessimistic of the best objective values. Subsequently, at each generation, the populations and thresholds are evolved progressively with more demanding values until reaching the best or optimal ones f_1^* and f_2^*. These latter remain unchanged in the last generations and are thereby used to re-evaluate the solutions according to their necessary β_N robustness.

Afterwards, in the next step, a Pareto-based fitness-assignment strategy is performed to evaluate the solutions in terms of convergence. It might be possible here to simply use our Pareto dominance relations proposed in Chapter 3 for ranking the fuzzy solutions and then to apply the robustness evaluation methodology on the solutions found. This means that robustness is applied only on the set of non-dominated solutions. In such context, solutions which are incomparable, equivalent or closest to the Pareto-optimal front are not considered. Our aim, however, is to enable the optimization algorithms achieving $\beta-$robust optimal solutions. To this end, the new robust dominance relations are then integrated into the search process of each algorithm, especially into the fitness-assignment strategy. The major advantage of these relations is that it can discriminate between some cases of indifference by using the robustness criteria. The environmental selection, variation and replacement steps of each algorithm remain unchanged. Finally, we denote by R-SPEA2 and R-NSGAII the robust version of E-SPEA2 and E-NSGAII algorithms.

5.5 Experiments in case of MO-VRPTW-FD

To illustrate our robustness approach, we have applied the R-SPEA2 and R-NSGAII algorithms for solving the MO-VRPTW-FD problem. The algorithms were implemented using the ParadisEO-MOEO module (Liefooghe et al., 2007) and with the same base parameters of our previous developments.

5.5.1 Experimental robustness analysis

Our aim of solving the MO-VRPTW-FD problem becomes here to find *robust optimal routes* for a fleet of identical vehicles which serve a set of customers within limited time windows and whose demands are assumed to be triangular fuzzy numbers. Then as the sequence of customers and service times depend primely on the amount of demands to be delivered, our both objectives of minimizing the *total traveled distance* and *total tardiness time* are clearly affected by the fuzziness of demands and so obtained as $D = [\underline{D}, \widehat{D}, \overline{D}]$ and $T = [\underline{T}, \widehat{T}, \overline{T}]$, respectively.

By following the steps of our robustness approach, the $\beta-$robust routes are interpreted as those having a certain confidence that the cost in terms of traveled distance D and tardiness time T will be less than given thresholds D^* and T^*, respectively. Next, the objective thresholds are computed based on the best found solutions for D and T (Equation 5.1) and a given tightness factor TF. The parametric analysis used to determine TF consists on:

- considering the best routes achieved using exact demands (i.e. the middle values) as estimate of the crisp situations;
- examining the variation of routes induced by particular triangular fuzzy demands comparing with the crisp ones;
- quantifying the degree of conservatism that less perturbed routes have as the tightness factor.

In our case, the most appropriate TF value for best possible performance is estimated at 0.75. Notice that, the thresholds are updated along a number of generations from relatively pessimistic values $D^0 > D^1 > D^2 \dots$ until optimal ones D^* and T^* that remain invariable in the last generations. Once the thresholds are fixed, the necessary robustness of each routes is conceptualized in two β_N degrees: $\beta_N^1 = N(D \leq D^*)$, $\beta_N^2 = N(T \leq T^*)$. Remember that, the route is called necessarily β-robust if it reaches the desired level of robustness (i.e. min $(\beta_N^1, \beta_N^2) \in [R, 1]$). At this level, we supposed that a sufficient robustness should be at least equal to $R = 0.4$. This conservative choice (neither extremely optimistic nor pessimistic) allows us to keep a large number of possible robust solutions.

The experiments were conducted by applying the proposed robust algorithms R-SPEA2 and R-NSGAII on the 56 sampled fuzzy Solomon's instances. We have also used the same algorithmic parameters namely, the initial population of size=100, crossover rate=0.8, mutation rate=0.1 and maximum number of generations=1000 from which the last 100 use the D^* and T^* values. Both algorithms have been executed 30 times on each of 56 test instances, thus we have $2 \times 56 \times 30 = 3360$ runs.

For empirical assessment, we have adapted the method given in (Palacios et al., 2014) to our problem. The principle of this method is to assess the "real" robustness of solutions using a Monte-Carlo simulation based on fuzzy semantics. In fact, suppose we have solved the MO-VRPTW-FD, the fuzzy routes found are often considered as *a-priori solutions* of the problem. However, in practical use of these routes, it is impossible to predict the exact total traveled distance or tardiness time since they depend on the demands which are not known yet. In other words, the "real" routes with the exact quantity of demands are only determined upon arrival at the customer's locations. Moreover, each fuzzy route corresponds to an ordering of customers that can be used to estimate the *possible a-posteriori realizations* of the problem.

In this setting, the behaviour of customers sequence found on a given fuzzy instance were evaluated on a set of deterministic samples, representing the K a-posteriori solutions of the fuzzy test instance. More precisely, we have randomly generated for each fuzzy instance, 10 deterministic versions by simulating exact demands according to probability distributions coherent with the triangular fuzzy demands. Thereafter, the found ordering of customers were used with the simulated exact demands to process our problem objectives. This allowed us to obtain precise routes with "real" total traveled distance and tardiness time that may be under or above the thresholds D^* and T^*. To this end, we have considered the whole set of sampled deterministic instances to obtain the different possible values of both objectives and then computed the proportion n of those values which are below actually bellow the fixed thresholds. This gives us an empirical evaluation of the real robustness n-rob of found routes, where a good degree of β_N should correspond to a high n.

5.5.2 Results

Table 5.1 summarizes the results of our algorithms R-SPEA2 and R-NSGAII on a set of 14 instances of the fuzzy Solomon's benchmark. Each row corresponds to one of these instances tested on each of both algorithms. Notice that, the remaining instances produce fairly similar results to those shown below. In fact, the column, with header best, reports for each algorithm and instance, the best solution found across 30 runs. Every solution corresponds to the minimum traveled distance $D_m in$ and tardiness time $T_m in$ obtained respectively as triangular fuzzy numbers (the triplet of values between braces). The next column shows the thresholds of both objectives, denoted D^* and T^*, that will be used later for estimating the robustness of solutions. Additionally, the fifth and sixth columns report the expected values of the average and worst solution in 30 runs (using Equation 4.3 in Chapter 4). The last column shows the average time (in seconds) taken by each algorithm on every tested instance in a single run.

As we can see from the available results, the R-SPEA2 algorithm obtains the best-so-far solutions in all instances, even the expected values of average and worst solutions found are better than those obtained with the R-NSGAII algorithm. According to run times, both algorithms require considerably less time (i.e. approximately between 6 and 10 seconds).

Table 5.2 contains the results of robustness evaluation on the same set of fuzzy instances. It first reports, for each test instance, the necessary β_N robustness of the best solution in 30 runs, by showing the $\beta_N^1[best]$ and $\beta_N^2[best]$ degrees computed for our two objectives $D_{\min}$ and $T_{\min}$ respectively, in addition to the min value between them $\min(\beta_N^1, \beta_N^2)[best]$. The next two columns correspond each to the min β_N values obtained on that instance for the average and worst solutions. Finally, the last column presents the robustness degree of simulated proportion n.

As we can observe, the β_N degrees of best solutions are very similar and greatly exceed our desired level of robustness $R = 0.5$. These high results (equitably ≥ 0.7) may be explained by the fact that the robustness is computed w.r.t the corresponding thresholds typically estimated using the best solutions. Obviously, these latter yield good necessary robustness, such as for the *Fuzz-RC208* instance that have $\min(\beta_N^1, \beta_N^2)[best] = 0.879$. Then observing the min of β_N degrees of the average solutions, we remark that R-SPEA2 provides better results than R-NSGAII in all cases: the min β_N values for R-SPEA2 are within the interval $[0.35, 0.6]$, while those for R-NSGAII are relatively less than 0.5. Thereby, even for the worst solutions, the necessary β_N robustness is always > 0 for both algorithms and so the possible robustness β_Π is 1 for all test instances. Besides, the simulated real robustness values n-rob are always 1 or ideally close to 1 even when β_N is low. This means that the traveled-distance and tardiness-time values for all simulations are below the fixed thresholds. In that sense, we may conclude that the robustness we are looking for in our solutions is satisfiable.

Table 5.3 presents the performance assessment of our two robust algorithms compared between each other and also with the previously proposed algorithms, namely E-SPEA2 and E-NSGAII, based on both quality indicators I_H^- and $I_{\epsilon+}$. The experimental protocol is as described in Chapter 4, where either the results of the algorithm located at a specific row are significantly better $\prec$ than those of the algorithm located at a specific column, either they are worse $\succ$ or there is no significant difference between them $\equiv$. The results show that R-SPEA2 and R-NSGAII outperform the algorithms E-SPEA2 and E-NSGAII for all the instances with regard to hypervolume and epsilon indicators. This leads us to conclude that thanks to the robustness improvement, the new algorithms provide more competitive and better solutions when compared to our previous algorithms, and obviously to their crisp versions. In addition, we remark that R-NSGAII is always outperformed by R-SPEA2 for all cases. The only exception is for the instances *Fuzz-RC204* and *Fuzz-R210* where there is no difference between both algorithms.

Table 5.1 – Results on fuzzy Solomon's instances

Instances	Algorithms	Best $(D_{\min}, T_{\min})$	D^*	T^*	E[Avg]	E[Worst]	T(s)
Fuzz-C102	R-SPEA2	[2129 2413 2507] [262635 284312 294875]	**2483.50**	**292234.25**	[2482 287963]	[2576 299843]	5.56
	R-NSGAII	[2205 2389 2607] [281441 292133 306015]	2552.50	302544.50	[2497 299203]	[2672 302310]	6.80
Fuzz-R102	R-SPEA2	[2880 3156 3430] [182149 209616 210160]	**3361.50**	**210024.00**	[3198 212882]	[3318 219764]	6.63
	R-NSGAII	[3069 3337 3531] [211509 236599 238249]	3482.50	237836.50	[3389 237093]	[3512 240121]	6.92
Fuzz-RC102	R-SPEA2	[3025 3427 3562] [202639 216923 220078]	**3528.25**	**219289.25**	[3471 219410]	[3534 224121]	9.05
	R-NSGAII	[3325 3628 3784] [241511 245359 264805]	3745.00	259943.50	[3599 249985]	[3647 260012]	8.25
Fuzz-C108	R-SPEA2	[2207 2515 2841] [261220 280740 291204]	**2759.50**	**288588.00**	[2587 284867]	[2719 291631]	7.27
	R-NSGAII	[2289 2647 2897] [288421 305147 313595]	2834.50	311483.00	[2652 303797]	[2842 312340]	7.29
Fuzz-R108	R-SPEA2	[2252 2351 2447] [136902 146717 152084]	**2423.00**	**150742.25**	[2355 149655]	[2498 152416]	7.94
	R-NSGAII	[2308 2401 2597] [157788 181982 183792]	2548.00	183339.50	[2466 178643]	[2576 184921]	8.03
Fuzz-RC108	R-SPEA2	[2786 3043 3103] [169400 199616 200720]	**3088.00**	**200444.00**	[3098 198231]	[3154 212431]	6.73
	R-NSGAII	[2788 3156 3304] [182149 202546 210160]	3267.00	208256.50	[3197 200350]	[3208 219918]	6.82
Fuzz-C204	R-SPEA2	[2053 2174 2410] [246105 266436 266773]	**2351.00**	**266688.75**	[2254 267413]	[2420 276191]	7.17
	R-NSGAII	[2246 2298 2718] [274781 293399 296129]	2613.00	295446.50	[2411 298472]	[2679 305359]	7.43
Fuzz-R204	R-SPEA2	[2234 2436 2641] [96838 115068 120569]	**2589.75**	**119193.75**	[2475 118185]	[2563 120171]	7.78
	R-NSGAII	[2477 2609 2658] [114350 120872 149695]	2645.75	142489.25	[2636 127644]	[2788 143123]	8.17
Fuzz-RC204	R-SPEA2	[2070 2307 2512] [200458 221067 235948]	**2460.75**	**232227.75**	[2338 223569]	[2398 231180]	9.12
	R-NSGAII	[2125 2312 2520] [200663 221503 235558]	2468.00	232044.25	[2376 228960]	[2416 239860]	8.93
Fuzz-C207	R-SPEA2	[1972 2085 2320] [280387 302111 306841]	**2261.25**	**305658.50**	[2186 302678]	[2263 308726]	6.63
	R-NSGAII	[2079 2093 2522] [300816 304768 335222]	2414.75	327608.50	[2213 319331]	[2397 329312]	7.04
Fuzz-R207	R-SPEA2	[2478 2567 2780] [140607 143727 159027]	**2726.75**	**155202.00**	[2670 149772]	[2797 151245]	8.21
	R-NSGAII	[2477 2609 2758] [152531 161085 169531]	2720.75	167419.50	[2683 169850]	[2731 179121]	8.54
Fuzz-RC207	R-SPEA2	[2496 2963 3075] [172808 198734 196968]	**3047.00**	**197409.50**	[2899 201811]	[3047 219425]	9.97
	R-NSGAII	[2555 2986 3269] [197901 219777 224925]	3198.25	223638.00	[3040 229505]	[3194 230085]	9.19
Fuzz-R210	R-SPEA2	[2471 2648 2908] [139557 150051 165074]	**2843.00**	**161318.25**	[2698 158311]	[2787 170012]	9.15
	R-NSGAII	[2461 2648 2912] [145001 150526 167455]	2846.00	163222.75	[2725 157733]	[2876 171290]	9.47
Fuzz-RC208	R-SPEA2	[2167 2422 2569] [300746 315986 353844]	**2552.50**	**344379.50**	[2472 326410]	[2559 346941]	8.75
	R-NSGAII	[2361 2607 2707] [356369 357209 370401]	2682.00	367103.00	[2597 369702]	[2607 381056]	8.92

Table 5.2 – Robustness evaluation of R-SPEA2 and R-NSGAII

Instances	Algorithms	$\beta_N^1[best]$	$\beta_N^2[best]$	$\min(\beta_N^1, \beta_N^2)[best]$	$\min(\beta_N^1, \beta_N^2)[Avg]$	$\min(\beta_N^1, \beta_N^2)[Worst]$	$n-$rob
Fuzz-C102	R-SPEA2	0.750	0.750	**0.750**	**0,548**	**0.339**	0.986
	R-NSGAII	0.749	0.750	0.749	0.499	0.291	0.955
Fuzz-R102	R-SPEA2	0.751	0.752	**0.751**	**0.443**	**0.254**	0.989
	R-NSGAII	0.750	0.750	0.750	0.403	0.185	0.941
Fuzz-RC102	R-SPEA2	0.754	0.750	**0.750**	**0.389**	**0.217**	0.964
	R-NSGAII	0.750	0.750	0.750	0.358	0.187	0.931
Fuzz-C108	R-SPEA2	0.750	0.754	**0.750**	**0.374**	**0.333**	0.975
	R-NSGAII	0.748	0.749	0.748	0.305	0.256	0.945
Fuzz-R108	R-SPEA2	0.750	0.750	**0.750**	**0.423**	**0.276**	0.898
	R-NSGAII	0.741	0.744	0.741	0.375	0.255	0.886
Fuzz-RC108	R-SPEA2	0.750	0.750	**0.750**	**0.499**	**0.326**	0.951
	R-NSGAII	0.750	0.749	0.749	0.428	0.304	0.902
Fuzz-C204	R-SPEA2	0.750	0.750	**0.750**	**0.370**	**0.213**	0.979
	R-NSGAII	0.746	0.750	0.746	0.364	0.189	0.945
Fuzz-R204	R-SPEA2	0.748	0.749	**0.748**	**0.435**	**0.285**	0.889
	R-NSGAII	0.743	0.745	0.743	0.356	0.154	0.874
Fuzz-RC204	R-SPEA2	0.813	0.754	**0.754**	**0.491**	**0.292**	0.923
	R-NSGAII	0.754	0.746	0.746	0.424	0.170	0.904
Fuzz-C207	R-SPEA2	0.750	0.750	**0.750**	**0.571**	**0.362**	0.897
	R-NSGAII	0.749	0.750	0.749	0.497	0.227	0.883
Fuzz-R207	R-SPEA2	0.750	0.750	**0.750**	**0.470**	**0.233**	0.878
	R-NSGAII	0.746	0.748	0.746	0.368	0.183	0.869
Fuzz-RC207	R-SPEA2	0.750	0.755	**0.750**	**0.416**	**0.248**	0.953
	R-NSGAII	0.750	0.753	0.750	0.365	0.195	0.947
Fuzz-R210	R-SPEA2	0.753	0.749	**0.749**	**0.531**	**0.257**	0.896
	R-NSGAII	0.750	0.745	0.745	0.448	0.169	0.884
Fuzz-RC208	R-SPEA2	0.887	0.879	**0.879**	**0.547**	**0.318**	0.996
	R-NSGAII	0.750	0.755	0.750	0.449	0.274	0.994

Table 5.3 – Algorithms performance comparison

		I_H^-				$I_{\epsilon+}$			
Instances	Algorithms	E-SPEA2	R-SPEA2	E-NSGAII	R-NSGAII	E-SPEA2	R-SPEA2	E-NSGAII	R-NSGAII
Fuzz-C102	R-SPEA2	$\prec$	-	$\prec$	$\equiv$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\equiv$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-R102	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-RC102	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-C108	R-SPEA2	$\prec$	-	$\prec$	$\equiv$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\equiv$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-R108	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-RC108	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-C204	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-R204	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-RC204	R-SPEA2	$\prec$	-	$\prec$	$\equiv$	$\prec$	-	$\prec$	$\equiv$
	R-NSGAII	$\prec$	$\equiv$	$\prec$	-	$\prec$	$\equiv$	$\prec$	-
Fuzz-C207	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-R207	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-RC207	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-
Fuzz-R210	R-SPEA2	$\prec$	-	$\prec$	$\equiv$	$\prec$	-	$\prec$	$\equiv$
	R-NSGAII	$\prec$	$\equiv$	$\prec$	-	$\prec$	$\equiv$	$\prec$	-
Fuzz-RC208	R-SPEA2	$\prec$	-	$\prec$	$\prec$	$\prec$	-	$\prec$	$\prec$
	R-NSGAII	$\prec$	$\succ$	$\prec$	-	$\prec$	$\succ$	$\prec$	-

5.6 Conclusion

In this chapter, we have pointed out the difficulties faced when ignoring robustness in our previous achievements and then focused on defining new robustness concepts in a fuzzy multi-objective context. In addition, we have addressed the algorithmic issues in order to incorporate the robustness into the search process of our previous algorithms. The proposed robustness approach have been applied on the same VRP problem as in chapter 4. The experimental study carried out on a set of fuzzy sampled instances shows that we have successfully get the robust optimal solutions of the problem.

Conclusion and future work

Amongst the most challenging scientific problems, multi-objective optimization under uncertainty is today present as an active research area reflecting reality. Indeed, real-life problems are multi-objective by nature as they usually consider several conflicting objectives simultaneously. Besides, they are often subject to different types of uncertainty whose ignorance may cause misleading results.

However, from our deep survey of existing works relative to this area, we have identified that almost all of them have been limited to reduce such an uncertain multi-objective problem into one or more mono-objective subproblems or even to transform it into a deterministic equivalent while neglecting the uncertainty propagation to the model parameters and/or outputs. Hence, there is a significant need for new concepts, techniques and methods capable to handle the problem as-is without any transformation.

Through this Ph.D. thesis, we have contributed to the design of a generic framework for combinatorial multi-objective problems with fuzzy data, especially expressed by means of triangular fuzzy numbers and propagated to the objective functions. Our major contributions are three-fold:

The first one focuses primarily on the limitation of classical multi-objective concepts in dealing with fuzziness. In fact, we have proposed a novel Pareto approach for ranking the fuzzy outcomes generated in our case. On the one hand, this approach offers the possibility to rank a pair of triangular fuzzy numbers in a mono-objective context. On the other hand, it provides a Pareto ranking between two fuzzy solutions represented by vectors of triangular fuzzy numbers.

As second contribution, we have introduced a fuzzy extension of two well-known Pareto-based evolutionary algorithms. In particular, we have first incorporated our Pareto dominance relations into their fitness assignment strategy. Thereafter, we have extended the specific diversity preservation and elitism techniques of each algorithm in order to enable them working in a fuzzy search space. The main advantage of our proposed algorithms is the flexibility of their application to any multi-objective problem with fuzzy-valued objectives. In our study, we have applied them to resolve a bi-objective vehicle routing problem with fuzzy demands. The experimental results show that our algorithms are efficient in terms of front approximation quality and outperform their classical crisp versions.

The third contribution concerns the definition of a new robustness approach in fuzzy multi-objective setting. The aim of our proposal was to find robust optimal solutions taking into account a given performance level. To this end, we have introduced new robustness concepts with an evaluation procedure based on the attitude of decision maker. Subsequently, we have improved our fuzzy Pareto dominance relations by incorporating a robustness criterion. All these refinements have been next integrated into the previous algorithms to lead them achieving robust outcomes. Our approach have been illustrated on the vehicle routing problem and evaluated using some experimental tests. Finally, we have obtained encouraging results in terms of solutions quality and efficiency.

These original contributions have managed to combine four important research fields, namely the field of multi-objective optimization, the fuzzy logic domain, the evolutionary computation domain and the field of robustness. Thereby due to the novelty of our proposals in these fields, there are still many open questions and perspectives to investigate.

As short-and-mean term perspectives, we intend to improve what has been already achieved in this thesis. Indeed, our proposed framework has proved to be successful and adaptable to a variety of fuzzy multi-objective problems. In that sense, it could be further enhanced for new issues such as:

- The refinement of our proposed Pareto approach and algorithms to consider other popular fuzzy shapes like trapezoidal fuzzy numbers.
- The fuzzy extension of commonly used multi-objective quality indicators like the hypervolume metric. This allows us not only evaluating the generated solutions without defuzzifying them, but also provides an accurate and better performance assessment.

- The comparison of our framework with existing works in order to ensure a complete experimental evaluation.

As long term perspectives, we attempt to apply our framework on more complex combinatorial optimization problems. For instance, we could consider a multi-objective vehicle routing problem containing more than one fuzzy inputs data, like fuzziness in both customer demands and number of vehicles. It would also be interesting to investigate a real-world application for example in the domain of *Electric Vehicle Routing Problems with Recharging Stations*.

References

Abbasbandy, S., & Asady, B. (2006). Ranking of fuzzy numbers by sign distance. *Information Sciences, Elsevier*, *176*(16), 2405-2416.

Agarwal, H. (2004). *Reliability based design optimization: Formulations and methodologies* (PHD Dissertation). University of Notre Dame, Indiana.

Aissi, H., Bazgan, C., & Vanderpooten, D. (2009). Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European journal of operational research*, *197*(2), 427-438.

Averbakh, I. (2000). Minmax regret solutions for minimax optimization problems with uncertainty. *Operations Research Letters*, *27*(2), 57-65.

Ayyub, B. M., & Klir, G. J. (2006). *Uncertainty modeling and analysis in engineering and the sciences.* Boca Raton, FL USA : CRC Press.

Bader, J., & Zitzler, E. (2010). *Robustness in hypervolume-based multiobjective search* (Rapport technique). Tech. Rep. TIK 317, Computer Engineering and Networks Laboratory, ETH Zurich.

Bahri, O., Ben, N. A., & El-Ghazali, T. (2014a). New pareto approach for ranking triangular fuzzy numbers. In Springer (Ed.), *International Conference on Information Processing and Management of Uncertainty (IPMU'14)* (p. 264-273).

Bahri, O., Ben, N. A., & El-Ghazali, T. (2014b). Optimization algorithms for multi-objective problems with fuzzy data. In IEEE (Ed.), *Computational Intelligence in Multi-Criteria Decision-Making (MCDM'14)* (p. 194-201).

Bahri, O., Ben, N. A., & El-Ghazali, T. (2016). New pareto approach for ranking triangular fuzzy numbers. In Springer (Ed.), *International Conference on Information Processing and Management of Uncertainty (IPMU'16)* (p. 225-237).

Baños, R., Ortega, J., et al. (2013). A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & Industrial Engineering*, *65*(2), 286-296.

Barrico, C., & Antunes, C. H. (2006). Robustness analysis in multi-objective optimization using a degree of robustness concept. In *Proceedings of the IEEE Conference on Evolutionary Computation CEC'02* (p. 1887-1892).

Basseur, M., Seynhaeve, F., & Talbi, E.-G. (2002). Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In *Proceedings of the IEEE Congress on Evolutionary Computation CEC'02* (Vol. 2, p. 1151-1156).

Basseur, M., & Zitzler, E. (2006). Handling uncertainty in indicator-based multiobjective optimization. *International Journal of Computational Intelligence Research*, *2*(3), 255-272.

Ben-Tal, A., Ghaoui, L. E., & Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

Bertoluzza, C., Blanco, N. C., & Salas, A. (1995). On a new class of distances between fuzzy numbers. *Mathware & soft computing*, *2*, 71-84.

Bertsimas, D., & Nohadani, O. (2010). Robust optimization with simulated annealing. *Journal of Global Optimization*, *48*(2), 323-334.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations research*, *52*(1), 35-53.

Beyer, H.-G., & Sendhoff, B. (2007). Robust optimization: a comprehensive survey. *Computer methods in applied mechanics and engineering*, *196*(33), 3190-3218.

Bleuler, S., Laumanns, M., Thiele, L., & Zitzler, E. (2003). PISA—a platform and programming language independent interface for search algorithms. In *International Conference on Evolutionary Multi-criterion Optimization EMO'03* (p. 494-508).

Bortolan, G., & Degani, R. (1985). A review of some methods for ranking fuzzy subsets. *Fuzzy sets and systems, Elsevier*, *15*(1), 1-19.

Boulmakoul, A., Laarabi, M. H., Sacile, R., & Garbolino, E. (2013). Ranking triangular fuzzy numbers using fuzzy set inclusion index. In *International Workshop on Fuzzy Logic and Applications, Springer* (p. 100-108).

Bowerman, R., Hall, B., & Calamai, P. (1995). A multi-objective optimization approach to urban school bus routing: Formulation and solution method. *Transportation Research Part A: Policy and Practice*, *29*(2), 107-123.

Cahon, S., Melab, N., & Talbi, E.-G. (2004). ParadisEO: A framework for the reusable design of parallel and distributed meta-heuristics. *Journal of Heuristics*, *10*(3), 357-380.

Carraway, R. L., Morin, T. L., & Moskowitz, H. (1980). Generalized dynamic programming for multicriteria optimization. *European Journal of Operational Research, Springer*, *44*(1), 95-104.

Chen, S.-H. (1985). Ranking fuzzy numbers with maximizing set and minimizing set. *Fuzzy sets and systems, Elsevier*, *17*(2), 113-129.

Cheng, C.-H. (1998). A new approach for ranking fuzzy numbers by distance method. *Fuzzy sets and systems, Elsevier*, *95*(3), 307-317.

Choobineh, F., & Li, H. (1993). An index for ordering fuzzy numbers. *Fuzzy sets and systems, Elsevier*, *54*(3), 287-294.

Chu, T.-C., & Tsao, C.-T. (2002). Ranking fuzzy numbers with an area between the centroid point and original point. *Computers & Mathematics with Applications*, *43*(1), 111-117.

Coelho, R. F., & Bouillard, P. (2011). Multi-objective reliability-based optimization with stochastic metamodels. *IEEE congress on Evolutionary Computation*, *19*(4), 525-560.

Coello, C. A. C. (2000). Handling preferences in evolutionary multiobjective optimization: A survey. In *IEEE congress on Evolutionary Computation* (Vol. 1, p. 30-37).

Coello, C. A. C., Aguirre, A. H., & Zitzler, E. (2005). Evolutionary multi-criterion optimization. In *International Conference on Evolutionary Multi-criterion Optimization EMO'05* (Vol. 3410). Guanajuato, Mexico.

Coello, C. A. C., Van-Veldhuizen, D. A., & Lamont, G. B. (2007). *Evolutionary algorithms for solving multi-objective problems (second edition)* (Vol. 242). Springer Science & Business Media.

Cordeau, J.-F., & Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, *37*(6), 579-594.

Curry, D. M., & Dagli, C. H. (2014). Computational complexity measures for many-objective optimization problems. *Procedia Computer Science*, *36*, 185-191.

Daniels, R. L., & Carrillo, J. E. (1997). β-robust scheduling for single-machine systems with uncertain processing times. *IIE transactions, Springer*, *29*(11), 977-985.

Dantzig, G. B. (1955). Linear programming under uncertainty. *Management science*, *1*(3-4), 197-206.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, *6*(1), 80-91.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms.* John Wiley & Sons.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE transactions on Evolutionary Computation*, *6*(2), 182-197.

Deb, K., & Gupta, H. (2005). Searching for robust Pareto-optimal solutions in multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization EMO'05* (Vol. 34, p. 150-164).

Deb, K., & Gupta, H. (2006). Introducing robustness in multi-objective optimization. *Journal of Evolutionary Computation*, *14*(4), 463-494.

Diwekar, U. (2008). Optimization under uncertainty. In *Introduction to applied optimization* (p. 1-54). Springer.

Dubois, D., & Prade, H. (1988). *Possibility theory.* Springer.

Dubois, D., & Prade, H. (1998). Possibility theory: qualitative and quantitative aspects. In *Quantified representation of uncertainty and imprecision* (p. 169-226).

Dubois, D., & Prade, H. (2000). Possibility theory in information fusion. In *Proceedings of the Third International Conference on Information Fusion* (p. 6-19).

Durillo, J. J., & Nebro, A. J. (2011). jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software, Elsevier*, *42*(10), 760-771.

Edgeworth, F. (1881). Mathemetical Physics. *P. Keagan, Londre, Angleterre*.

Ehrgott, M., Ide, J., & Schöbel, A. (2014). Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research*, *239*(1), 17-31.

Ezzati, R., Allahviranloo, T., Khezerloo, S., & Khezerloo, M. (2012). An approach for ranking of fuzzy numbers. *Expert Systems with Applications, Elsevier*, *39*(1), 690-695.

Fieldsend, J. E., & Everson, R. M. (2005). Multi-objective optimisation in the presence of uncertainty. In *IEEE Congress on Evolutionary Computation* (Vol. 1, p. 243-250).

Fishburn, P. C. (1974). Lexicographic orders, utilities and decision rules: A survey. *Management science*, *20*(11), 1442-1471.

Fonesca, C. M., Fleming, P. J., Zitzler, E., et al. (2003). Evolutionary multi-criterion optimization. In *International Conference on Evolutionary Multi-Criterion Optimization EMO'03* (Vol. 2632). Berlin.

Gabrel, V., Murat, C., & Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, *235*(3), 471-483.

Gandibleux, X., Marc, S., Sörensen, K., et al. (2004). *Metaheuristics for multiobjective optimisation.* Springer Science & Business Media.

Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*, *88*(1), 3-12.

Ghannadpour, S. F., Noori, S., et al. (2014). A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. *Applied Soft Computing*, *14*, 504-527.

Ghoseiri, K., & Ghannadpour, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing Journal*, *10*(4), 1096-1107.

Goh, C.-K., & Tan, K. C. (2009). *Evolutionary multi-objective optimization in uncertain environments* (Vol. 186). Issues and Algorithms, Studies in Computational Intelligence.

Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). *The vehicle routing problem: latest advances and new challenges* (Vol. 43). Springer Science & Business Media.

Goncalves, G., Hsu, T., & Xu, J. (2009). Vehicle routing problem with time windows and fuzzy demands: an approach based on the possibility theory. *International Journal of Advanced Operations Management*, *1*(4), 312-330.

Gunawan, S., & Azarm, S. (2005). Multi-objective robust optimization using a sensitivity region concept. *Structural and Multidisciplinary Optimization*, *29*(1), 50-60.

Gupta, R., Singh, B., & Pandey, D. (2010). Multi-objective fuzzy vehicle routing problem: a case study. *Int. J. Contemp. Math. Sciences*, *5*(29), 1439-1454.

Handl, J., Kell, D. B., & Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology. *Computational Biology and Bioinformatics, IEEE Computer Society Press*, *4*(2), 279-292.

Hansen, M. P. (1997). Tabu search for multiobjective optimization: MOTS. In *International Conference on Multiple Criteria Decision Making* (p. 574-586).

Hansen, M. P., & Jaszkiewicz, A. (1998). *Evaluating the quality of approximations to the non-dominated set.* Technical rapport IMM-REP-1998-7, Department of Mathematical Modelling, University of Denmark.

Haubelt, C., & Teich, J. (2003). Accelerating design space exploration using pareto-front arithmetics. In *ACM Confererence on Asia and South Pacific Design Automation* (p. 525-531).

Heilpern, S. (1997). Representation and application of fuzzy numbers. *Fuzzy sets and systems, Elsevier*, *91*(2), 259-268.

Hendriks, M., Geile, M., & Basten, T. (2011). Pareto analysis with uncertainty. In *9th International Conference on Embedded and Ubiquitous Computing (EUC)* (p. 189-196).

Hughes, E. J. (2001). Evolutionary multi-objective ranking with uncertainty and noise. In *International Conference on Evolutionary Multi-criterion Optimization EMO'01* (p. 329-343).

Hwang, C.-L., & Mausud, A. S. M. (2012). *Multiple objective decision making—methods and applications: A state-of-the-art survey* (Vol. 164). Springer Science & Business Media.

Ismail, S. B., Legras, F., & Coppin, G. (2011). *Synthèse du problème de routage de véhicules* (PHD thesis). Institut Mines-Télécom-Bretagne-UEB, Occidentale-Université de Bretagne Sud.

Jaszkiewicz, A. (2002). Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, *137*(1), 50-71.

Jin, Y., & Sendhoff, B. (2003). Trade-off between performance and robustness: an evolutionary multiobjective approach. In *International Conference on Evolutionary Multi-Criterion Optimization (EMO'03)* (pp. 237–251).

Jozefowiez, N., Semet, F., & Talbi, E.-G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research*, *189*(2), 293-309.

Kao, G. K., & Jacobson, S. H. (2008). Finding preferred subsets of pareto optimal solutions. *Computational optimization and applications, Springer*, *40*(1), 73-95.

Kasperski, A., & Kulej, M. (2009). Choosing robust solutions in discrete optimization problems with fuzzy costs. *Fuzzy Sets and Systems*, *160*(5), 667-682.

Kaufmann, A., & Gupta, M. M. (1988). *Fuzzy mathematical models in engineering and management science* (Elsevier Science Inc., Ed.).

Kim, S., & hyun Ryu, J. (2011). The sample average approximation method for multi-objective stochastic optimization. In *Proceedings of the Win-*

ter Simulation Conference (WSC) (p. 4021-4032).

Klein, G., Moskowitz, H., & Ravindran, A. (1990). Interactive multiobjective optimization under uncertainty. *Management Science, INFORMS*, *36*(1), 58-75.

Klir, G., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic* (Vol. 4). Prentice hall New Jersey.

Knowles, J., & Corne, D. (2002). On metrics for comparing nondominated sets. In *Proceedings of the IEEE Congress on Evolutionary Computation CEC'02* (Vol. 1, p. 711-716).

Knowles, J., Thiele, L., & Zitzler, E. (2006). A tutorial on the performance assessment of stochastic multiobjective optimizers. *Tik report*, *214*, 327-332.

Kouvelis, P., & Yu, G. (1997). Robust discrete optimization and its applications. *luwer Academic Press, Boston*.

Kovacs, A. A., Golden, B. L., et al. (2014). Vehicle routing problems in which consistency considerations are important: A survey. *Networks, Wiley Online Library*, *64*(3), 192–213.

Lacomme, P., Prins, C., & Sevaux, M. (2006). A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers & Operations Research*, *33*(12), 3473-3493.

Li, F., Wu, T., Hu, M., & Dong, J. (2010). An accurate penalty-based approach for reliability-based design optimization. *Research in Engineering Design*, *21*(2), 87-98.

Li, M., Azarm, S., & Aute, V. (2005). A multi-objective genetic algorithm for robust design optimization. *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 771–778.

Li, P., Arellano-Garcia, H., & Wozny, G. (2008). Chance constrained programming approach to process optimization under uncertainty. *Computers & Chemical Engineering*, *32*(1), 25-45.

Liefooghe, A. (2009). *Methodes pour l'optimisation multiobjectif: Approche cooperative, prise en compte de l'incertitude et application logistique* (PHD thesis). Université de Lille1, Lille. (cited on pages 13-20)

Liefooghe, A., Basseur, M., Jourdan, L., & Talb, E.-G. (2007). ParadisEO-MOEO: A framework for evolutionary multi-objective optimization. In *International Conference on Evolutionary Multi-criterion Optimization EMO'07* (p. 386-400).

Liefooghe, A., Jourdan, L., & Talbi, E.-G. (2010). Indicator-based approaches for multiobjective optimization in uncertain environments. *Uncertainty and Robustness in Planning and Decision Making (UR-PDM)*.

Limbourg, P. (2005). Multi-objective optimization of problems with epistemic uncertainty. In *International Conference on Evolutionary Multi-criterion Optimization EMO'05* (p. 413-427).

Limbourg, P., & Aponte, D. E. S. (2005). An optimization algorithm for imprecise multi-objective problem functions. In *IEEE congress on evolutionary computation* (p. 459-466).

Maeda, S., & Kawachi, T. (2001). Epsilon-constraint approach for robust optimization of wasteload allocation in river systems. *Transactions of the Japanese Society of Irrigation, Drainage and Reclamation Engineering (Japan)*.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, *26*(6), 369-395.

Mavrotas, G., & Diakoulaki, D. (1998). A branch and bound algorithm for mixed zero-one multiple objective linear programming. *European Journal of Operational Research, Springer*, *107*(3), 530-541.

Meng, Z., Shen, R., Jiang, M., et al. (2011). An objective penalty functions algorithm for multiobjective optimization problem. *American Journal of Operations Research*, *1*(4), 229.

Meunier, H., Talbi, E.-G., & Reininger, P. (2000). A multiobjective genetic algorithm for radio network optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation CEC'2000* (Vol. 1, p. 317-324).

Palacios, J. J., González-Rodríguez, I., Vela, C. R., & Peinador, J. P. (2014). β-robust solutions for the fuzzy open shop scheduling. In *International Conference on Information Processing and Management of Uncertainty (IPMU'14)* (Vol. 442, pp. 447–456).

Paquete, L., & Stützle, T. (2007). Stochastic local search algorithms for multiobjective combinatorial optimization. *Handbook of Approximation Algorithms and Metaheuristics*, *13*.

Pareto, V. (1971). *Manual of political economy, Translated by AS Schwie.* Macmillan, London.

Park, J., & Kim, B.-I. (2010). The school bus routing problem: A review. *European Journal of Operations Research*, *202*(2), 311-319.

Park, Y. B., & Koelling, C. P. (1989). An interactive computerized algorithm for multicriteria vehicle routing problems. *Computers & Industrial Engineering*, *16*(4), 477-490.

Pedrycz, W. (1994). Why triangular membership functions? *Fuzzy sets and Systems, Elsevier*, *64*(1), 21–30.

Petrone, G. (2011). *Optimization under uncertainty: theory, algorithms and industrial applications* (PHD thesis). Università degli Studi di Napoli Federico II. (cited on pages 77-122)

Pishevar, A., & Tavakkoi-Moghaddam, R. (2014). β-robust parallel machine scheduling with uncertain durations. *Universal Journal of Industrial and Business Management*, *2*(3), 69-74.

Prins, C., & Bouchenoua, S. (2005). A memetic algorithm solving the vrp, the carp and general routing problems with nodes, edges and arcs. In *Recent advances in memetic algorithms* (p. 65-85).

Rachmawati, L., & Dipti, S. (2006). Preference incorporation in multi-objective evolutionary algorithms: A survey. In *IEEE Congress on Evolutionary Computation, CEC'06* (p. 962-968).

Ribeiro, R., & Lourenço, H. R. D. (2001). A multi-objective model for a multi-period distribution management problem. *SSRN, New York USA*, 1-22.

Rivaz, S., & Yaghoobi, M. (2013). Minimax regret solution to multiobjective linear programming problems with interval objective functions coefficients. *European Journal of Operations Research*, *21*(3), 625–649.

Rockafellar, R. (s. d.). Optimization under uncertainty. *Lecture Notes*.

Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering, Elsevier*, *28*(6), 971-983.

Segev, D. (2007). *Approximation algorithms for NP-hard problems in combinatorial optimization* (PHD Thesis). Tel-Aviv University.

Sevaux, M., & Sörensen, K. (2004). Robustness analysis: optimisation. *Newslett Eur Working Group "Multiple Criteria Decision Aiding" Ser*, *3*.

Shafer, G., et al. (1976). *A mathematical theory of evidence* (Vol. 1). Princeton university press Princeton.

Shaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *First international conference on genetic algorithms* (p. 93-100). L. Erlbaum Associates Inc.

Sheskin, D. J. (2003). *Handbook of parametric and nonparametric statistical* *procedures.* CRC PressINX.

Silva, R. C., & Yamakami, A. (2009). Definition of fuzzy pareto-optimality by using possibility theory. In *IFSA/EUSFLAT conference, citeseer* (p. 1234-1239).

Soares, G., Adriano, R., Maia, C., et al. (2009). Robust multi-objective TEAM 22 problem: A case study of uncertainties in design optimization. *IEEE Transactions on Magnetics*, *45*(3), 1028-1031.

Soares, G., Guimarães, F. G., Maia, C. A., et al. (2009). Interval robust multi-objective evolutionary algorithm. *Evolutionary Computation*, 1637-1643.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, *35*(2), 254-265.

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, *21*(5), 1154-1157.

Srinivas, N., & Deb, K. (1994). Multi-objective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, *2*(3), 221-248.

Steele, J. M. (1997). *Probability theory and combinatorial optimization* (Vol. 69). Siam.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation.* John Wiley & Sons.

Tan, K., Lee, T., Khoo, D., Khor, E., & Kannan, R. S. (2000). MOEA toolbox for computer aided multi-objective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation CEC'2000* (p. 38-45).

Tan, K. C., Cheong, C. Y., & Goh, C. K. (2007). Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research*, *177*(2), 813-839.

Tavakkoli, M. R., Norouzi, N., et al. (2012). A vehicle routing problem condidering a balance of goods based on the vehicle capacity by improved practical swarm optimization. *Journal of Transportation Research.*

Teich, J. (2001). Pareto-front exploration with uncertain objectives. In *International Conference on Evolutionary Multi-criterion Optimization* (p. 314-328).

Toth, P., & Vigo, D. (2014). *Vehicle routing: problems, methods, and appli-*

cations (2nd éd., Vol. 18). Siam.

Veldhuizen, D. A. V., & Lamont, G. B. (1998). *Multiobjective evolutionary algorithm research: A history and analysis* (Technical Report). Department of Electrical and Computer Engineering, Air Force Institute of Technology.

Wang, Y.-M. (2009). Centroid defuzzification and the maximizing set and minimizing set ranking based on alpha level sets. *Computers & Industrial Engineering, Elsevier*, *57*(1), 228-236.

Wang, Z., & Tian, F. (2010). A note of the expected value and variance of fuzzy variables. *International Journal of Nonlinear Science*, *9*(4), 486-492.

Yager, R. R. (1986). A characterization of the extension principle. *Fuzzy sets and systems, Elsevier*, *18*, 205-217.

Yao, J.-S., & Wu, K. (2000). Ranking fuzzy numbers based on decomposition principle and signed distance. *Fuzzy sets and Systems, Elsevier*, *116*(2), 275-288.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, *8*(3), 338-353.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems, Elsevier*, *100*, 9-34.

Zhou, J., Yang, F., & Wang, K. (2014). Multi-objective optimization in uncertain random environments. *Fuzzy Optimization and Decision Making, Springer*, *13*(4), 397-413.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, *8*(2), 173-195.

Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In *8th international conference on parallel problem solving from nature, springer* (p. 832-842). Birmingham UK.

Zitzler, E., Laumanns, M., Thiele, L., et al. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. In *Eurogen* (Vol. 3242, p. 95-100).

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Fonseca, V. G. D. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE transactions on evolutionary computation*, *7*(2), 117-132.