

gen 200 2126

50376  
1998  
379

N° d'ordre :



# THESE

présentée à

**L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE**

pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITE**

en

**PRODUCTIQUE, AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE**

par

**Paul BARATOU**

**GESTION REACTIVE**

**DE STOCKS INTERMEDIAIRES**

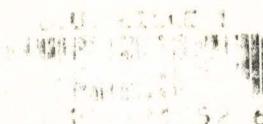
**D'UN FLUX DE PRODUCTION AUTOMOBILE**

Soutenue le 9 mars 1998

Devant la Commission d'examen :

M. Y. FREIN	Rapporteur	Professeur à L'INP de Grenoble
M. R. SOENEN	Rapporteur	Professeur à l'Université de Valenciennes
M. E. CASTELAIN	Directeur de thèse	Maître de Conférences, H.d.R à l'EC de Lille
M. J.C. GENTINA	Examinateur	Professeur à l'Ecole Centrale de Lille
M. J.P CASSAR	Examinateur	Professeur à l'Université de Lille I
M. P. BONNEAU	Examinateur	Docteur, Ingénieur à PSA Peugeot Citroën

Thèse préparée au Laboratoire d'Automatique et Informatique Industrielle de Lille  
L.A.I.L. URA CNRS D1440 - Ecole Centrale de Lille



## Remerciements

Cette thèse a été préparée dans le cadre d'une convention CIFRE entre PSA Peugeot-Citroën et le Laboratoire d'Automatique et d'Informatique Industrielle de Lille.

Je tiens à remercier M. DUMONT pour m'avoir accueilli au sein de la division dont il est le responsable.

Je tiens à remercier le Professeur Jean-Claude GENTINA, directeur de l'Ecole Centrale de Lille, de m'avoir accueilli au sein de son équipe.

Je remercie tout particulièrement mon directeur de recherche, Emmanuel CASTELAIN, pour la confiance et les conseils qu'il m'a donnés au long de ces trois ans.

J'adresse mes remerciements les plus vifs à Patrick BONNEAU qui a suivi mon travail de recherche pendant ces trois ans. Je le remercie pour ses conseils, ses apports et ses encouragements pendant ces trois ans.

Je tiens à remercier tout particulièrement les deux rapporteurs pour l'honneur qu'ils me font en acceptant d'examiner ce travail et pour le temps qu'ils ont consacré à l'évaluer :

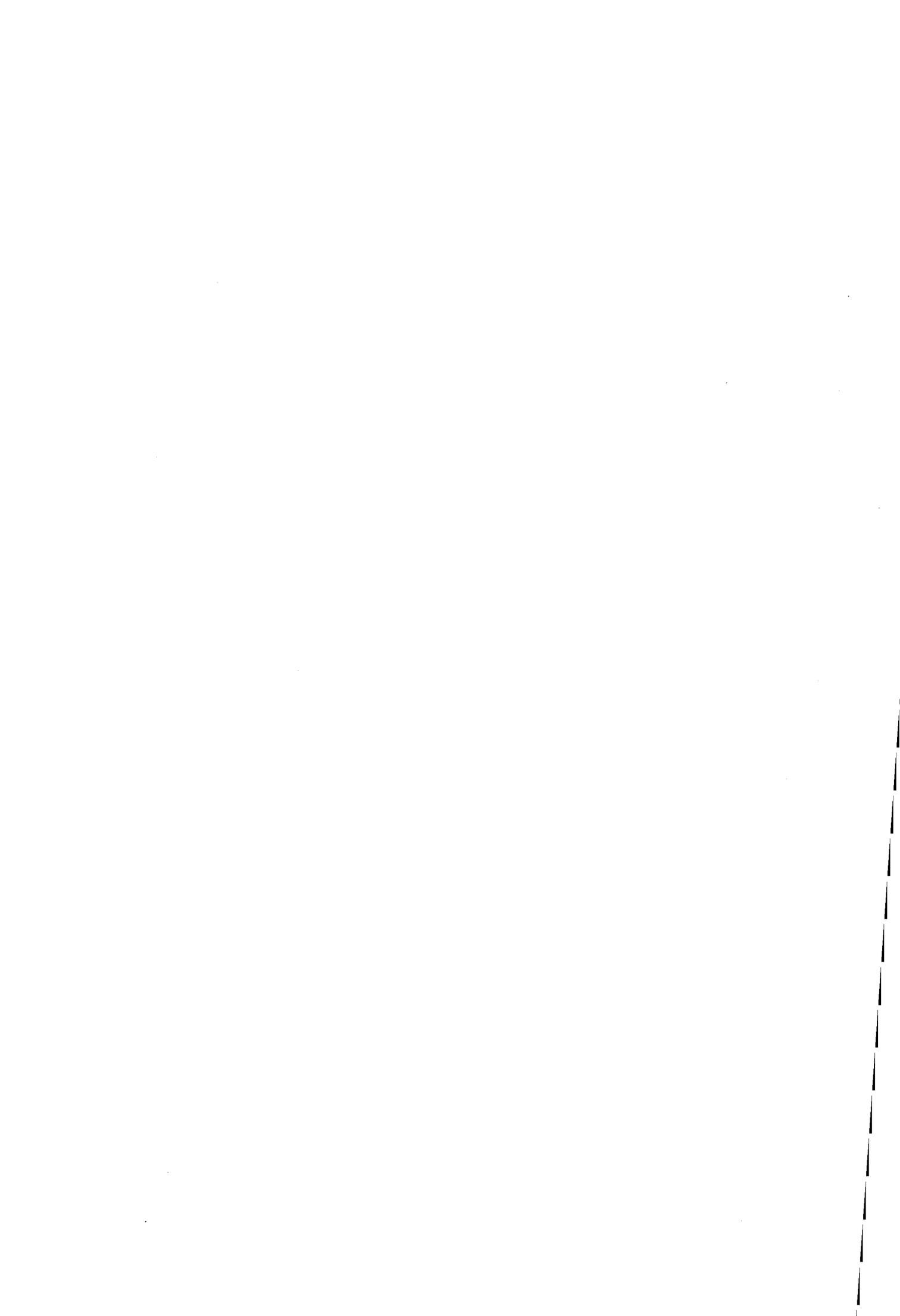
- Monsieur Yannick FREIN, Professeur à l'Institut National Polytechnique de Grenoble, Directeur du laboratoire GILCO.

- Monsieur René SOENEN, Professeur à l'Université de Valenciennes, Directeur du laboratoire LAMIH.

Je remercie le Professeur Jean-Philippe CASSAR pour sa présence dans le jury.

Je remercie toute l'équipe SAA et mes collègues du LAIL pour leur intérêt, soutien et apports à ce mémoire.

Ma reconnaissance va de même vers toutes celles et tous ceux qui m'ont apporté leur aide et leur soutien pendant ces trois années.



---

**TABLE DES MATIERES**


---

<b>INTRODUCTION.....</b>	<b>7</b>
<b>PARTIE 1 : PRESENTATION DES LIGNES D'ASSEMBLAGE A MODELES MELANGES ET DES PROBLEMES D'ORDONNANCEMENT ASSOCIES .....</b>	<b>15</b>
1. INTRODUCTION GENERALE DU PROBLEME.....	17
2. CONTEXTE ET EVOLUTION DE LA PRODUCTION .....	17
2.1. <i>Contexte actuel de la production</i> .....	17
2.2. <i>La production artisanale</i> .....	19
2.3. <i>La production de masse</i> .....	20
2.4. <i>La production au plus juste</i> .....	20
2.5. <i>La production réfléchie</i> .....	21
2.6. <i>Conclusion</i> .....	22
3. LIGNES DE PRODUCTION .....	22
3.1. <i>Système de transport</i> .....	23
3.2. <i>Atelier ferrage</i> .....	24
3.3. <i>Atelier peinture</i> .....	24
3.4. <i>Atelier montage</i> .....	26
3.5. <i>Les stocks</i> .....	29
3.5.1. <i>Stocks à tiroirs</i> .....	29
3.5.2. <i>Autostockeurs (ou MGH)</i> .....	31
3.5.3. <i>Stocks mixtes</i> .....	31
3.5.4. <i>Comparaisons</i> .....	31
4. LISSAGE DES CHARGES SUR LES LIGNES MULTIPRODUITS A FORTE DIVERSITE : EQUILIBRAGE ET SEQUENCEMENT .....	32
4.1. <i>Optimisation du fonctionnement d'une ligne d'assemblage</i> .....	32
4.2. <i>Optimisation à moyen terme (Equilibrage)</i> .....	34
4.3. <i>Optimisation à court terme</i> .....	35
4.3.1. <i>Les différents types de contraintes</i> .....	36
4.3.1.1. <i>Contraintes de production au plus juste</i> .....	36
4.3.1.2. <i>Respecter les capacités de production</i> .....	36
4.3.1.3. <i>Lisser la charge de production</i> .....	37
4.3.1.4. <i>Minimiser les coûts de changement d'outils</i> .....	37
4.3.2. <i>Compromis</i> .....	38
4.4. <i>Problème statique : séquençement</i> .....	39
4.5. <i>Problème dynamique : cadencement</i> .....	41
4.6. <i>Autres stratégies de gestion</i> .....	41
5. CONCLUSION .....	44

---

<b>PARTIE 2 : OUTILS, METHODES, MODELISATION.....</b>	<b>45</b>
1. INTRODUCTION .....	47
2. ORDONNANCEMENT DES LIGNES A MODELES MELANGES : TYPOLOGIE DES REGLES DE SEQUENCEMENT.....	47
2.1. <i>"Répartition des produits" ou lissage des critères</i> .....	49
2.1.1. Taux de consommation de composant (PART USAGE GOAL).....	50
2.1.2. Taux de production de chaque type de produit fini (MODEL PRODUCTION RATE).....	52
2.1.3. Consommation cumulée.....	53
2.1.4. Définition de positions cibles.....	55
2.2. <i>Respect des capacités de production (ou "Ratios")</i> .....	56
2.3. <i>Ratio orienté</i> .....	60
2.4. <i>Position imposée</i> .....	60
2.5. <i>Position souhaitée</i> .....	60
2.6. <i>Placer au plus tôt</i> .....	61
2.7. <i>Placer au plus tard</i> .....	61
2.8. <i>Succession impaire</i> .....	61
2.9. <i>Suivi des modifications</i> .....	61
2.10. <i>Respect des "RAFALES"</i> .....	61
2.11. <i>Conclusion</i> .....	63
3. MODELISATION DU PROBLEME PAR CONTRAINTES .....	64
3.1. <i>De la notion de charge de travail à celle d'espacement</i> .....	64
3.2. <i>Indicateurs de cohérence et de résultats</i> .....	65
3.2.1. Indicateurs a priori .....	65
3.2.2. Indicateurs a posteriori .....	67
4. LES METHODES D'OPTIMISATION RETENUES .....	68
4.1. <i>Introduction</i> .....	68
4.2. <i>Simulations</i> .....	70
4.3. <i>Méthodes exactes</i> .....	70
4.3.1. Branch & Bound.....	71
4.3.2. Propagation par contraintes.....	73
4.3.3. Programmation linéaire .....	73
4.4. <i>Heuristiques</i> .....	74
4.5. <i>Heuristiques de construction progressive</i> .....	75
4.5.1. Méthode à un pas .....	75
4.5.2. Méthode à P pas.....	75
4.5.3. Autres approches.....	76
4.6. <i>Heuristiques par affinements successifs</i> .....	77
4.6.1. Méthode du gradient.....	78
4.6.2. Recuit simulé .....	78
4.6.3. Méthode Tabou .....	79
4.6.4. Algorithme génétique .....	80
5. CONCLUSION .....	81

<b>PARTIE 3 : SEQUENCEMENT .....</b>	<b>83</b>
1. SEQUENCEMENT DES PRODUITS .....	85
1.1. <i>Les objectifs du séquençement</i> .....	85
1.2. <i>Le problème posé</i> .....	85
2. APPROCHE DE RESOLUTION PROPOSEE .....	85
2.1. <i>Les méthodes d'optimisation retenues</i> .....	86
2.1.1. Heuristiques de construction progressive .....	86
2.1.2. Heuristiques par affinements successifs .....	87
2.1.2.1. Définition des voisinages .....	87
2.1.2.2. Fonction de coût ou métrique .....	88
2.1.2.3. Configuration initiale .....	90
2.1.2.4. Plan de recuit .....	91
2.1.3. Méthode combinée .....	92
2.2. <i>Description de la méthode combinée</i> .....	93
2.2.1. Introduction .....	93
2.2.2. La partition .....	94
2.2.3. Méthodes de partition .....	95
2.2.4. Recherche par faisceaux .....	98
2.2.5. Recuit simulé .....	100
2.3. <i>Complexité des algorithmes</i> .....	100
2.3.1. Goal Chasing .....	100
2.3.2. Recuit Simulé .....	100
3. METRIQUES DE SEQUENCEMENT .....	101
3.1. <i>Métriques constructives et évaluatives</i> .....	102
3.2. <i>Ratio</i> .....	102
3.3. <i>Rafales</i> .....	103
3.4. <i>Date</i> .....	104
4. RESULTATS ET COMPARAISON ENTRE LES METHODES .....	104
4.1. <i>Description des jeux de tests</i> .....	104
4.2. <i>Comparaison des méthodes</i> .....	106
4.2.1. Simulation 1 .....	107
4.2.2. Simulation 2 .....	110
4.2.3. Jeu de test réel .....	112
4.3. <i>Remarques</i> .....	112
5. CONCLUSION .....	113
<b>PARTIE 4 : CADENCEMENT .....</b>	<b>115</b>
1. PROBLEMATIQUE .....	117
2. MODELISATION DES DIFFERENTS STYLES DE STOCKS ET PARTICULARITES .....	118
2.1. <i>Autostockeurs MGH</i> .....	119
2.2. <i>Tiroirs</i> .....	120
2.3. <i>Architectures mixtes</i> .....	125

2.4. Conclusion .....	126
3. CARACTERE REACTIF ET OPTIMISATION LOCALE POUR UNE VUE GLOBALE.....	126
3.1. Réactif et Dynamique .....	127
3.2. Problème instantané et local .....	128
3.3. Influences sur l'algorithme de cadencement.....	129
4. ALGORITHME DE PARCOURS D'ARBRE .....	132
4.1. Introduction .....	132
4.2. Politiques de gestion .....	134
4.2.1. Gestion simultanée de l'entrée et de la sortie des tiroirs .....	135
4.2.2. Gestion de l'entrée indépendante de la sortie.....	137
4.3. Algorithme d'entrée.....	139
5. METRIQUES DE CADENCEMENT .....	144
5.1. Convexité des métriques .....	144
5.2. Répartition des produits .....	145
5.3. Ratio .....	147
5.4. Position souhaitée et date.....	148
5.5. Rafales .....	149
5.6. Interaction entre les règles.....	149
6. APPROCHE DE RESOLUTION PROPOSEE .....	151
6.1. Structure de l'algorithme.....	152
6.2. Heuristiques et paramètres pour le parcours d'arbre .....	154
6.2.1. Comparaisons des méthodes sur un jeu de test .....	154
6.2.2. Paramétrage de l'arbre .....	157
6.2.3. Paramétrage du stock.....	160
6.3. Conclusion .....	163
7. GESTION AMONT : APPROCHE CURATIVE.....	163
7.1. Cadencement Réactif pur .....	165
7.2. Remise en séquence.....	165
7.3. Cadencement Réactif avec ancienneté.....	166
7.4. Remise en séquence avec contraintes .....	166
7.5. Résultats.....	167
8. GESTION AVALE PREDICTIVE.....	169
8.1. Introduction .....	169
8.2. Cadencement reséquençable.....	170
8.3. Résultats et modes de fonctionnement.....	173
9. CONCLUSION .....	175
<b>CONCLUSION .....</b>	<b>177</b>
<b>REFERENCES BIBLIOGRAPHIQUES .....</b>	<b>183</b>
<b>LISTES DES FIGURES.....</b>	<b>189</b>

# INTRODUCTION



## INTRODUCTION

Les constructeurs automobiles ont fabriqué des véhicules en grandes séries afin de minimiser les coûts de production unitaire en utilisant les principes de production de masse de Ford. Les produits sont standardisés et les gains sont réalisés grâce à des économies d'échelle. Le système de production adapté à ce style de production est la ligne cadencée. Les lignes d'assemblages regroupent des centaines de postes de travail qui correspondent à un tronçon continu de la ligne sur lequel des opérations peuvent et doivent être faites. L'intérêt de ce type de ligne est la facilité de manutention des véhicules sur la ligne et la rationalisation de la production (en facilitant le pilotage et la gestion des encours). L'inconvénient majeur de cette conception de la production de masse est la rigidité du système dans son ensemble.

L'évolution de la demande et la concurrence croissante ont développé la diversité des produits offerts sur le marché : les modèles de véhicules sont très nombreux, personnalisés (chaque client peut choisir les options qu'il souhaite). Ces différents modèles sont assemblés sur la même ligne car le coût de telles installations est élevé. Elles s'appellent des lignes d'assemblage à modèles mélangés ("Mixed Model Assembly Lines" dans la littérature anglo-saxonne) et permettent de produire des véhicules satisfaisant la demande des clients : la production de masse est devenue la production au plus juste.

La diversité des modèles sur la ligne entraîne d'importantes variations dans la durée des tâches sur les différents modèles. Pour gérer la diversité de la ligne, il existe deux degrés de liberté [OKA 79]:

- L'équilibrage de la ligne permettant de fabriquer les volumes prévisionnels issus des prévisions commerciales. Nous supposons dans cette thèse que la ligne a été correctement équilibrée.

- La détermination de l'ordonnancement des produits sur la ligne : c'est la détermination d'une séquence de véhicules qui devront être fabriqués dans cet ordre.

Les gains de productivité sont alors obtenus par une bonne gestion de cette ligne à l'aide d'une politique de lancement et d'ordonnancement. Ces problèmes d'ordonnancement sont complexes.

En effet cet ordonnancement doit garantir :

- la faisabilité de la production (contraintes des moyens physiques)
- le coût minimum de fabrication (lissage de la charge de travail)
- la synchronisation des flux de composants

La détermination de cette séquence idéale de véhicules à produire peut être obtenue de deux façons différentes. La première approche consiste à résoudre quotidiennement ce problème avec l'ensemble des commandes des clients devant être fabriquées ce jour : c'est le **séquencement**. Cet ordonnancement prévisionnel va prendre en compte un grand nombre de contraintes de production et va essayer de déterminer un compromis satisfaisant entre elles. Mais des aléas de production viennent perturber le séquencement prévisionnel et l'ordonnancement prédictif ne peut être conservé tout au long du flux de l'usine. Des stocks intermédiaires de production permettent de réordonnancer le flux de véhicules au niveau de ces stocks : c'est le **cadencement**.

Dans cette thèse menée dans le cadre d'une convention CIFRE entre PSA Peugeot Citroën DTII (Direction des Technologies de l'Information et de l'Informatique) et le laboratoire LAIL (Laboratoire d'Automatique et d'Informatique Industrielle de Lille) de l'Ecole Centrale de Lille, nous avons cherché à améliorer le séquencement prévisionnel des véhicules et la gestion des stocks intermédiaires de production à travers le cadencement, en particulier dans le cas des architectures à tiroirs.

La première partie de ce mémoire va décrire le problème industriel du séquencement des lignes d'assemblage à modèles mélangés. Le contexte spécifique au monde automobile est étudié ainsi que l'évolution de la production automobile pour répondre à l'évolution de la demande. Les lignes de production automobile sont ensuite décrites avec leurs particularités : convoyeur, stocks et ateliers. Nous terminons par une approche du problème du séquencement proprement dit.

La deuxième partie aborde les outils, les méthodes et la modélisation du problème que nous avons retenu. En effet le problème de séquencement sur une ligne d'assemblage à modèles mélangés est fortement dépendant du contexte industriel. Il est important de définir clairement les objectifs de l'optimisation. Dans un premier temps, nous étudions les différentes règles de séquencement qui peuvent être utilisées dans le monde automobile en nous appuyant essentiellement sur les travaux de Comby [COM 96], qui ont été menés chez RVI, dans le cadre d'une usine d'assemblage de camions. Puis, nous essayons de décrire des indicateurs concrets qui permettent de représenter la qualité du flux et nous étudions plus finement les modélisations que nous pouvons retenir. Pour finir, une revue des différentes méthodes d'optimisations (locales ou globales) est faite avec une étude des différentes applications qui en résultent.

La troisième partie propose une méthode de résolution du séquencement. L'objectif du séquencement d'une ligne d'une usine consiste à créer une séquence de référence. Cet

ordonnancement est effectué en statique et une fois par période de référence (en général une journée de production).

Dans le cas d'un problème industriel, de nombreuses contraintes apparaissent et sont parfois concurrentes. Deux méthodes classiquement utilisées sont décrites et comparées : il s'agit du Goal Chasing et du Recuit Simulé. Puis nous proposons une nouvelle approche, appelée méthode combinée, plus adaptée à l'espace de recherche chaotique du monde automobile.

La quatrième partie s'attaque au problème de cadencement : c'est une approche essentiellement dynamique.

Cette approche consiste à ordonnancer dynamiquement les produits entrant et sortant des stocks intermédiaires de production. En effet, pour optimiser le flux de production les règles et les contraintes à respecter sont les mêmes que pour le problème statique de séquençement. En revanche, seules les contraintes et les règles des ateliers avals au stock sont encore intéressantes à gérer. La différence apparaît au niveau de contraintes physiques liées au stock qui n'existaient pas dans le cadre du problème statique (configuration et contenu du stock). Elles apparaissent également au niveau des contraintes temporelles car l'algorithme de gestion du stock doit pouvoir assurer les choix en garantissant un temps de réponse compatible avec la vitesse de la ligne en sortie de stock.

Dans cette quatrième partie, nous allons décrire le problème lié à la configuration des stocks et la façon dont nous l'avons modélisé (en particulier pour les stocks à tiroirs). Dans un deuxième temps, nous allons préciser les contraintes temporelles du problème et les difficultés rencontrées lors d'une optimisation dynamique d'un stock. Nous aborderons ensuite les algorithmes que nous avons utilisés, leur implémentation et les métriques utilisées.

Nous décrirons en fin de partie, les deux approches concurrentielles qui peuvent être utilisées pour gérer un stock. L'approche amont est une méthode curative et traditionnelle. Elle consiste à générer une séquence pour l'atelier suivant et uniquement pour celui-ci en corrigeant les aléas du flux issus de l'atelier amont. L'approche avale est une approche prédictive. Elle crée une séquence pour optimiser l'atelier aval en garantissant que les perturbations engendrées par cette optimisation pourront être corrigées avec le stock suivant.

L'ensemble de ces travaux a été mené en vue d'une implantation sur le site de Rennes où les stocks sont des stocks à tiroirs.



*Partie 1*

PRESENTATION DES LIGNES D'ASSEMBLAGE

A MODELES MELANGES

ET

DES PROBLEMES D'ORDONNANCEMENT

ASSOCIES



## **Partie 1 : Présentation des lignes d'assemblage à modèles mélangés et des problèmes d'ordonnancement associés**

### ***1. Introduction générale du problème***

Cette première partie va décrire le problème industriel du séquençement des lignes d'assemblage à modèles mélangés<sup>1</sup>. C'est en effet le type de lignes d'assemblage utilisé dans les usines automobiles. Dans un premier temps, le contexte spécifique au monde automobile sera étudié. Puis la ligne de production automobile et ses spécificités seront abordées, avant de terminer par une approche du problème de séquençement proprement dit.

### ***2. Contexte et Evolution de la production***

#### **2.1. Contexte actuel de la production**

Les cinquante millions de voitures fabriquées chaque année dans le monde et les treize millions vendues en Europe montrent l'importance de l'industrie automobile dans l'économie européenne et mondiale. La fabrication des automobiles remonte au dix-neuvième siècle. A cette époque, elle était artisanale. Mais depuis, cette production industrielle n'a cessé d'évoluer et de se diversifier en devenant de plus en plus complexe. Dans notre société de consommation, l'évolution de la demande contribue à cette évolution et s'alimente ainsi elle-même. Les évolutions industrielles associées à ces évolutions de la demande peuvent se décomposer en trois phases.

La première de ces trois phases est le passage à une production de masse pour augmenter les volumes d'une production artisanale limitée. C'est l'instauration du "Fordisme" et l'avènement de la consommation de masse. Dans un deuxième temps, la réduction des coûts liée à la production de masse n'étant plus suffisante pour conquérir des parts de marché, les entreprises ont cherché à élargir l'offre proposée au client. La troisième phase est apparue au début des années 80, avec une adéquation plus juste entre la demande et le client. La transformation du marché en Amérique et en Europe en un marché de renouvellement a amené les fabricants à proposer en plus au client LE produit désiré dans le délai le plus court. Womack dans son livre [WOM 90] décrit l'évolution de la production automobile :

---

<sup>1</sup> appelé en langue anglaise "mixed-model assembly line "

### Progression de la variété de l'offre et du volume de production dans l'industrie automobile

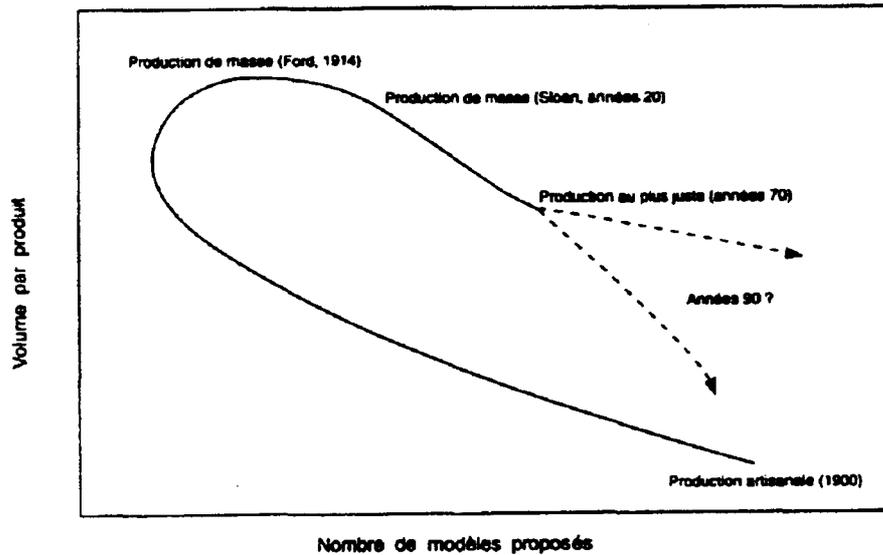


Figure I.1 - Evolution de l'offre automobile au XX<sup>e</sup> siècle

Pour satisfaire ces différentes politiques, quatre façons différentes de produire des voitures ont été développées. Elles correspondent à l'évolution logique de la production, d'abord en volume puis en diversité :

- La production artisanale (fabrication unitaire ou de petites séries).
- La production de masse (fabrication de grandes séries sans diversité).
- La production au plus juste (fabrication de grandes séries avec une grande diversité).
- La production réfléchie (volumes plus faibles mais grande diversité).

Ces méthodes sont liées chacune aux différentes évolutions de la production et de la demande décrites précédemment.

Dans la revue "Le Revenu Français", la logistique de Peugeot est décrite pour garantir au client un délai dans la livraison de son automobile. L'objectif est de satisfaire le client en lui proposant le véhicule de son choix et en étant capable de lui garantir une date à laquelle il pourra venir chercher son véhicule chez le concessionnaire. Pour cela, l'ensemble des maillons nécessaires doit être robuste et garantir le délai de réalisation de leur tâche, en particulier la fabrication proprement dite du véhicule dans l'usine.

de travail sont découpés en "équipes" autonomes de petites tailles responsables d'une partie de l'assemblage.

De manière simpliste, cela correspond à cinq zéros :

- zéro temps de changement d'outils (réduire les temps de changement de série)
- zéro défaut (instaurer l'autocontrôle)
- zéro délai
- zéro stock (réduire les délais en ne fabriquant que le strict nécessaire)
- zéro panne (développer la maintenance préventive)

Ces évolutions permettent d'atteindre les trois principaux objectifs caractérisant la production au plus-juste : grande productivité, gain en trésorerie et grande qualité du produit. Mais ces méthodes de travail nécessitent une remise en cause complète des fonctionnements de l'entreprise.

## **2.5. La production réfléchie**

Ellgard et al. [ELL 92] dans leur article décrivent leur conception de la production réfléchie<sup>4</sup>, c'est l'évolution de la production de masse dans l'environnement socio-économique suédois. Techniquement ce type de production consiste à paralléliser au maximum les flux de matières dans les usines d'assemblage final. Des équipes de travail autonomes de grande taille sont responsables de l'assemblage d'un véhicule complet : les tâches assignées à un individu sont donc nombreuses et variées et le temps de cycle est donc beaucoup plus long.

Cette production, en plus des objectifs économiques présents chez les autres, a un objectif de "satisfaction sociale". L'ergonomie des postes de travail est étudiée en détail. Elle permet aux employés d'avoir une importante autonomie dans leur action.

Le concept de production réfléchie est associé par ses détracteurs à un néo-artisanat. Ce terme est excessif car les modes de fonctionnement sont évolués et utilisent des technologies avancées [JOH 90]. Néanmoins, ce type de production semble réservé à la fabrication de véhicule de haut de gamme où la qualité du produit est plus importante que le coût d'assemblage. En effet ce style de production ne permet pas les mêmes cadences de production que les deux systèmes précédents. En revanche, les ouvriers très qualifiés sont capables de garantir une qualité d'assemblage supérieure. Ce style de production existe dans l'usine suédoise de Volvo : Uddevalla.

---

<sup>4</sup> "reflective production" en anglais

Ce type de production n'est pas nécessairement adapté à tous les types d'usines et à tous les constructeurs. La plupart des constructeurs automobiles, comme PSA, ont choisi une production semblable à la production au plus juste.

## 2.6. Conclusion

Dans l'industrie automobile, les modes de production les plus utilisés sont la production de masse et la production au plus juste. A la différence des modes de production artisanales et réfléchies, ils utilisent des lignes de production cadencées. Ils permettent de standardiser le travail et de diminuer les en-cours entre postes.

D'autre part il s'agit aujourd'hui de fabriquer dans un délai le plus court possible le véhicule commandé par le client avec les options qu'il a choisies. Cette diversité dans les produits à fabriquer transforme quasiment la production en une production unitaire de part la diversité des équipements existants sur les véhicules. Pour améliorer la réactivité et diminuer les stocks, la diversité des véhicules est concentrée sur l'atelier d'assemblage final. On est donc confronté à des lignes d'assemblage à modèles mélangés qui requièrent un ordre précis dans lequel les tâches doivent être effectuées et dans lequel les produits doivent être fabriqués. Ces problèmes d'ordonnancement sont très semblables entre les usines de production de masse et dans les usines de production au plus juste. Ce sont les problèmes qui vont être abordés dans cette thèse. Mais avant de les décrire un peu plus précisément, il est important de comprendre le fonctionnement de ces lignes d'assemblage.

## 3. Lignes de production

Une usine de production automobile est un ensemble de lignes de production qui sont peu flexibles et très sensibles aux aléas de part la nature des moyens de convoyage. Le processus de fabrication enchaîné est composé de 3 ateliers consécutifs : l'atelier ferrage, l'atelier peinture et l'atelier montage.

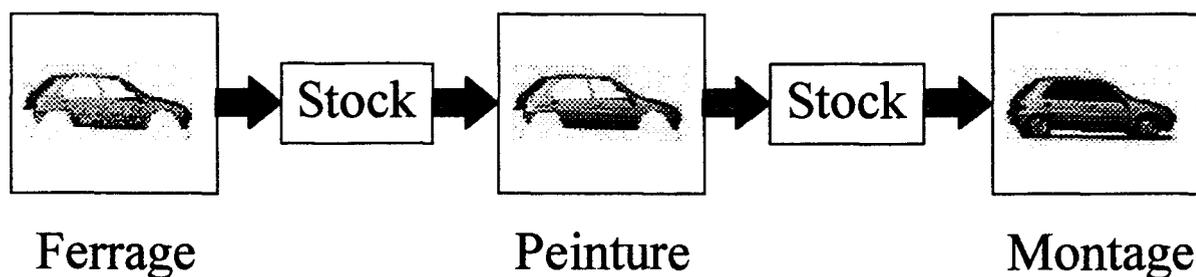


Figure I.3 - Flux global d'une usine de production automobile

L'atelier ferrage fabrique la caisse de la voiture par soudure de tôles embouties. C'est un atelier fortement automatisé. Cette caisse brute est ensuite envoyée dans l'atelier peinture où elle est traitée (anticorrosion, étanchéité, ...) puis peinte. Une fois peinte, l'atelier de montage effectue l'assemblage des pièces mécaniques (bloc moto-propulseur, ...) et l'habillage du véhicule (sellerie, câblerie, ...) pour obtenir un produit fini. Ce dernier atelier est celui où la valeur ajoutée est la plus forte et où les contraintes (en particulier logistiques) sont les plus fortes.

Nous allons présenter les caractéristiques des lignes de production. Dans un premier temps, nous allons décrire les systèmes de transports utilisés pour expliquer les problèmes de flexibilité. Dans un deuxième temps, les différents ateliers seront analysés pour comprendre leur fonctionnement, l'impact que peuvent avoir sur eux les objectifs économiques et les perturbations du flux rendant leur fonctionnement irrégulier. Pour terminer, nous étudierons les différentes typologies de stocks pouvant être rencontrés dans des usines pour comprendre leur rôle dans la flexibilité du flux.

Pour avoir un exemple précis d'une usine automobile, on pourra se reporter à la description de l'usine Citroën d'Aulnay faite par Guerre-Chaley dans sa thèse [GUE 95]. Il analyse le fonctionnement de chacun de ses ateliers, ainsi que leurs spécificités (en particulier celles des flux) et l'ensemble des opérations effectuées sur la ligne.

### **3.1. Système de transport**

Les véhicules à produire sont des produits lourds et volumineux (4 mètres de long, une tonne), de plus ils ne sont pas faciles à manipuler car ils sont fragiles (les caisses sont peintes). L'intérêt d'un convoyeur est le nombre limité de manipulations qu'il impose à l'ensemble du processus de fabrication (un accrochage et un décrochage). Il existe trois types de transport possibles permettant d'assurer le déplacement de la caisse à travers l'usine.

Le premier type est le transport continu : les véhicules se déplacent tous de façon synchronisée mais à une vitesse constante. Les opérateurs et les machines doivent suivre le mouvement de la ligne pour effectuer leurs tâches. Le cycle de travail est donc composé de deux temps différents. Le premier consiste à se déplacer dans le sens de la ligne pour effectuer le travail, et le deuxième à remonter la ligne sans rien faire pour se repositionner en vue du produit suivant.

Le deuxième type est le transport synchrone : tous les véhicules se déplacent en même temps et s'arrêtent à un poste de travail. Les travaux sont effectués à un poste entre deux déplacements d'un produit. Les hommes et les moyens de production ne se déplacent plus.

Le dernier type est le transport asynchrone : dès qu'une opération est terminée sur un produit celui-ci est convoyé vers le poste suivant indépendamment des autres produits. Les postes de travail doivent être indépendants et avoir des poumons entre eux pour masquer les différences de temps de travail.

Nous voyons donc que dans tous les cas les systèmes de transports sont rigides. En effet les véhicules ne peuvent se dépasser sur un tronçon de part la topologie même du système de production. Les seules flexibilités dans le flux apparaissent au niveau des aiguillages divergents et convergents existant dans les ateliers, ou au niveau du parallélisme entre deux lignes de production ou au niveau des stocks intermédiaires de production. Il apparaît donc très important que sur un même segment l'ordre des véhicules soit choisi avec précision car il ne pourra pas évoluer.

### **3.2. Atelier ferrage**

Nous ne nous attarderons pas sur cet atelier. En effet, il est fortement automatisé et le processus est robuste. Peu d'aléas existent et la diversité du produit à ce niveau est faible. Les contraintes de fonctionnement de cet atelier sont en général des contraintes de formes de véhicules et de changements de configuration des presses effectuant l'emboutissage des tôles et des robots les soudant.

### **3.3. Atelier peinture**

En entrant dans l'atelier peinture, les caisses sont dites en blanc (c'est une armature de tôles non traitées). Le traitement dans cet atelier ne consiste pas uniquement à peindre le véhicule mais aussi à effectuer des traitements de surface pour que la carrosserie du véhicule résiste à différents types d'agressions : chimiques (sel, pluie, carburants, ...) et mécaniques (gravillons, ...). Le traitement est effectué en plusieurs étapes et à différents postes dans l'atelier de peinture. Tout d'abord, la tôle est traitée par pulvérisation (c'est la phosphatation). Puis la caisse est immergée pour recevoir une couche de cataphorèse. Ces deux traitements sont effectués sur tous les véhicules et protègent les tôles des agressions physiques. Ensuite, le véhicule subit une étape d'étanchéité qui consiste à déposer des joints de mastics. Cette étape peut introduire une diversité en fonction de la silhouette des véhicules (3 portes, 5 portes, break, berlines pour lesquels le travail n'est pas le même). Puis une couche d'apprêt est déposée

pour assurer une protection contre les agressions mécaniques que pourra subir le véhicule. Finalement la peinture est appliquée au véhicule sous forme d'une couche de laque ou d'une couche de base recouverte de vernis. C'est ici qu'apparaît la grande diversité existant dans l'atelier de peinture. Il faut en effet gérer plusieurs dizaines de couleurs différentes sur les véhicules. C'est à cet endroit que les gains d'un bon ordonnancement sont perceptibles. La dernière étape du parcours dans l'atelier peinture est le contrôle. Un homme s'assure de la bonne qualité de la peinture à ce point du flux et décide si le véhicule peut continuer vers l'atelier de montage ou si il doit être dirigé vers un atelier de retouches. Les retouches sont de deux types : partielles (si une pièce de la carrosserie doit être repeinte) ou totales (si elle doit être poncée pour être entièrement repeinte). Le flux dans l'atelier peinture devient complexe à cet endroit. Les charges de travail sur les véhicules dépendent du type de retouche. Elles ne peuvent donc pas être déterminées à l'avance. Des dysfonctionnements existent et créent donc des perturbations plus ou moins importantes dans le flux principal. Le taux de retouches est d'environ 20%. Guerre-Chaley [GUE 95] donne un exemple pour l'usine d'Aulnay : 14 à 21 % de retouches réparties en deux tiers de retouches partielles plus ou moins longues et un tiers de retouches totales (ou revoilages).

Le séquençement pour l'atelier de peinture doit permettre de réduire les coûts au niveau de la peinture elle-même. En effet, le changement de teinte (rinçage des pistolets, ou des autres moyens) coûte cher. La perte de matière (peinture et solvant) et leur retraitement coûtent environ 100 francs. Pour illustrer les gains possibles, prenons l'exemple d'une usine fabriquant 200 000 véhicules et dont le coût de changement de teinte est de 100 francs et étudions l'impact de la longueur des convois de véhicules ayant la même teinte<sup>5</sup>:

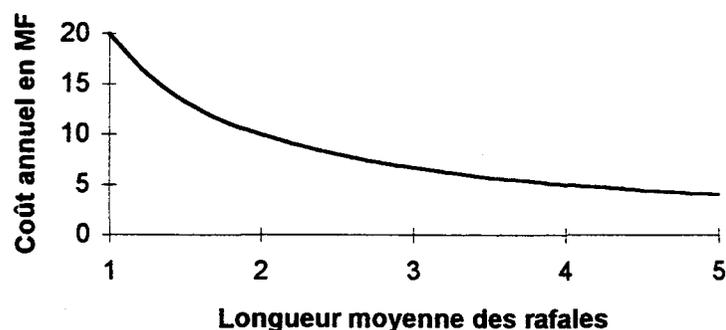


Figure 1.4 - Coûts des changements de teintes en fonction de la longueur des rafales.

<sup>5</sup> appelé aussi "rafales"

Il est évident que plus les rafales sont grandes plus le gain est important. Mais il faut néanmoins être capable de limiter cette taille car l'acuité du contrôle du censeur diminue si les rafales sont trop longues. La qualité de l'enlèvement des véhicules avant la peinture a donc un impact direct sur les gains qui peuvent être réalisés dans cet atelier.

Néanmoins la taille moyenne des rafales (ou le nombre de changements de teinte) n'est pas la seule règle qu'il est nécessaire de suivre pour générer un bon ordonnancement. Il faut tenir compte aussi des aléas de production (une teinte est indisponible) ou des configurations d'atelier : certains types de véhicules ne peuvent être fabriqués que sur une des lignes de l'usine. De manière plus importante, il faut remarquer que l'atelier peinture n'est pas le dernier atelier de l'usine. L'atelier suivant, celui de montage est plus difficile à piloter car il gère plus de diversité. Il ne faut donc pas, dans l'optique d'optimiser localement l'atelier peinture, hypothéquer le pilotage de l'atelier montage à la fois pour ses flux constituants et pour le séquençement que l'on sera amené à faire pour le stock suivant.

### **3.4. Atelier montage**

Les caisses arrivant de l'atelier peinture sont donc prêtes à être montées. La particularité des ateliers de montage d'usine automobile est le faible taux d'automatisation. De plus, cet atelier concentre le plus grand nombre de diversité dans les produits et les plus grands problèmes de logistique au niveau des approvisionnements de pièces fabriquées chez les fournisseurs ou dans d'autres usines (moteurs, boîtes de vitesse, sièges, ...).

L'atelier montage se caractérise par un convoyeur à vitesse constante (c'est le premier système de transport décrit en 3.1). Il n'y a pas de zones de stockage intermédiaire. La durée de fabrication dépend donc uniquement de la longueur de la ligne et de la vitesse de celle-ci. De plus, la séquence de véhicules ne varie pas tout au long de la ligne, donc l'ordre de passage des véhicules à chaque poste de travail est le même. Il faut donc que la charge de travail d'une séquence soit équilibrée pour l'ensemble des postes.

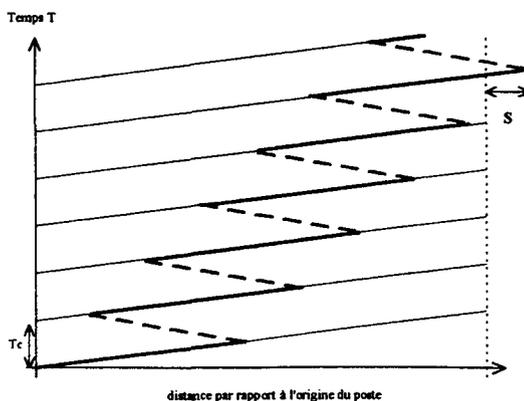
L'atelier de montage est composé de plusieurs centaines de postes de travail. Chaque poste correspond à une zone sur la ligne (délimitée par une limite amont et une limite aval) dans laquelle l'opérateur doit effectuer sa tâche. Cette zone peut être ouverte ou fermée suivant le type de tâche, le matériel utilisé et la succession des opérations sur la ligne. Les termes "ouverte" ou "fermée" correspondent à la possibilité pour l'opérateur de franchir la limite du poste pour travailler sans gêner les autres opérateurs. Cette notion peut être utilisée à droite (limite aval) ou à gauche (limite amont).

Les travaux à effectuer sur un véhicule sont décomposés en tâches élémentaires qui sont ensuite réparties entre les différents postes. Elles peuvent être de deux types différents. Dans le premier type, les tâches sont celles à effectuer sur tous les véhicules et dont la durée est la même quelque soit le véhicule. Ces tâches ne sont donc pas un problème pour l'équilibrage de la ligne. Il suffit que le poste soit bien dimensionné pour le taux de production. Dans le deuxième type, les tâches sont celles qui sont variables en fonction des véhicules en particulier au niveau de la durée de l'opération de montage. Delaval [DEL 97] propose de les séparer en deux groupes : celles à durée variable et celles à présence variable. Mais on peut considérer que le deuxième groupe est un cas particulier du premier.

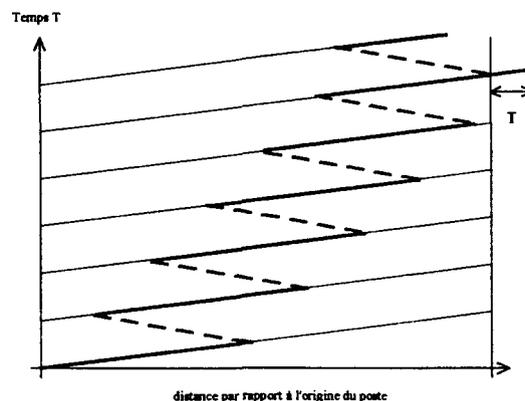
Si la durée de montage d'un véhicule est la même quelque soit son type, la somme des durées des opérations pour le monter n'est pas la même entre deux véhicules. Un véhicule haut de gamme avec tous les équipements demandera plus de travail qu'un véhicule bas de gamme avec peu d'équipement. La charge de travail le long de la ligne va donc varier en fonction du véhicule à monter. Pour des raisons économiques, il faut essayer de lisser la charge de travail pour éviter des pertes d'efficacité de l'outil de production : en effet il ne doit pas être surdimensionné sinon il fonctionnera à vide, mais il ne doit pas être sousdimensionné sinon il ne pourra pas fabriquer le volume de production souhaité et commandé.

Thomopoulos [THO 67] a défini quatre sortes de pertes d'efficacité :

- La surcharge de travail quand l'opérateur doit sortir de son poste pour finir son travail.
- Le travail non terminé quand l'opérateur ne peut pas terminer son travail.
- L'inactivité quand l'opérateur attend un véhicule pour travailler.
- Le manque de travail quand l'opérateur doit sortir de son poste pour travailler.



*Surcharge de travail*



*Travail non terminé*

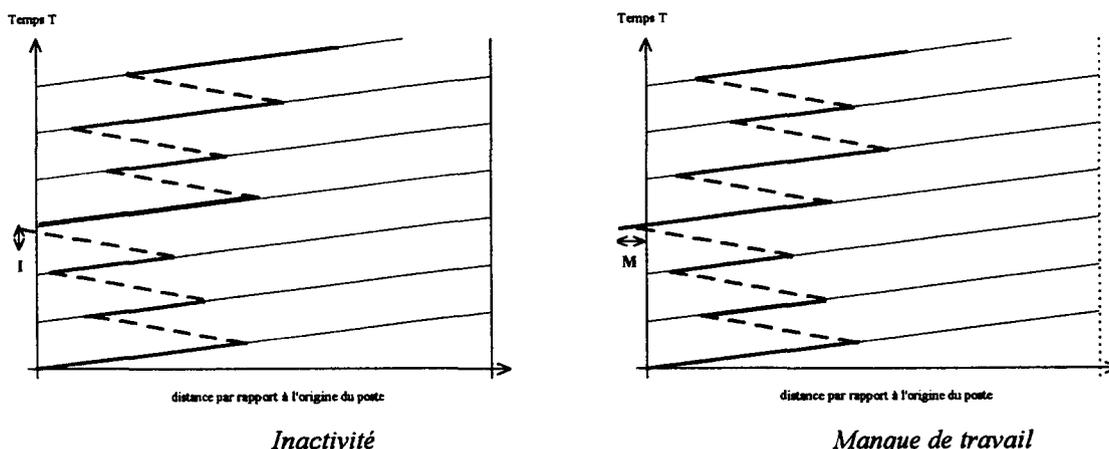


Figure 1.5 - Tableaux d'activité (perte d'efficacité)

Ces tableaux d'activité illustrent la charge d'un opérateur sur son poste de travail. Partant du point de départ (limite amont de son poste), il va effectuer son travail sur le premier véhicule (trait continu), puis il va remonter la ligne pour effectuer son travail sur le véhicule suivant (trait pointillé). Il va ainsi alterner des phases de travail sur des véhicules plus ou moins longues et des périodes où il va remonter le flux (celles ci sont toujours les mêmes car la vitesse du convoyeur est constante). Si la charge de travail est trop faible, l'opérateur commencera toujours son travail près de la limite amont. En revanche si celle-ci est trop importante, il travaillera toujours près de la limite aval en ayant peu de marge pour effectuer sa tâche.

Les limites du poste de travail peuvent être ouvertes ou fermées. Sur un poste ouvert (limites en pointillé), l'opérateur peut commencer ou terminer son travail sur le poste voisin. Sur un poste fermé (limites en trait continu), les opérateurs ne peuvent pas empiéter sur le poste adjacent.

La première contrainte que devra respecter le problème de séquençement est donc une contrainte de faisabilité du travail sur chaque poste donc de charge de travail.

La deuxième contrainte rencontrée dans ces ateliers est liée d'une part à la diversité des constituants et à leur nombre et d'autre part au caractère réactif des systèmes de production juste à temps.

En effet, il faut être d'une part capable de fabriquer le produit lui même (temps d'assemblage de produits à respecter) mais d'autre part il faut également être capable d'acheminer et de fabriquer en amont les constituants assemblés en dehors de la ligne principale. Cette dernière production doit, elle aussi, être optimisée pour minimiser les stocks internes ou externes, pour ne fabriquer que ce qu'il est nécessaire de fabriquer et pour le fabriquer au moment où l'on en a besoin.

### 3.5. Les stocks

Entre les divers ateliers d'une usine, des moyens de stockage intermédiaires existent. Ces stocks intermédiaires de production ont deux rôles principaux. Le premier consiste à réguler le flux en volume. Suite à des aléas de production ou à des rythmes de travail qui ne sont pas synchronisés, le flux pourrait se désamorcer entre deux ateliers successifs. Le deuxième rôle consiste à réguler le flux en qualité. Ils permettent de satisfaire les critères d'optimisation du flux en respectant les charges de travail par exemple. L'utilisation de ces stocks peut ainsi permettre l'optimisation des tailles des rafales peinture (comme le montre Guerre-Chaley dans sa thèse [GUE 95]), ou de lisser la charge des postes de travail [CAS 95].

Différents types de stocks existent. Mais les solutions matérielles peuvent se résumer à trois familles de stocks qui ont chacune leurs spécificités.

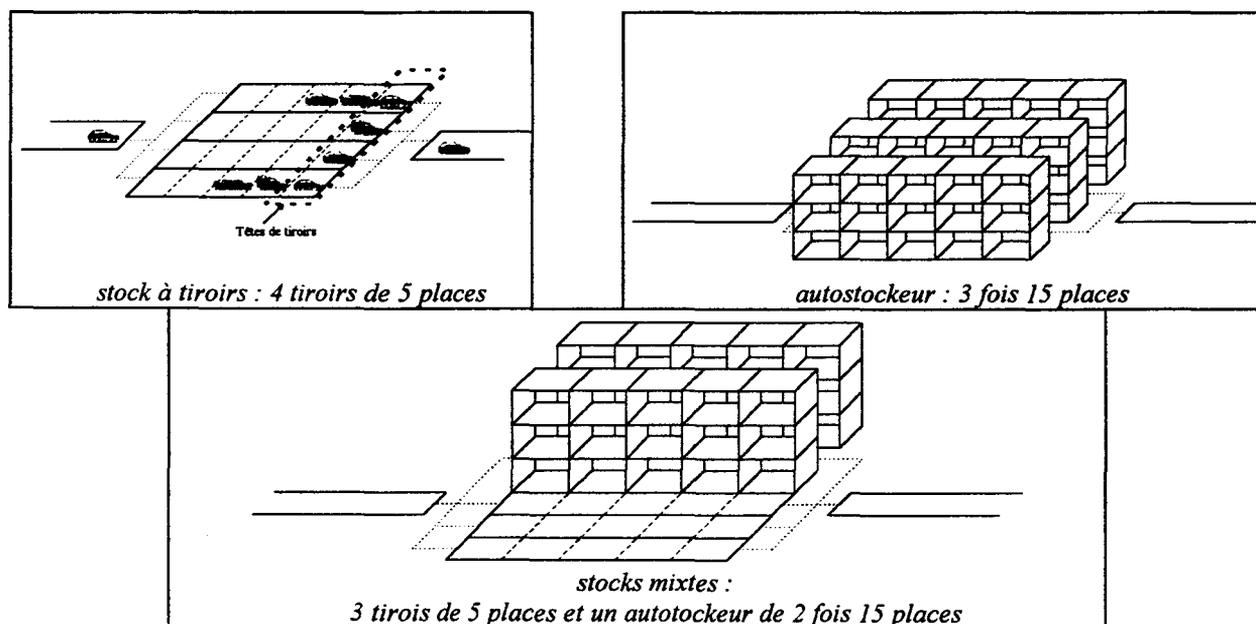


Figure I.6 - Les différentes architectures des stocks intermédiaires.

#### 3.5.1. Stocks à tiroirs

Le stock à tiroirs est composé de files d'attente en parallèles fonctionnant en FIFO<sup>6</sup>. Ce fonctionnement garantit un écoulement régulier du flux. En entrée du stock, le véhicule peut aller dans une des différentes files d'attente de ce stock (appelée aussi tiroir). Il ne pourra pas dépasser les véhicules se trouvant dans le même tiroir que lui mais pourra dépasser ou être dépassé par les véhicules des autres tiroirs. En sortie, le choix doit être fait entre les véhicules

<sup>6</sup> First In, First Out : Premier Entré, Premier Sorti

se trouvant en tête de tiroirs : ce sont les véhicules qui peuvent être manipulés pour être enlignés.

Ce type de stock est, en général, robuste vis-à-vis des pannes. Mais sa capacité à réordonner est limitée vis-à-vis de l'ensemble des ordonnancements possibles. De plus il faut gérer deux types d'événements et de décisions :

- l'arrivée de véhicules en entrée de stock et leur orientation
- la demande de véhicule en sortie et le choix parmi les disponibles

Cela augmente la complexité des algorithmes de gestion de ce type de stocks car il faut optimiser l'entrée en fonction de la sortie.

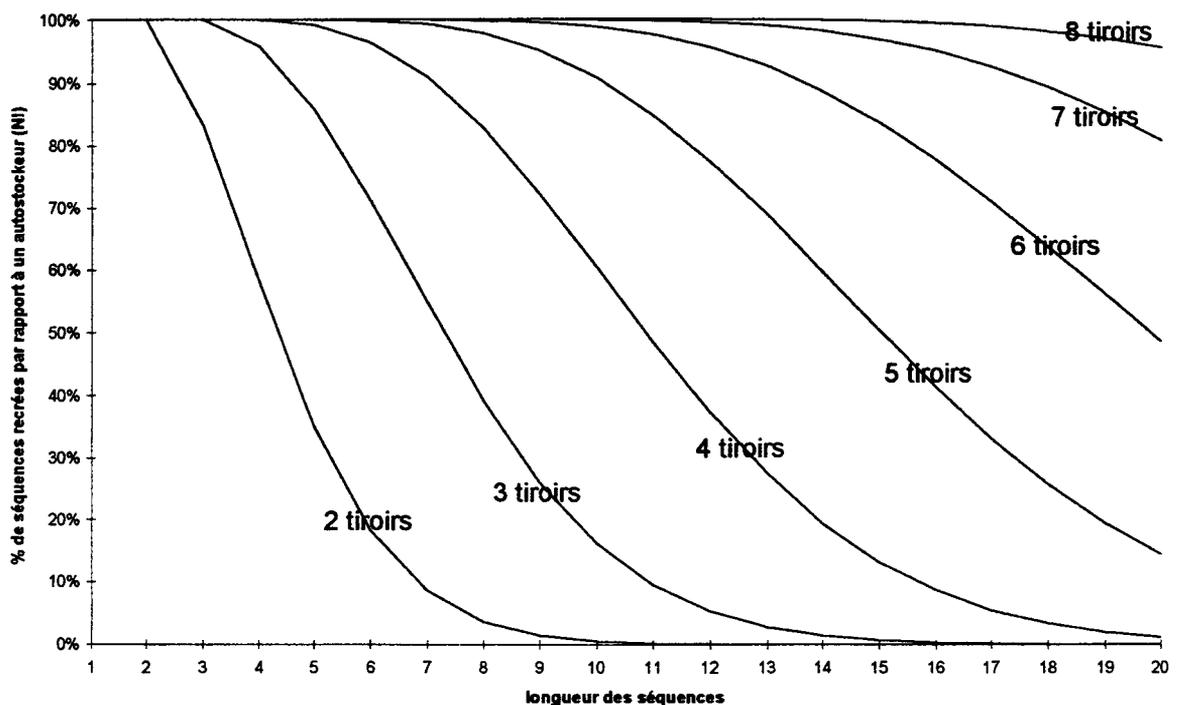


Figure 1.7 - Graphique de capacité de réordonnement d'un stock à tiroir en fonction du nombre de tiroirs et de la taille de la liste.

Dans le graphique ci-dessus, nous indiquons pour un nombre de tiroirs donné (de taille illimitée) et pour un nombre  $N$  de véhicules, le pourcentage de séquence que nous sommes capables de recréer (par rapport au  $N!$ ). La capacité de réordonnement augmente avec le nombre de tiroirs mais diminue très rapidement avec la taille de la liste que l'on souhaite réordonner. Un phénomène de seuil apparaît au delà duquel il n'est pas envisageable d'accéder à l'ensemble des combinaisons qui seraient accessibles sans les contraintes du moyen physique.

Il faut également prendre en compte l'occurrence d'apparition des aléas de fabrication. En effet, si un véhicule ne peut être engagé dans l'atelier suivant suite à un problème

d'approvisionnement, il faut pouvoir le retenir dans le stock. L'architecture à tiroirs permet de le faire à l'aide de boucles de retour permettant ainsi de vider les tiroirs de leurs véhicules "interdits"<sup>7</sup>. Néanmoins ce fonctionnement avec les boucles de retour peut rendre fortement complexe la gestion du stock et ne plus garantir l'écoulement régulier du flux à travers l'usine.

### **3.5.2. Autostockeurs (ou MGH)**

Un autostockeur est un grand hangar où tous les véhicules sont stockés dans des alvéoles indépendantes. L'intérêt d'un tel stock est sa capacité de réordonnancement total. En effet, en entrée les véhicules peuvent être rangés dans n'importe quelle alvéole vide et en sortie, n'importe quel véhicule peut être choisi. Il faut toutefois souligner quelques limites à cette description de fonctionnement. En effet, ce style de magasin est, en général, pour des questions de place occupée, extrêmement compact. Chaque alvéole est desservie par un transbordeur qui est capable d'atteindre des véhicules en hauteur. L'expérience montre que la disponibilité (pour des raisons de panne ou de maintenance) des autostockeurs n'est pas aussi importante que celle des tiroirs. Mais pour des raisons de simplification nous ferons l'hypothèse que tous les véhicules sont disponibles et que la topologie de l'autostockeur n'influe pas sur les capacités d'ordonnancement. Cette hypothèse n'est remise en question que dans deux cas. Soit le stock est saturé et le temps de cycle ne peut être garanti par les transbordeurs, soit un transbordeur tombe en panne et une partie des véhicules est temporairement inaccessible. Ces deux problèmes peuvent être masqués par une politique d'entrée permettant une répartition des véhicules entre les transbordeurs et leurs travées.

Les difficultés d'ordonnancement du stock à tiroirs sont gommées. En effet, il ne reste à gérer que la sortie de l'autostockeur. De plus en sortie, les choix sont plus nombreux et ils permettent donc a priori un ordonnancement plus proche de l'optimum.

### **3.5.3. Stocks mixtes**

D'autres formes de stocks existent, mais sont dans la plupart des cas des combinaisons des topologies décrites précédemment. On peut avoir par exemple en parallèle un autostockeur de grande contenance et des petits tiroirs pour augmenter la capacité globale du stock.

### **3.5.4. Comparaisons**

La capacité de réordonnancement d'un stock à tiroirs est plus faible.

---

<sup>7</sup> véhicules ne pouvant être engagés pour des raisons d'approvisionnement.

Un autostockeur contient  $N$  véhicules, un stock à  $T$  tiroirs contient aussi  $N$  véhicules répartis entre les  $T$  tiroirs ( $N_i$  dans le tiroir  $i$ ). Le tableau ci-dessous indique le nombre d'ordonnements différents possibles et illustre le résultat pour  $N=12$ ,  $T=4$  et  $N_i=3$

Autostockeur	Stock à tiroirs	Comparaison
$N!$	$\frac{N!}{\prod_{i=1}^T N_i!}$	$\prod_{i=1}^T N_i!$
$12! \approx 479 \cdot 10^6$	$\frac{12!}{(3!)^4} \approx 370000$	1296 fois plus ou 0.07%

Figure I.8 - Tableau comparatif des stocks en statique.

Ces résultats correspondent à un ordonnancement statique en sortie du stock.

Dans le cas d'un ordonnancement dynamique du stock (gestion en entrée et en sortie du stock), la différence entre les deux stocks diminue très sensiblement.

En effet, dans l'exemple précédent, si la taille des tiroirs est suffisamment importante (au moins 9 places), la Figure I.7 nous indique que, pour 4 tiroirs et 12 véhicules, nous sommes capables de recréer 37% des  $12!$  séquences possibles (c'est-à-dire 178 098 474 séquences différentes). Il faut comparer ce résultat avec les 0.07 % de l'ordonnement en statique (cf. Figure I.8) pour comprendre qu'un ordonnancement en entrée de stocks est indispensable pour bien gérer des stocks à tiroirs.

Ce résultat n'est valable que pour une liste de petite taille (12 véhicules seulement). La capacité d'ordonnement diminue rapidement (cf. Figure I.7). Il donne néanmoins une idée de l'importance de la gestion de l'entrée en même temps que la sortie.

Il est également possible de comparer le comportement limite des stocks à tiroirs et des autostockeurs en indiquant qu'un autostockeur à  $N$  places a le même comportement qu'un stock de  $N$  tiroirs de 1 place.

#### **4. Lissage des charges sur les lignes multiproduits à forte diversité : équilibrage et séquençement**

##### **4.1. Optimisation du fonctionnement d'une ligne d'assemblage**

Comme nous avons pu le voir dans le chapitre précédent, les moyens physiques impliquent des contraintes très fortes sur l'écoulement du flux dans l'ensemble de l'usine. D'autre part de nombreuses contraintes sont liées à la fois à l'importance des moyens mis en oeuvre dans une

usine automobile (moyens humains et moyens matériels) et à la fois aux approvisionnements importants et diversifiés liés à l'évolution de la demande dans le monde automobile.

Okamura et Yamashima indiquent dans leur article [OKA 79] que l'utilisation efficace d'une ligne d'assemblage à modèles mélangés nécessite de résoudre deux problèmes différents mais liés :

1. répartition de la charge entre les opérateurs
2. séquençement des produits

Miltenburg découpe le problème en quatre parties [MIL 89a] :

1. détermination des temps de cycle
2. détermination du nombre et de la séquence des postes de travail sur la ligne
3. équilibrage de la ligne
4. détermination de l'ordonnancement des produits sur la ligne

L'approche entre les deux est la même mais la première étape est décrite plus finement, chez Miltenburg, avec des critères liés à la conception de la ligne et des paramètres qui peuvent être réglés pendant le fonctionnement de la ligne.

Les deux premiers problèmes remontent à la conception de la ligne d'assemblage elle-même. Pour cela, il faut agencer les postes pour être capable d'effectuer l'ensemble des tâches sur l'ensemble des modèles de produits en garantissant un temps de cycle donné. La solution choisie pour ces problèmes va directement influencer les choix pour les deux suivants car les capacités de production vont être limitées par les capacités physiques. En effet, le dimensionnement de postes sur des installations lourdes comme le remplissage des climatisations peut limiter le nombre de véhicules ayant cet équipement sur la ligne. Un choix au niveau de la conception de la ligne limitera donc les capacités de production lors du fonctionnement nominal de la ligne.

Dans notre approche, nous considérons que ces deux premiers problèmes ont été résolus et que nous en connaissons les résultats. (temps de cycle fixé, contraintes matérielles précisées).

Les deux derniers problèmes à résoudre sont des problèmes de planification. L'équilibrage est une planification à long terme alors que le séquençement est une planification à court terme. L'intérêt d'une bonne planification est l'utilisation optimale des ressources mises à disposition sur la ligne. Le terme optimal est lié à la capacité à fabriquer tous les véhicules qui seront commandés par les clients en engageant le minimum de ressources. En reprenant la classification de Sethi [SET 90], cela correspond à deux des flexibilités globales :

- la flexibilité de production qui correspond à l'ensemble des pièces que le système pourrait produire sans investissement matériel important, et ceci indépendamment des temps de préparation.
- la flexibilité de marché qui est associée à la facilité d'un système à s'adapter à un environnement de marché variable.

#### 4.2. Optimisation à moyen terme (Equilibrage)

L'équilibrage est l'affectation des opérations aux ressources et aux postes de la ligne. Cela consiste à déterminer **qui** réalise l'opération (choix de la ressource) et **où** elle sera réalisée (choix du poste). Réduire les écarts de charge entre les postes permet d'optimiser le taux d'engagement et l'utilisation des ressources.

Un mauvais équilibrage de la ligne conduit à une sous-utilisation des opérateurs et à la création de goulots d'étranglement inutiles et inappropriés.

Ghosh [GHO 89] propose une classification et une étude bibliographique des différentes approches que l'on peut trouver dans la littérature.

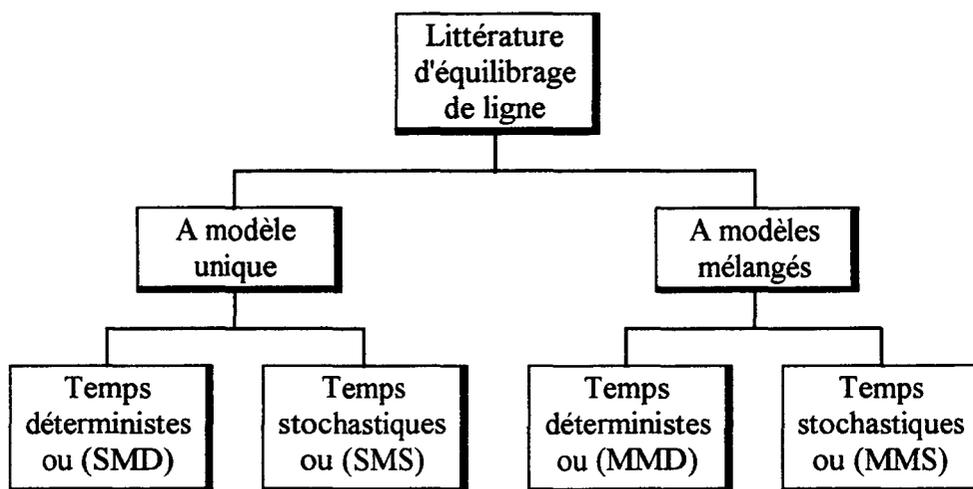


Figure 1.9 - Classification des problèmes d'équilibrage [GHO 89]

La périodicité de ce problème dépend de la production de la ligne et des modifications apportées à celle-ci. Tout produit nouveau fabriqué sur la ligne nécessitera une nouvelle répartition des tâches pour ce véhicule sur les postes et donc de résoudre une nouvelle occurrence de ce problème. De même toute modification apportée à un poste obligera à résoudre ce problème pour optimiser le processus et limiter les goulots d'étranglement à ce nouveau poste. D'autre part, l'équilibrage est également effectué en fonction des prévisions commerciales sur une période envisagée. En effet l'équilibrage de certains postes (comme la climatisation) dépendent fortement de la période et la demande est fluctuante.

Notre étude s'intéresse au séquençement des produits sur la ligne de production. Il est donc nécessaire que le problème d'équilibrage soit résolu pour permettre une bonne utilisation des ressources. Il est évident que notre problème de séquençement aura des contraintes dérivant directement de l'équilibrage et des goulots d'étranglement (postes critiques) que l'équilibrage aura identifié.

### **4.3. Optimisation à court terme**

Le problème du séquençement des ordres de fabrication sur la ligne est au centre de notre étude. Le séquençement consiste à ordonnancer les produits à fabriquer pour déterminer l'ordre de lancement des véhicules (dans le cadre d'une cadence de lancement des véhicules constante) voire la date de lancement (dans le cadre d'une cadence de lancement des véhicules variable). Ce paragraphe va décrire rapidement les différentes classes de contraintes prises en compte dans ce problème.

En effet, comme nous avons pu le voir dans les chapitres précédents, la concurrence sur le marché automobile a amené les constructeurs à élargir leur offre au niveau des gammes et des équipements, à réduire les coûts de production et les délais, et à augmenter la qualité des véhicules. La diversité des produits sur les lignes de production et la politique d'organisation et d'équilibrage de la ligne ont une grande influence sur la faisabilité d'une séquence de produits engagés et sur la charge de travail des ressources utilisées.

L'ordonnancement s'effectue à deux niveaux. En effet des ordonnancements prévisionnels sont nécessaires pour l'approvisionnement en flux tendus de certains sous-ensembles (bloc moto-propulseur en particulier). Et d'autre part il est nécessaire d'équilibrer au mieux les charges des différents postes des différents ateliers. Les ordonnancements effectués doivent être le plus robuste possible car si la production "au plus juste" tend à supprimer les aléas d'approvisionnement et de fabrication, il en existe toujours en pratique.

Les objectifs du séquençement sont de différents types et ont des priorités plus ou moins importantes pour pouvoir garantir à la fois la satisfaction du client (au niveau du délai et de la qualité), et la faisabilité de la production à moindre coût. Il faut pour cela :

- respecter une position ou date de référence en minimisant les écarts entre la référence et la réalité.
- respecter les capacités de production
- lisser la charge de production
- minimiser les coûts de changement d'outils

Comby dans sa thèse [COM 96] prend en compte d'autres règles comme :

- sélectionner en priorité des produits ayant un critère particulier (véhicules prioritaires)
- minimiser l'encours de production
- gérer la charge au niveau de chaque ressource.

Cette approche, même si elle possède des aspects intéressants, n'a pas été retenue car elle demande un très grand effort de modélisation pour une précision des résultats trop importante; de plus, elle exclue l'intervention humaine de la boucle de décision.

Chew [CHE 91] propose également des règles supplémentaires liées aux changements de production, comme la prise en compte des nouveaux véhicules, les préséries et les fins de séries.

#### ***4.3.1. Les différents types de contraintes***

Revenons sur les différents types de contraintes en reprenant leur ordre de priorité.

##### **4.3.1.1. Contraintes de production au plus juste**

Dans un système juste à temps, les taux de consommation de chaque pièce (produits finis, sous ensembles, composants et matières premières [MIL 89b]) doivent être maintenus constants. En effet, les flux des constituants sont tirés vers le flux principal et il est nécessaire de garder pour chacun de ces flux annexes une charge constante. Donc pour cela, il faut que chaque type de véhicule soit réparti de manière régulière dans la séquence. (cf. chapitre 3.4)

La deuxième contrainte liée à cette conception de la production est de garantir un délai de livraison. Chaque produit doit donc pouvoir être fabriqué dans un intervalle de temps donné. Pour cela on peut déterminer une date au plus tôt et une date au plus tard que l'enlèvement du produit devra garantir. Cette notion de fenêtre d'enlèvement est également nécessaire pour gérer au plus juste les flux hors usine comme les approvisionnements en moteurs et boîtes de vitesse. Cette notion de fenêtre peut être ramenée à la présence dans une journée de production de l'ordre de fabrication associé [GUE 95], ou dans une plage horaire plus restreinte comme le partage en équipe de production (demi journée de production) [COM 96], voire à une date précise pour correspondre avec le flux à synchroniser.

##### **4.3.1.2. Respecter les capacités de production**

Cette contrainte est typiquement rencontrée dans l'atelier de montage. En effet pour éviter des surcharges sur certaines ressources, on est capable d'évaluer la quantité de véhicules ayant

un équipement<sup>8</sup> qui peut être assemblée sur la ligne. Cette quantité est souvent exprimée sous forme de ratios : pour des raisons de capacité et d'équilibrage de la charge des opérateurs, les toits ouvrants ne peuvent être par exemple montés que sur seulement 25% des véhicules ; cela veut dire que entre deux véhicules ayant un toit ouvrant, il en faut trois sans cet équipement. Le rôle du séquençement est donc de répartir les véhicules dit "lourds"<sup>9</sup> sur l'ensemble de la séquence. Dans la deuxième partie nous verrons la manière de modéliser ces contraintes de production pour prendre en compte les pertes d'efficacité ou les non respects de contraintes de ratios.

Cette modélisation de contraintes de ratios est étroitement liée à la contrainte suivante ("lisser la charge de production"). Elle peut même paraître redondante, mais les deux approches sont sensiblement différentes.

Cette limite de ratios correspond en général à des niveaux de production qui ne peuvent être dépassés. En effet, pour des raisons économiques, l'équilibrage a dimensionné la capacité de la ligne en fonction des prévisions commerciales pour répondre à une demande moyenne : le besoin instantané doit donc être limité (au niveau de la fabrication) pour éviter de trop fortes fluctuations de charges qui nécessiteraient un surdimensionnement de l'outil de production.

#### 4.3.1.3. Lisser la charge de production

Le respect des ratios est une contrainte forte car elle est liée à une faisabilité de la production. Le lissage de la charge consiste à équilibrer au mieux le temps de travail des opérateurs pour permettre un lissage des taux de consommation de pièces et éviter que des goulots d'étranglement se forment de manière excessive. La complémentarité de cette contrainte avec la précédente et sa modélisation seront décrites en détail dans la deuxième partie.

#### 4.3.1.4. Minimiser les coûts de changement d'outils

Les contraintes de groupement permettent de réduire le nombre de changements d'outils. Ces contraintes se retrouvent dans les ateliers de peinture où l'objectif est de réduire les purges des pistolets en formant des rafales de même teinte. Elle peut également se retrouver dans l'atelier ferrage pour diminuer les temps de reconfiguration des presses.

L'objectif global est en général de réduire les coûts dus aux changements, en diminuant la variance de la séquence du flux.

---

<sup>8</sup> ou critère

<sup>9</sup> avec beaucoup de critères

### 4.3.2. *Compromis*

L'objectif du séquençement va être d'optimiser les différentes contraintes avec plusieurs critères. Le problème qui peut en résulter est un conflit entre différentes contraintes ou les différents critères.

Nous avons vu que la contrainte de changement d'outils tend à regrouper les véhicules ayant le même critère, à la différence des contraintes de ratios ou de lissage de charge qui ont tendance à espacer les produits ayant les mêmes critères. Il est donc évident qu'un critère donné ne peut être optimisé suivant deux contraintes antagonistes comme celles décrites ici.

Ces contradictions sont en général faciles à détecter. Mais des problèmes moins évidents peuvent exister. Certains critères peuvent être croisés. Par croisé, on entend le fait qu'un produit ne possède pas qu'un seul critère à optimiser mais un ensemble. Prenons un exemple pour illustrer ce problème : supposons que les clients qui commandent des voitures à caractère sportif (un bloc moto-propulseur difficile à monter nécessite une contrainte de lissage de charge) choisissent un coloris rouge. L'optimisation de l'atelier peinture va entraîner des rafales de véhicules rouges avec un moteur spécifique qui ne pourra pas être optimisé pour le montage. De même on peut voir apparaître des critères uniquement sur certains types de véhicules (les climatisations sont en général commandées sur des véhicules hauts de gammes) qui possèdent déjà de nombreuses contraintes à optimiser.

Pour résoudre ce problème, il est donc nécessaire d'arriver à un compromis entre l'ensemble des contraintes et des critères. En général, les contraintes de ratios sont privilégiées par rapport aux contraintes de lissage de flux ou de changement d'outil car elles sont liées directement à la faisabilité de la séquence de produits et ont donc un impact direct sur la qualité de la production. Les deux autres types de contraintes sont là pour optimiser le fonctionnement de l'outil de production en réduisant les coûts directs (rafales de teintes pour la peinture) ou induits (diminution des stocks pour les flux constituants). Les contraintes de dates sont en général indépendantes des autres si une planification à moyen terme a été faite (le volume d'ordre de fabrication journalier doit respecter les capacités de production de l'usine en terme de critères).

Les contraintes des ateliers de montage et de peinture ne sont pas toujours compatibles (et peuvent être antagonistes comme dans l'exemple précédent), il faut donc également recourir à un compromis. Mais pour atteindre cet objectif, il existe deux types d'optimisation possibles :

- résolution du problème en statique (appelé "Séquençement" par la suite) : il s'agit de trouver un compromis entre les différentes contraintes des différents ateliers, avec une vue

globale de l'ensemble du flux de production et de l'ensemble des ordres de fabrication sur une période donnée (en général un jour ou une semaine).

- résolution du problème en dynamique (appelé "Cadencement" par la suite) : il s'agit pour un stock donné situé entre deux ateliers d'optimiser le fonctionnement de l'atelier suivant en prenant en compte la topologie aval de l'usine. Mais cette optimisation n'a qu'une vue partielle du flux de production (la partie aval par rapport au stock) et une vue locale des ordres de fabrication (le contenu du stock et les pousmons d'accès), tout en ayant des informations sur le volume global de la production sur une longue période (volumes prévus pour la journée en général).

Les deux approches ont chacune leur intérêt. La deuxième est cependant plus réactive et permet une souplesse plus importante et une robustesse aux aléas plus forte. En revanche, elle est plus compliquée à mettre en place car elle doit prendre des décisions pour chaque événement lié au flux (véhicule arrivant en entrée, demande de sortie de véhicule, apparition d'aléas, ...).

#### **4.4. Problème statique : séquençement**

Cette approche, si elle est retenue, doit prendre en compte plusieurs contraintes qui peuvent être antagonistes. Il faut donc réaliser une optimisation globale et multicritère de la séquence. Et il est nécessaire d'établir des règles de priorité entre les ateliers, entre les contraintes et entre les critères. Cette optimisation peut être effectuée simultanément entre toutes les contraintes comme l'a fait Bolat [BOL 94] ou par étapes en prenant d'abord les contraintes les plus importantes comme l'ont fait Burns [BUR 87] et Chew [CHE 91]. Burns et Daganzo proposent une succession de regroupements élémentaires permettant de respecter les contraintes de groupement puis dans chaque groupe les produits sont espacés régulièrement.

Sur la Figure I.10, nous illustrons la méthode de Burns. A partir d'une séquence de départ, deux approches sont possibles : on peut choisir soit de regrouper suivant le critère '1' (la couleur) d'abord, soit de regrouper suivant le critère '2' (la forme). Dans un premier temps, on montre le résultat que l'on obtient en regroupant d'abord suivant la couleur puis suivant la forme. Dans un deuxième temps, le résultat que l'on obtient en regroupant d'abord suivant la forme puis suivant la couleur est décrit. Les différences sont sensibles et le premier critère optimisé est avantagé. Finalement, à partir de l'optimisation de la séquence suivant la couleur puis la forme, on cherche à respecter la contrainte d'espacement sur le critère '3' sans dégrader les deux précédentes optimisations. On s'aperçoit que l'on ne peut pas nécessairement garantir un espacement régulier.

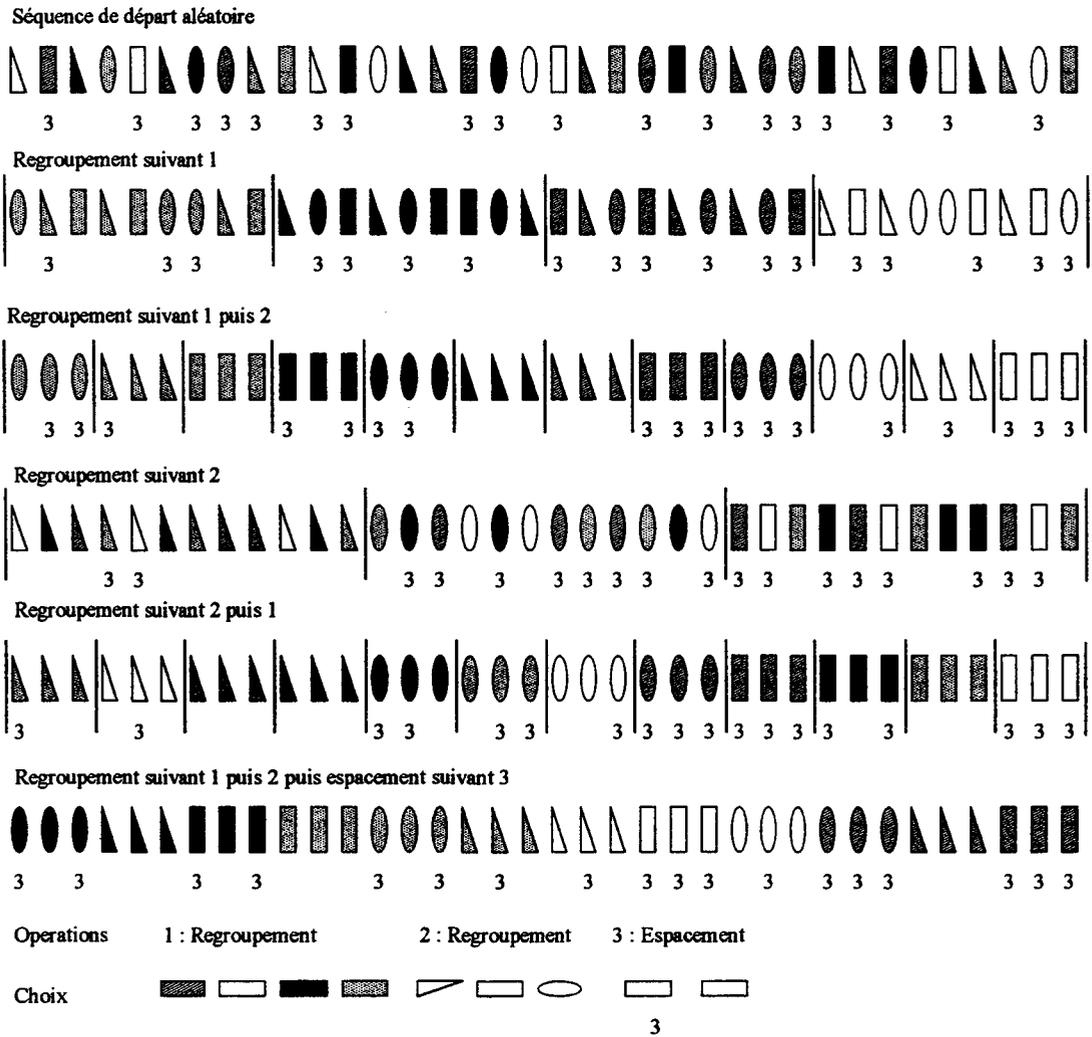


Figure I.10 - Technique de regroupement et d'espacement pour Burns

Cette approche est intéressante pour la partie regroupement mais devient beaucoup plus délicate à mettre en place pour les espacements car les ratios entre critères peuvent varier énormément.

Cette planification très fine permet cependant de prévoir les dates de livraison et de générer des flux d'approvisionnement pour la période considérée.

Dans la troisième partie, nous étudierons cette approche plus en détail et nous proposerons une méthode de résolution qui essaye de concilier qualité du résultat et temps de réponse de l'algorithme. Nous appellerons ce problème séquencement par la suite à la différence de l'approche suivante car nous avons une vision globale du problème qui est plus proche des études traditionnelles.

#### **4.5. Problème dynamique : cadencement**

La présence de stocks intermédiaires de production introduit une flexibilité supplémentaire dans les usines. En effet ils ont une capacité de réordonnancement plus ou moins importante en fonction de leur type et de leur taille (cf. chapitre 3.5). Au niveau de notre problème, leur intérêt peut être double. Dans un premier temps, le problème dynamique, que nous appellerons cadencement par la suite, consiste à enligner les véhicules en temps réel en respectant les contraintes de l'atelier aval et des ateliers suivants. Le terme temps réel correspond au temps de réponse des algorithmes de décision vis-à-vis du temps de cycle et de la cadence de lancement de la ligne.

Les différences notables entre les problèmes de cadencement et de séquençement sont au nombre de quatre :

- pour le problème du cadencement, l'algorithme a une vision locale du flux mais doit toujours effectuer une optimisation globale.
- le problème du cadencement est dynamique : l'ensemble des véhicules à ordonnancer varie avec les événements liés au flux.
- pour le problème du cadencement, plus nous nous positionnons en aval sur le flux de l'usine, moins le nombre de contraintes à prendre en compte est important.
- pour le problème du cadencement, l'architecture du stock doit être prise en compte et influe sur la capacité de réordonnancement.

#### **4.6. Autres stratégies de gestion.**

Jusqu'à présent, nous avons décrit une stratégie en flux poussé. Une approche en flux tiré existe également.

Chan [CHA 96] compare les performances des trois systèmes manufacturiers flowshop : poussé, tiré ou hybride. Mertins [MER 96] établit une comparaison entre trois systèmes de gestion de production dans le cadre des industries automobiles.

Dans un premier temps, il est décrit l'approche classique du flux poussé. Cette approche consiste plus en une coordination entre les différents plans de production qu'une synchronisation des différents flux. La philosophie de ce type de système de production consiste à planifier en central les produits à fabriquer à chaque étape. Puis à partir de ces prévisions, les matières premières nécessaires à cette production sont donc anticipées et envoyées périodiquement sur la ligne en quantités suffisantes.

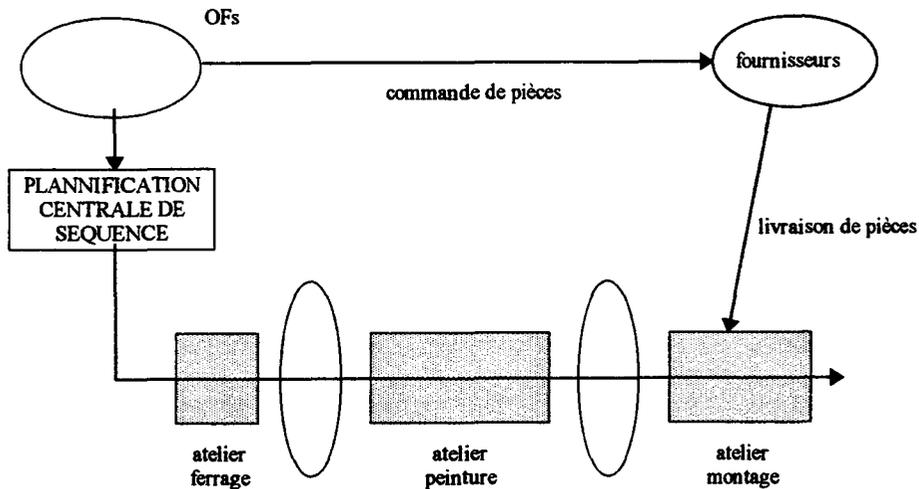


Figure 1.11 - Systèmes de gestion de production en flux poussé

Cette approche entraîne un manque de réactivité et nécessite des stocks importants à cause de la fiabilité des livraisons. Toutes ces raisons "affaiblissent la compétitivité" de l'usine.

La deuxième approche est un système de gestion de production en flux tiré, aussi bien pour le contrôle des constituants que pour le contrôle de la production. Pour atteindre ces objectifs, différentes stratégies doivent être mises en place pour décrire le cœur du système. Dans un premier temps, Mertins conçoit une "segmentation" de l'usine. En effet chaque atelier doit être indépendant et doit donc posséder à la fois un outil de planification et un outil de contrôle du système pour chacun d'eux. Dans un deuxième temps, il est nécessaire de concevoir la notion de "produits d'atelier". Chaque atelier doit fabriquer des produits correspondant à la demande de l'atelier suivant. Le ferrage fabrique des véhicules dont la définition est la forme. L'atelier peinture peint les véhicules, et ajoute donc l'attribut couleur. Cet atelier fabrique donc un ensemble de caisses possédant les attributs forme et couleur. La troisième stratégie à mettre en place est la création d'une "production anonyme" de véhicules en utilisant plusieurs "points de baptêmes" pour garder une flexibilité de production la plus grande possible. En effet, la définition définitive des attributs d'un véhicule doit être reportée le plus tard possible. Les véhicules en entrée de l'atelier ferrage sont anonymes. En sortie de l'atelier de ferrage, un point de baptême permet d'attribuer à chaque véhicule un attribut : sa forme. Il en est de même pour l'atelier peinture. En entrée, les véhicules sont classés en fonction des attributs définis dans l'atelier précédent. En sortie un point de baptême attribue à chaque véhicule une couleur.

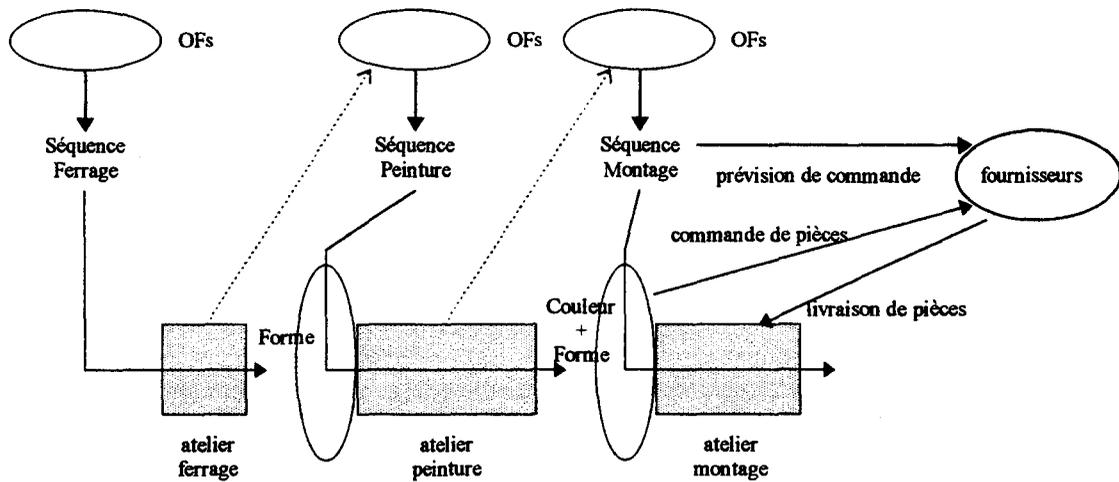


Figure 1.12 - Systèmes de gestion de production en flux tiré

Cette philosophie consiste donc à avoir une production de types de véhicules plutôt que de véhicules uniques et différenciés. La production d'un véhicule est déclenchée par un client. Cet ordre de fabrication se répercute sur l'ensemble de la ligne de fabrication en remontant le flux. Pour terminer la fabrication du véhicule demandé, il faut que les pièces soient disponibles en sortie de l'avant dernier atelier. Si ce n'est pas le cas, une commande est passée pour qu'elles soient fabriquées au plus tôt.

La troisième approche consiste en un système de gestion de production "hybride" qui allie à la fois le caractère poussé du contrôle de la production et le caractère tiré du contrôle de l'approvisionnement en pièces. Cette approche permet une plus grande flexibilité, une diminution des stocks et une plus grande disponibilité des pièces sur la ligne d'assemblage. En revanche, elle repousse les difficultés sur les approvisionnements en pièces, d'après Mertins.

L'approche hybride va permettre de déterminer la nature d'un véhicule le plus tôt possible en l'identifiant à un client. Le véhicule sera défini très tôt. Cela permet de mieux maîtriser le flux, de s'assurer des délais. En revanche cette approche peut reporter sur les flux de constituants et les approvisionnements les difficultés intrinsèques liées au "juste-à-temps".

L'approche flux tiré totale, prônée par Mertins, permet de s'affranchir a priori de ces contraintes sur les flux convergeants, mais en contrepartie elle nécessite la création de relations client/fournisseurs entre les différents ateliers du flux principal de l'usine et la définition la plus tardive possible des équipements du véhicule. Cela correspond donc à gérer la diversité des "produits d'atelier" dans les stocks intermédiaires et à ne pas identifier un véhicule précisément et à ne pas pouvoir garantir son délai car on ne sait pas à quelle caisse potentielle il sera associé.

## **5. Conclusion**

Dans cette première partie, nous avons présenté la problématique associée aux lignes de fabrication multi-produits à modèles mélangés.

En effet, la demande de la clientèle, l'offre des constructeurs et la concurrence ont fait évoluer les systèmes de production dans le monde automobile. Dans le cadre de la production de masse et de la production au plus juste, les lignes d'assemblage (Mixed Model Assembly Lines) peuvent être assimilées à des flow-shop sans attente. La difficulté de gestion de ces lignes consiste à générer une séquence de produits fabricables au moindre coût et dans les délais.

Pour intégrer les difficultés intrinsèques aux usines automobiles, il est important de comprendre la topologie d'une usine. Les trois ateliers (ferrage, peinture et montage) imposent des contraintes de fabrication sur la succession des produits dans la séquence et sur la charge des différents postes. Les moyens de transport expliquent la nécessité de générer une séquence et les stocks intermédiaires de production influencent directement la capacité de production et sont les seules flexibilités maîtrisables de l'outil de production, permettant de modifier ou de corriger le flux en fonction d'une séquence optimum.

Dans un deuxième temps, nous avons montré l'intérêt d'une optimisation de ces séquences de produits pour des raisons économiques et commerciales (réduire les coûts et garantir les délais). Et nous avons décrit les moyens de résoudre ce problème (soit de manière statique par un séquençement prévisionnel, soit de manière dynamique par un cadencement réactif) et les philosophies de gestion de flux qui en découlent (flux poussé ou flux tiré) et qui permettent ou non une gestion juste-à-temps.

## *Partie 2*

OUTILS, METHODES, MODELISATION



## Partie 2 : Outils, méthodes, modélisation

### 1. Introduction

Le problème de séquençement sur une ligne d'assemblage à modèles mélangés est fortement dépendant du contexte industriel. Nous avons abordé ce problème dans la première partie. D'après Okamura [OKA 79] et Miltenburg [MIL 89a], pour une bonne utilisation d'une ligne d'assemblage il est nécessaire de résoudre quatre problèmes indépendants mais fortement complémentaires :

- Détermination des temps de cycle.
- Détermination du nombre et de la séquence des postes de travail sur la ligne.
- Equilibrage de la ligne.
- Détermination de l'ordonnancement des produits sur la ligne.

Nous nous intéressons dans cette étude au quatrième point. Nous allons aborder dans cette partie les différents outils que nous allons utiliser par la suite dans les parties trois et quatre. Il est important de définir clairement les objectifs que l'optimisation doit chercher à améliorer. Dans un premier temps, nous allons étudier plus précisément trois approches fréquemment utilisées dans la littérature. Puis nous allons répertorier les différentes règles de séquençement qui peuvent être utilisées dans le monde automobile en s'appuyant sur les travaux de Comby [COM 96]. Ensuite, nous allons essayer de décrire des indicateurs concrets qui permettront de représenter la qualité du flux et nous étudierons plus finement les modélisations que nous pouvons retenir. Pour finir, une revue des différentes méthodes d'optimisations (locales ou globales) sera faite avec une étude des différentes applications qui ont pu en être faites.

### 2. Ordonnancement des lignes à modèles mélangés : Typologie des règles de séquençement

Le séquençement de produits sur une ligne d'assemblage a fait l'objet de nombreuses études au niveau académique ainsi qu'au niveau industriel. Les différentes contraintes que l'on peut rencontrer obligent à concevoir une méthode capable de s'adapter aux différents environnements de production. Une étude de Ghosh et Gagnon [GHO 89] a montré que les problèmes d'équilibrage de lignes d'assemblage peuvent être regroupés en quatre grandes familles. La classification des problèmes d'équilibrage et de séquençement des véhicules est la

semblable. Les difficultés se retrouvent dans les deux types de problèmes en fonction du nombre de modèles et de la modélisation des temps opératoires.

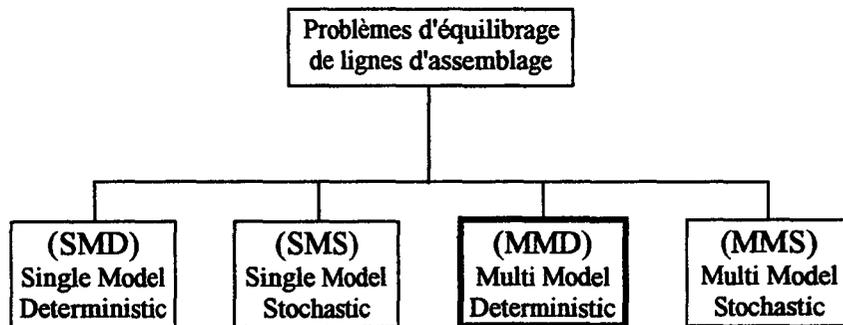


Figure II.1 - Rappel de la classification de la littérature sur l'équilibrage des lignes d'assemblage (Assembly Line Balancing)

De nombreux critères ont été utilisés dans le cadre de l'équilibrage, critères à la fois techniques et économiques. Ces critères peuvent consister à :

- Minimiser le nombre de postes de travail pour un temps de cycle donné.
- Minimiser le temps de cycle pour un certain nombre de postes de travail.
- Minimiser le temps d'inactivité sur l'ensemble de la ligne.
- Minimiser l'écart sur l'équilibrage.
- Minimiser la longueur de la ligne.
- Minimiser le throughput time (ou temps de défilement).
- Minimiser la probabilité de dépassement du temps de cycle pour chaque poste de la ligne.
- Minimiser le coût combiné de la main d'oeuvre, des postes de travail et de l'incomplétude des tâches non terminées.
- Minimiser le coût du travail par voiture produite.
- Minimiser les coûts des stockages, de réglages et d'inactivité.
- Minimiser les coûts d'encours.
- Maximiser le profit net.

Dans le cadre de notre étude, nous nous intéressons au domaine MMD (pour Multi Model Deterministic), pour lesquels Arcus [ARC 66] est un précurseur. Plusieurs produits peuvent être fabriqués en même temps sur une ligne d'assemblage et la durée des tâches est considérée comme déterministe. La différence avec les MMS (pour Multi Model Stochastic) apparaît sur le caractère stochastique des durées des tâches des opérateurs, variabilité que nous ne prenons pas en compte. A la différence des SMD (Single Model Deterministic) ou SMS (Single Model Stochastic), le séquençement des produits à fabriquer devient critique.

Pour résoudre ce problème de séquençement, on peut être amené à prendre en compte deux types d'objectifs [COM 96].

D'une part les objectifs peuvent être indépendants de la configuration de la ligne, et les types de règles de séquençement sont de la forme :

- Répartition des produits en lissant le taux de production : c'est-à-dire minimiser l'écart entre la position réelle du produit et la position moyenne.

- Lisser le taux de consommation des sous-ensembles : c'est-à-dire minimiser l'écart entre la position réelle d'un produit contenant le sous-ensemble et la position moyenne.

- Respect d'un ratio de production pour certains équipements ou certains types de produits. C'est le respect des capacités de production.

- Minimiser l'encours de production.

- Minimiser le coût de changement d'outils.

D'autre part, les objectifs peuvent être dépendants de la configuration de la ligne en lissant la charge des ressources de la ligne. Cet objectif peut être atteint :

- En minimisant les surcharges de travail ou/et les souscharges.

- En minimisant les variations de charges.

- En minimisant les surpotentiels (ce qui correspond à réduire l'oisiveté de ces ressources).

Les différentes règles abordées dans la littérature pour le séquençement des produits sur une ligne de montage peuvent être classées suivant trois familles principales. Une répartition des produits sur l'ensemble de la production est l'approche la plus fréquente. Mais le respect d'un ratio de production à satisfaire a également été étudié. Une troisième famille regroupe les règles minimisant les coûts de changement d'outils (augmentation de la taille de lots ou de rafales).

Comby [COM 96] a regroupé de manière exhaustive les règles de séquençement qu'il a recensées dans le monde industriel automobile. Il a repris et particularisé les trois types de règles précédentes et en a défini de nouvelles. Nous décrirons aussi les différences, lorsqu'elles existent, entre les métriques évaluatives (qui évaluent une liste déjà construite) et les métriques constructives (qui permettent de choisir le produit suivant dans la liste en construction).

### **2.1. "Répartition des produits" ou lissage des critères**

La règle de répartition des produits consiste à lisser la consommation des composants ou la production des produits finaux sur l'ensemble de l'horizon de production. La première

approche de ce type ainsi que les suivantes s'inspirent de l'algorithme de Toyota le Goal Chasing décrit par Monden en 1983 ([MON 83] p181).

La répartition des produits sur l'ensemble de la séquence à produire est nécessaire dans le cas d'une production en juste à temps. En effet elle requiert une consommation régulière de chaque composant approvisionné par une boucle Kanban.

Cette règle consiste à répartir sur l'ensemble de la séquence les produits qui peuvent être caractérisés par un type, une classe, un ensemble de composants ou un ensemble de matières premières nécessaires à leur fabrication [BAR 92]. Cette approche peut également servir à gérer les problèmes de charge des postes : les goulots d'étranglement sont transformés en ressources dont il faut étaler les tâches critiques.

### 2.1.1. Taux de consommation de composant (PART USAGE GOAL)

L'approche décrite par Monden s'appuie sur deux objectifs. Le premier objectif consiste à s'assurer que la charge de travail à chaque poste pondérée par la quantité des produits est inférieure au temps cycle.

La formalisation mathématique de cette contrainte est :

$Q_i$  = quantité de produits  $A_i$  ( $i=1, \dots, \alpha$ ) planifiée

$T_{il}$  = temps opératoire du produit  $A_i$  sur le poste  $l$

$C$  = temps cycle =  $\frac{\text{temps de travail journalier}}{\sum_{i=1}^{\alpha} Q_i}$

$$\max \left\{ \frac{\sum_{i=1}^{\alpha} Q_i T_{il}}{\sum_{i=1}^{\alpha} Q_i} \right\} \leq C \quad \forall l$$

Ce premier objectif est pris en compte dans l'algorithme global mais son importance est moindre que le second.

Ce deuxième objectif est une nécessité due au système Kanban mis en place pour répondre au fonctionnement en juste-à-temps. Dans ce système "tiré", la variation des quantités produites ou des temps de transfert entre les postes doit être minimisée. Donc les encours doivent l'être aussi. Pour cette raison, la vitesse de consommation (c'est à dire la quantité de pièces utilisées) doit être gardée aussi constante que possible.

On va définir les notations nécessaires à la compréhension du problème :

$Q_i$  = quantité de produits  $A_i$  ( $i=1, \dots, \alpha$ ) à produire.

$Q = \sum_{i=1}^{\alpha} Q_i$  quantité totale à produire de tous les produits  $A_i$  ( $i=1, \dots, \alpha$ )

$N_j$  = Quantité totale nécessaire du composant  $a_j$  pour produire tous les produits  $A_i$  ( $i=1, \dots, \alpha$ ), ( $j=1, \dots, \beta$ ).

$X_{jk}$  = Quantité totale nécessaire du composant  $a_j$  pour produire tous les produits d'une séquence déterminée de la première position à la  $K$  ème.

D'où les deux valeurs suivantes :

$N_j/Q$  = Quantité moyenne nécessaire du composant  $a_j$  par unité à produire.

$\frac{K \cdot N_j}{Q}$  = Quantité moyenne nécessaire du composant  $a_j$  pour produire  $K$  unités.

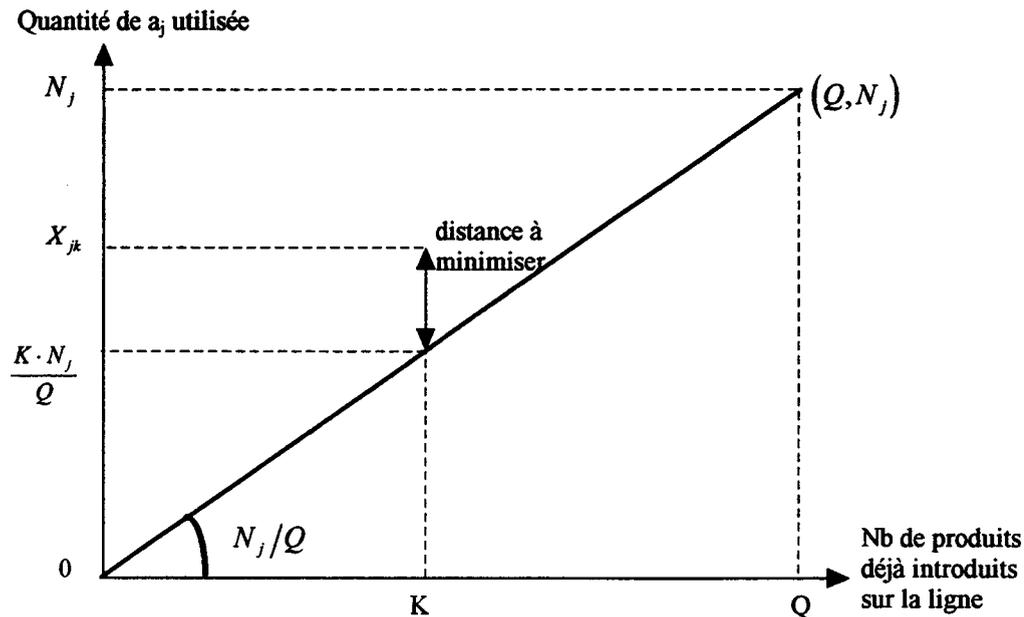


Figure II.2 - Relation entre  $X_{jk}$  et  $kN_j/Q$  dans la formulation de Monden

Donc pour maintenir la vitesse de consommation des composants constante, il faut essayer de maintenir la position de  $X_{jk}$  le plus près possible de la droite de consommation idéale (c'est à dire de la valeur  $K \cdot N_j/Q$ ). Cela revient à minimiser la distance entre les deux points :

$$G_k = (K \cdot N_1/Q, K \cdot N_2/Q, \dots, K \cdot N_\beta/Q)$$

$$P_k = (X_{1k}, X_{2k}, \dots, X_{\beta k})$$

dont la distance est :

$$D_k = \|G_k - P_k\| = \sqrt{\sum_{j=1}^{\beta} \left( \frac{k \cdot N_j}{Q} - X_{jk} \right)^2} \quad (\text{distance euclidienne})$$

Au niveau mathématique le problème de séquençement revient à minimiser pour l'ensemble de la liste de  $Q$  éléments,

$$\sum_{k=1}^Q D_k = \sum_{k=1}^Q \sum_{j=1}^{\beta} \left( k \cdot N_j / Q - X_{jk} \right)^2$$

L'heuristique de Monden consiste alors à choisir parmi les  $Q+1-k$  choix au niveau  $k$  le meilleur. C'est une méthode gloutonne de rang 1 : c'est-à-dire qu'à chaque niveau de décision, seul le meilleur choix est retenu. C'est une méthode myope qui donne des résultats moyens mais qui est très rapide (la complexité pour un ensemble d'Ordre de Fabrication de taille  $N$  est :  $O(N)$  pour chaque choix, c'est à dire  $O(N^2)$  pour l'ensemble des voitures à ordonnancer).

### 2.1.2. Taux de production de chaque type de produit fini (MODEL PRODUCTION RATE)

Cette approche est peu différente de la précédente. Mais elle est plus rapide car elle ne descend pas jusqu'au niveau des composants mais reste au niveau des sous produits.

Pour résoudre ce problème, Miltenburg [MIL 89a] a proposé une approche à un niveau, en utilisant une modélisation mathématique proche de celle de Monden, mais ne prenant en compte que le lissage de la consommation des produits.

Une nouvelle notation apparaît :

$X_{ik}$  = Quantité de produits  $A_i$  séquencés entre les positions 1 et  $k$

L'optimisation consiste alors à minimiser la distance :

$$D'_k = \|G'_k - P'_k\| = \sqrt{\sum_{i=1}^{\alpha} \left( X_{ik} - \frac{k \cdot Q_i}{Q} \right)^2}$$

et donc l'optimum global consiste à minimiser :

$$\sum_{k=1}^Q D'_k = \sum_{k=1}^Q \sum_{i=1}^{\alpha} \left( X_{ik} - k \cdot Q_i / Q \right)^2$$

Miltenburg propose également différentes distances non euclidiennes pour modéliser le problème. Pour le résoudre, il propose 3 algorithmes différents et deux heuristiques; la première à un pas, la deuxième à deux pas. Sumichrast, Russel et Taylor [SUM 92] ont retenu l'algorithme 3 et l'heuristique 2 noté (M-A3H2) et les ont comparés à leur algorithme.

D'autres approches existent dans la littérature et dérivent en général de celle de Miltenburg.

Inman et Bulfin [INM 92] proposent de maintenir constant le taux de consommation de tous les sous-ensembles. Cet objectif est atteint en calculant pour chaque élément de chaque produit une date de livraison idéale puis en les triant par ordre croissant. La séquence ainsi générée est comparée alors aux résultats des algorithmes de Miltenburg [MIL 89a], [MIL 89b] et permet d'obtenir de très bon résultats. Les auteurs les considèrent meilleurs dans les cas "industriels". En effet dans les hypothèses de Miltenburg, il y a peu de produits et les problèmes sont de petite taille, ce qui ne correspond pas à des problèmes réalistes. Dans ces cas "réalistes", 10 produits et 500 éléments à fabriquer, l'heuristique proposée EDD (Earliest Due Date [INM 92]) donne de manière surprenante de meilleurs résultats que Miltenburg en un temps de calcul nettement plus faible.

Groeflin, Luss, Rosenwein et Wahls [GRO 89] proposent un algorithme utilisant une procédure de recherche locale basée sur des échanges dans la séquence permettant de résoudre une formulation lexicographique du problème de séquençement des lignes d'assemblage.

Kubiak et Sethi [KUB 94] développent un algorithme semblable à celui de Miltenburg en ne considérant que le niveau des produits finis. L'algorithme est polynomial. L'apport le plus important de cet étude est le fait que le problème de séquençement se comporte comme un problème d'affectation lorsque la fonction de pénalisation ou métrique est convexe et non négative.

### *2.1.3. Consommation cumulée*

Miltenburg et Sinnamon [MIL 89b] ont modélisé le problème par une approche à quatre niveaux : Produits, Sous-ensembles, Composants et Matières premières. L'objectif est de lisser la production à chaque niveau.

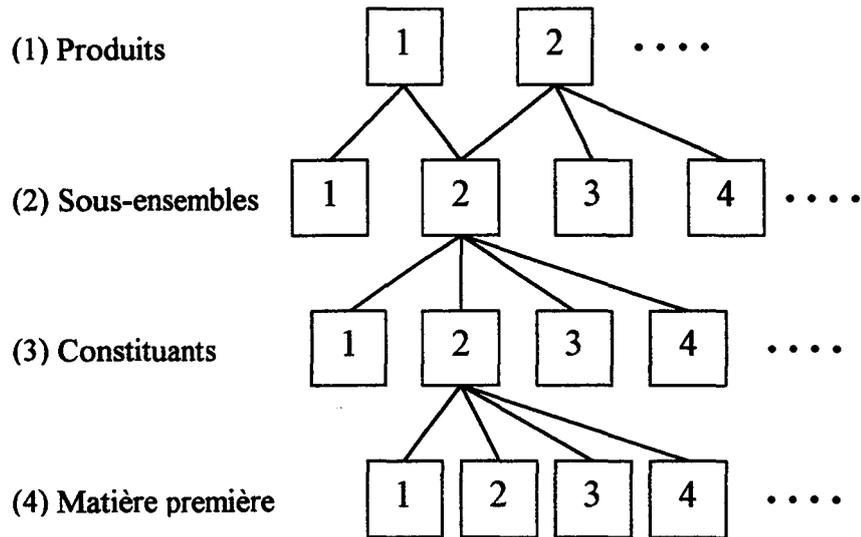


Figure II.3 - Niveaux de production d'un système juste-à-temps [MIL 89b]

Pour décrire cette approche, Miltenburg définit les variables suivantes

$j$  = numéro du niveau (1, produits; 2, sous-ensembles; ...)

$n_j$  = nombre d'éléments au niveau  $j$  ( $j = 1, 2, 3, 4$ )

$d_{i1}$  = demande de produit  $i$  ( $i = 1, 2, 3, 4$ )

$t_{ijl}$  = nombre d'éléments  $i$  au niveau  $j$  utilisés pour produire une unité du produit  $l$ ,

( $i = 1, 2, \dots, n_j; j = 1, 2, 3, 4; l = 1, 2, \dots, n_1$ )

On en déduit les notions suivantes :

$$d_{ij} = \sum_{h=1}^{n_1} t_{ijh} d_{h1} = \text{demande d'éléments } i \text{ au niveau } j, (i = 1, 2, \dots, n_j; j = 1, 2, 3, 4)$$

$$DT_j = \sum_{i=1}^{n_j} d_{ij} = \text{demande totale au niveau } j, (j = 1, 2, 3, 4)$$

$$r_{ij} = d_{ij} / DT_j = \text{ratio de production au niveau } j \text{ pour l'élément } i, (i = 1, 2, \dots, n_j; j = 1, 2, 3, 4)$$

L'horizon de production est  $DT_1$ .

$$x_{i1k} = \text{nombre de produits } i \text{ fabriqués pendant les } k \text{ premiers pas } (j=1)$$

$$x_{ijk} = \sum_{h=1}^{n_1} t_{ijk} x_{h1k} = \text{nombre d'éléments } i \text{ au niveau } j \text{ fabriqués pendant les } k \text{ premiers pas,}$$

$$\text{avec } x_{ij0} = 0$$

$$XT_{jk} = \sum_{i=1}^{n_j} x_{ijk} = \text{production totale au niveau } j \text{ pendant les } k \text{ premiers pas}$$

$$w_j = \text{poids du niveau } j$$

L'objectif, semblable aux autres approches, est de minimiser l'écart entre la droite de consommation moyenne et la consommation réelle. D'où la formule mathématique :

$$\sum_{k=1}^{DT_i} \sum_{j=1}^4 \sum_{i=1}^{n_j} w_j (x_{ijk} - XT_{jk} r_{ij})^2 \text{ pour une distance euclidienne.}$$

Si l'on fait une analogie entre cette approche globale et celles décrites précédemment, Miltenburg [MIL 89a] ne prend en compte que le premier étage de cette modélisation (il ne lisse que le taux de production des produits). De même, Kubiak et Sethi [KUB 94] ne prennent en compte que le deuxième niveau. De son côté, Monden [MON 83] ne prend en compte que les deux premiers étages en lissant le taux de consommation des sous-ensembles.

Dans la problématique juste-à-temps, il est intéressant d'optimiser au mieux la consommation à l'étage le plus bas. Ce lissage ne se répercute pas nécessairement sur les étages supérieurs. Le problème d'une telle approche est l'explosion combinatoire qui résulte de la modélisation et la complexité de la fonction d'évaluation. Les deux approches décrites précédemment permettent de limiter cette complexité en lissant la consommation à un étage plus élevé. L'approximation en résultant peut permettre d'obtenir de bons résultats en un temps de calcul beaucoup plus faible.

#### 2.1.4. Définition de positions cibles

Cette règle ne nécessite que peu de paramètres. En effet, seuls le niveau de détail recherché (produit, sous-ensemble, ...) et une caractérisation réduite des produits (nomenclature restreinte) sont nécessaires. Le taux de production à respecter dépend alors de l'ensemble des produits à séquencer.

Cette approche a été proposée par Inman et Bulfin [INM 92] et a été reprise par Comby [COM 96]. Elle consiste à déterminer des positions cibles pour chaque véhicule ayant le critère et à minimiser les écarts entre les positions de ces véhicules et les positions idéales calculées.

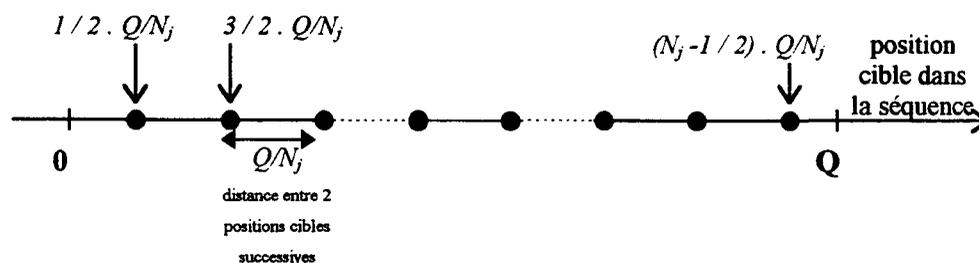


Figure II.4 - Positions cibles proposées par Inman et Bulfin [INM 92]

La vitesse de consommation idéale pour le critère "j" est :  $Q/N_j$ . Donc en positionnant les véhicules ayant ce critère à des positions espacées de  $Q/N_j$  places, le lissage des critères sera

respecté. Pour respecter la notion d'intervalle entre véhicule, le premier de ceux-ci est positionné à la place  $0,5 \cdot Q/N_j$ , les suivants s'échelonnent de manière régulières.

Il s'agit par la suite de positionner les véhicules ayant le critère "j", le plus près possible de la position définie précédemment. Cette méthode est néanmoins peut robuste si de nombreux critères sont croisés car elle se base sur une définition statique des espacements entre les véhicules ayant le même critère.

Une dernière approche consiste à définir une vitesse de consommation moyenne qui permet d'atteindre le volume journalier  $N_j$ , sans pour autant répondre trop violemment aux écarts par rapport à la moyenne. En effet, des approches comme celles de Monden ou de Inman peuvent poser problème lors de fonctionnements critiques (aléas de fabrication, contraintes croisées, ...). Si un retard est pris, il est immédiatement corrigé, cela implique sur le flux principal une forte variation de l'espacement entre deux produits ayant le même critère. En revanche une politique plus souple, pour laquelle l'objectif est à long terme (cible journalière), permet d'atténuer les fluctuations, en particulier pour les flux très perturbés.

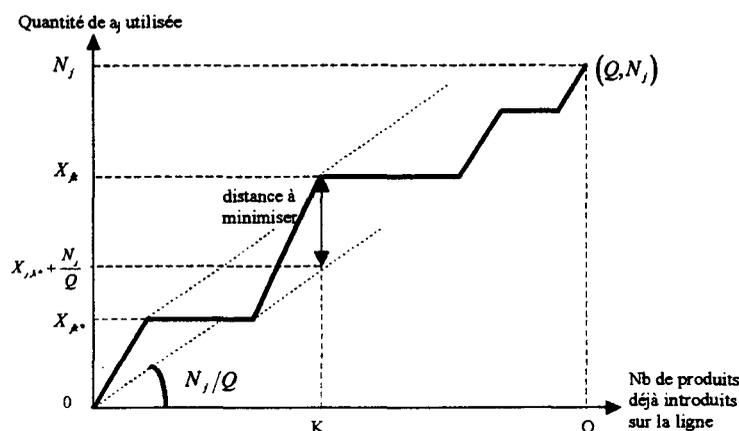


Figure 11.5 - Approche de Monden avec des vitesses de consommation instantanée constantes

Dans cette approche, deux objectifs peuvent être pris pour cible. L'algorithme peut soit conserver une vitesse instantanée constante et égale à  $N_j/Q$ , soit recentrer sa cible sur le volume journalier  $(Q, N_j)$  à chaque étape.

Dans cette approche, les métriques évaluatives et constructives sont identiques.

## 2.2. Respect des capacités de production (ou "Ratios")

Le deuxième type de règle de séquençement rencontré dans la littérature est le respect de ratio qui est un problème rencontré souvent en production automobile. Parello, Kabat et Wos [PAR 86] sont les premiers à avoir introduit cette idée. Cette notion correspond à garantir que pour des voitures nécessitant le montage d'un équipement particulier, l'usine est capable d'en

assurer la production. Par exemple si 60% des voitures ont cet équipement (ou critère dans la modélisation), l'usine devra être capable de le monter sur au moins 3 véhicules sur 5. Cette notion de ratio peut apparaître sur deux formes semblables : soit la production doit garantir un fonctionnement le plus proche possible du ratio de production demandé (Dincbas, Simonis et Van Hentenryck [DIN 88]), soit la ligne d'assemblage a une contrainte de capacité "capacity constraint" qui correspond à un ratio que la ligne ne peut pas dépasser pour des raisons de performance [PAR 86].

Le séquençement des produits doit donc garantir le ratio de chacune des options prises en compte sur la ligne d'assemblage, cette contrainte est très forte car, en général, il s'agit de garantir la capacité à produire de la ligne.

Cette notion de ratio de production de  $N_r$  sur  $D_r$  pour un critère donné (ou noté  $N_r/D_r$ , par abus) indique que dans toute sous-séquence de taille  $D_r$ , on ne souhaite pas plus de  $N_r$  véhicules ayant le critère.

La difficulté du problème réside dans le fait que les véhicules ont évidemment plusieurs critères concurrents. Il faut donc optimiser la séquence en essayant de respecter les ratios pour chacun des critères.

Pour cela, Dincbas, Simonis et Van Hentenryck [DIN 88] ont introduit la notion de fenêtres glissantes. Il faut respecter le ratio pour l'ensemble des fenêtres de taille  $D_r$  de la séquence:

Par exemple pour une séquence de 10 véhicules devant respecter le ratio 2/3, les fenêtres seront donc de la forme :

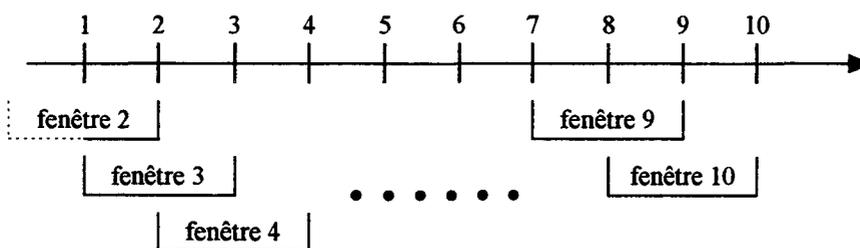


Figure II.6 - Position des fenêtres glissantes

La position de la fenêtre  $k$  a donc comme bornes les positions  $k + 1 - D_r$  et  $k$ . Il faut ensuite minimiser la fonction de coût, où le terme  $\max(0, x)$  indique le nombre de violations :

$$\sum_j \sum_k w_j \cdot f \left( \max \left( 0, -N_j + \sum_{l=k+1-B_j}^k FC(s(l), j) \right) \right)$$

avec :

$j$  = le critère retenu

$k$  = la position dans la séquence

$w_j$  = le poids du critère

$f(x)$  = fonction de démerite des violations (en général affine)

$s(l)$  = produit à la  $l$ ème position

$FC(y, j) = 1$  si le produit  $y$  à le critère  $j$ , sinon  $FC(y, j) = 0$

Dincbas a résolu ce problème en utilisant le langage CHIP (langage de propagation par contraintes).

[CHE 91] a utilisé cette modélisation avec une méthode par affinements successifs (recuit simulé). [CAS 94] l'a utilisé dans le cadre d'une construction progressive de séquence.

Guerre-Chaley [GUE 95] modélise le problème des ratios sans utiliser les fenêtres glissantes, mais il se limite au cas où  $N_r = 1$ . Pour cela, il associe à chaque critère une variable  $r(n, i)$  pour le critère  $n$  à la place  $i$  : elle correspond au nombre de véhicules sans le critère qu'il faut encore enligner pour respecter le ratio. Cette variable est calculée comme suit :

$$r(n, i) = \max(r(n, i-1) - 1, 0) \quad \text{si le critère est absent à la place } i$$

$$r(n, i) = D_n - 1 \quad \text{si le critère est présent à la place } i$$

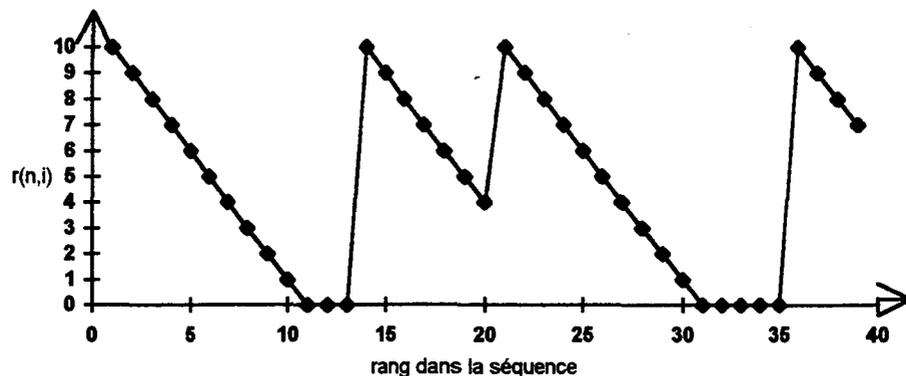


Figure 11.7 - Exemple d'évolution de  $r(n, i)$  en fonction de  $i$  avec  $D_n = 11$

Dans l'exemple ci dessus, il y a violation des ratios au rang 20.

Cette modélisation ne peut être appliquée qu'à deux types de ratio : soit  $1/D$  soit  $D - 1/D$ . Dans les autres cas elle s'avère trop restrictive. Il est alors nécessaire d'utiliser l'approximation proposée par Guerre-Chaley consistant à découper chaque ratio en deux ou plusieurs ratios

complémentaires. Par exemple, un ratio de 2/5 ne peut être pris en compte par cette méthode, en revanche, il est possible d'utiliser une succession de ratios 1/2 puis 1/3 (c'est-à-dire d'espacement de 1 puis de 2, qui permettent de respecter le volume prévisible). En revanche, le ratio 2/5 permet d'avoir deux véhicules ayant le critère se succédant pourvu que trois véhicules ne l'ayant pas suivent :

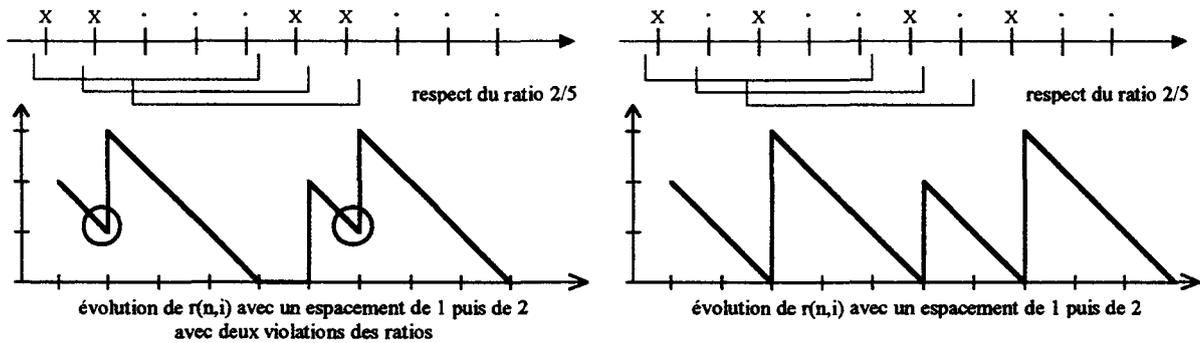


Figure II.8 - Limites de la notation  $r(n,i)$

Elle ne permet pas non plus de lisser la charge : tout retard pris dans la production n'est pas rattrapé car la longueur des paliers où  $r(n,i) = 0$  n'est pas limitée.

Mais Guerre-Chaley montre qu'un espacement non entier peut être ramené à une succession d'espacements entiers. Cette modélisation est néanmoins plus contraignante que la modélisation avec fenêtre glissante.

La notion de ratio peut être transformée en espacement entre produits ayant le critère (comme l'indique Guerre-Chaley [GUE 95]). En effet un ratio de  $1/D$  correspond à un espacement entre deux produits de  $E = D - 1$ , c'est à dire qu'il doit y avoir  $D - 1$  produits sans le critère entre deux produits l'ayant.

Cette règle de respect des ratios peut ressembler à la règle de lissage. En effet si la répartition des produits est effectuée de manière parfaite, si toutes les options sont espacées de manière régulière, le séquençage respecte alors un ratio pour chacune des options de  $Q_i/Q$  (où  $Q_i$  est le nombre de produits ayant l'option  $i$  et  $Q$  le nombre total de produits). La règle des ratios diffère néanmoins sur le fait que l'on doit garantir un ratio maximal et qu'il n'y a pas de contraintes sur la stabilité des options dans la séquence si le ratio souhaité laisse une marge vis à vis du taux de production demandée.

La règle de répartition des produits permet d'obtenir a priori un meilleur résultat si l'on s'approche de l'optimum global. En revanche la règle de respect des ratios est plus robuste vis à vis des contraintes de production que doit respecter la séquence.

La notion d'espacement (à valeur réelle) peut dériver de cette notion de ratio. La formule suivante permet la conversion :  $N_r/D_r \rightarrow E_r = D_r/N_r - 1$

### **2.3. Ratio orienté**

La notion de ratio orienté proposée par [COM 96] est semblable à la précédente mais introduit un ordre de précedence entre plusieurs critères. Cette approche décrit a priori les problèmes que l'on peut rencontrer sur des postes où l'ouverture n'est pas la même en entrée et en sortie (soit "closed" soit "open" cf. [BAR 92] et [DAR 75] cf. I.1.3). Dans le cadre de notre étude, elle n'a pas été retenue.

### **2.4. Position imposée**

Un produit défini et personnalisé doit pouvoir être positionné à une place bien précise. Cette approche peut correspondre à des véhicules préséries ou prototypes nécessitant des équipes ou des ressources spéciales.

Cette règle (nouvelle dans la littérature) est une règle dure, dans le sens où il faut respecter cette position de manière absolue. Le fait que des véhicules aient un traitement spécial sur une ligne de montage, entraîne en général un suivi précis tout au long de la ligne de montage et une allocation spéciale de ressource. Cette règle n'a donc pas été retenue.

### **2.5. Position souhaitée**

Cette règle est semblable à la précédente, mais la contrainte est moins rigide et une dérive de cette position est tolérée.

Comby associe cette règle à une notion de logique floue sur la position du produit. Elle est utilisée en remplacement de la règle "position imposée" quand il existe une certaine marge de manoeuvre.

Cette règle est plus souple que la précédente, mais aussi plus pernicieuse. En effet il faut s'assurer avant de les utiliser que l'on ne rentre pas en conflit avec une autre règle (par exemple des véhicules différents peuvent avoir la même position ou bien le positionnement de certains produits peut perturber l'agencement de règles plus importantes comme les "ratios" ou la "répartition des produits"). Elle est liée à un nombre limité de produits et cela doit limiter son importance pour ne pas perturber l'ensemble de la séquence.

## **2.6. Placer au plus tôt**

Pour Comby, cette règle permet de limiter le retard que certains produits ont pu prendre pour une raison quelconque. Les produits concernés par une règle "au plus tôt" doivent être placés avant un rang limite " $r_{max}$ ".

## **2.7. Placer au plus tard**

Cette règle est identique à la précédente et permet de limiter l'avance prise par certains produits dans le cas de rejalonnements.

## **2.8. Succession impaire**

Cette règle consiste à laisser un nombre pair de positions entre deux produits successifs ayant ce critère.

Cette règle peut servir dans le cas où il y a des dégroupements et des regroupements de flux. L'intérêt en est limité, car cela correspond à mixer deux flux dont les contraintes sont différentes. Cette règle n'a pas été retenue dans le cadre de notre étude.

## **2.9. Suivi des modifications**

Pour Comby, lorsque plusieurs produits sont équivalents du point de vue des règles de séquençement définies précédemment, cette règle permet de les différencier. Le choix est fait sur la base de leur numéro de fabrication qui dépend de la date de commande du véhicule.

L'intérêt de cette règle me paraît limité car elle consiste à gérer des classes de véhicules équivalents. Si on gère une date de livraison du véhicule (date au plus tôt), on est capable de faire une différence entre tous les véhicules.

## **2.10. Respect des "RAFALES"**

La dernière famille de règles de séquençement gère le coût de changements d'outils en regroupant au maximum les familles de produits ayant le même critère. Burns et Daganzo [BUR 87] traitent ce problème en effectuant des regroupements imbriqués de manière itérative au sein de sous-séquences déjà regroupées. (cf. figure I3.5)

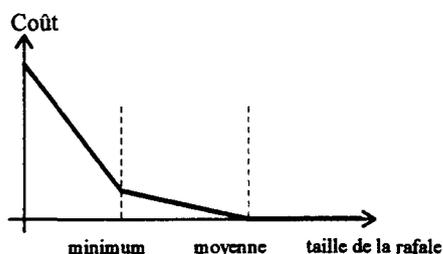
Ce problème a été couplé avec l'un des précédents par Chew et al. [CHE 91] en prenant en compte les coûts de changement d'outils et le respect des ratios de fabrication. Bolat et al. [BOL 94] traitent pour leur part le problème en couplant le coût de réglage des postes et la charge des postes dans une ligne d'assemblage à modèles mélangés.

Pour optimiser le moyen de production, il peut être souhaitable de regrouper des produits ayant le même critère. Une règle de type rafale ou convoi consiste à minimiser les pertes dues à un changement d'outils (purges pour la peinture, setup time pour une machine d'usinage, ....)

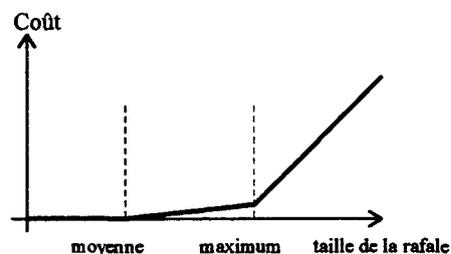
Deux approches peuvent être perçues. La première consiste à minimiser le nombre de changements, donc à minimiser le nombre de rafales ou à maximiser la taille moyenne de ces rafales. La deuxième approche qui est rarement abordée consiste à optimiser une taille de rafale que l'on appelle convoi. On cherche toujours à minimiser le nombre de changements d'outils, mais avec une borne que l'on ne souhaite pas franchir. Cela correspond à avoir une borne minimum et une borne maximum pour la taille des convois. La borne minimum permet d'optimiser au mieux l'outil de production et la borne maximum est une limite qu'il est préférable de ne pas franchir pour des raisons de coût. Dans le cas contraire, elle nécessiterait des encours de production trop importants, entraînant des problèmes de qualité ou de contrôle, ou bien des réglages qui sont de toute façon nécessaires au bout d'un certain temps.

La première approche est intéressante pour une métrique évaluative : il est facile de dénombrer les changements de rafales. En revanche dans le cas d'une métrique constructive, le fait d'avoir une notation discontinue ne va pas faciliter l'optimisation, en particulier si les rafales sont très longues. Le coût associé vaut 1 si les deux produits n'ont pas le même critère et 0 sinon.

Pour la deuxième approche, la notation consiste à évaluer la taille des rafales et à la limiter par une borne inférieure et une borne supérieure entourant une valeur cible moyenne.



*coût pour les véhicules sans le critère*



*coût pour les véhicules avec le critère*

*Figure II.9 - Métrique des rafales*

Cette métrique décrite ci-dessus permet de favoriser la création de rafales possédant des tailles particulières (on les appelle aussi convoi). On cherche à avoir une taille comprise entre minimum et maximum avec une moyenne donnée. Pour toute rafale commencée, si le véhicule n'a pas le critère, la pénalisation sera forte si la taille courante de la rafale est inférieure au minimum requis (en effet on arrête la création de la rafale). Cette pénalisation sera beaucoup moins forte si la taille de la rafale est supérieure au minimum mais inférieure à la moyenne

souhaitée. Elle sera nulle si la taille est supérieure à la moyenne car on souhaite interrompre la rafale au plus tôt. Si le véhicule a le critère, la fonction de pénalisation est inverse car on cherche à avoir des rafales dont la taille est de l'ordre de la moyenne mais avant tout inférieure à la valeur maximale que l'on a fixée.

## 2.11. Conclusion

La modélisation des règles de séquençement utilise principalement trois types de contraintes : les espacements entre critères, les rafales et les dates.

Les contraintes d'espacements regroupent les deux premiers paragraphes (II.2.1 et II.2.2) : la répartition des produits et le respect de ratios. Dans la suite de notre étude, ces deux contraintes seront souvent utilisées en collaboration car elles peuvent être complémentaires. Leur domaine d'utilisation est principalement l'atelier de montage.

Les rafales (II.2.10) sont principalement utilisées dans le cadre de l'atelier de peinture.

Le troisième groupe de contraintes est lié aux dates. Nous avons choisi de regrouper pour des raisons de cohérence les notions de positionnement dans la séquence (II.2.4 à II.2.7) en une seule référence : la position par rapport à une date ou à une position de référence pour chaque véhicule.

Dans notre approche, cette règle peut être perçue de deux façons différentes : soit comme une règle indépendante des autres, soit comme une règle complémentaire des autres.

Dans le cas d'une métrique évaluative cette distinction n'est pas nécessaire : la pénalisation dépend directement de l'écart par rapport à une position souhaitée :

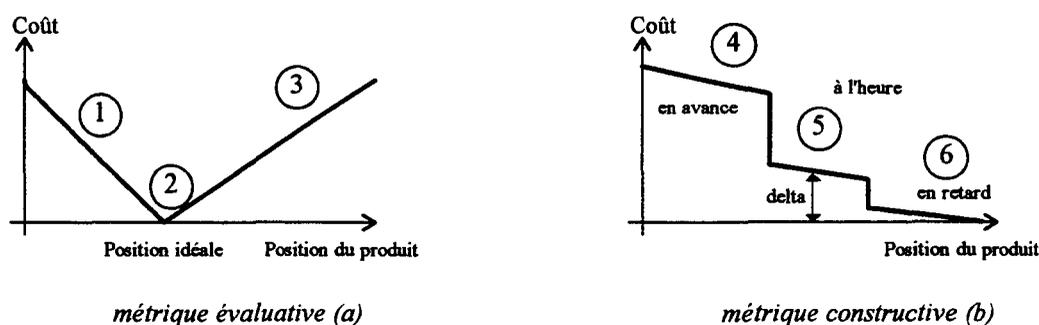


Figure II.10 - Métrique temporelle

Sur la Figure II.10.a, la métrique évaluative pénalise autant les véhicules en avance (1) que les véhicules en retard (3). Seuls les véhicules ayant la position idéale (2) ont un surcoût nul.

Dans le cas d'une métrique constructive (Figure II.10.b), une approche similaire à la précédente peut introduire des effets néfastes dans les algorithmes utilisés. En effet, il est nécessaire d'avoir une fonction en escalier, pour favoriser le placement des véhicules les plus en retard (6), c'est-à-dire ceux les plus pénalisants. D'autre part, l'apparition de surcoût pour les

véhicules à l'heure ne donne pas une note nulle quand la solution est idéale (5) (une dérive de delta apparaît à chaque itération). Kubiak et Sethi [KUB 94] ont montré qu'une telle notation était inadaptée. La solution qui peut être apportée à ce problème est de prendre cette règle en complément des autres : au lieu d'ajouter le coût de cette règle, on multiplie le coût des autres règles par celui-ci.

Le domaine d'application de ce type de règles est très vaste. Il s'applique aussi bien à l'atelier ferrage, qu'à l'atelier peinture ou à l'atelier montage. Nous illustrerons son application dans la quatrième partie.

### **3. Modélisation du problème par contraintes**

#### **3.1. De la notion de charge de travail à celle d'espace**

L'approche du problème au niveau des ratios est moins fine que l'approche utilisée par Bard ou Thomopoulos. Ces derniers cherchent à équilibrer directement la charge de chaque poste de manière fine. Mais les règles de ratios permettent d'agréger la vision de la ligne de production (poste par poste). De plus l'approche actuelle est de considérer les potentiels de lignes en capacité à monter un équipement sur une partie de la production (un ratio de véhicules). Une analogie est possible entre les deux approches. En effet les règles de ratio pour une option consistent à analyser les points critiques et les goulots d'étranglement à un niveau macroscopique. Une fois répertoriés les postes limitant la production d'une certaine option, il reste à les transformer en règles de ratio associé à une option ou critère.

Yano et Bolat [YAN 89] et Yano et Rachamadugu [YAN 91] ont modélisé le problème de l'ordonnancement des produits sur une ligne en revenant au problème avec une seule machine.

Pour avoir une surcharge à un poste dans un temps de cycle, il faut avoir :

$$o_j k_j + b_j m_j \geq (k_j + m_j) t_c$$

où le ratio est "pas plus de  $k_j$  sur  $n_j$ " pour l'option  $j$

$b_j$  = est le temps de fabrication pour une tâche de base

$o_j$  = est le temps de fabrication pour une tâche avec option

$t_c$  = est le temps de cycle

$k_j$  = nombre de produits ayant l'option dans la fenêtre  $n_j$

$m_j$  = nombre de produits de base dans la fenêtre  $n_j$

avec évidemment  $b_j < t_c < o_j$  et  $k_j + m_j = n_j$

On en déduit donc qu'il faut être capable de supporter pour chaque option le ratio suivant :

$$x_j = \frac{k_j}{n_j} \leq \frac{t_c - b_j}{o_j - b_j}$$

Nous pouvons illustrer ce concept par un exemple :

- le temps de cycle est de  $t_c = 1'$
- les temps opératoires sont  $b_j = 30''$  et  $o_j = 3$

il est alors possible de calculer le ratio critique à garantir :

$$x_j = \frac{1' - 0.5'}{3' - 0.5'} = \frac{0.5'}{2.5'} = \frac{1}{5} \Rightarrow Eacc = 4$$

Il faut donc au plus enligner un véhicule ayant le critère sur 5 ou avoir 4 véhicules sans la contrainte entre deux véhicules l'ayant.

Cette notation pour  $x_j$  se simplifie quand il n'y a pas de travail à faire sur les véhicules qui n'ont pas le critère ( $b_j = 0$ ). En effet la formule devient :

$$x_j = \frac{k_j}{n_j} \leq \frac{t_c}{o_j}$$

On peut illustrer ce cas particulier par un exemple :

- le temps de cycle est de  $t_c = 1'$
- le temps opératoire est  $o_j = 3'$

$$x_j = \frac{1'}{3'} = \frac{1}{3} \Rightarrow Eacc = 2$$

C'est le ratio critique qu'il ne faut pas dépasser pour résoudre le problème à une seule machine ou un seul poste. Pour l'ensemble de la ligne, il reste à agréger l'ensemble des critères ainsi que le cas le plus défavorable pour chaque.

### 3.2. Indicateurs de cohérence et de résultats

#### 3.2.1. Indicateurs a priori

Un problème peut apparaître quand on essaye de vérifier la cohérence des règles utilisées dans le problème.

En effet, il faut dans le cas des ratios s'assurer qu'ils sont compatibles avec le volume des options à produire. Il faut que la valeur du ratio qu'on peut fabriquer soit supérieure au ratio à produire. Cette condition doit également être vérifiée si une règle de répartition de produit et

une règle de ratio sont utilisées en même temps. Cette redondance est utile quand l'un des objectifs à atteindre est à la fois le respect des capacités de production et le lissage de la charge des postes critiques. Cette condition n'est pas nécessaire dans l'absolu, les algorithmes utilisés peuvent être robustes, mais les résultats obtenus risquent d'être mauvais car la solution optimale devra nécessairement faire des violations sur les ratios critiques.

En deuxième point, il faut également vérifier la cohérence des critères que l'on veut prendre. C'est à dire des phénomènes d'inclusion et d'exclusion mutuelle entre certaines règles, qui entraînent des contraintes de ratio. Dans le cas d'une exclusion, le ratio devient la somme des deux ratios, dans le cas de l'inclusion, c'est le plus grand qui l'emporte et l'optimisation se fera sur deux critères antinomiques.

Par exemple, une ligne produit deux options O1 et O2, notre problème d'ordonnancement comporte trois règles de ratios : R1, R2 et R3. La règle R1 est un ratio acceptable de l'option O1, la règle R2 est un ratio acceptable de l'option O2, la règle R3 est un ratio acceptable d'une option fictive O3 (le produit a l'option O1 ou l'option O2). Les options O1 et O2 sont exclusives avec  $R1 = 1/2$ ,  $R2 = 1/2$ . La règle R3 devra supporter un ratio de 1 : elle est donc inutile (voire dangereuse en cas de dérives de l'algorithme).

Autre cas néfaste qui peut apparaître en fonction de la demande. Deux options sont produites sur la même ligne : O1 (ratio acceptable  $r1=1/10$ ) et O2 (ratio acceptable  $r2=1/15$ ). La demande est telle que tous les produits ayant l'option O2 ont l'option O1. La ligne n'est pas capable de produire au volume maximal chaque option. Si le volume de O2 est inférieur à  $1/20$ , la production de O1 ne sera pas perturbée, sinon la production de O1 devra suivre celle de O2.

L'exemple précédent illustre le problème de l'optimisation suivant plusieurs critères qui pris individuellement est possible, mais par le fait du croisement de ces critères rend l'optimisation beaucoup plus compliquée voire impossible. Pour cette raison, il peut être utile d'introduire des indicateurs de difficulté à respecter une contrainte.

Pour Comby [COM 96], un indicateur est :  $I_r = \frac{B_r}{A_r} \times \left( \frac{A_r}{B_r} - \frac{Q_r}{Q} \right)$  que l'on appellera par la

suite, indicateur de difficulté à respecter les ratios.

où  $A_r/B_r$  est le ratio à respecter

et  $Q_r/Q$  est le taux de produits à fabriquer.

Une condition nécessaire à respecter est un ratio de  $I_r \geq 0$ . Cette condition n'est pas suffisante car il faut prendre en compte le croisement des contraintes comme dans l'exemple précédent.

Si  $I_r$  tend vers 0, le problème devient très difficile, il faut à chaque choix respecter le ratio sinon toute dérive ne pourra pas être rattrapée : la marge est nulle. En revanche, si  $I_r$  tend vers 1, le problème devient très facile; cela correspond à un taux de production nul qui est facile à respecter.

On peut faire l'analogie avec les espacements :

$$I_r = \left( \frac{Eth - Eacc}{Eth + 1} \right)$$

où  $Eth = Q/Q_r - 1$  correspond à l'espacement de répartition des produits (volume à fabriquer), appelé par la suite Espacement Théorique.

et  $Eacc = B_r/A_r - 1$  correspond au ratio à respecter, appelé par la suite Espacement Acceptable.

On voit réapparaître cette notion dans les métriques pour normaliser les différentes contraintes les unes par rapport aux autres.

Le paramètre  $B_r$  correspond à la taille des fenêtres pour les ratios. Donc il est intéressant d'avoir un ratio réduit pour limiter la taille des fenêtres des ratios. Plus  $B_r$  sera petit plus facile sera l'optimisation des ratios.

### 3.2.2. Indicateurs a posteriori

D'autres indicateurs peuvent être utilisés pour vérifier la qualité du séquençement obtenu, après résolution du problème.

Pour Guerre-Chaley [GUE 95], la notion retenue est le *CV* (Coefficient de Variation) qui est le ratio entre l'écart type et la moyenne réalisée pour chaque critère retenu. C'est une note évaluative pour indiquer si la contrainte est "chahutée" ou non; c'est-à-dire s'il a été difficile de suivre cette contrainte ou non. Cet indicateur ne peut être utilisé qu'a posteriori car il prend en compte l'écart-type effectivement réalisé et qui ne peut être évalué a priori à la différence de la moyenne qui doit être conservée (volume à garantir).

Une autre notion que l'on peut introduire est la notion de capacité du procédé :

*CAP* est le Coefficient d'Aptitude du Processus

$$CAP = \frac{IT}{6\sigma} \quad \text{avec} \quad IT = Ts - Ti$$

*IT* = est l'intervalle de tolérance

$T_i$  = la tolérance inférieure

$T_s$  = la tolérance supérieure

(la valeur  $6\sigma$  correspond aux 99.6% des échantillons d'une distribution normale).

$CPK$  est le coefficient de position de la dispersion de la Production

$$CPK = \min\left(\frac{T_s - \bar{X}}{3\sigma}; \frac{\bar{X} - T_i}{3\sigma}\right)$$

Il doit être le plus grand possible et en tout cas supérieur à 1.

En prenant comme hypothèse que le volume prévu est respecté, l'espacement réalisé moyen est égal à la cible  $E_{th}$ . Dans notre cas, nous n'avons qu'une borne inférieure  $E_{acc}$  qui est l'espacement associé au ratio à respecter; d'où :

$$CPK = \frac{E_{th} - E_{acc}}{3\sigma} \geq 1$$

On peut obtenir ainsi une valeur du Coefficient de Variation  $CV$  que l'on doit obtenir pour avoir un procédé capable :

$$\frac{I_r}{3} \left(1 + \frac{1}{Eth}\right) = \frac{Eth - E_{acc}}{3 \times Eth} \geq \frac{\sigma}{Eth} = CV$$

La difficulté à garantir un critère donné à l'avance peut être évaluée.

Supposons un critère dont le ratio à respecter est 1/3 donc l'espacement acceptable associé est  $E_{acc} = 2$ . Le volume à produire correspond à un taux de 1/4, ce qui donne un espacement théorique de  $E_{th} = 3$ .

On calcule  $I_r = 0.25$  et en faisant l'hypothèse que le volume prévu est atteint, on peut calculer une borne du Coefficient de Variation que l'on doit garantir.

$$CV \leq \frac{I_r}{3} \left(1 + \frac{1}{Eth}\right) = \frac{1}{9}$$

$$\Rightarrow \frac{\sigma}{Eth} \leq \frac{1}{9} \quad \text{ou} \quad \sigma \leq \frac{1}{3}$$

On est donc capable d'avoir une idée des résultats que l'on peut obtenir.

## 4. Les méthodes d'optimisation retenues

### 4.1. Introduction

Delaval [DEL 97] propose une revue détaillée et une critique des méthodes rencontrées dans la littérature sur le Car Sequencing Problem.

Bel et Bensana [BEL 96] proposent une revue d'ensemble d'études et de réalisations de systèmes d'information industriels utilisant des techniques d'Intelligence Artificielle, à base de

systèmes experts, de systèmes à bases de connaissances, de programmation par contraintes, d'algorithmes génétiques, etc. Ce bilan partiel met en évidence les moyens par lesquels certaines techniques d'Intelligence Artificielle peuvent être intégrées dans des composantes décisionnelles d'applications industrielles.

Les principes liés aux méthodes d'optimisation que nous souhaitons retenir doivent prendre en compte le caractère multi-critères de notre problème (plusieurs types de paramètres comme la date, les rafales et les nombreuses contraintes) et également le caractère réactif du problème (répondre en un temps relativement court, voire être interruptible). Dans la littérature, différentes approches ont été étudiées sur le séquençement des lignes d'assemblage à modèles mélangés ou sur l'ordonnancement de ces lignes.

articles	Méthodes	Critères d'optimisation
<b>Simulations</b>		
[ARC 66] [BHA 93] [MAC 73]	Simulation Monte-Carlo Simulation Simulation	temps d'attente et longueur de ligne interférences entre opérateurs temps d'attente et travail utile
<b>méthodes exactes</b>		
[BEN 94] [BOL 94] [NG 94] [BAR 92] [BAR 94] [WIT 85]	Branch&Bound Branch&Bound Branch&Bound programmation linéaire programmation linéaire en nombres entiers Branch&Bound et heuristiques Recherche Tabou énumérations cycliques	violation de contraintes de précedence temps de réglage et surpotentiels(coût main d'oeuvre) lisser la consommation des composants longueur de la ligne et débit longueur de ligne et consommation de composants  maximiser le débit et minimiser les encours
<b>méthodes de construction progressive et heuristiques</b>		
[CAS 94] [DEL 97] [GUE 95] [INM 92] [KUB 94] [MIL 89a] [MON 83] [OW 88] [PLE 93] [SUM 92] [THO 67] [WES 64] [YAN 89] [YAN 91] [CHA 86] [DAR 75] [SHI 90] [OHL 94]	construction progressive recherche par faisceaux filtrée construction progressive à un pas positionnement par rapport à cible construction progressive à un pas construction progressive à un ou deux pas et heuristiques construction progressive à un pas (GCM) recherche par faisceaux filtrée construction progressive par morceaux construction progressive (GCM et M-A3H2) construction progressive construction progressive cyclique construction progressive avec évaluation du reste construction progressive (gloutonne) heuristique et shortest path heuristiques construction progressive heuristiques placement	optimisation des ratios optimisation des ratios : <b>Cadencement</b> ratio(espacements) ou rafales : <b>Cadencement</b> lisser le taux de production lisser le taux de production lisser le taux de production  lisser la consommation des composants  minimiser les retards et avances lisser la consommation des composants lisser la charge ou la consommation  oisiveté et sous/sur-charges temps de fabrication minimiser les sur charges  travaux non terminés coût total de production longueur de la ligne minimiser le temps de fabrication faisabilité - goulot d'étranglement
<b>méthodes par affinements successifs</b>		
[PAR 86] [BUR 87]	Automated Reasoning (programmation par contraintes) regroupements successifs hiérarchisés	respect d'espacement  temps de réglage longueur des rafales

[CHE 91]	recuit simulé	ratio et rafales
[COM 96]	recuit simulé	Beaucoup de critères
[DEL 96]	recuit simulé	ratio : <b>Cadencement</b>
[DIN 88]	Programmation par contrainte	respect de ratio
[GRO 89]	procédure de recherche locale par permutation	minimiser l'encours de production
[OKA 79]	affinement par permutations ou insertion	éviter arrêt de la ligne

Figure II.11 - Tableau synthétique des articles sur le séquençement

Il est à noter que peu d'études s'intéressent à un problème dynamique.

De cette description rapide et synthétique de la littérature, quatre familles de méthode apparaissent. La première est la simulation. Elle permet d'évaluer les solutions trouvées pour vérifier si elles sont bonnes. Le problème est la durée de ces simulations.

La deuxième famille est celle des méthodes exactes. L'approche est de type énumérative.

La troisième famille est celle des heuristiques constructives. Fondée sur une approche du type énumérative, pour laquelle toutes les solutions sont évaluées. Ces méthodes consistent à construire la liste au fur et à mesure par des optimisations locales.

La quatrième famille est celle des affinements successifs. A partir d'une séquence globale initiale, des transformations permettent d'obtenir une nouvelle séquence voisine qui peut-être évaluée.

Après une description rapide des méthodes exactes, nous allons décrire les deux dernières familles de méthodes: leurs caractères spécifiques, leurs principes, leurs avantages et leurs inconvénients. Ce sont celles qui seront retenues pour la suite des travaux

#### 4.2. Simulations

La recherche de la solution se fait à l'aide de l'évaluation des choix effectués par une simulation du fonctionnement de l'usine. Le meilleur résultat de simulation est alors retenu [BHA 83].

#### 4.3. Méthodes exactes

La recherche de solution s'appuie sur le fait que pour explorer un espace de recherche de grande taille, il est préférable d'adopter une stratégie qui vous aide à naviguer. Pour comprendre la résolution informatisée de ce problème, il faut faire une analogie entre l'espace de recherche et un arbre. La recherche commence à la racine et chaque point de décision (ou noeud) crée plusieurs branches qui vont croître de façon indépendante. Si l'optimum est unique, trouver la branche optimale nécessite une organisation précise et sans faille.

Les méthodes exactes correspondent à une énumération contrôlée de l'ensemble des solutions possibles. Dans le cas d'une séquence de  $N$  produits, l'ensemble des solutions est au nombre de  $N!$ . Les méthodes de Recherche Opérationnelle permettent de réduire cette combinatoire en fonction du problème [DAW 95].

#### 4.3.1. Branch & Bound

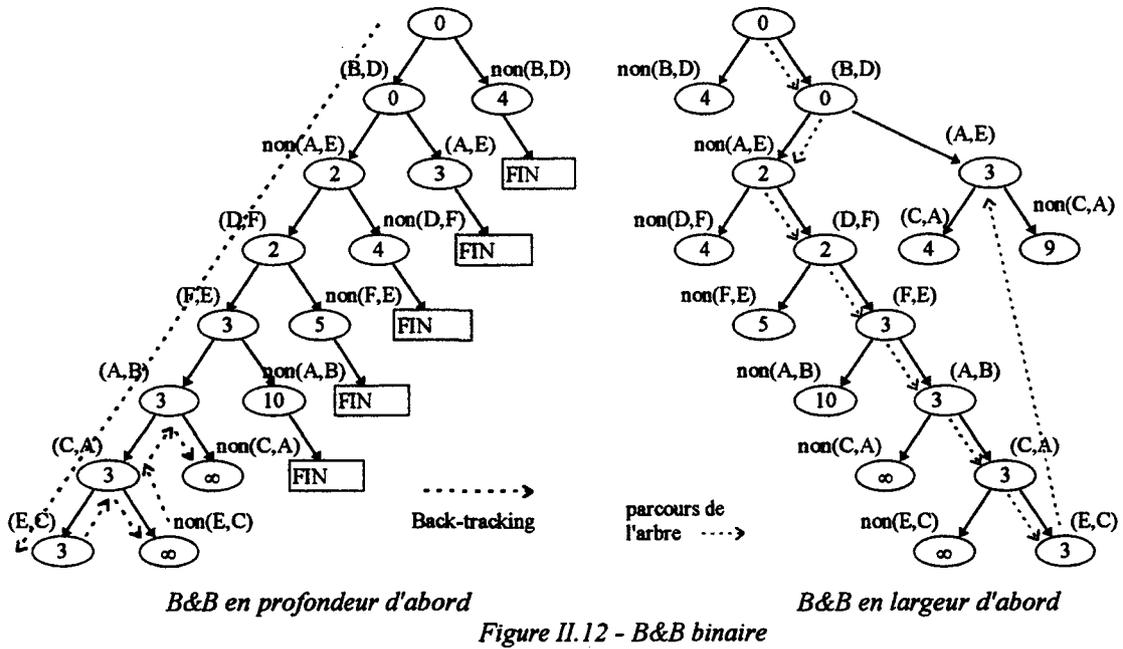
La première approche est le Branch and Bound (ou Procédure de Séparation et d'Evaluation en français). L'objectif de cette méthode est de parcourir l'arbre de recherche suivant deux stratégies concurrentes : soit en profondeur d'abord (Depth-first Search), soit en largeur d'abord (Breadth-first search).

La recherche en profondeur d'abord consiste à trouver le chemin direct vers une solution (proche d'une méthode gloutonne) puis à effectuer des retours en arrière (Back-tracking) pour remettre en cause les choix faits à chaque niveau de l'arbre, pour des variantes potentiellement moins mauvaises que la solution déjà trouvée.

La recherche en largeur d'abord consiste à explorer toutes les branches possibles à un noeud donné, puis de repartir du meilleur noeud et de continuer jusqu'à atteindre une feuille qui sera la meilleure solution.

Pour illustrer ce problème, prenons l'exemple du PVC (problème du voyageur de commerce) déjà réduit. C'est le problème d'un voyageur de commerce qui doit visiter un certain nombre de villes, sans visiter deux fois la même tout en minimisant le trajet à parcourir. Cela consiste à trouver le trajet hamiltonien reliant l'ensemble des points A, B, C, D, E et F, avec la matrice des distances suivante :

	A	B	C	D	E	F
A	-	2	4	2	0	2
B	2	-	4	0	7	5
C	1	6	-	9	0	6
D	4	2	2	-	4	0
E	0	0	0	3	-	0
F	3	2	3	2	0	-



Le parcours d'arbre exhaustif (c'est-à-dire qu'à chaque noeud tous les arcs sont explorés), se fait de haut en bas et de gauche à droite (voir le dessin ci dessous). Le parcours démarre du point B et à chaque noeud il passe par un point, auquel est associé un coût (celui de la distance depuis B en passant par tous les points précédents) par exemple pour le premier noeud du niveau 3, le parcours de B à D puis à F vaut 0.

Les flèches indiquent, quand elles sont continues, le parcours effectué dans l'arbre et, quand elles sont pointillées, une évaluation d'un noeud qui n'est pas gardé car trop mauvais.

Pour l'arbre en largeur d'abord, le procédé est le même. Le parcours démarre du point B et à chaque noeud, tous les arcs sont explorés. Puis, la recherche repart du noeud qui a le coût le plus faible et qui n'a pas encore été développé.

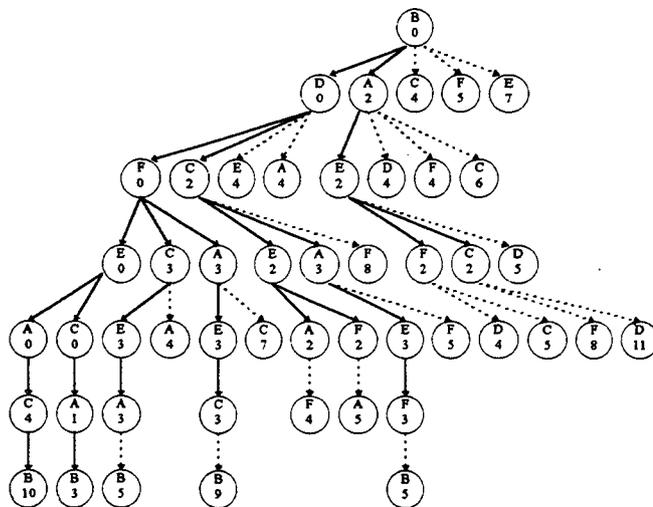


Figure II.13 - B&B non binaire en profondeur d'abord

Dans cet exemple relativement simple, le B&B en profondeur d'abord va trouver la bonne solution au début et le Back-tracking va permettre de s'assurer que cette solution était la meilleure. Le B&B en largeur d'abord va trouver tous les sous-parcours dont le coût est inférieur à celui de la solution optimale et développer ces embryons de solution avant de découvrir la solution optimale en atteignant la première feuille. Le B&B en largeur d'abord est plus rapide car pour un parcours exhaustif, il développe le minimum de noeuds. En revanche, le B&B en profondeur d'abord nécessite moins de ressources informatiques : il ne garde en mémoire qu'un nombre polynomial de noeuds de l'arbre  $O(N^2)$  soit en développement binaire ou en développement exhaustif, à la différence du B&B en largeur d'abord qui doit garder en mémoire l'ensemble des noeuds non développés en  $O(N!)$ .

#### 4.3.2. Propagation par contraintes

Cette approche étudiée par Parello et al. [PAR 86] consiste à parcourir l'espace de recherche en éliminant tous les débuts de solutions qui ne permettent pas de respecter un ensemble de règles (par exemple des ratios). Cette approche peut être intéressante quand une solution existe, sinon il faut se résoudre à faire de la relaxation de contraintes. Des difficultés apparaissent également lorsque les critères décrivant l'optimum sont concurrentiels (poids relatifs).

#### 4.3.3. Programmation linéaire

L'objet de la programmation linéaire, appelée aussi optimisation linéaire, consiste à résoudre le type de problèmes suivants :

Avec  $N$  variables indépendantes  $x_1, \dots, x_N$  (noté  $\vec{X}$ ), il s'agit de maximiser la fonction (de coût) :  $z = \vec{A}_0 \cdot \vec{X}$ , avec les contraintes :

$$\begin{aligned} \vec{X} &\geq 0, \\ [A_1] \cdot \vec{X} &\leq \vec{B}_1, \text{ où } \vec{B}_1 \geq 0, \\ [A_2] \cdot \vec{X} &\geq \vec{B}_2, \text{ où } \vec{B}_2 \geq 0, \\ [A_3] \cdot \vec{X} &= \vec{B}_3, \text{ où } \vec{B}_3 \geq 0. \end{aligned}$$

Le vecteur répondant aux contraintes et maximisant la fonction objectif est appelé vecteur faisable optimal. En 1948, Dantzig publia la "Méthode du Simplexe", qui est un moyen d'organiser la procédure d'optimisation. Une série de combinaisons est effectuée qui augmente la fonction objectif à chaque pas, et le vecteur faisable optimal est atteint en un nombre fini d'itérations (soit la dimension de  $\vec{X}$ , soit la somme des dimensions des  $\vec{B}_i$ ).

Cette méthode peut-être étendue à des problèmes entiers (à l'aide d'un parcours d'arbre excluant les solutions non entières trouvées pour le vecteur  $\vec{X}$ ). Elle permet alors de résoudre des problèmes d'ordonnancement en prenant en compte les charges de travail aux différents postes de travail. En revanche cette approche est difficilement applicable dans le cadre d'une ligne d'assemblage où il est nécessaire de déterminer une séquence avec une vitesse de lancement constante.

#### 4.4. Heuristiques

L'énumération des solutions même contrôlée est incapable si le problème est de trop grande taille de répondre dans un temps imparti. Pour cela, des méthodes approchées du B&B ont été développées et sont appelées communément B&B tronqués. Elles consistent à se limiter dans le parcours d'arbre de manière artificielle, en ne gardant en général qu'un certain nombre de choix différents à chaque profondeur.

Dans le cas d'un B&B exhaustif, à chaque noeud de l'arbre, le nombre de choix est celui de chaque produit non encore séquencé. La combinatoire est donc de  $N!$  pour une séquence de taille  $N$  (c'est le nombre de noeuds à parcourir dans le pire des cas). Les heuristiques consistent à ne garder à chaque niveau de l'arbre qu'un nombre limité de noeuds (ceux qui sont potentiellement les meilleurs).

Castelain et Al. [CAS 94] proposent dans un parcours d'arbre en profondeur d'abord de limiter à chaque noeud-père, le nombre de noeuds-fils. Ow et Al. [OW 88] et Dawn [DAW 95] décrivent le fonctionnement du BeamSearch ou recherche par faisceaux. Cette méthode, dérivée du B&B en largeur d'abord, consiste à ne garder que les  $P$  meilleurs parcours déjà trouvés à cette profondeur. La détermination des meilleurs parcours peut être définie de deux manières différentes : estimer ou non le reste de la séquence qui n'a pas encore été choisie. Cette approche ne garantit pas la meilleure solution car on fait un élagage sans Back-tracking. En revanche, elle élimine deux gros problèmes du B&B en largeur d'abord : la complexité est limitée en  $O(NP)$  (d'où le temps de calcul) et surtout la taille de l'arbre à garder en mémoire est devenue polynomiale en  $O(NP)$ .

L'intérêt des méthodes arborescentes est le Back-tracking qui permet d'éviter des verrouillages (dead locks) liés à des configurations bizarres. On se permet de remettre en cause des choix passés. En revanche, il faut dans la pratique utiliser des B&B tronqués pour limiter le temps de calcul. Dans ce cas, il faut donc trouver un juste équilibre entre l'élagage nécessaire pour limiter la combinatoire et la qualité de la solution que l'on souhaite obtenir.

#### 4.5. Heuristiques de construction progressive

Le principe de la construction progressive de séquence consiste, comme son nom l'indique, à construire une séquence pas à pas, produit par produit, à partir d'un lot de produits à séquencer.

A partir d'une position  $k$  atteinte (c'est-à-dire la séquence de la position 1 à  $k - 1$  construite par le principe), on va choisir parmi les  $N - k + 1$  produits restants à séquencer celui qui permet d'obtenir la meilleure sous-séquence 1 à  $k$ . Le problème est donc de déterminer une métrique (constructive) qui permet d'évaluer le début de la séquence. Diverses approches apparaissent dans la littérature.

##### 4.5.1. Méthode à un pas

Cette méthode consiste à évaluer le meilleur véhicule parmi ceux non séquencés et à le positionner à la place  $k$ . Elle s'apparente à un B&B en profondeur tronqué qui ne prend que le meilleur choix à chaque niveau (appelé aussi méthode gloutonne).

Le principe de cette méthode utilisée par Monden [MON 83] et Guerre-Chaley [GUE 95] est le suivant :

<b>Etape 1 :</b>	$k = 1$ et $S_{k-1} = \{P_1, P_2, \dots, P_N\}$ , l'ensemble des $N$ produits à séquencer
<b>Etape 2 :</b>	A la position $k$ dans la séquence, on positionne le produit $P_i^* \in S_{k-1}$ tel que $f(P_i^*) = \min(f(P_i)), \forall P_i \in S_{k-1}$ où $f(x)$ est une métrique constructive qui permet d'évaluer la séquence.
<b>Etape 3 :</b>	on met à jour l'ensemble des véhicules non séquencés : $S_k = S_{k-1} - \{P_i^*\}$
<b>Etape 4 :</b>	Si $S_k = \emptyset$ alors l'algorithme est terminé, sinon reprendre à l'étape 2 avec $k \leftarrow k + 1$ .

Figure II.14 - Goal Chasing Method

Cette méthode est intéressante car elle est très rapide, mais elle est myope et donc souvent peu robuste car elle n'explore qu'une partie des solutions possibles.

Sa complexité est faible  $O(N^2)$  par rapport au  $N!$  d'une méthode énumérative.

##### 4.5.2. Méthode à P pas

La méthode à  $P$  pas se rapproche beaucoup de la méthode précédente. Mais au lieu de ne choisir un produit qu'avec une vue locale, le choix va être effectué en fonction de la meilleure sous-séquence ( $k$  à  $k + P - 1$ ), celle qui a le plus faible coût pour cette sous-séquence.

C'est cette méthode qu'utilisent Miltenburg et Sinnamon dans [MIL 89b] pour leur heuristique H2.

Le principe est le suivant :

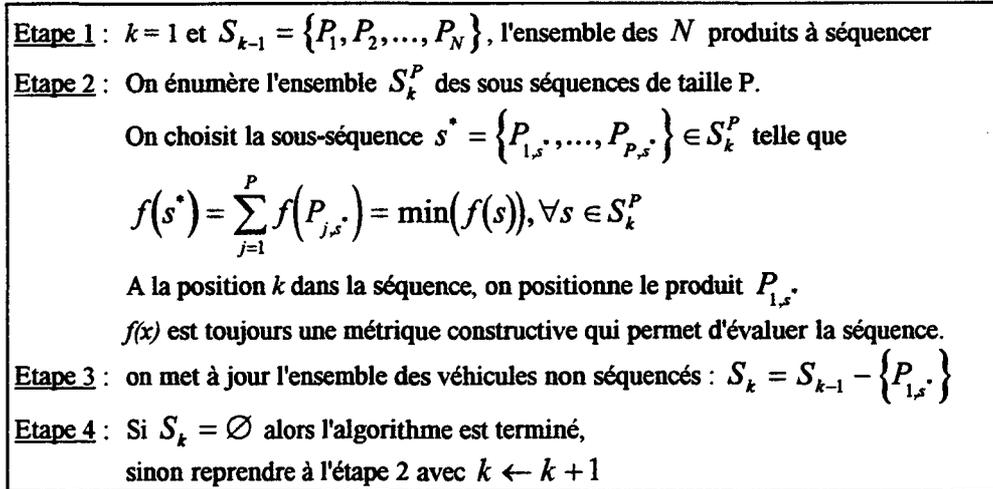


Figure II.15 - Méthode à  $P$  pas

Cette méthode est intéressante car même si elle est moins rapide que la précédente, elle élimine la plupart des problèmes de robustesse car elle est moins myope.

Sa complexité est toujours polynomiale  $O(N^{P+1})$  et plus rapide qu'une méthode énumérative en  $N!$ .

#### 4.5.3. Autres approches

Yano et Bolat ont proposé [YAN 89] une méthode de construction progressive plus évoluée par rapport aux précédentes en évaluant le séquencement du reste des produits à séquencer.

Pour cela il utilise le principe suivant :

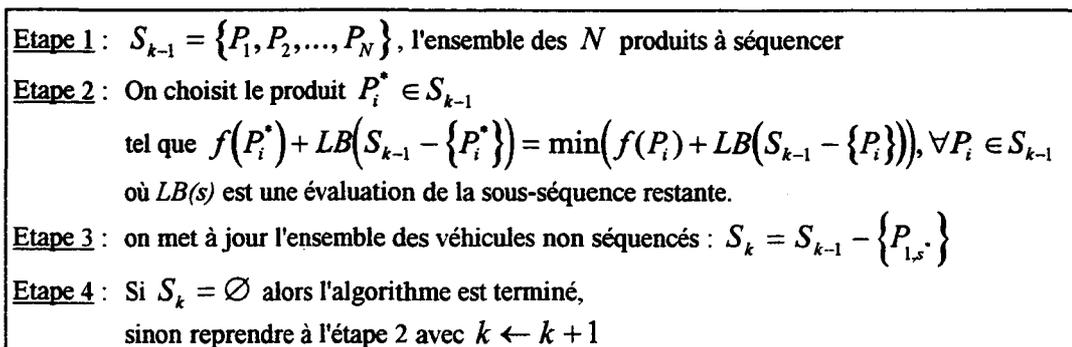


Figure II.16 - Méthode à un pas avec évaluation de la borne inférieure [YAN 89]

Dans les problèmes d'ordonnancement, cette méthode est plus connue sous le nom de  $A^*$  [WIN 84].

Pour Yano et Bolat, le choix de la fonction  $LB(s)$  est une valeur approchée de la borne inférieure du travail utile. Ils gardent donc la solution la plus prometteuse. Les plus pessimistes peuvent garder la solution la moins mauvaise en utilisant pour calculer  $LB(s)$  une méthode gloutonne (une méthode à un pas calculable en  $O(N^2)$ ).

#### 4.6. Heuristiques par affinements successifs

Les heuristiques par affinements successifs consistent à améliorer une solution initiale quelconque en parcourant son voisinage et en retenant la ou les meilleures solutions. Dans ces méthodes trois types de problèmes se posent.

Le premier est la liste initiale. Les méthodes sont en général assez robustes pour accepter une solution initiale quelconque, mais la performance de l'algorithme s'en trouve modifiée de manière significative. Cette solution initiale peut être obtenue en prenant la liste des produits en vrac, s'il n'y a pas de limitations dans la faisabilité de la séquence. Elle peut être obtenue également par une heuristique de construction progressive très rapide (méthode gloutonne ou Goal Chasing Method), ou par une génération aléatoire.

Le deuxième problème est l'algèbre associée à l'espace de recherche. En effet, la solution initiale est l'un des éléments de l'ensemble des séquences faisables du problème. Cet ensemble doit être connexe, c'est à dire qu'à partir de n'importe quelle séquence, toute autre séquence peut être atteinte. L'algèbre va correspondre à la forme des voisinages de l'espace. Elle est définie par  $(\mathcal{E}, \mathcal{D})$ , où  $\mathcal{E}$  est l'espace et  $\mathcal{D}$  est la distance. Un voisinage  $\mathcal{V}$  d'un point  $x_0$  est défini comme suit :  $\mathcal{V}(x) = \{x \in \mathcal{E} \text{ tq } \mathcal{D}(x_0, x) < \varepsilon, \forall \varepsilon\}$ . En général, dans les méthodes prenant en compte des listes, le voisinage est défini comme étant l'ensemble des séquences ayant subi une transformation élémentaire. Cette transformation peut être évoluée (une sélection est faite à ce niveau parmi les voisins) mais dans ce cas le risque de non connexité de l'espace apparaît : toutes les séquences ne seront pas nécessairement atteintes et en particulier la meilleure solution.

Le troisième problème est lié à la condition d'arrêt de telles méthodes. En effet, la qualité de la solution obtenue va être améliorée de manière itérative, mais il est nécessaire de s'arrêter à un moment donné. A la différence des heuristiques de construction progressive ou des B&B tronqués pour lesquels le cheminement et le nombre d'itération sont bien définis, les heuristiques par affinements successifs n'ont pas de conditions d'arrêt bien définies. L'utilisateur peut choisir comme critère le temps de calcul : au bout d'un certain temps, le calcul est

interrompu et la solution retenue sera la meilleure trouvée jusque là. Le deuxième critère d'arrêt peut être lié à la qualité de la solution obtenue : soit le coût de la solution est proche de la borne inférieure si elle a pu être calculée (écart inférieur à  $x\%$ ), soit il n'y a pas eu d'amélioration depuis un nombre donné d'itérations.

#### 4.6.1. Méthode du gradient

La première de ces méthodes est celle du gradient. Elle consiste à choisir dans le voisinage de la solution initiale la meilleure des solutions. La condition d'arrêt est l'arrêt des améliorations quand on a atteint un minimum qui peut être local.

Pour une telle méthode, la solution trouvée est le premier minimum local rencontré par la procédure. La méthode ne permet pas de sortir de ce premier puits rencontré. Elle est donc rapide mais en général peu adaptée aux environnements de recherche chaotiques.

#### 4.6.2. Recuit simulé

Le recuit simulé (Kirkpatrick et al. [KIR 83]) consiste à accepter des dégradations locales de la qualité de la solution pour sortir des minimums locaux.

Décrivons l'algorithme du recuit simulé :

- Choisir une configuration initiale : séquence  $s_0$
- Tant que le critère d'arrêt n'est pas atteint (système figé (1)) faire :
  - tant que la fin du palier n'est pas atteinte (équilibre thermodynamique (2)) faire :
    - faire une transformation élémentaire de la séquence (3)
    - calculer la variation du coût  $\Delta C$
    - faire le test d'acceptation de Metropolis (4)
      - si  $\Delta C \leq 0$ , la transformation est acceptée
      - sinon, la transformation est acceptée avec une probabilité  $\exp(-\Delta C/T)$
  - diminuer la température  $T$ .
- La solution est atteinte

Figure II.17 - Recuit simulé

Cette méthode converge de manière asymptotique vers l'optimum global si certaines conditions sont vérifiées. Le livre de Siarry et Dreyfus [SIA 87], ou l'article de Bonomi et Lutton [BON 89], donneront de bonnes précisions sur le fonctionnement de cette méthode et le réglage de certains paramètres. Chew [CHE 91] l'a déjà utilisé avec succès pour les problèmes de séquencements dans le monde automobile. Néanmoins précisons certains points.

Les critères d'arrêt (remarques (1) et (2)) correspondent à une analogie avec la physique statistique. En général, la taille des paliers (2) est une constante assez grande et le nombre de paliers est déterminé pour avoir un temps de réponse compatible avec l'application.

Les transformations élémentaires (3) sont de plusieurs sortes. En général, il s'agit soit de permutations entre deux éléments de la liste [COM 96], soit d'insertions d'un élément de la liste [GUE 95], voire de combinaisons entre les deux [OKA 79]. Dans certains cas la notion de voisinage pose problème. Comby [COM 96] et Okamura [OKA 79], pour réduire le nombre de voisins et accélérer la convergence, ont stipulé que l'un des éléments de la permutation devait être un produit correspondant à une surcharge. Comby a montré qu'un choix trop dirigiste (échange entre une période trop chargée et une période peu chargée) amène à une performance dégradée de l'algorithme. Guerre-Chaley propose le principe des insertions car le voisinage est moins "chaotique". Après des tests, il s'avère plus intéressant de prendre une notion de permutation sur l'ensemble de la liste, ou sur un tronçon. Le fait de garantir la connexité de l'espace de recherche (et en particulier la réversibilité) donne des résultats de meilleure qualité mais avec une vitesse de convergence moins importante.

Dans la règle d'acceptation de Metropolis, le paramètre important est la température. Cette notion de température permet de garantir un taux de dégradation de la solution courante de l'algorithme. Au début, l'algorithme doit être capable de sortir des minima locaux, et la température doit être importante : quand  $\max(\Delta C)/T \rightarrow 0$  le taux d'acceptation tend vers 1. A la fin quand  $\max(\Delta C)/T \rightarrow \infty$ , le taux d'acceptation tend vers 0, la méthode se transforme en une méthode du gradient, pour trouver le minimum global.

L'utilisation des paramètres standards, définis par Siarry, donne toujours de bons résultats. Ces résultats garantissent la robustesse de l'algorithme.

Le problème majeur que l'on rencontre avec cet algorithme est le réglage des paramètres (température initiale et paliers de températures) pour avoir un bon compromis entre la vitesse de convergence et la qualité du résultat. Ce réglage semble dépendre du type de problème. Delaval [DEL 96] réussit à obtenir de bons résultats dans un problème de cadencement avec un recuit simulé.

#### **4.6.3. Méthode Tabou**

La méthode Tabou proposée par Glover [GLO 86], consiste à choisir, parmi les voisins de la solution courante, la meilleure solution. Pour éviter les bouclages, la liste des solutions tabou est constituée de l'ensemble des  $K$  dernières solutions visitées. La convergence de cette méthode est assurée pour un paramètre  $K$  infini.

Décrivons l'algorithme tabou :

- Choisir une configuration initiale : séquence  $s_0$
- Initialiser la liste tabou
- Tant que le critère d'arrêt n'est pas atteint, faire :
  - générer les solutions voisines
  - déterminer la meilleure solution  $s$  non tabou
  - $s$  devient la solution courante
  - mettre à jour la liste tabou avec  $s$ .
- La solution est atteinte, c'est la meilleure rencontrée depuis le début.

Figure II.18 - Méthode Tabou

Deux problèmes coexistent pour cette méthode. Le premier est la définition de la liste tabou. Tout d'abord, il faut régler la taille de cette liste, et également sa modélisation (en général, on ne garde en mémoire que les permutations que l'on a effectuées).

Le deuxième problème, qui est peu souvent abordé, est la définition du voisinage. En effet, il doit toujours permettre une connexité de l'espace de recherche et cela peut poser des problèmes de temps de réponse. Par exemple, prendre l'ensemble des permutations entre deux éléments quelconque d'une séquence de taille  $N$  donne  $N(N-1)$  voisins. Si l'on prend les permutations de deux éléments adjacents le nombre des voisins est encore de  $2(N-1)$ .

Dans le cas du CSP, où il n'y a pas une sélectivité facile à établir au niveau des voisins, cette méthode semble peu adaptée.

#### 4.6.4. Algorithme génétique

L'idée sous-jacente des algorithmes génétiques est de reproduire l'évolution naturelle des organismes en respectant notamment une loi de survie où les plus forts survivent et se reproduisent. Dans une population, chaque individu est caractérisé par une empreinte génétique issue de la *recombinaison* de l'empreinte de ses deux parents, obtenue par *croisement* (ou "*cross-over*") ou par *mutation*. Le croisement correspond à la reproduction de l'espèce et permet de transmettre à la génération suivante les caractères héréditaires qui permettent une bonne sauvegarde. La mutation est une évolution de l'empreinte génétique qui évite la dégénérescence de la population.

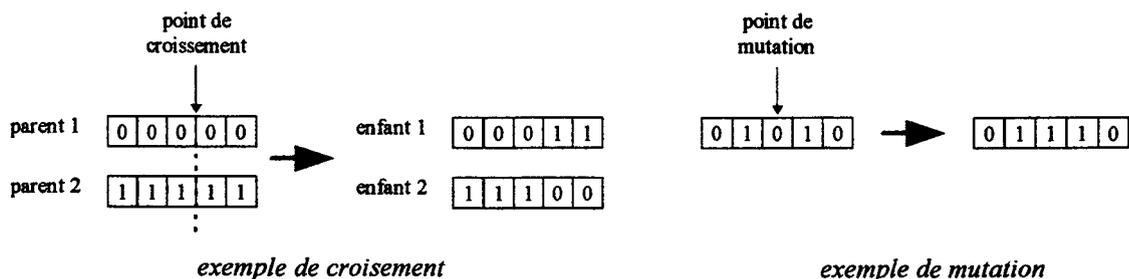


Figure II.19 - Croisements et mutation pour un algorithme génétique

Au niveau algorithmique et au niveau de l'implémentation, les opérateurs de croisement et de mutation sont des algorithmes qui agissent sur la structure des données, associée aux

individus. Le renouvellement de la population est effectué à chaque itération (de l'algorithme génétique) par la création d'une nouvelle génération, qui va créer et détruire des individus, en effectuant une sélection "naturelle".

Décrivons l'algorithme génétique :

- Définir un codage du problème
- Créer une population initiale de  $N$  individus:  $P(0) = x_1, x_2, \dots, x_N$  et  $t = 0$
- Tant que le critère d'arrêt n'est pas atteint, faire :
  - Evaluer chaque individu :  $F(x_i), \forall x_i \in P(0)$
  - Sélectionner  $N$  individus de  $P(t)$  (à l'aide d'une roulette pondérée) pour créer  $S(t)$
  - Recombinaison : création de  $S(t+1)$ 
    - Pour chaque paire de  $S(t)$ , croisement avec la probabilité  $p_{crois}$
  - Mutation :
    - Pour chaque individu de  $S(t+1)$ , mutation avec la probabilité  $p_{mut}$
- Une population de solutions est obtenue.

Figure II.20 - Algorithme génétique

Pour mettre en œuvre un algorithme génétique, il faut coder le problème sous forme génétique: c'est-à-dire sous forme de chromosomes en général binaires, créer une population initiale, une fonction d'évaluation  $F(x_i)$  et des paramètres pour les opérateurs de sélection, de croisement et de mutation.

Dans le cadre des problèmes d'ordonnancement peu d'études ont été menées en utilisant les algorithmes génétiques. Caux, Pierreval et Portmann proposent dans [CAU 94] un codage possible pour les problèmes d'ordonnancement.

Cette approche semble néanmoins peu adaptée au CSP, en raison des difficultés à modéliser le problème et à créer les opérateurs de croisement et de mutation. Une étude a cependant été menée [BAR 93].

## 5. Conclusion

Les méthodes exactes permettent de garantir la meilleure qualité de la liste globale, au détriment d'un temps de calcul très long voire non borné (c'est-à-dire très grand par rapport à nos critères). Les heuristiques associées à ces méthodes permettent de limiter le temps de calcul ou de le borner, en revanche la qualité de la liste est légèrement dégradée. Les problèmes de petite dimension sont le domaine d'application de ces méthodes qui permettent d'obtenir l'optimum global de ces petits problèmes.

Les méthodes de construction progressive vont permettre d'obtenir pour un temps de calcul faible une séquence de bonne qualité (mais résultant d'une succession d'optimisations locales). Leur domaine d'application sont les problèmes de grande dimension, peu complexes, dont on

souhaite un résultat correct de manière immédiate. Le temps de réponse peut être réglé en fonction de la profondeur de calcul utilisée.

Les méthodes par affinements successifs permettent d'obtenir des séquences de meilleure qualité que les méthodes de construction progressive car elles optimisent l'ensemble de la liste. Mais en contrepartie, elles nécessitent un temps de calcul supérieur qui peut être facilement maîtrisé (la méthode est interruptible à chaque itération). Leur domaine d'utilisation se situe entre les deux précédents : pour un problème de grande taille, on souhaite un résultat de très bonne qualité, tout en s'assurant que le temps de calcul n'est pas excessif.

Voici un tableau récapitulatif des performances que l'utilisateur peut espérer de chacune de ces méthodes pour notre Car Sequencing Problem :

Méthode	Qualité du résultat	Temps de calcul	Robustesse	Ressources mémoire
Construction Progressive	bon	excellent	très bonne	très bonne
B&B tronqué	bon	très bon à bon	bonne	bonne à mauvaise
B&B	très bon	mauvais	bonne	bonne à mauvaise
Programmation par Contraintes	bon	bon	mauvaise	variable
Méthode du Gradient	moyen	très bon	très bonne	très bonne
Recuit Simulé	bon	bon	moyenne	bonne
Méthode tabou	bon	bon	moyenne	moyenne
Algorithme génétique	moyen	bon	moyenne	mauvaise

Figure II. 21 - Comparaison des différentes méthodes

Certaines de ces méthodes sont adaptées à des optimisations globales en considérant l'ensemble des ordres de fabrication à produire. Elles nécessitent en général beaucoup de ressources (temps CPU, mémoire) mais donnent de bons résultats. Les autres méthodes sont adaptées à des optimisations locales; la séquence de produits peut être construite pas à pas pour obtenir une liste complète. Cette optimisation peut être effectuée avec un retour arrière partiel ou sans. Ces méthodes peuvent être rapides si des approximations dans l'ensemble des solutions sont tolérées, et en général, elles ne requièrent que peu de ressources. De plus, elles sont robustes et réactives aux aléas car elles ne prennent en compte qu'une partie du problème.

Dans les deux parties suivantes, nous allons utiliser ces méthodes pour résoudre nos problèmes de séquençement et de cadencement. Pour les problèmes de séquençement prévisionnel, nous avons retenu, comme méthode globale, le recuit simulé et, comme méthodes locales, le Goal Chasing et la recherche par faisceaux. Pour les problèmes de cadencement qui demandent une réactivité beaucoup plus importante, la méthode de construction progressive a été retenue. En effet, elle permet un juste compromis entre la qualité du résultat, la robustesse de la solution obtenue et surtout le temps de calcul imparti à l'algorithme.

## *Partie 3*

### SEQUENCEMENT



## **Partie 3 : Séquencement**

### **1. SEQUENCEMENT DES PRODUITS**

#### **1.1. Les objectifs du séquencement**

L'objectif du séquencement d'une ligne d'une usine consiste à créer une séquence de référence. Cet ordonnancement est effectué en statique et une fois par période de référence (en général une journée de production). Il doit garantir le respect d'un maximum de contraintes liées aux critères des véhicules. Ces contraintes (lissage des critères, capacité de production et rafales) ont été décrites dans le chapitre précédent.

Dans le cas d'un problème industriel, de nombreuses contraintes apparaissent parfois concurrentes.

#### **1.2. Le problème posé**

Le problème que l'on souhaite résoudre est un problème d'ordonnancement sur une ligne à modèles mélangés. Il est composé d'une part d'un ensemble d'ordre de fabrication et d'autre part des contraintes de séquencement décrites dans le chapitre précédent. Les ordres de fabrications sont des véhicules possédant ou ne possédant pas un certain nombre de critères. Avec ces ordres de fabrication, l'algorithme doit créer une séquence de véhicules dont les critères respectent au mieux les contraintes de séquencement. Dans ce problème, on ne prend pas en compte l'accessibilité des véhicules ni leur déplacement à travers l'usine.

### **2. Approche de résolution proposée**

Pour résoudre ce genre de problème, différentes approches sont possibles : des méthodes arborescentes (Goal Chasing, Branch and Bound tronqué) ou des méthodes par voisinages (Recuit simulé, méthodes tabou), comme décrites dans le chapitre précédent.

Nous allons tester deux méthodes relativement courantes dans la littérature pour ce genre de problème : le Goal Chasing et le Recuit Simulé. Le Goal Chasing (répartition de composant) a été retenu de préférence à M2H3 [MIL 89b] car les produits étant très différenciés, il est difficile de les regrouper en classes d'équivalence. En effet dans l'exemple industriel que nous étudions (cf. III.3), il y a 46 classes dont la taille varie de un millier de produits à quelques uns. Dans son étude, Guerre-Chaley [GUE 95] a répertorié 25 classes différentes et 90% des véhicules étaient concentrés dans seulement 5 classes. D'autre part, les classes ne sont pas

représentatives de tous les croisements de critères. Parmi les méthodes de voisinages, le recuit simulé a été retenu pour ses qualités de robustesse et pour sa facilité de mise en oeuvre.

Pour finir, nous proposons une méthode mixte qui va essayer de concilier les avantages et les faiblesses de ces deux méthodes (rapidité et qualité du résultat) pour obtenir une solution de bonne qualité en un temps de réponse faible [BAR 98b].

## 2.1. Les méthodes d'optimisation retenues

### 2.1.1. Heuristiques de construction progressive

La méthode retenue est le Goal Chasing décrit dans la partie II [MON 83]. Cette méthode suppose que tous les critères sont des composants dont la vitesse de consommation doit être constante.

L'algorithme peut démarrer à un instant défini arbitrairement comme l'origine, mais il ne prend pas alors en compte les produits déjà présents sur la ligne (production de la veille ou de la semaine passée et non encore terminée). Comby [COM 96] propose une adaptation pour prendre en compte la dérive induite par les derniers véhicules engagés lors du calcul précédent.

L'algorithme mis en oeuvre correspond à la description du chapitre II.2.1.1. La métrique associée est légèrement différente des applications traditionnelles, car chaque critère va être pondéré en fonction de son importance.

Elle s'exprime sous la forme :

$$f(k) = \sum_{c=1}^{n_c} P_c \cdot (k \cdot V_c - X_{ck})$$

où  $k$  est la position courante dans la recherche.

$V_c = N_c/Q$  est la vitesse de consommation du composant  $c$ .

$X_{ck}$  est le nombre de produits ayant le composant  $c$  parmi les  $k$  premiers produits.

$P_c$  est le poids relatif du composant  $c$ .

Les résultats sont obtenus très rapidement. Il faut une dizaine de secondes pour séquencer 3000 véhicules. En revanche, comme prévu, ces résultats ne sont pas très bons pour le respect des ratios ou des dépassements d'espacement acceptable car il cherche à lisser, en moyenne, la consommation.



La permutation consiste à choisir au hasard deux véhicules et à les échanger. Cela correspond à deux insertions successives.

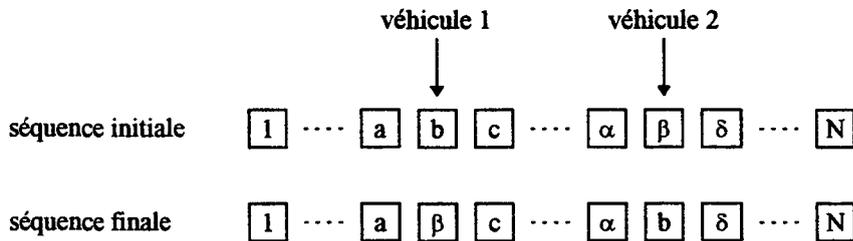


Figure III.2 - Exemple de permutation

Cette transformation est plus difficile à mettre en oeuvre que la précédente, car de nombreuses limites peuvent exister dans les permutations. En revanche d'après la littérature, les résultats sont meilleurs.

#### 2.1.2.2. Fonction de coût ou métrique

Pour que l'algorithme du recuit soit efficace, une condition nécessaire est la création d'une procédure rapide de calcul du coût d'un voisin. Le temps de calcul de cette procédure doit être indépendant de  $N$  (la taille de la liste à séquencer avec  $C$  contraintes).

Le coût d'une séquence  $s'$  voisine de  $s$  obtenue par la transformation  $t$  est calculé de la façon suivante :

$$f(s') = f(s) + g(t, s \rightarrow s')$$

où  $f$  est la fonction d'évaluation d'une séquence, avec un temps de calcul en  $O(N \cdot C)$

et  $g$  est la procédure d'évaluation de la transformation  $t$  de  $s$  en  $s'$  avec un temps de calcul en  $O(C)$ . Elle peut être décomposée en deux :

$g(t, s \rightarrow s') = h(t, s') - h(t, s)$  où  $h$  est la fonction de coût de la séquence sur le tronçon concerné par la transformation  $t$ .

Il faut donc une métrique  $f$  pour évaluer la liste initiale et une métrique  $h$  pour évaluer l'impact local d'une transformation.

Deux approches sont possibles. La première globale agrège les données en utilisant les espacements entre critère et en vérifiant le respect des ratios ou des espacements acceptables. La seconde, locale, évalue le coût de chaque élément de la séquence (approche semblable à une construction progressive). La deuxième approche a deux avantages. Elle permet une comparaison plus précise avec un algorithme de construction progressive et elle permet de mieux définir les éléments de la séquence concernés par les transformations. En contrepartie, elle nécessite plus de mémoire pour être implémentée.

Ayant retenu la deuxième approche, la métrique "h" d'évaluation des modifications consiste à trouver tous les éléments de la séquence dont les paramètres sont perturbés par la transformation.

Pour cela, la fonction "f" va calculer le coût de chaque produit dans la séquence. Lors d'une transformation, le coût de la liste peut être modifié pour chaque contrainte. Dans le cas d'une permutation, si les deux produits ont la contrainte ou si les deux ne l'ont pas, il est évident que le coût de la séquence ne change pas à cause de cette contrainte. En revanche si l'un des véhicules a la contrainte et l'autre non, la séquence se trouve modifiée par cette permutation. La modification va concerner au moins un nombre de véhicules correspondant à la taille de la fenêtre glissante associée (cf. Figure III.3) pour le respect des ratios. Pour le lissage de la charge, l'influence va apparaître pour tous les véhicules se situant entre le véhicule permuté et le véhicule suivant ayant la contrainte. Cette influence sur la séquence apparaît évidemment aux deux points de la permutation.

Dans l'exemple ci-dessous, où un ratio de 1/3 doit être respecté, trois fenêtres (c'est le dénominateur du ratio) au niveau du point de permutation sont influencées. De même le coût de lissage d'un nombre de véhicules variable est influencé. En général, ce nombre est faible (de l'ordre de l'espacement moyen entre deux véhicules ayant la contrainte). Cependant dans le cas de listes initiales très perturbées, il dépend de la configuration locale de la séquence.

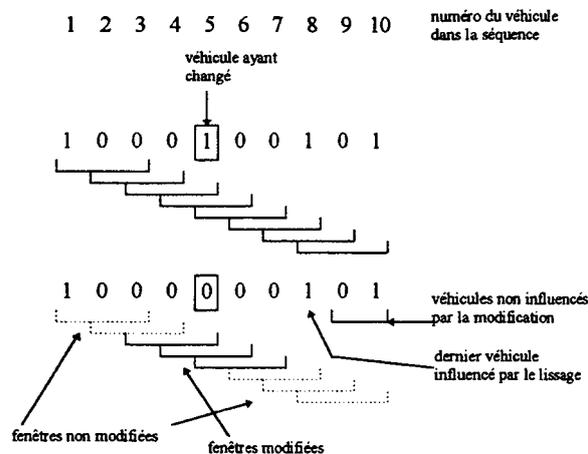


Figure III.3 - Exemple d'une modification dans une liste respectant un ratio de 1/3

Le cinquième véhicule de la séquence a changé. Il avait le critère, le nouveau ne l'a plus. Pour vérifier le respect des ratios, seules les trois fenêtres comprenant les véhicules 3-4-5, 4-5-6 et 5-6-7 peuvent être modifiées. Le véhicule "8" est le premier véhicule après la modification

à avoir le critère, il est donc le dernier véhicule à être influencé par la modification. Donc le coût de seulement 4 véhicules change : les "5", "6", "7" et "8".

Dans le cas d'une insertion, le comportement est à peu près le même. Il est néanmoins antisymétrique, car au niveau du véhicule choisi, une place est perdue et au point d'insertion, une place est créée. Il s'agit donc comme dans l'exemple ci-dessous de déterminer les véhicules dont le coût est changé par la modification (en fait les deux modifications élémentaires).

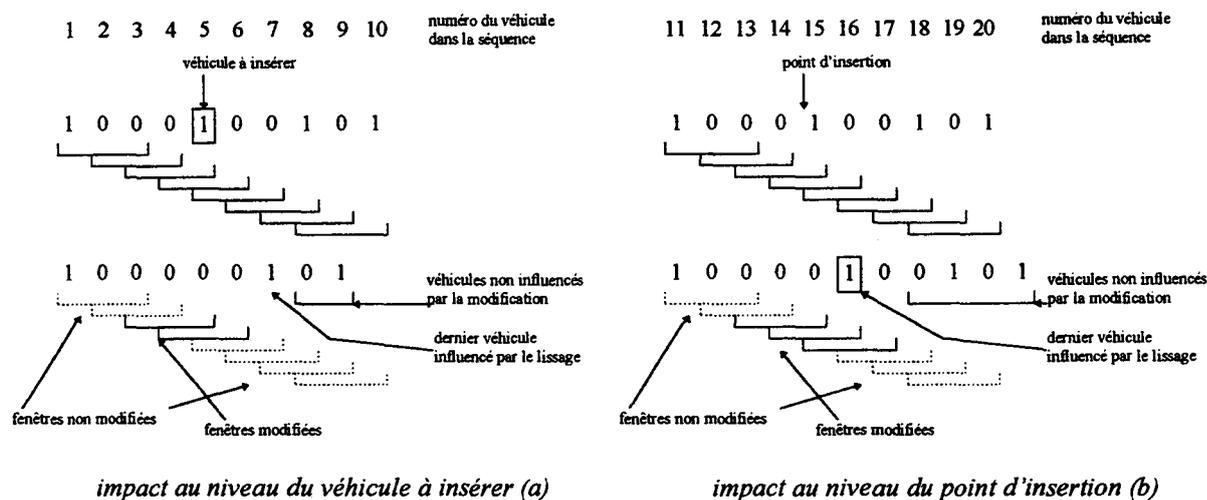


Figure III.4 - Exemples de transformations élémentaires : insertion

Dans l'exemple de la Figure III.4, le véhicule "5" est retiré de la liste pour être inséré entre les véhicules "14" et "15". Dans la première partie (a), les deux fenêtres 3-4-5 et 4-5-6 sont modifiées et le véhicule "7" est le dernier à être affecté par la modification. Le coût des autres véhicules n'est pas modifié (à part un décalage d'une place). Dans la deuxième partie (b), les fenêtres 13-14-15, 14-15-16 et 15-16-17 sont modifiées. Le dernier véhicule influencé par la modification pour le lissage est le "16". Nous avons donc 6 véhicules dont le coût est modifié par l'insertion : "5", "6" et "7" d'une part et "15", "16" et "17" d'autre part.

### 2.1.2.3. Configuration initiale

La configuration initiale peut être obtenue de différentes manières, comme nous avons pu le voir dans la partie précédente.

Une génération aléatoire peut être retenue.

Une autre solution consiste à prendre les véhicules séquencés dans l'ordre où l'algorithme les reçoit : c'est en général l'ordre dans lequel les commandes ont été prises. Cette approche, qui pourrait être comparée à la précédente, s'avère être radicalement contradictoire, car le flux de donnée arrivant ne répond pas à un processus aléatoire sans mémoire. En effet, l'extraction des véhicules des portefeuilles d'ordres de fabrication provient du traitement des offres

commerciales (il peut y avoir des commandes de flotte, de série spéciale, de regroupement de commande), et ce traitement n'est pas aléatoire.

Une troisième approche consiste à créer une liste initiale à l'aide d'un algorithme ultra rapide et simple. Le résultat n'est pas très bon en absolu, mais il est nettement meilleur en qualité que celui obtenu par les deux précédentes approches. Ces algorithmes sont en général des algorithmes gloutons. Dans notre approche, nous avons retenu le Goal Chasing de Monden [MON 83]. L'intérêt de cette approche est de concilier temps de réponse et qualité de la liste initiale. Le résultat obtenu permet de gagner de la réactivité sur l'ensemble de l'algorithme de recuit avec une solution initiale qui essaye de répartir les différentes contraintes sur l'ensemble de la séquence.

Dans ce style d'approche, on trouve également des algorithmes plus sophistiqués mais qui prennent plus de temps. Nous reviendrons sur ce genre d'algorithme à la fin de cette partie.

#### 2.1.2.4. Plan de recuit

Les plans de recuit sont les différents réglages des paramètres de l'algorithme de recuit simulé. Comme décrit dans la partie précédente, ils sont au nombre de quatre : la température initiale, le critère d'arrêt, la longueur des plateaux (ou chaîne de Markov) et la loi de décroissance de la température.

Dans un premier temps, les choix retenus ont été ceux préconisés par Siarry [SIA 87]. L'intérêt d'une telle approche permet de tester la robustesse de l'algorithme par rapport à notre problème. En effet, l'algorithme de recuit simulé a des performances (à la fois en qualité de la solution et en temps de réponse) qui dépendent du plan de recuit. Il faut donc s'assurer qu'il existe des plans de recuit qui permettent une certaine indépendance entre les paramètres à régler et la qualité des résultats obtenus.

Pour cela, nous avons retenu pour  $T_0$ , température initiale du recuit une valeur qui permet une acceptation d'une violation la plus grave avec une probabilité de 1/3. Cette approche est un bon compromis entre n'importe quelle modification et une solution initiale que l'on ne souhaite pas trop dégrader.

Le critère d'arrêt de l'algorithme est fixé par la durée maximale de l'algorithme. Dans un premier temps, le critère d'arrêt consiste à atteindre une température finale donnée, ce qui correspond à un nombre de paliers donnés et à un temps de réponse borné.

La longueur des paliers influe aussi sur le temps de calcul. En effet, la combinaison des deux donne le nombre de transformation que l'algorithme effectue, et donc le temps de réponse de l'algorithme. Siarry propose une longueur constante proportionnelle au maximum de

transformations possibles à partir d'un état donné. Cette approche donne pour une liste de taille 1000, près de un million de permutations possibles, d'où une taille de palier de la même longueur. Par la suite, nous avons essayé des critères qui ne garantissent plus un temps de réponse, mais qui indiquent qu'une certaine stabilité a été atteinte après un certain nombre d'itération sans améliorations. Cette approche doit donner de meilleurs résultats mais le temps de réponse de l'algorithme ne peut être garanti, ce qui peut provoquer des problèmes dans certains cas d'utilisation (cf. Delaval [DEL 97]) : dans le cas du cadencement le temps de calcul est un paramètre critique.

Le dernier paramètre à régler est la loi de décroissance de la température. Ce paramètre est important car il force la convergence de l'algorithme vers un optimum local. En effet, plus la température diminue et tend vers 0, plus la probabilité d'accepter une solution dégradée (et donc la possibilité de sortir d'un puits) diminue, l'algorithme de recuit se transforme alors en une descente stochastique (méthode du gradient où le voisin est choisi de manière aléatoire).

### ***2.1.3. Méthode combinée***

Un troisième type d'approche que nous allons utiliser consiste à coupler les deux approches précédentes pour obtenir un bon compromis entre le temps de réponse et la qualité de la solution obtenue.

En effet, les méthodes par construction progressive sont très rapides, mais la qualité de la solution engendrée est souvent médiocre en fonction des problèmes rencontrés. En outre, ces méthodes ne sont pas interruptibles (elles ne peuvent être interrompue en cours d'exécution).

Les méthodes par affinements successifs permettent d'obtenir des solutions meilleures en un temps plus long ce qui nécessite en général un bon réglage des paramètres pour avoir un fonctionnement optimal. L'intérêt de ces méthodes est d'être interruptibles à tout moment : l'algorithme peut donc fonctionner longtemps et ainsi améliorer la solution, la possibilité d'arrêter la recherche reste entière.

L'objectif de cette troisième approche est donc d'utiliser les forces des deux méthodes pour palier aux faiblesses respectives de l'autre. Elle consiste à obtenir dans un premier temps une bonne solution à l'aide d'une méthode par construction progressive. Cette première partie est incompressible au niveau temps de réponse. En contrepartie, elle donne une solution initiale à la méthode par affinement successif. Cette solution initiale ne doit pas nécessairement avoir un coût minimum vis-à-vis de la métrique retenue mais doit principalement garantir une bonne répartition de toutes les contraintes sur l'ensemble de la séquence.

Dans un deuxième temps, la méthode par affinements successifs va améliorer cette première solution en ayant comme cible la métrique retenue. L'algorithme commençant sa recherche dans une partie de l'espace intéressante, la convergence est plus rapide. Cette partie de l'algorithme est plus lente, mais est interruptible.

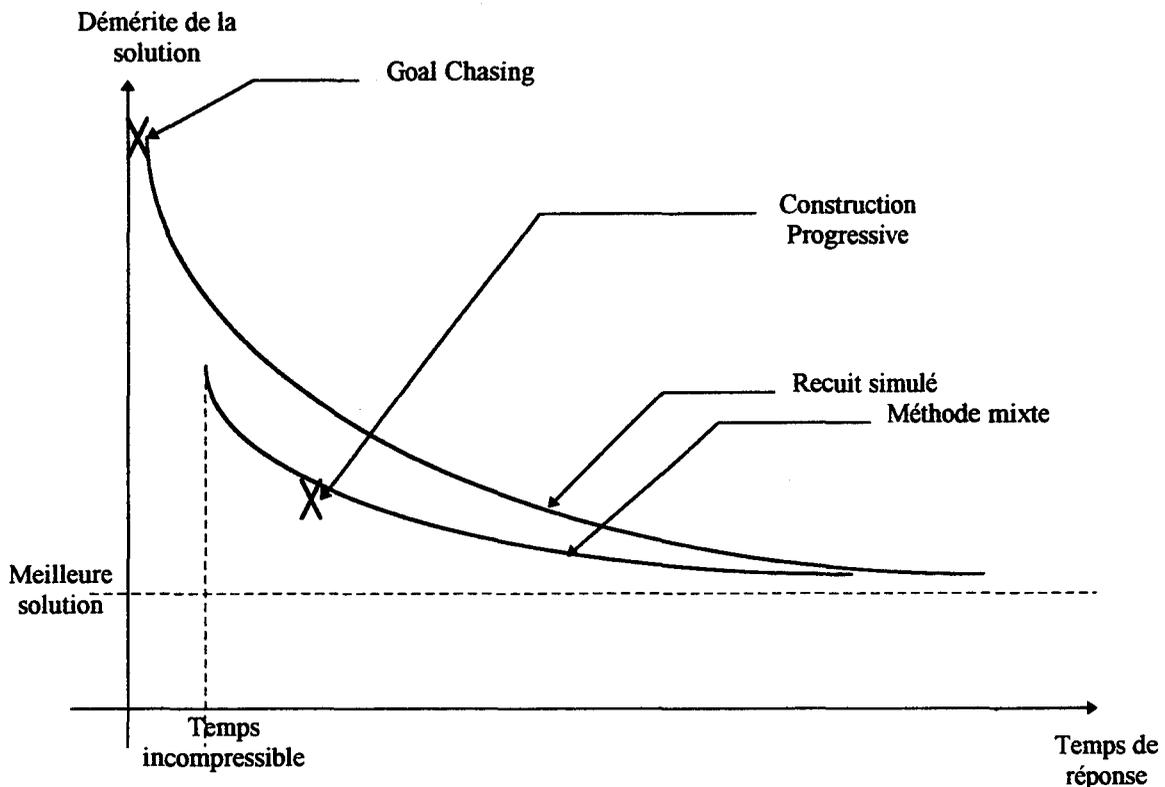


Figure III.5 - Comparaison entre les diverses méthodes

## 2.2. Description de la méthode combinée

### 2.2.1. Introduction

Nous allons décrire une méthode qui va permettre d'obtenir rapidement un séquencement pour une ligne à modèles mélangés.

Dans la littérature, les différentes méthodes existantes pour générer des ordonnancements globaux pour un flux de production n'utilisent pas de combinaisons entre les méthodes de construction progressive et les méthodes par affinements successifs.

Pour résoudre ces problèmes, j'ai été amené à étudier des méthodes décrites par Miltenburg [MIL 89a]. L'approche décrite y est intéressante, car la complexité de l'algorithme est réduite par l'extraction de sous-listes répétitives : on effectue alors un ordonnancement cyclique. L'optimisation de la séquence se réduit à celle d'une séquence de taille plus petite et compatible avec des méthodes énumératives ou donnant de très bons résultats avec des méthodes par construction progressive.

La difficulté liée à une telle approche est la configuration du problème : s'il est répétitif avec un grand dénominateur commun entre les classes, cette méthode est applicable. En revanche, quand ce n'est pas le cas, l'utilisateur se trouve devant une optimisation beaucoup plus complexe qu'il ne pourra pas résoudre de manière convenable.

Dans le cas du CSP, les classes de véhicules (véhicules ayant les mêmes contraintes) sont souvent très disparates. Certains types de véhicules peuvent être très fréquents comme des séries limitées, et d'autres n'apparaissent qu'épisodiquement, correspondant à des options particulières.

Le concept de sous-listes à optimiser localement a été gardé. En effet, il est concevable et intéressant d'utiliser une méthode énumérative ou s'en approchant sur une petite séquence. Il faut néanmoins que chaque sous-liste soit représentative de l'ensemble de la liste pour garantir le lissage des contraintes. Pour cela une partition de la liste en sous listes doit respecter au mieux le volume de chaque critère, les ratios à respecter restant les mêmes. Dans un deuxième temps les sous-listes sont optimisées les unes après les autres pour générer une séquence complète, et finalement une optimisation globale est effectuée par recuit simulé pour améliorer l'ensemble de la liste.

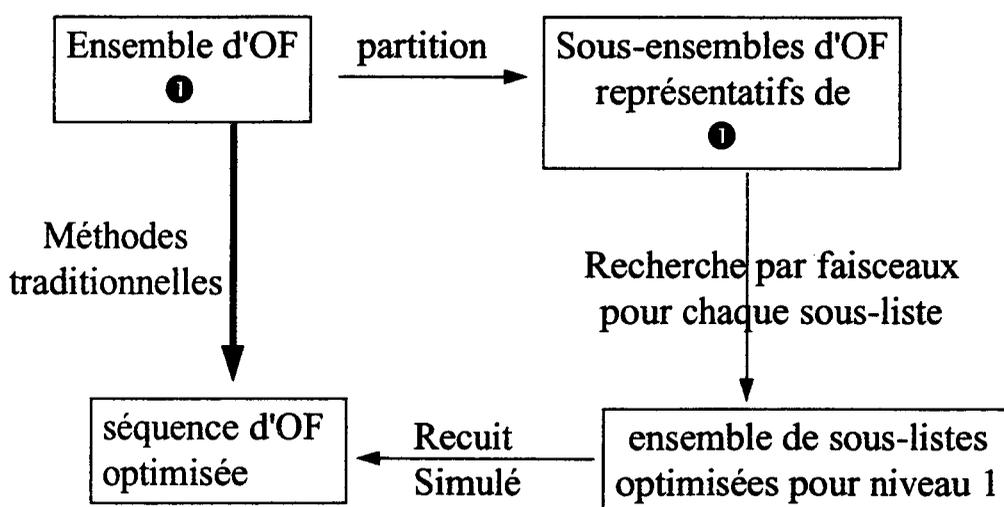


Figure III.6 - Enchaînement des procédures de la méthode mixte

### 2.2.2. La partition

Le paramètre à régler pour la partition en sous-listes est la taille de ces sous-listes. En effet des sous-listes trop petites seront faciles à optimiser mais ne pourront pas être représentatives de l'ensemble des éléments à ordonnancer. Et de même si les sous-listes sont trop grandes, elles seront trop difficiles (au niveau temps de calcul) à optimiser, mais l'ensemble des sous-listes sera plus représentatif du problème.

L'exemple de Miltenburg peut être repris pour illustrer cette question :

La demande est : 6000 unités pour le produit A; 6000 unités pour le produit B; 1000 unités pour le produit C.

Dans ce cas idyllique, la taille des sous-listes est 13, avec 6 produits A, 6 B et 1 C.

Supposons maintenant que la demande fluctue, 6500 unités pour le produit A; 5300 unités pour le produit B; 1200 unités pour le produit C. Les ratios sont encore relativement proches des précédents et faciles à décomposer, mais le PGCD est 100 et non plus 1000 comme auparavant. La taille des sous-listes devient 130 avec 65 produits A, 53 B et 12 C. En revanche pour avoir des sous-listes de taille 13, il est nécessaire d'arriver au compromis suivant :

A	6	6	6	6	6	7	7	7	7	7
B	6	6	6	5	5	5	5	5	5	5
C	1	1	1	2	2	1	1	1	1	1

Cette approche donne donc 10 sous-listes pour un regroupement semblable au 130 décrits précédemment.

Dans le cas général, la situation est néanmoins délicate.

Un autre cas, qui est rarement traité, est le caractère cyclique de ces sous-séquences qui peut poser problème au niveau des joints entre listes (la thèse de Camus [CAM 97] pourra apporter des éclaircissements et des éléments de réponse).

### 2.2.3. Méthodes de partition

La première partie de l'algorithme consiste donc à diviser l'ensemble des ordres de fabrication en sous-groupes représentatifs de l'ensemble de la liste. Notre problème consiste à respecter des ratios de critères et à minimiser le nombre de rafales. Cette question se différencie donc un peu de celle approchée par Miltenburg.

Deux méthodes ont été retenues pour réaliser cette partition.

La première est une méthode gloutonne à un pas. Elle consiste à répartir les critères de manière équitable entre les différentes sous-listes.

Le problème est constitué d'une liste de  $N$  véhicules ayant  $K$  contraintes, et pour chaque contrainte  $k$ ,  $N_k$  véhicules ayant cette contrainte.

La partition de  $\{V_1, \dots, V_N\}$  en  $L$  sous-listes ayant les mêmes proportions de contraintes va être définie par des vitesses de consommation  $V_k = N_k/L$  et  $V = N/L$ . Chaque sous-liste devra avoir les paramètres suivants : (la notion  $\lceil \cdot \rceil$  est l'arrondi à la valeur supérieure)

nombre de véhicules dans la liste  $l$  :

$$N_l = \left[ l \times V - \sum_{i=1}^{l-1} N_i \right]$$

nombre de véhicules dans la liste "l" ayant la contrainte k :

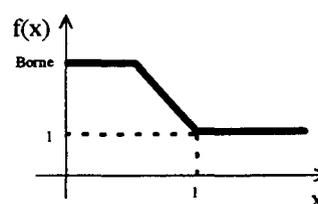
$$N_{k,l} = \left[ l \times V_k - \sum_{i=1}^{l-1} N_{k,i} \right]$$

Par la suite à l'aide d'une méthode gloutonne chacune des listes est remplie à son tour par le véhicule ayant le plus de contraintes (pondérées) sans dépasser les quantités requises.  $N_{k,l}$  et  $N_l$

Les coûts sont pondérés par l'importance relative des contraintes et par la difficulté "relative" à s'approcher du quota :

$$P'_k = P_k \times f(\text{Esp}_{\text{théorique}} - \text{Esp}_{\text{acceptable}}) \quad \Leftrightarrow$$

pondération



fonction de pondération  $f(x)$

Figure III.7 - Importance relative des contraintes

$P_k$  est l'importance initiale de la contrainte.  $P'_k$  est l'importance pondérée de la contrainte en fonction de la difficulté à effectuer l'enlèvement. Si la différence entre l'espacement théorique moyen et l'espacement acceptable qu'il faut garantir est supérieure à 1, aucune complication n'est à noter pour cette contrainte : son importance reste inchangée. En revanche si cette différence est inférieure à 1 et tend vers 0, la contrainte est d'autant plus difficile à respecter : son importance relative augmente (tout en restant bornée).

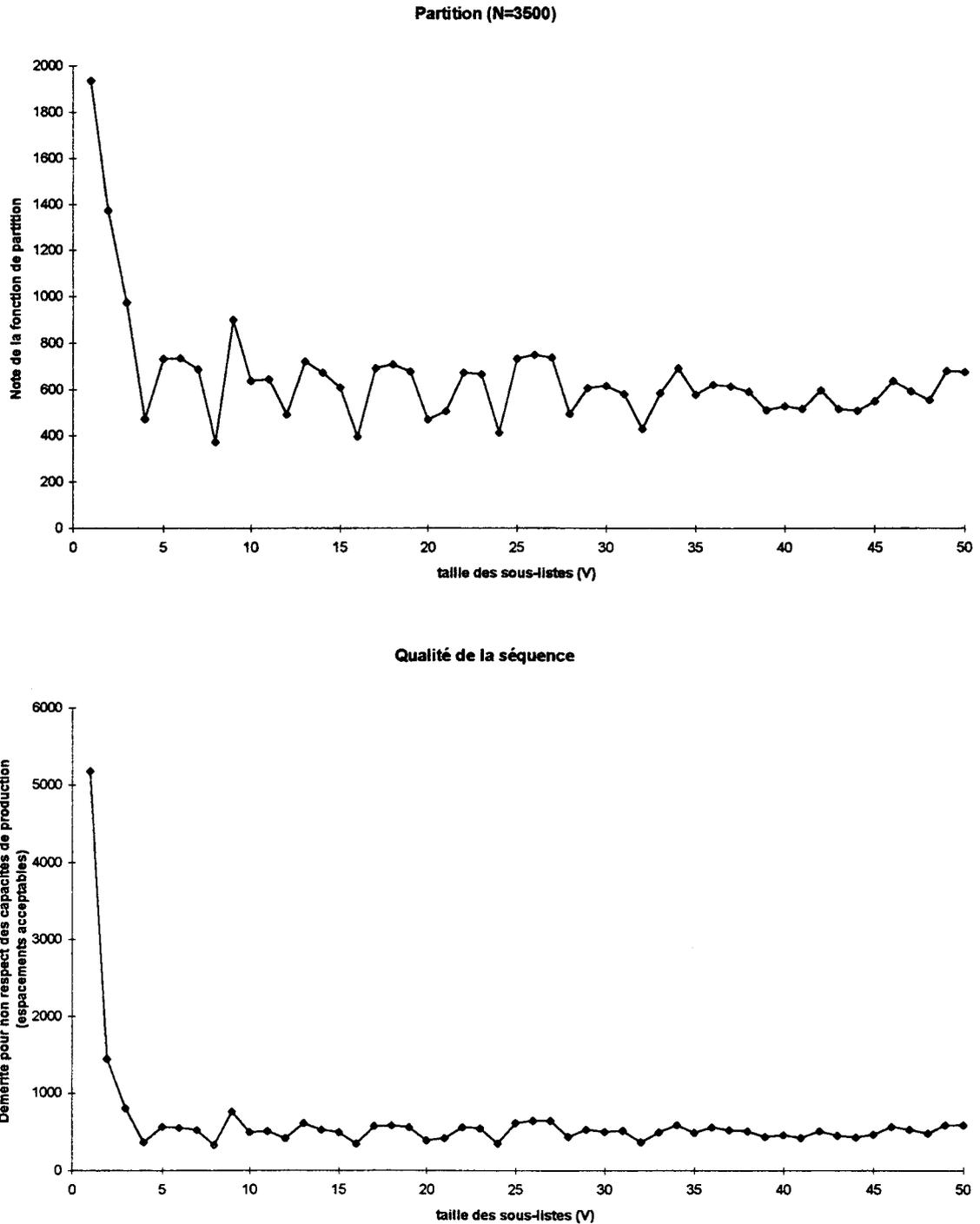
Ensuite, la partition effectuée est notée en fonction des différences entre les valeurs obtenues ( $V_{k,l \text{ obtenue}}$ ) et celles souhaitées ( $N_{k,l}$ ):

$$\text{Note} = \sum_{l=1}^L \sum_{i=1}^K (V_{k,l \text{ obtenue}} - N_{k,l})^2$$

Le dénombrement des valeurs obtenues ( $V_{k,l \text{ obtenue}}$ ) s'effectue sous-liste par sous-liste ("l") et contrainte par contrainte ("k").

De manière expérimentale, les minimums locaux de la fonction *Note* (en fonction de la taille des sous-listes) donnent une bonne partition pour l'ensemble du problème : le résultat de l'optimisation finale est alors un minimum de cette appréciation (cf. graphique ci dessous).

Mais il est à noter également que le choix de la taille des sous-listes influence peu la qualité finale sauf si ce choix est trop petit : dans l'exemple illustré ci-dessous la taille minimum à choisir est 4.



*Figure III.8 - Influence de la taille de la partition sur la qualité de la séquence*

On peut remarquer sur la Figure III.8 que les deux courbes ont la même tendance. Les minimums de la fonction de partition sont les minimums du démerite pour non respect des capacités de production obtenu avec une recherche par faisceaux.

Donc une méthode pour choisir une partition consiste à évaluer la fonction *Note* en fonction de la largeur et à développer les solutions les plus prometteuses. L'inconvénient d'une telle

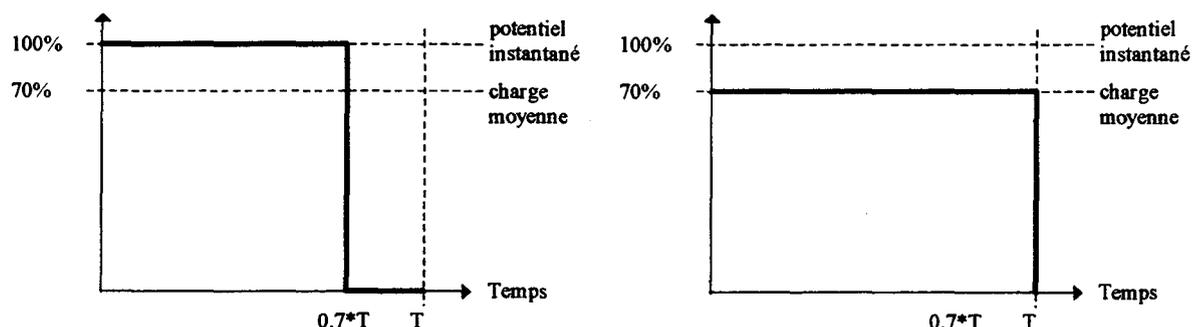
approche est le nombre de solutions potentiellement bonnes qu'il est nécessaire de développer avant de ne garder que la meilleure.

La deuxième méthode qui peut être employée consiste à minimiser la fonction de coût en effectuant un recuit simulé (ou une méthode du gradient) sur la partition complète pour lequel les voisinages sont constitués par des permutations entre deux éléments appartenant à deux sous-listes différentes et où la métrique évaluative est la fonction *NOTE* décrite précédemment.

#### 2.2.4. Recherche par faisceaux

Une fois la partition effectuée, chacune des sous-listes est optimisée. La méthode retenue est un Beam Search (recherche par faisceau) qui est un parcours d'arbre en largeur d'abord avec un élagage important comme décrit dans la partie 2.

Deux types de paramètres sont retenus pour le Beam Search. Comme les optimisations des sous-listes sont faites les unes après les autres, il est nécessaire de garder une fin de liste de bonne qualité pour éviter de reporter, sur la sous-liste suivante, les difficultés d'ordonnancement rencontrées avec la sous-liste présente. Pour cette raison, l'objectif ne va pas être de lisser la charge, mais de minimiser l'écart entre la charge et le potentiel instantané.



minimiser les écarts entre la charge et le potentiel (a)

lisser la charge (b)

Figure III.9 - Objectifs de la recherche par faisceaux

Dans l'exemple de la Figure III.9, la charge moyenne est de 70%. Sur la figure (a), on cherche à charger au maximum l'outil de production (donc charge de 100% sur 70% de la période et de 0% sur les 30% de la période qui reste). Sur la deuxième figure (b), la charge est régulière : 70% sur l'ensemble de la période.

Le deuxième paramètre du Beam Search est l'élagage à chaque niveau. Le filtre est à deux niveaux. Les débuts de liste qui ont le coût le plus faible sont gardés mais ceux qui ont le coût prédictif sur l'ensemble de la liste le plus faible, sont également retenus.

Le Beam Search ne pose pas beaucoup de problèmes sauf pour le réglage de la largeur. Ici une étude en fonction des largeurs montre un résultat relativement stable.

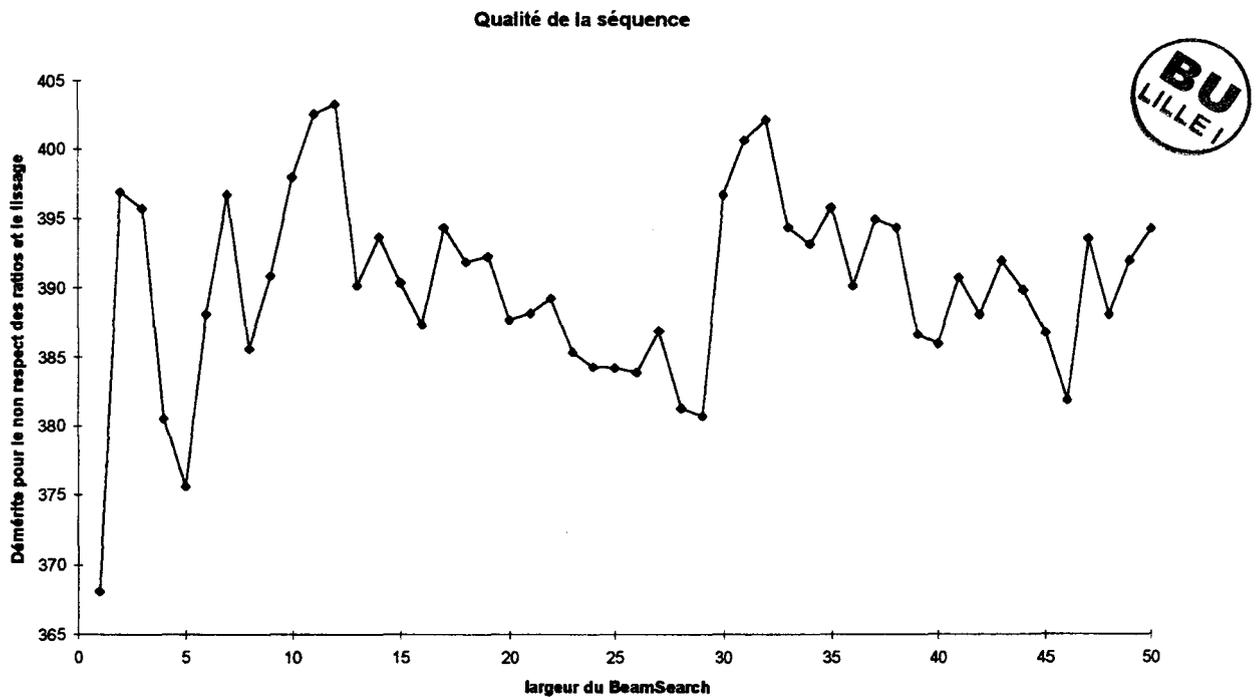


Figure III.10 - Influence de la largeur du Beamsearch sur la qualité de la séquence

Les écarts ci dessus ont été calculés pour des sous-listes de taille 20.

L'expérience montre qu'il est préférable d'avoir une largeur de Beam Search supérieure à la taille de la liste (il faut prendre en compte au moins tous les choix possibles au premier niveau de décision de l'arbre).

A la fin de cette étape, le résultat au niveau des violations de l'espacement acceptable est satisfaisant mais celui au niveau du lissage ne l'est pas car la procédure de recherche par faisceaux n'a pas orienté le résultat dans ce sens.

La deuxième approche pour la partition consiste à améliorer la partition à l'aide d'une descente stochastique ou d'un recuit simulé. Nous cherchons à améliorer par ordre de priorité, le respect des capacités de production puis le lissage des critères. Elle semble plus fiable et plus robuste même si la méthode n'est plus déterministe (utilisation du recuit simulé). Le résultat est en général meilleur et le temps de calcul plus faible, car un seul Beam Search est développé même si un recuit simulé a été fait au préalable. Le recuit simulé converge très vite vers une solution quasi-optimale (valeur de la note de partition proche de 0). Cette note de 0 est rarement atteinte car l'équipartition des critères n'est pas toujours possible (par exemple des critères peuvent être exclusifs ou inclusifs).

### 2.2.5. Recuit simulé

La troisième étape de l'algorithme consiste à compenser les déficiences liées à la fonction d'évaluation du Beam Search (minimisation due aux écarts instantanés entre la charge et le potentiel au lieu d'un lissage de la charge).

Pour cela l'utilisation d'un recuit simulé partant d'une bonne solution initiale (au niveau des viols) permet une convergence assez rapide malgré l'environnement de recherche très chaotique (cf. Guerre-Chaley 95), car le nombre de violations est relativement faible par rapport aux approches traditionnelles. Le recuit simulé utilisé est le même que celui utilisé au chapitre 2.2.

Le voisinage est effectué par permutation entre deux véhicules, le surcoût de la séquence est calculé pour chaque véhicule comme pour l'approche classique du recuit simulé (cf. chapitre II.2.1.2) et les paramètres du recuit utilisés sont standard : décroissance géométrique de la température avec un facteur classique pour  $a=0.9$ .

## 2.3. Complexité des algorithmes

Nous avons déjà vu dans la partie précédente que le nombre de séquencements différents pour une liste de taille  $N$  est  $N!$ . Les différentes méthodes décrites dans cette partie permettent d'obtenir des résultats en un temps de calcul beaucoup plus faible que ce que l'on pourrait espérer avec une énumération exhaustive.

### 2.3.1. Goal Chasing

Pour cette méthode à un pas, il faut évaluer pour chacune des  $N$  places le coût des véhicules. Il faut donc utiliser la métrique d'évaluation  $N(N-1)$  fois.

La complexité de la métrique étant en  $N_C$ , la complexité totale de cette méthode est en :  $O(N^2.N_C)$ .

### 2.3.2. Recuit Simulé

L'algorithme du recuit simulé s'exécute en effectuant des permutations dont le nombre est  $Nb_{permut}$ . Dans le cas standard, où l'algorithme a un nombre de permutations standard : ( $Nb_{permut} = Nb\_paliers \times Longueur\_paliers$ ), il ne reste qu'à évaluer la complexité d'une permutation.

Pour les permutations, il s'agit de déterminer le nombre de produits influencés par une permutation. Le calcul s'effectue sur l'ensemble des  $N_C$  critères. Le nombre de fenêtres

modifiées par une permutation est  $2.B_r$ . Pour le lissage, il est nécessaire de rajouter les véhicules suivants qui sont au plus au nombre de  $2.E_{theo}$ . Tous les critères ne sont pas influencés par une permutation. La probabilité que cette permutation ait une influence sur ce critère est la somme des probabilités que les deux véhicules ont le critère ou que les deux véhicules n'ont pas le critère :  $p_{\pm} = (Q_r/Q)^2 + (1-Q_r/Q)^2$ . De même la probabilité d'avoir une influence est :  $p_{\pm} = 2 \times (Q_r/Q) \times (1-Q_r/Q)$ . Le nombre moyen de véhicules dont le coût local est modifié est donc au maximum :

$$\begin{aligned} n &= 2 \times p_{\pm} \times (B_r + (Q/Q_r) - 1) \\ &= 4 \times (1 - Q_r/Q) \times (Q_r/Q) \times (B_r + (Q/Q_r) - 1) \\ &\leq B_r + 4 \times (1 - Q_r/Q)^2 \\ &\leq B_r + 4 \end{aligned}$$

L'intérêt de ce résultat assez complexe est le fait qu'il est indépendant de la taille de la liste et est proportionnel à la taille des fenêtres ( $B_r$ ) que l'on prend pour les ratios, mais est indépendante du volume de véhicules ayant le critère. En effet si un critère est peu fréquent, il aura une influence sur beaucoup de véhicules mais peu souvent.

La complexité du problème est donc :  $Nb_{permut} \times \sum_{k=1}^{N_c} B_r(k)$

où  $B_r(k)$  est la taille de la fenêtre associée au critère  $k$ .

D'où la nécessité d'avoir des contraintes réalistes (c'est-à-dire des contraintes de fabrication) qui n'aient pas des dénominateurs de ratios trop grands.

### 3. Métriques de séquencement

Dans le cadre des méthodes utilisées deux types de métriques peuvent être employées. Il faut de préférence utiliser des métriques évaluatives, qui ont une vue globale de la liste, que des métriques constructives.

Il existe deux approches possibles. La première évaluera la liste véhicule par véhicule et elle sera semblable aux métriques constructives dans sa forme (voir Guerre-Chaley [GUE 95] et Comby [COM 96]). La seconde, complètement différente, notera la liste dans son ensemble en prenant en compte les valeurs effectives des grandeurs mesurées (intervalles entre véhicules, longueur des rafales par exemple).

Si la deuxième peut paraître plus adaptée, nous avons retenu la première car elle est plus simple à appréhender et permet de la comparer aux méthodes itératives et de générer la métrique  $h$  évaluant l'impact local d'une transformation avec une complexité plus faible.

### 3.1. Métriques constructives et évaluatives

Nous allons dans ce paragraphe décrire les trois métriques retenues et leurs adaptations aux différentes méthodes utilisées. Cette adaptation est contraignante pour l'algorithme lui-même et il est nécessaire d'être prudent avec leur mise en place. Nous avons cherché à obtenir des métriques qui puissent être à la fois adaptées à des approches constructives (construction progressive, parcours d'arbre) et des approches par optimisations successives (recuit simulé par exemple).

### 3.2. Ratio

La métrique liée aux ratios va chercher à optimiser l'outil de production (cf. partie II.2.2).

Pour respecter les ratios proprement dits, il est nécessaire d'utiliser une fenêtre glissante de taille  $D_i$ . La variable  $M_i$  indique le nombre de produits dans la fenêtre glissante ayant le critère "i", il suffit de la comparer à  $N_i$  pour savoir si un dépassement du ratio à respecter (ou de l'espace acceptable) a lieu ou non. L'intérêt de cette notation est de prendre en compte les produits déjà enlignés avant le début de la procédure.

exemple : l'algorithme cherche à garantir un ratio de  $2/3$ .

La notation  $C_i(X)$  indique si le produit  $X$  a le critère  $i$  (valeur 1) ou non (valeur 0).

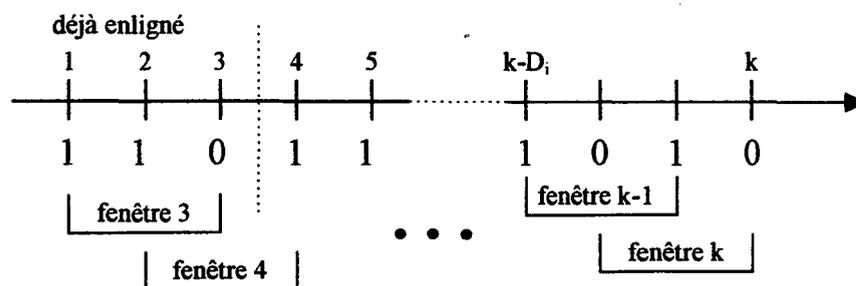


Figure III.11 - Métrique des ratios

Dans cet exemple, l'initialisation se fait en calculant  $M_i$  pour la fenêtre 3.

$$M_i(3) = 2$$

Pour calculer la valeur suivante de  $M_i$ , il suffit de savoir si le produit 1 a le critère  $i$  ou non et si le produit 4 remplit les mêmes conditions.

$$M_i(4) = M_i(3) + C_i(4) - C_i(1)$$

Plus généralement, on peut noter :

$$M_i(k) = M_i(k-1) + C_i(k) - C_i(k - D_i)$$

Ensuite, il suffit de comparer les valeurs de  $M_i$  à  $N_i$ .

Si  $M_i(k) > N_i$ , le produit  $k$  fait une violation sur le critère  $i$

Pour le lissage des critères, deux approches peuvent être utilisées. Pour le Goal Chasing, Monden a utilisé une fonction de coût qui peut être réutilisée dans la même forme (cf. partie II.2.2.1.1). La deuxième approche consiste à essayer de maintenir constante la vitesse de consommation instantanée d'un critère, à la différence de la précédente qui essaye de corriger brutalement tout écart avec la droite de consommation moyenne.

La première approche est la plus appropriée pour le problème statique (optimisation de la liste entière) et c'est la notation que nous avons retenue.

Les deux règles ratios et lissage peuvent être antinomiques néanmoins, car si la règle de ratio est prioritaire (et c'est en général le cas), la capacité à rattraper les fluctuations s'en trouve limitée par la pente du ratio imposé :  $N_i/D_i$ .

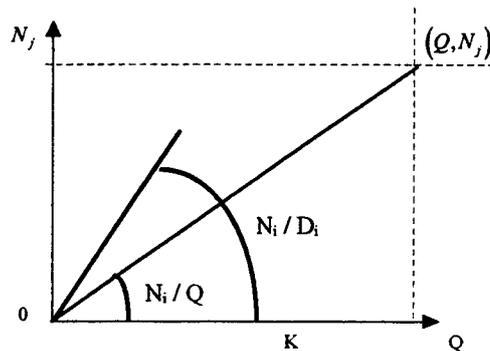


Figure III.12 - Limitation de la vitesse de consommation en fonction des ratios à respecter

Pour les approches évaluatives, il est nécessaire d'avoir une métrique locale pour évaluer les transformations effectuées. La prise en compte des ratios et du lissage des critères est fait de la même manière, mais pour optimiser les algorithmes, les coûts locaux sont stockés et modifiés en fonction des transformations. L'implémentation est décrite dans le chapitre suivant pour le recuit simulé.

### 3.3. Rafales

Deux approches sont également possibles pour cette règle. La première consiste à compter le nombre de changements d'outils (nombre de rafales) et à essayer de le minimiser. Pour atteindre l'objectif, il suffit de minimiser cette valeur.

La deuxième approche consiste à optimiser la taille des rafales (ou des convois). C'est l'approche qui a été retenue car elle permet une plus grande flexibilité de la règle. La notation utilisée est celle décrite dans la deuxième partie.

Il est à noter que les contraintes de rafales et de ratios ou de lissages sont difficiles à mettre en oeuvre ensemble. Une solution pour éviter les problèmes de concurrence est d'effectuer deux étapes dans l'optimisation (cette solution se rapproche des regroupements successifs effectués par Burns [BUR 87]).

### **3.4. Date**

Dans le cadre d'un problème de séquencement en statique, l'idée est de définir des plages de dates. Cette approche correspond à la notion de jalonnement décrite par Comby. Si un produit doit être fabriqué dans la plage horaire  $I$ , il fera partie de l'ensemble des produits qui peuvent être choisis dans la période donnée. La création des partitions se fera en prenant en compte les dates de livraisons et en créant ainsi des paquets d'ordres de fabrication.

Le fait de choisir une fonction de coût semble mal adapté car il sera difficile de pondérer l'influence de cette règle par rapport aux précédentes. De plus, comme nous l'avons décrit dans la partie précédente (II.3.7), les dérives induites par la fonction de coût peuvent pénaliser des algorithmes comme la recherche par faisceaux ou le Branch&Bound.

Dans le cadre d'une métrique évaluative, il suffit de comptabiliser les dérives par rapport aux valeurs cibles souhaitées. Les phénomènes de dérives liés aux métriques constructives ne sont plus à l'ordre du jour.

## **4. Résultats et comparaison entre les méthodes**

Pour comparer les diverses méthodes et valider les concepts retenus, nous avons été amené à tester les diverses méthodes à l'aide de jeux d'essai.

Dans un premier temps ces jeux d'essai correspondent à des jeux aléatoires et dans un deuxième temps, les algorithmes sont testés avec un jeu de test réel.

### **4.1. Description des jeux de tests**

La première série de jeux de tests va permettre de mettre en évidence le comportement des algorithmes dans une situation critique. La deuxième va étudier les réactions de ceux-ci dans un cadre beaucoup plus souple, où les contraintes sont plus espacées. Pour finir, un jeu de tests avec des données réelles permettra de mettre en évidence les comportements des différents types d'algorithmes devant des agencements réels qui peuvent poser problèmes.

Les jeux de tests aléatoires sont constitués de 20 séries de 1000 véhicules avec 20 critères. Pour chaque critère, le volume moyen dans chaque série est le même.

La première série correspond à un flux dense. Ce qui donne par exemple pour les vingt contraintes :

minimum	ratio effectif		ratio à respecter	ratio effectif		ratio à respecter
	moyenne	maximum		minimum	moyenne	
0.08	0.10	0.12	0.10	0.07	0.08	0.10
0.88	0.90	0.92	0.90	0.78	0.80	0.82
0.08	0.10	0.11	0.80	0.09	0.10	0.13
0.01	0.01	0.02	0.05	0.03	0.04	0.05
0.01	0.01	0.02	0.17	0.11	0.13	0.15
0.25	0.26	0.28	0.33	0.23	0.24	0.27
0.16	0.20	0.23	0.50	0.17	0.20	0.22
0.12	0.14	0.16	0.80	0.12	0.14	0.16
0.23	0.25	0.28	0.25	0.22	0.25	0.26
0.33	0.35	0.38	0.50	0.30	0.33	0.37

En convertissant ces données avec l'indicateur  $I_r$  (cf. partie II.3.2.1), il est plus facile d'appréhender la difficulté du flux<sup>1</sup> :

	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
Min.	-0.18	-0.03	0.86	0.58	0.86	0.16	0.54	0.80	-0.12	0.24
Moy.	-0.01	0.00	0.88	0.77	0.92	0.21	0.60	0.82	-0.01	0.29
Max.	0.19	0.02	0.90	0.88	0.95	0.26	0.67	0.85	0.08	0.34
Ecart Type	0.09	0.01	0.01	0.08	0.02	0.03	0.03	0.01	0.05	0.03
	C 11	C 12	C 13	C 14	C 15	C 16	C 17	C 18	C 19	C 20
Min.	0.02	0.08	0.84	-0.02	0.12	0.21	0.56	0.80	-0.04	0.27
Moy.	0.16	0.11	0.87	0.23	0.22	0.27	0.60	0.82	0.01	0.34
Max.	0.31	0.13	0.89	0.42	0.32	0.33	0.66	0.85	0.14	0.40
Ecart Type	0.08	0.01	0.01	0.13	0.06	0.03	0.02	0.01	0.05	0.03

La deuxième série correspond à un flux peu contraint :

minimum	ratio effectif		ratio à respecter	ratio effectif		ratio à respecter
	moyenne	maximum		minimum	moyenne	
0.06	0.07	0.09	0.10	0.07	0.08	0.10
0.68	0.70	0.73	0.90	0.78	0.80	0.83
0.08	0.10	0.12	0.80	0.09	0.10	0.12
0.00	0.01	0.02	0.05	0.04	0.04	0.06
0.01	0.01	0.02	0.17	0.11	0.13	0.15
0.23	0.25	0.28	0.33	0.22	0.24	0.27
0.17	0.20	0.23	0.50	0.18	0.20	0.22
0.13	0.14	0.16	0.80	0.12	0.14	0.16
0.18	0.20	0.22	0.25	0.18	0.20	0.22
0.31	0.33	0.36	0.50	0.31	0.33	0.36

<sup>1</sup> quand  $I_r$  tend vers 0, aucune fluctuation n'est permise pour la contrainte  
quand  $I_r$  tend vers 1, la contrainte devient inexistante.

	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
Min.	0.14	0.19	0.86	0.68	0.87	0.16	0.53	0.80	0.12	0.27
Moy.	0.28	0.22	0.87	0.80	0.93	0.25	0.60	0.82	0.20	0.33
Max.	0.41	0.24	0.90	0.92	0.95	0.32	0.66	0.84	0.28	0.38
Ecart Type	0.08	0.01	0.01	0.07	0.02	0.04	0.03	0.01	0.05	0.03
	C 11	C 12	C 13	C 14	C 15	C 16	C 17	C 18	C 19	C 20
Min.	0.05	0.08	0.85	-0.10	0.12	0.19	0.55	0.80	0.11	0.28
Moy.	0.17	0.11	0.87	0.13	0.24	0.26	0.60	0.82	0.19	0.34
Max.	0.30	0.14	0.89	0.30	0.32	0.34	0.65	0.85	0.28	0.39
Ecart Type	0.08	0.01	0.01	0.12	0.05	0.04	0.02	0.01	0.04	0.03

Ces tableaux montrent la différence entre les deux types de listes. La première a la moitié de ses contraintes difficiles à respecter (voire impossible à respecter quand l' $I_r$  est négatif). La deuxième a des contraintes difficiles mais celles-ci sont moins contraignantes que dans le premier cas.

Le troisième jeu de données qui est utilisé est un jeu de données réelles.

	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10
$I_r$	0.074	-0.008	0.879	0.785	0.921	0.236	0.611	0.824	-0.027	0.291

Ce jeu de données est très contraint (deux contraintes ont un  $I_r$  négatif, ce qui veut dire que les contraintes ne pourront pas être respectées) et de plus les deux premières sont croisées (elles sont complémentaires).

#### 4.2. Comparaison des méthodes

Les résultats obtenus dans les diverses simulations permettent de mettre en évidence les avantages et les inconvénients de chaque méthode. Pour chacune des méthodes et pour chaque type de jeux de simulations, le tableau de la page suivante donne une idée synthétique des résultats obtenus.

Pour chaque jeu de test, nous allons comparer les quatre méthodes décrites dans cette partie :

- Le Goal Chasing
- Le Recuit Simulé
- La recherche par faisceaux (première partie de la méthode combinée)
- La méthode combinée (partition et recuit simulé)

Pour comparer les différentes méthodes, nous avons mis en place trois indicateurs qui permettent d'évaluer les performances de chaque méthode pour :

- Le temps de calcul de l'algorithme est exprimé en secondes sur une DEC Alpha 1000.

- Le respect des capacités de production est exprimé par la somme pondérée des dépassements d'espacements acceptables.
- La répartition des produits est contrôlée en prenant la somme pondérée des Coefficients de Variation (cf. II.3.2.2).

Pour chaque indicateur des simulations 1 et 2, nous donnons les valeurs minimum , moyenne et maximum ainsi que l'écart type pour les 20 données.

Ces résultats permettent de comparer les performances de chaque méthode et de confirmer les critiques faites au début de cette partie.

#### 4.2.1. Simulation 1

##### Temps de calcul

	Min.	Moy.	Max.	Ecart Type
Goal Chasing	2.33	2.44	3.19	0.21
Recuit Simulé	4815.33	4982.10	5202.43	93.22
Recherche par Faisceaux	404.92	408.54	411.71	1.78
Méthode Combinée	2022.92	2072.70	2142.23	31.59

##### Violations

	Min.	Moy.	Max.	Ecart Type
Goal Chasing	12316	22483.00	37296	5898.85
Recuit Simulé	4287	12229.50	28122	5413.72
Recherche par Faisceaux	9793	18319.15	32357	6307.98
Méthode Combinée	2823	9939.00	27158	6018.37

##### Lissage

	Min.	Moy.	Max.	Ecart Type
Goal Chasing	482.22	535.00	602.06	26.16
Recuit Simulé	548.70	602.16	666.69	27.02
Recherche par Faisceaux	738.31	785.80	856.06	30.79
Méthode Combinée	564.56	617.88	675.12	27.08

Ce premier jeu de données est dense. Les résultats sont donc de qualité moyenne.

Le temps de calcul de l'algorithme est néanmoins relativement constant.

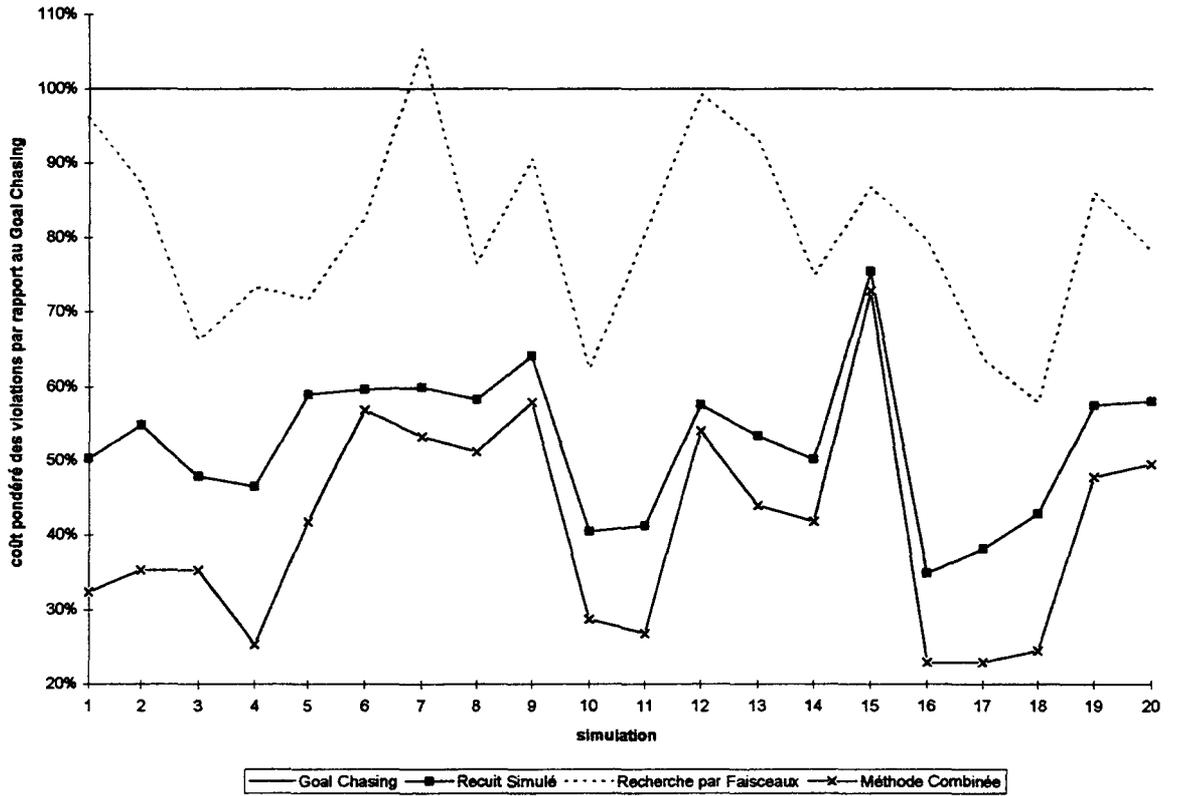


Figure III.13 - Simulation 1 : respect des ratios

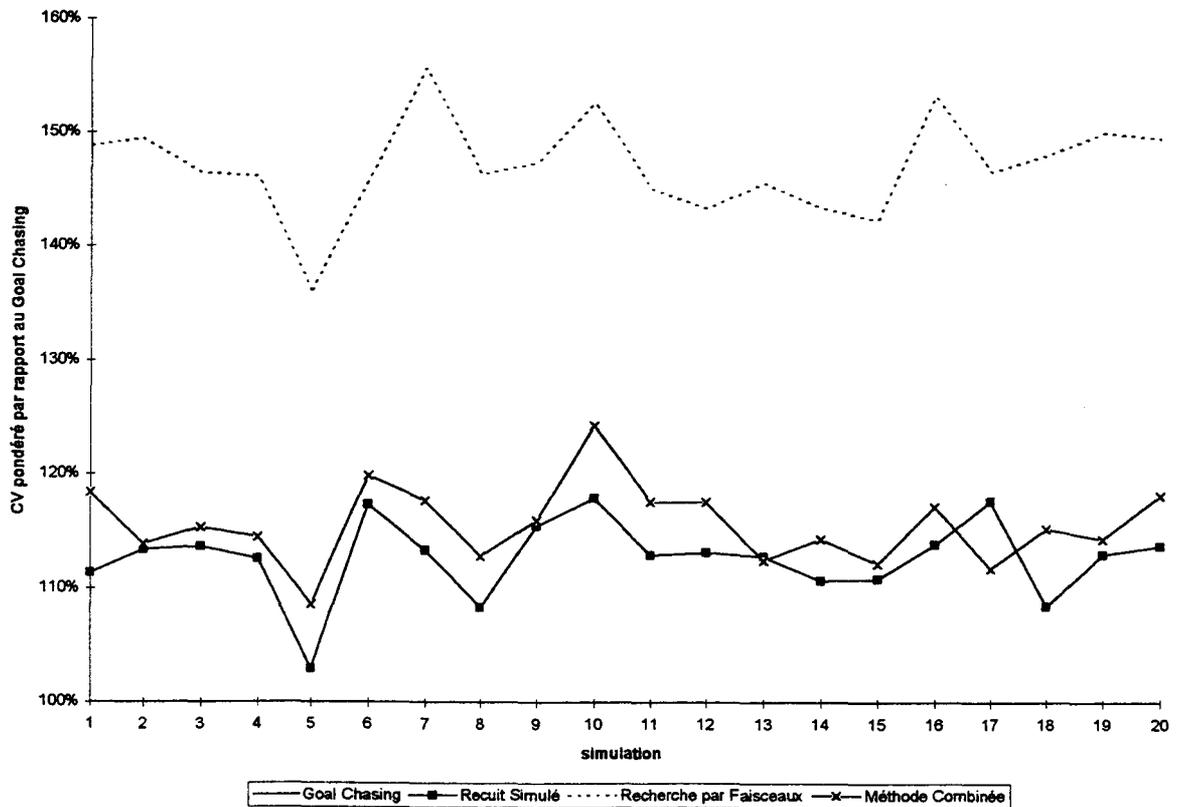


Figure III.14 - Simulation 1 : respect du lissage des critères

Pour le respect des ratios, la qualité de résultat ne subit que peu de fluctuations. On peut voir le détails des résultats sur la Figure III.13. Pour le lissage des critères, le détail se trouve sur la Figure III.14.

Au niveau temps de réponse, le Goal Chasing est évidemment très performant. La structure de l'algorithme l'explique : elle est très simple. La recherche par faisceaux donne des résultats rapidement (environ 400 secondes). La méthode combinée est plus rapide que le recuit simulé seul car la solution initiale (obtenue par la recherche par faisceaux) est de très bonne qualité.

Au niveau de la qualité de la séquence, les résultats sont différents suivant les indicateurs. Le lissage est très bon pour le Goal Chasing. C'est tout à fait compréhensible car c'est le critère que l'on cherche à optimiser. La recherche par faisceaux donne de mauvais résultats car ce n'est pas le critère d'optimisation. Les deux autres méthodes ont des résultats semblables : le Recuit Simulé est légèrement meilleur que la Méthode combinée (environ 3%). Ils sont de bonne qualité car seulement 10% moins bons que ceux du Goal Chasing.

Le critère le plus important pour la qualité d'une séquence est le respect des ratios. Le Goal Chasing est mauvais. La recherche par faisceaux donne des résultats très moyen. La Méthode combinée est la meilleure et de loin (30% meilleure que le recuit simulé).

Les résultats confirment donc les suppositions faites au chapitre III.2. La Méthode combinée donne en un temps de calcul limité (inférieur au recuit simulé) de très bons résultats : meilleurs que le recuit simulé pour le respect des ratios et presque aussi bons pour le lissage des critères.

## 4.2.2. Simulation 2

## Temps de calcul

	Min.	Moy.	Max.	Ecart Type
Goal Chasing	2.31	2.33	2.37	0.02
Recuit Simulé	3157.50	3246.53	3387.44	54.31
Recherche par Faisceaux	407.11	410.31	413.76	1.90
Méthode Combinée	1479.08	1506.09	1559.64	19.39

## Violations

	Min.	Moy.	Max.	Ecart Type
Goal Chasing	2613	3966.75	5588	734.23
Recuit Simulé	109	337.45	928	222.55
Recherche par Faisceaux	1027	1989.40	2956	449.02
Méthode Combinée	81	207.40	535	108.42

## Lissage

	Min.	Moy.	Max.	Ecart Type
Goal Chasing	333.05	357.32	375.48	10.64
Recuit Simulé	385.41	422.76	448.84	13.50
Recherche par Faisceaux	620.24	650.98	684.52	17.62
Méthode Combinée	422.46	446.74	477.14	16.05

La deuxième simulation porte sur un jeu de test beaucoup moins contraint. Pour des méthodes déterministes comme le Goal Chasing ou la recherche par faisceau, le temps de calcul n'est pas modifié par rapport aux simulations précédentes. Pour les deux méthodes utilisant le recuit simulé, le temps de calcul est sensiblement plus court tout en restant stable quelques soient les jeux de données.

Sur la Figure III.15, on a les détails du respect des ratios et sur la Figure III.16, le détail du lissage des critères. La qualité des résultats a également la même tendance que pour le jeu de simulation 1. Mais ces tendances sont accentuées car les jeux de données permettent plus de libertés. Le recuit simulé et la Méthode Combinée sont très nettement meilleurs que le Goal Chasing pour le respect des ratios. Cette flexibilité permet d'atteindre des résultats proches du "zéro" dépassement d'espacement acceptable dans certains jeux.

Il est à noter que l'écart entre les méthodes déterministes et celles avec recuit simulé s'accroît de manière très sensible.

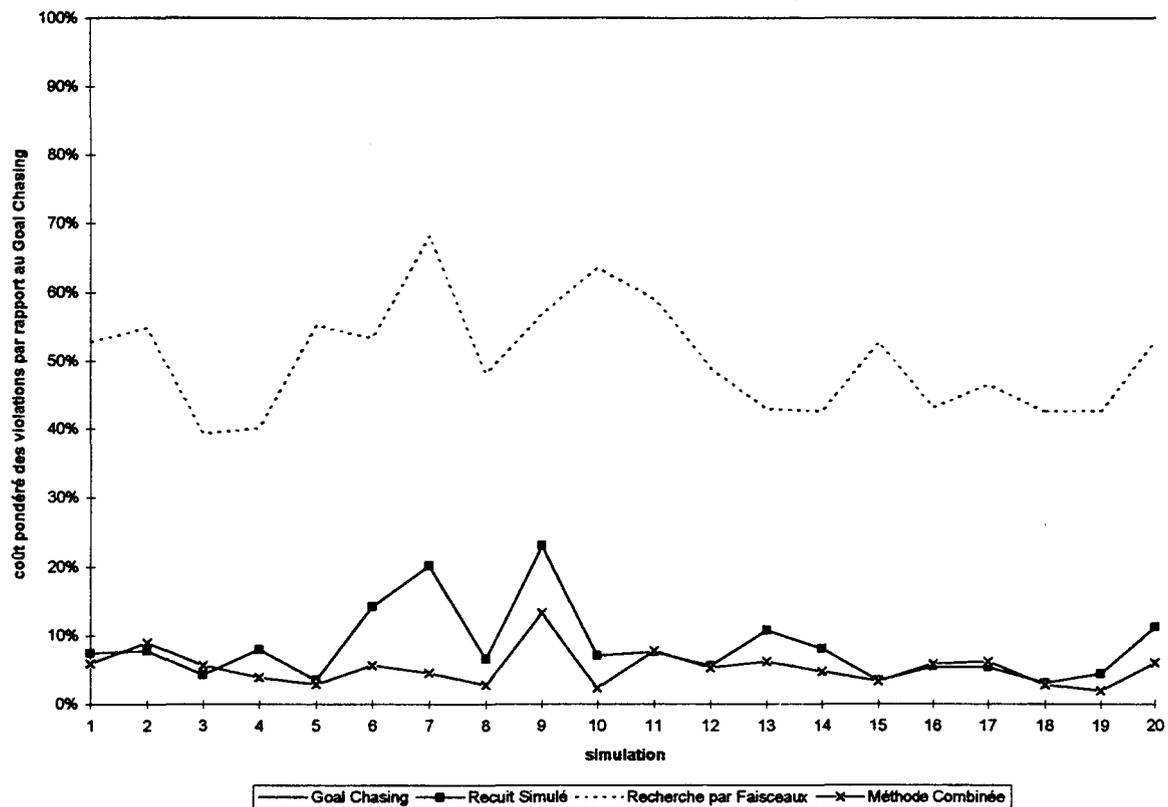


Figure III.15 - Simulation 2 : respect des ratios

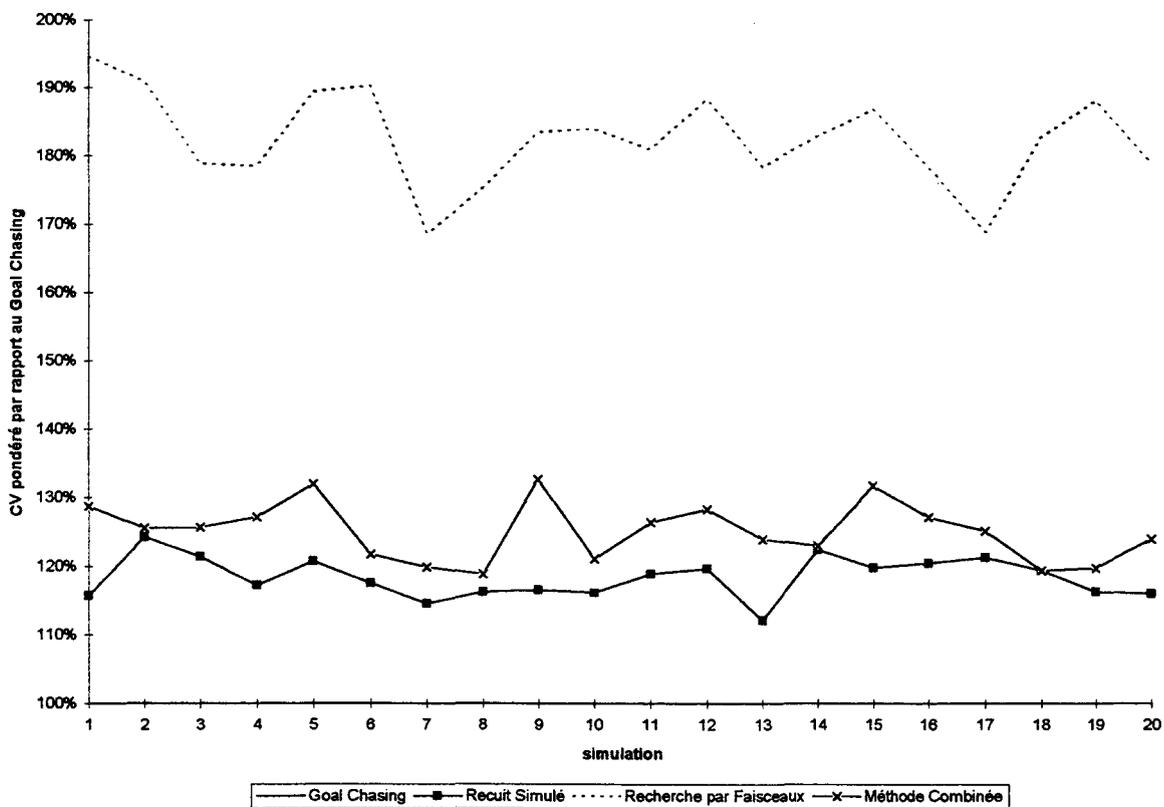


Figure III.16 - Simulation 2 : respect du lissage des critères

### 4.2.3. Jeu de test réel

	Temps Cal
Goal Chasing	14.47
Recuit Simulé	3749.09
Recherche par Faisceaux	352.09
Méthode Combinée	1246.67

	Violations
Goal Chasing	52020
Recuit Simulé	50380
Recherche par Faisceaux	38400
Méthode Combinée	30780

	Lissage
Goal Chasing	458.17
Recuit Simulé	484.12
Recherche par Faisceaux	595.25
Méthode Combinée	489.11

La troisième simulation est un jeu de test avec des données réelles très contraint.

Cette approche permet de prendre en compte des croisements de contraintes qui peuvent apparaître dans la réalité.

Les résultats sont semblables dans leur ensemble aux précédents.

Les temps de calcul sont relativement proches sauf pour le Goal Chasing.

La qualité des résultats a la même tendance sauf pour le respect des ratios. La recherche par faisceaux est nettement meilleure que le recuit simulé. L'explication provient du faible indicateur  $I_r$  mais aussi du croisement de certaines contraintes qu'il est difficile d'apprécier avec un indicateur. Le taux d'acceptation des permutations du recuit simulé est alors très faible, ce qui explique la faible performance du recuit simulé. Elle est confirmée par le faible gain de la méthode combinée par rapport à la recherche par faisceaux : elle est de l'ordre de 20% alors qu'elle est de 50% pour la simulation 1 et de 90% pour la simulation 2.

### 4.3. Remarques

Les quatre méthodes décrites précédemment ont toutes un intérêt et un domaine d'application privilégié.

Le Goal Chasing permet d'obtenir rapidement une bonne solution qui lisse la consommation de chaque critère, sans garantir le respect des contraintes d'espacement. Le temps de calcul est

très faible. Mais la qualité de la séquence se dégrade fortement vers sa fin car le nombre de véhicules disponibles diminue et ceux qui restent sont difficiles à positionner (par le nombre de contraintes et leur croisement).

Le recuit simulé permet d'obtenir de bons résultats quand la solution initiale n'est pas trop mauvaise et que l'espace de recherche n'est pas trop chaotique. Il permet en général un bon lissage des contraintes et surtout une optimisation de l'ensemble de la séquence. En contrepartie, l'algorithme nécessite un temps de calcul important. Mais cela n'est pas toujours incompatible avec les ressources à mettre en place. Il est à noter que l'espace de recherche devient de plus en plus chaotique quand l'indicateur  $I_r$  tend vers 0. De même si un croisement des contraintes est fort, les performances du recuit simulé sont moins bonnes.

La recherche par faisceau sur de petites séquences permet d'obtenir des résultats de bonne qualité au niveau du respect des contraintes pour l'ensemble des critères mais sans garantir un bon lissage des critères sur l'ensemble de la liste (on peut voir une dégradation de l'ordre de 50% par rapport au Goal Chasing). En revanche le temps de calcul n'est pas excessif et peut donc être un bon compromis entre temps de calcul et qualité de la liste. La qualité des résultats est robuste, en particulier quand l'espace de recherche est chaotique. Le fait de forcer la répartition des produits par une partition permet de contourner ce problème et de rendre la recherche par faisceaux meilleure que le recuit simulé dans certains cas.

La dernière méthode, la méthode combinée permet de regrouper les avantages des deux méthodes précédentes : le recuit simulé et la recherche par faisceaux. La recherche par faisceau permet d'avoir une bonne solution initiale quelque soit l'espace de recherche (même et surtout si celui ci est très chaotique) et de réduire ainsi le temps de calcul de la deuxième étape. La deuxième étape à base de recuit simulé permet d'affiner le résultat au niveau du respect des ratios et surtout au niveau du lissage en particulier quand l'espace de recherche n'est pas trop chaotique.

## **5. Conclusion**

Nous avons proposé une méthode permettant de résoudre les problèmes de séquencement. Des méthodes existent dans la littérature de l'ordonnancement, mais la notre semble bien adaptée à des environnements de recherche chaotiques où les méthodes par affinement successif fonctionnent avec difficulté. En associant à une procédure de partition une recherche par faisceau, notre algorithme permet de respecter facilement et de manière déterministe les contraintes de ratios. En associant une procédure de recuit simulé, l'algorithme permet de lisser

la charge des différents critères. La démarche n'est plus déterministe, mais la qualité de la solution obtenue est très bonne vis-à-vis du temps de réponse.

De l'analyse présentée, nous avons vu qu'il était possible d'améliorer de manière sensible les résultats du séquencement prévisionnel par rapport à ce qui est fait avec des méthodes comme le Goal Chasing.

## *Partie 4*

### CADENCEMENT



## Partie 4 : Cadencement

### 1. Problématique

Nous avons vu dans la première partie qu'il existait deux façons différentes de séquencer un flux de production : en statique ou en dynamique.

La première méthode, appelée séquençement, consiste à séquencer de manière prévisionnelle le flux de production. Les approches de cette méthode ont été décrites dans la troisième partie.

La deuxième méthode, appelée cadencement, consiste à séquencer dynamiquement les produits entrant et sortant des stocks intermédiaires de production. Pour ce deuxième type d'approche, la problématique est presque identique à celle du problème statique. En effet, pour optimiser le flux de production les règles et les contraintes à respecter sont les mêmes. Les contraintes juste-à-temps, les surcharges aux différents postes sont toujours présentes. En revanche, seules les contraintes et les règles des ateliers aval au stock sont encore intéressantes à gérer. Les autres ont été respectées pour les ateliers amonts et ne sont plus à l'ordre du jour. La différence apparaît au niveau de contraintes physiques liées au stock qui n'existaient pas dans le cadre du problème statique (configuration et contenu du stock). Elles apparaissent également au niveau des contraintes temporelles car l'algorithme de gestion du stock doit pouvoir assurer les choix en garantissant un temps de réponse compatible avec la vitesse de la ligne ou des lignes en sortie de stock.

Dans cette quatrième partie, nous allons décrire le problème lié à la configuration des stocks et la façon dont nous l'avons modélisé. Dans un deuxième temps, nous allons dépeindre les contraintes temporelles du problème et les difficultés rencontrées lors d'une optimisation dynamique d'un stock. Nous aborderons ensuite les algorithmes que nous avons utilisés, leur implémentation et les métriques utilisées.

Nous décrirons en fin de partie, les deux approches concurrentielles qui peuvent être utilisées pour gérer un stock. L'approche amont est une méthode curative. Elle va corriger les défauts rencontrés dans les ateliers amonts et gérer les aléas de production qui se sont déclenchés : en aval elle va optimiser le flux sortant du stock en fonction des règles et contraintes encore en vigueur à ce point du flux. La deuxième approche, l'approche aval, consiste à optimiser localement l'atelier aval du stock sans perturber le flux global : l'algorithme s'assure que sans aléas dans l'atelier aval, le flux pourra être remis en séquence grâce au stock aval.

## 2. Modélisation des différents styles de stocks et particularités

Dans la première partie nous avons vu qu'il existait trois types de stock. Les stocks à tiroirs sont des rails en parallèles. Les autostockeurs ou MGH (pour Magasin Grande Hauteur) sont des stocks dont tous les emplacements sont accessibles dans le temps-cycle de la ligne. Les stocks dits mixtes sont une combinaison des deux types de stocks précédents. Ces trois stocks ont des comportements différents vis-à-vis de leur capacité de réordonnancement local : les stocks à tiroirs réduisent beaucoup cette capacité, les MGH sont capables de recréer toutes les sous-séquences possibles pourvu que le véhicule soit présent dans le stock au moment où il doit être enligné, les stocks mixtes ont un comportement intermédiaire dont la capacité de réordonnancement dépend à la fois de la répartition entre les capacités des MGH et des tiroirs d'une part et d'autre part des capacités individuelles de chaque tiroir.

Dans notre étude, nous avons cherché à modéliser les différents stocks pour obtenir un algorithme générique pour chacun d'entre eux. L'objectif est d'avoir un ensemble cohérent qui puisse prendre en compte à la fois leur configuration et les données spécifiques au site.

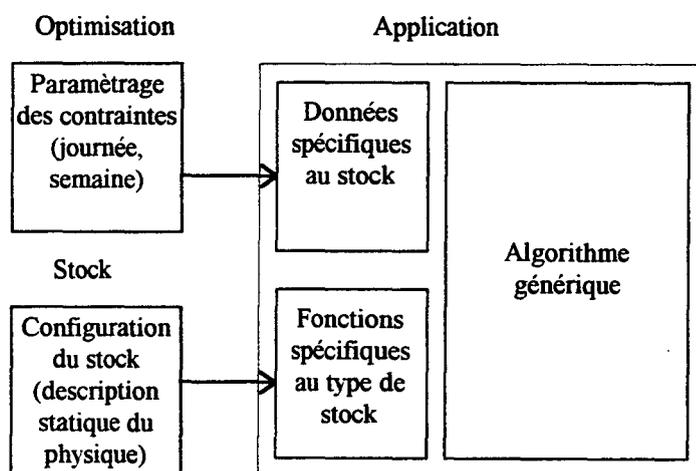


Figure IV.1 - Architecture de l'application.

Un noyau doit être développé pour la méthode d'optimisation. Des fonctions spécifiques sont adaptées aux types de stock (en particulier pour simuler le fonctionnement différent entre un tiroir et un MGH). Et des méthodes d'optimisation (métriques évaluatives) sont adaptées aux problèmes à résoudre (rafales, lissage, ratios, ...). Ces métriques sont paramétrables en fonction du nombre de règles, des contraintes et des données associées.

Nous allons revenir sur les différents types de stocks et leur complexité intrinsèque.

## 2.1. Autostockeurs MGH

Dans le cas d'un MGH, la complexité est facile à calculer. Si le MGH contient  $N$  véhicules différenciés, la combinatoire en sortie est  $N!$ . Si les véhicules ne sont plus différenciés, mais regroupés en classes de véhicules, la combinatoire diminue et elle vaut :

$$\frac{N!}{\prod_i N_i!} \text{ où } N_i \text{ est la taille de chaque classe de véhicules } (N = \sum_i N_i)$$

Si l'on souhaite créer une séquence prévisionnelle de  $L$  véhicules, la combinatoire devient dans le premier cas le nombre d'arrangements de  $L$  objets pris parmi  $N$  :

$$A_N^L = \frac{N!}{(N-L)!}$$

Cette étude porte uniquement sur la sortie du stock. Guerre-Chaley [GUE 95] et Delaval [DEL 97] ont montré que seule la gestion de la sortie était nécessaire. D'autre part, leurs études respectives ont prouvé qu'il n'est pas intéressant d'optimiser l'ensemble du stock. En effet le contenu de celui-ci n'est pas nécessairement représentatif de l'ensemble de la journée de production. Il sert néanmoins à atténuer des aléas du flux de production : la proportion de chaque critère n'est donc pas toujours représentative du volume journalier. Il faut donc être prudent car le roulement du stock (l'alternance entre les entrées et les sorties de véhicules) permet de faire varier ces proportions relativement rapidement. La topologie du stock permet également de choisir n'importe quel véhicule. Cela facilite l'optimisation de la ligne car le choix pour le meilleur véhicule se fait parmi un grand nombre de possibilités différentes.

Delaval a montré qu'une approche par recuit simulé pouvait être intéressante mais qu'elle nécessite la création de poids fictifs pour favoriser les premiers véhicules de la séquence prévisionnelle. En effet le recuit simulé optimise l'ensemble de la séquence et il est nécessaire de privilégier le début de celle-ci, en particulier le premier, celui qui va être **effectivement** enligné. Guerre-Chaley propose de son côté de ne faire le choix que sur le premier véhicule de la séquence. Cela peut paraître restrictif mais apparaît bon vis-à-vis des résultats proposés. Le temps de calcul est très faible. Le roulement du stock et la grande diversité des véhicules disponibles en sortie de stock permettent d'obtenir dans l'ensemble de bons résultats. Castelain [CAS 95] et Delaval [DEL 97] ont étendu cette approche à une procédure à  $N$  pas qui permet de prendre en compte plusieurs véhicules dans la séquence prévisionnelle.

Nous avons retenu cette dernière approche sans prendre en compte l'entrée. On peut faire une analogie entre le fonctionnement du stock et une pelote de laine : la séquence prévisionnelle est le fil que l'on tire de la pelote et la pelote est l'ensemble des véhicules du

stock (on ne sait pas exactement la façon dont ils pourront sortir du stock). Le risque est de voir un noeud apparaître dans la pelote et bloquer l'ensemble. Dans le cas du MGH, cette éventualité est peu probable car le choix des véhicules est important et un véhicule "gênant" peut rester dans le stock (on dit qu'il "culotte").

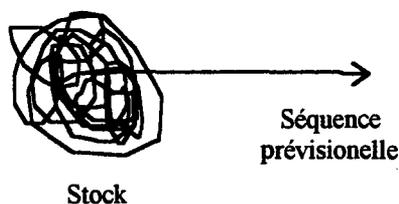


Figure IV.2 - Pelote de laine.

Les risques liés à une telle approche sont de deux sortes. Le premier est le culottage de véhicules dans le stock. Il peut être évité en attribuant une date d'entrée dans le stock à chaque véhicule (s'il n'a pas de date de livraison). Cet attribut permet d'accélérer la sortie de véhicules difficiles à enligner (ils peuvent avoir beaucoup de contraintes dont les croisements ne facilitent pas le choix de ce véhicule). Le coût du véhicule diminuant avec son ancienneté dans le stock, l'insertion de ce véhicule sera d'autant plus facile dans une séquence prévisionnelle que celle-ci est longue. Cette approche ressemble aux méthodes décrites dans la troisième partie comme le Goal Chasing (GCM), des méthodes plus évoluées prenant en compte plusieurs niveaux (EGCM et M1H1 de Miltenburg) ou des méthodes moins myopes comme le M3H2 de Miltenburg [MIL 89a]. Ces méthodes donnent de bons résultats car le choix entre les classes de véhicules est très important.

## 2.2. Tiroirs

Dans le cas des tiroirs la complexité est d'environ  $T^N$ , où  $T$  est le nombre de tiroirs et  $N$  le nombre de véhicules au total. La formule exacte est en fait pour  $T$  tiroirs possédant  $N_i$  véhicules : le nombre de séquences qui peuvent être enlignées à partir d'une configuration du stock donnée :

$$C = \frac{\left( \sum_{i=1}^T N_i \right)!}{\prod_{i=1}^T (N_i !)} \text{ nous le noterons par la suite } C(N_1, N_2, \dots, N_T)$$

où  $i$  est l'indice du tiroir. Cette notation ressemble à celle des MGH, l'explication en est la même puisque l'on choisit le premier parmi les  $N_i$  de chaque tiroir, on a donc  $T$  classes (les

tiroirs) dont on prend le premier véhicule disponible.

Cela permet de faire l'approximation  $N! \approx N^N e^{-N} \sqrt{2\pi N}$  (formule de Stirling)

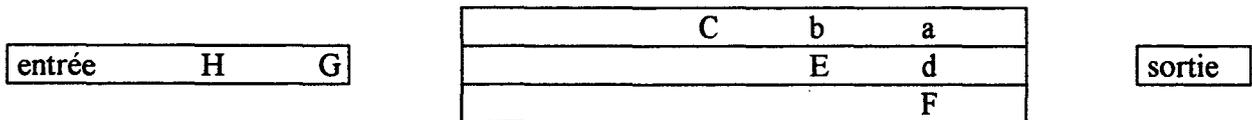
et sachant que  $\prod_{i=1}^T (N_i!) \geq \left[ \left( \frac{\sum_{i=1}^T N_i}{T} \right)! \right]^T$  et que  $\sum_{i=1}^T N_i = N$

La combinatoire peut être approchée par :  $C \approx \frac{\left(\frac{N}{e}\right)^N \sqrt{2\pi N}}{\left(\left(\frac{N}{Te}\right)^{N/T} \sqrt{\frac{2\pi N}{T}}\right)^T} = T^N (2\pi N)^{\frac{1-T}{2}} T^{\frac{T}{2}}$

La combinatoire est plus faible. Mais on a montré dans la première partie qu'il fallait prendre également en compte l'entrée. Dans le cas d'un MGH, cette entrée est transparente car on n'a pas besoin de la gérer. Dans le cas des tiroirs, l'affectation d'un véhicule dans un tiroir a des conséquences importantes. En effet le fait que ce soit des FIFO va contraindre le flux.

La combinatoire devient  $T^{N+2Ne}$  où Ne est le nombre de véhicules non affectés en entrée des tiroirs.

Le petit exemple ci dessous va décrire la complexité des tiroirs :



Dans cet exemple la combinatoire en sortie est  $C(3,2,1) = 60$ .

Si l'on veut prendre en compte l'entrée des tiroirs, deux approches sont possibles.

La première consiste à déterminer le nombre de rangements possibles pour les véhicules en entrée de stock :

- si on prend en compte le premier véhicule "G", on a 3 rangements possibles et la combinatoire explorée est alors :

$C = C(4,2,1) + C(3,3,1) + C(3,2,2) = 455$

- si on prend en compte le véhicule "H", on a 9 rangements possibles et la combinatoire est:

$C = C(5,2,1) + 2 \times C(4,2,2) + 3 \times C(4,3,1) + 3 \times C(3,2,3) = 3528$

Ces résultats sont largement supérieurs aux solutions différentes existantes. Cette approche n'est donc pas assez précise : l'approximation est trop importante et des calculs seront inutiles. En effet, l'insertion du véhicule "G" dans les 60 séquences différentes donne 420 solutions différentes (60 x 7). L'insertion du véhicule "H" dans les 420 séquences différentes donne 3360 solutions différentes (60 x 7 x 8).

La deuxième approche consiste à dénombrer les différentes séquences que l'on peut générer à travers le stock décrit ci-dessus : elles sont au nombre de 256 si l'on prend en compte l'entrée de "G" uniquement. La répartition de ses séquences s'établit comme suit :

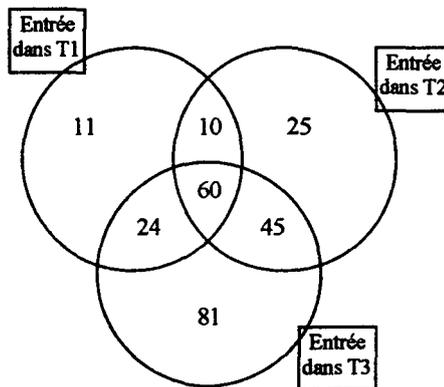


Figure IV.3 - Répartition des séquences en fonction des entrées dans les tiroirs.

Si l'on prend en compte à la fois les produits "G" et "H", le nombre de séquences possibles atteint 1213 possibilités différentes. Si l'on fait le choix d'entrer le produit "G" dans le meilleur tiroir (ici c'est "T3", celui permettant de générer le plus de séquences différentes), avant de faire le choix pour le produit "H", le nombre de séquences que l'on peut générer diminue à 948. La perte de plus de 20% peut paraître faible. Mais il faut prendre en compte le fait que d'une part, le choix pour le premier véhicule était le moins restrictif, et que d'autre part le problème décrit est un exemple de très petite taille.

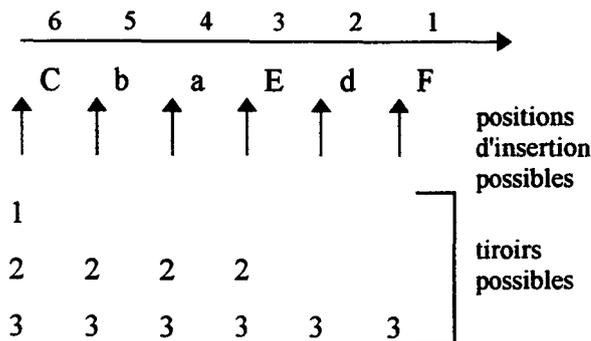


Figure IV.4 - Points d'insertion possibles dans la séquence

Sur la Figure IV.4, nous montrons les points d'insertion possibles du véhicule "G" dans une séquence hypothétique de sortie : C b a E d F. L'affectation du véhicule "G" dans le premier

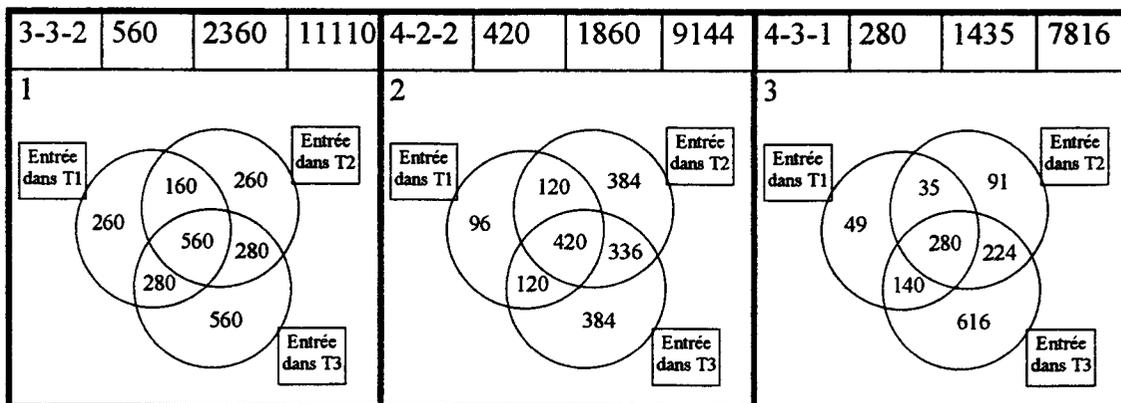
tiroir "T1" est inintéressante si la séquence prévisionnelle de sortie est "C b a E d F" car les mêmes résultats peuvent être obtenus en affectant le véhicule "G" dans les tiroirs "T2" ou "T3". En faisant l'un de ces deux choix, les possibilités d'ordonnancement seront même plus importantes.

Nous venons de voir la capacité d'ordonnancement des tiroirs indépendamment de la qualité de la séquence. En conclusion, il est important de noter que plus le nombre de véhicules pris en compte en entrée est important, plus on a de probabilités de trouver la meilleure solution si on est capable d'explorer l'espace des solutions. Une deuxième remarque indique que l'orientation des véhicules en entrée dans les tiroirs les moins remplis (on cherche à équilibrer le stock) permet de générer potentiellement le plus de séquences différentes. L'équilibrage entre les tiroirs peut donc être une politique intéressante de rangement dans le stock. La troisième remarque est la plus importante. Pour gérer l'entrée du stock, il est quasi impératif de séquencer l'ensemble du stock pour pouvoir orienter les véhicules dans les bons rails. L'autre alternative consiste à utiliser une heuristique en entrée pour orienter le véhicule dans un rail. Le problème d'une telle approche est de découpler l'entrée et la sortie.

Dans le tableau ci-dessous le comportement de trois tiroirs remplis avec 8 véhicules est décrit. Les indications sont :

- la répartition dans les tiroirs,
- le nombre de séquences en sortie pour les 8 véhicules,
- le nombre de séquences en sortie pour les 8 véhicules du stock et un véhicule en entrée
- le nombre de séquences en sortie pour les 8 véhicules du stock et deux véhicules en entrée

Le dessin illustre le nombre de séquences en sortie en fonction de l'orientation du véhicule en entrée.



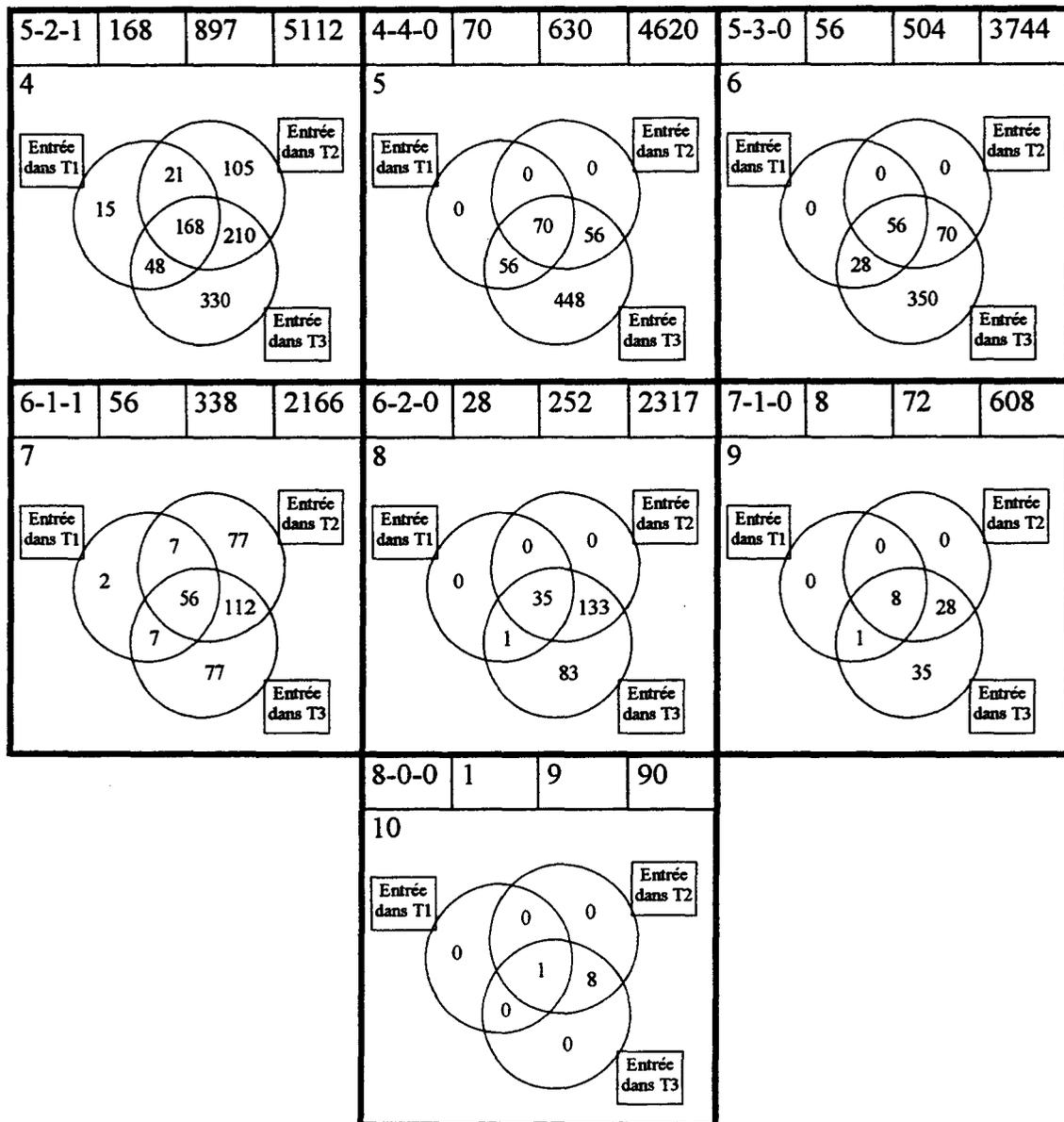


Figure IV.5 - Capacité de réordonnement des tiroirs.

On s'aperçoit dans cet exemple complet que l'équilibrage des différents tiroirs (cas n°1) permet d'augmenter sensiblement la capacité de réordonnement du stock. Le choix, en cas de solutions multiples en entrée, pourra donc être d'équilibrer le taux de remplissage des différents tiroirs (on pourra diriger le véhicule vers le tiroir le moins rempli).

La prise en compte de deux véhicules en entrée au lieu d'un seul nécessite d'explorer un espace de recherche plus important : il est 5 à 6 fois plus grand. En revanche cette approche permet d'accéder à un nombre de séquencements supérieur qui ne pourrait pas être atteint en ne prenant en compte qu'un seul véhicule en entrée. Dans cet exemple, en rangeant le premier véhicule dans le tiroir le plus vide (celui qui permet de générer le plus grand nombre de séquences différentes en sortie), le gain en prenant en compte le deuxième véhicule est

d'environ 25% de séquences générées différentes en plus. Pour le premier rangement, 3-3-2, en effectuant le choix en entrée véhicule par véhicule (méthode à un pas), le nombre de séquencements peut atteindre 8680. En revanche en prenant en compte deux véhicules (méthode à deux pas), on atteint 11110 séquencements différents, soit un gain de 27,8%.

Ces exemples montrent la difficulté qu'aura l'algorithme à gérer les stocks à tiroirs. Il devra trouver un compromis entre le temps de calcul nécessaire à explorer l'espace des solutions et la qualité souhaitée des séquencements. A la différence des autostockeurs, le choix en entrée aura directement un impact sur le séquencement en sortie de stock. De plus la configuration du stock (qui est une donnée statique sauf en cas de panne, ce qui est rare) et le remplissage du stock (qui est une donnée dynamique) influencent directement le séquencement possible en sortie de stock. Des phénomènes de saturation (dans un tiroir donné, ou dans l'ensemble du stock) et des contraintes d'ordre apparaissent, phénomènes qui n'existent pas dans le cas des autostockeurs.

### **2.3. Architectures mixtes**

Les stocks mixtes ont une topologie intermédiaire entre les autostockeurs et les tiroirs. Cette approche implique plusieurs problèmes au niveau de la gestion des entrées et sorties du stock. Ce type de stock conjugue à la fois les difficultés de gestion des autostockeurs (beaucoup de véhicule à gérer, grande diversité en sortie) et celles des tiroirs (gestion de l'entrée du stock). En effet il faut, en entrée, prendre des décisions pour orienter les véhicules d'une part vers l'autostockeur et d'autre part vers les tiroirs.

Si l'algorithme reprend l'approche des tiroirs, il faut réordonnancer tout le stock pour gérer l'entrée en particulier. La complexité du problème est alors gigantesque car la plupart des véhicules sont potentiellement disponibles en sortie (en particuliers ceux de l'autostockeur).

Si l'algorithme reprend l'approche des autostockeurs, la gestion des véhicules dans les tiroirs devient critique car si un véhicule reste bloqué en sortie de tiroir, il bloque l'ensemble des véhicules de ce tiroir.

L'approche qui peut être retenue est de gérer l'entrée des tiroirs à l'aide d'une métrique évaluative comme l'a proposé Calatayud [CAL 95]. Une heuristique en entrée permet d'évaluer l'orientation des véhicules dans les tiroirs. Les véhicules trop difficiles à cadencer sont dirigés vers l'autostockeur et les plus faciles (ceux qui ont peu de contraintes et des contraintes faciles à cadencer) sont dirigés vers l'un des tiroirs.

## 2.4. Conclusion

Dans l'ensemble, nous avons une approche suivant deux axes possibles. D'une part les autostockeurs très flexibles permettent de recréer un très grand nombre de séquences différentes. Leur flexibilité permet en général d'utiliser efficacement des heuristiques pour réduire la combinatoire sans trop dégrader la qualité de l'enlèvement. D'autre part, les architectures à tiroirs, beaucoup moins flexibles, ne permettent pas d'obtenir de nombreux séquençements différents. Cette limitation due à la topologie du stock oblige l'algorithme à explorer une partie très importante de l'espace des solutions possibles car toute décision approximative peut avoir des effets néfastes sur l'enlèvement. Ces effets n'apparaissent pas nécessairement immédiatement. En revanche, parce qu'il y a des contraintes fortes d'ordre, elles peuvent apparaître une dizaine de minutes plus tard car un mauvais choix en entrée ne générera un grumeau en sortie que lorsque ce véhicule arrivera en extrémité de tiroir. Le troisième type de stock, dit mixte, est une forme hybride des deux autres.

Sur la figure IV.6, nous montrons une façon de modéliser les différentes architectures des stocks à base de tiroirs et de MGH. Un **stock**, pour lequel on a choisi une **méthode** d'optimisation, est constitué d'un ensemble de **tiroirs** et de **MGH** en parallèle. Chaque tiroir et chaque MGH contient des **véhicules**.

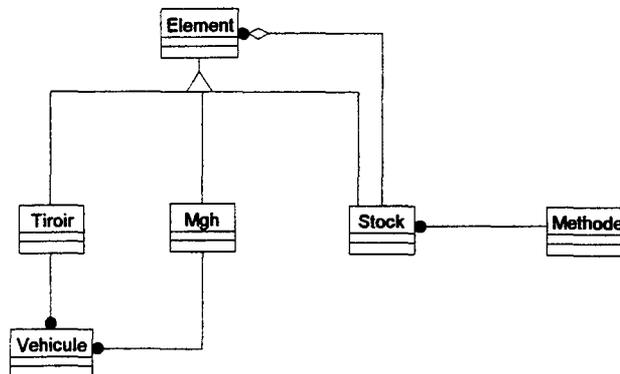


Figure IV.6 - Modélisation OMT des stocks.

## 3. Caractère réactif et optimisation locale pour une vue globale

Dans cette partie, nous allons expliciter les particularités du cadencement par rapport au séquençement. Dans la première partie, nous avons différencié ces deux problèmes essentiellement par leurs caractères dynamiques et statiques. Nous nous intéresserons également aux conséquences au niveau de l'algorithme.

### 3.1. Réactif et Dynamique

Le séquençement est statique et prévisionnel : il doit optimiser un ensemble d'ordres de fabrication pour créer une séquence compatible avec les plages choisies (équipe, journée ou semaine). La durée impartie à l'algorithme peut être relativement importante (quelques minutes à quelques heures). De plus il n'y a pas de contraintes sur l'accessibilité des ordres de fabrication (comme dans la gestion des stocks). Le temps de réponse n'est donc pas un paramètre critique.

Le problème du cadencement est un problème dynamique, dans le sens où une réponse doit être donnée dans le temps cycle de la ligne d'assemblage. Il faut souligner les deux termes qui me semblent les plus importants : **une réponse** et le **temps cycle**.

En effet le cadencement est directement imbriqué avec le fonctionnement de la ligne. L'algorithme doit avant tout garantir un temps de réponse : pendant le temps cycle de la ligne, c'est-à-dire entre deux événements provenant de la gestion du stock (une entrée, une confirmation d'entrée, une demande de sortie ou une confirmation de sortie), l'algorithme doit donner une réponse. Cette décision pourra alors être mise en oeuvre par le gestionnaire de stock.

Dans le cas de notre étude, le temps de cycle de la ligne est de l'ordre de la minute. Pendant cet intervalle de temps, un véhicule va entrer dans le stock et un véhicule va sortir du stock (en moyenne). De plus il faut prendre en compte tout le traitement physique de l'opération ainsi que le traitement informatique de l'information (transfert des événements au système d'information, récupération au près du système d'information des critères de chaque véhicule). Ce chiffre est une moyenne. En effet, le taux d'arrivée des véhicules en entrée de stock dépend du comportement de l'atelier amont : des aléas de production peuvent ralentir les flux, la synchronisation peut être décalée. En outre en sortie de stock, s'il existe un poumon de véhicule avant l'entrée de l'atelier suivant, la cadence ne doit être respectée qu'en moyenne. L'algorithme peut alors conseiller d'enligner un véhicule qui n'est pas encore arrivé à l'extrémité du tiroir, il faut donc l'attendre si c'est possible. La cadence de la ligne n'est pas garantie en sortie de stock mais en sortie du poumon. L'algorithme pourra alors rattraper le retard pris avec l'attente.

Dans ce cadre, le comportement de l'algorithme peut être de deux types. D'une part l'algorithme peut répondre instantanément à une requête de demande de décision et la réponse doit être quasiment instantanée (de l'ordre de la seconde). D'autre part on peut choisir un algorithme qui utilise le temps d'attente entre deux requêtes pour prendre des décisions. Le temps d'attente entre une requête d'entrée et une requête de sortie est en moyenne de 30

secondes, ce qui permet de calculer plus longuement la meilleure décision à prendre. Pour permettre de répondre à toutes les requêtes, le temps de réponse ne doit pas néanmoins dépasser les 10 à 20 secondes. De plus cette méthode doit pouvoir être interrompue et paramétrable au niveau du temps de réponse. En effet en cas d'arrivée massive de véhicules en entrée de stock ou de demande importante en sortie de stock (suite à une panne), l'algorithme doit être capable de répondre très vite (comme dans le premier cas). D'autre part si la réponse doit être obtenue plus tôt que prévue, il faut pouvoir interrompre l'exécution de l'algorithme pour extraire une réponse.

Le terme "une" réponse est également important : on ne souhaite pas nécessairement la meilleure solution en absolu, mais une solution "bonne" vis à vis des contraintes qui ont été fixées et si possible une solution robuste. La robustesse de la solution doit apparaître en particulier au niveau de la stabilité de la séquence prévisionnelle que l'on peut calculer. Il est difficile de décrire l'optimum du problème du cadencement car la résolution est locale alors que l'appréciation du flux est globale.

Soulignons également le caractère réactif vis-à-vis des événements extérieurs (aléas, pannes, défaillance, ...) qui est aussi une caractéristique locale du flux.

### **3.2. Problème instantané et local**

Les aléas de fabrication peuvent interdire l'enlignement de certains véhicules ou autoriser de nouveau leur enlignement. L'algorithme de cadencement doit donc être capable de prendre en compte l'apparition possible de tels événements. Pour cela, il n'est pas nécessaire de planifier à long terme l'enlignement des véhicules. Seul le premier véhicule est très important car il sera effectivement enligné. Les suivants ont seulement une probabilité plus ou moins importante de l'être mais ce n'est pas une certitude.

Le deuxième point important à noter est le fait que l'algorithme effectue une optimisation locale (un véhicule est enligné en sortie de stock ou un véhicule est orienté en entrée) en vue d'une optimisation globale sur une période beaucoup plus longue (équipe ou journée entière) : c'est une suite de décisions instantanées. Cette différence entre les horizons d'optimisation et les horizons d'évaluation pose un problème de cohérence entre les résultats. En effet il est connu qu'une succession d'optimisations locales ne permet pas d'obtenir un optimum global (cette remarque s'applique aussi au Goal Chasing ou à la méthode M3H2 de Miltenburg).

La troisième remarque sur le caractère instantané du problème est le roulement de stock. En effet le contenu du stock et du poumon d'entrée du stock (qui est visible par l'algorithme) est constamment renouvelé par les véhicules enlignés dans l'atelier aval et par les véhicules

provenant de l'atelier amont. Les données du problème changent en permanence, ce qui influence directement l'algorithme de cadencement. En effet dans le cadre d'un autostockeur, un véhicule arrivant en entrée peut améliorer le coût de la liste prévisionnelle de sortie calculée au pas précédent. Dans le cas d'un stock à tiroirs, la stabilité de la liste est plus importante en raison de la topologie du stock.

La quatrième remarque qui peut être faite est liée à la composition même du stock. En effet, le contenu du stock n'est pas nécessairement représentatif du volume journalier de critères à produire. Il est donc nécessaire d'optimiser en fonction des objectifs journaliers alors que les véhicules dans le stock ne permettent pas de le faire. Un dilemme apparaît donc : faut-il privilégier le court terme en optimisant la séquence de sortie du stock en prenant en compte les véhicules présents dans le stock ou en prenant en compte les objectifs à long terme (la journée de production)? Il apparaît évident que l'optimisation d'une séquence complète de sortie du stock n'est pas une approche concevable ([DEL 95], [GUE 95]) car une telle approche ne prend pas en compte l'une des fonctionnalités des stocks intermédiaires de production, la régulation du flux. Le stock permet de retenir un surcroît de véhicules ayant un critère par rapport à la vitesse de consommation moyenne de la journée. De même, il est capable de réduire au minimum le nombre de véhicules ayant un critère si les contraintes liées à ce critère commencent à ne plus être respectées. Le rôle du stock est alors de ralentir ou d'accélérer le flux de véhicules qui ont ou qui n'ont pas les critères posant problème.

### **3.3. Influences sur l'algorithme de cadencement**

Les deux phénomènes (réactif et instantané) vont influencer très fortement sur la conception de l'algorithme de cadencement. Dans un premier temps, il est nécessaire d'avoir une méthode très rapide et dans un deuxième temps, il est préférable d'avoir une méthode locale évoluant pas à pas.

Pour chaque décision d'enlèvement, l'algorithme doit prendre en compte l'ensemble des véhicules disponibles en stock à l'instant donné. Guerre-Chaley [GUE 95] propose de ne prendre en compte que le premier véhicule de la liste prévisionnelle car les changements prévisibles dans l'autostockeur ne sont pas connus à l'avance. Delaval [DEL 97] fait remarquer qu'une telle approche est myope et ne permet pas de détecter des blocages potentiels par la suite. En effet un véhicule peut paraître optimal vis-à-vis des véhicules préalablement enlignés mais s'il n'existe pas de véhicules dans le stock pouvant être enligné par la suite (sans faire de dépassement d'acceptable, en respectant les ratios, ...), le choix du premier véhicule s'avère à moyen terme (2 pas de calcul) très mauvais. Dans cette remarque apparaît l'idée sous-jacente

utilisée dans l'heuristique 2 de Miltenburg : utiliser un algorithme à deux étages qui permet de prendre le premier véhicule de la meilleure séquence prévisionnelle de longueur 2.

Le réglage effectif de cette longueur de séquence prévisionnelle est lié à plusieurs aspects du problème pour éviter à la fois des calculs inutiles et une myopie excessive:

- Le choix de véhicules, c'est-à-dire la **diversité** de véhicules **disponibles** en sortie de stock. Plus la diversité des véhicules est grande, plus les choix seront différents et permettront de s'approcher de l'optimum local qui a été défini. D'autre part cette diversité permettra d'avoir pour l'enlèvement du prochain véhicule un grand choix de véhicules différents permettant de trouver encore une "bonne" solution. Le terme "disponible" est également important car seuls les véhicules disponibles peuvent être enlignés. Des véhicules interdits en grand nombre ou une topologie de stock (comme les tiroirs) réduisent ce nombre de manière importante et nécessitent de prendre en compte une séquence plus importante pour garantir l'obtention d'un résultat de qualité.

- La diversité des contraintes à respecter. Cette diversité quant à elle augmente la difficulté à obtenir une bonne solution car les causes d'inter-blocage sur les contraintes augmentent avec le carré du nombre de contraintes (ce sont les contraintes croisées dont on a parlé dans le chapitre 2).

- Le temps de réponse de l'algorithme est un des éléments impératifs à garantir : il faut pouvoir garantir le temps cycle de la ligne.

Delaval [DEL 97] [DEL 95] propose pour calibrer la longueur  $L$  de la séquence dans un autostockeur d'utiliser un vecteur de stabilité de séquence  $S$ .

Pour conclure, il est important de faire deux remarques. Une des solutions aux difficultés décrites précédemment pourrait être de calculer la séquence optimale la plus longue possible pendant le temps de calcul imparti. Cette approche est néanmoins risquée car le calcul effectué peut être inutile. En effet, l'incertitude régnant sur les données du problème, n'est pas négligeable, or l'algorithme doit connaître précisément l'environnement dans lequel il travaille pour planifier à long terme. La stabilité de la séquence n'étant pas garantie, l'algorithme peut même tomber dans des effets pervers : il n'est donc pas raisonnable de planifier les décisions à long terme. En effet, l'algorithme va prévoir une longue séquence prévisionnelle qui est optimisée : c'est la meilleure façon de sortir les  $L$  prochains véhicules (où  $L$  est une longueur que l'on fixe à l'avance). Mais ce que l'on souhaite, c'est sortir un seul véhicule, le premier. De plus, des séquences prévisionnelles trop longues en sortie de stock ont de faibles probabilités

d'être conservées en l'état, en particulier pour des stocks aussi flexibles que les autostockeurs. L'exemple suivant va illustrer le problème :

L'objectif est de respecter le ratio de 1/3 pour le critère noir et de lisser ce critère.

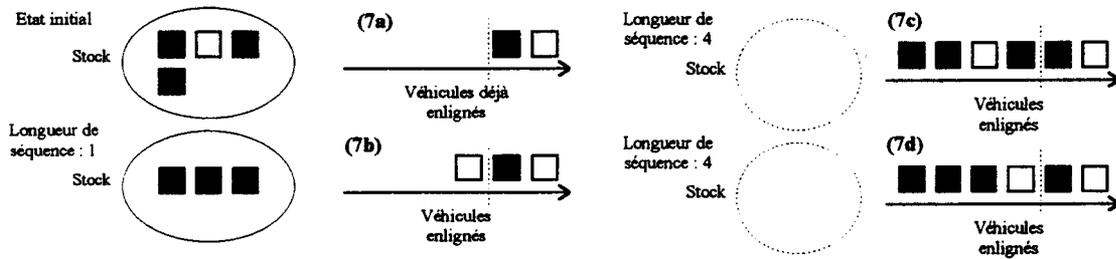


Figure IV.7 - Influence de la longueur de la séquence prévisionnelle.

Dans cet exemple, nous allons montrer l'influence de la longueur sur les choix d'enlignement en prenant deux valeurs de  $L$  : 1 et 4. La figure (7a) illustre l'état initial du stock. La séquence de longueur 1 donne la réponse souhaitée : on respecte le ratio de 1/3 (figure 7b). Dans le cas de l'optimisation sur une séquence de longueur 4 (figure 7c et 7d), on sait que l'on ne peut pas respecter le ratio avec les véhicules actuellement présents dans le stock. Deux solutions sont possibles. La première (figure 7c) fait quatre dépassements du ratio (à 2/3 au lieu de 1/3). La deuxième solution (figure 7d) fait trois dépassements du ratio (à 2/3 au lieu de 1/3 pour deux d'entre eux et 3/3 pour le dernier) et obtient le même lissage pour l'espacement entre les critères noirs. Donc, l'algorithme va choisir la première des solutions (7c) qui, sur un horizon de quatre véhicules, va donner la meilleure solution si aucun véhicule sans le critère n'entre dans le stock pendant ce temps là. Or la deuxième solution peut à long terme être beaucoup plus intéressante si des véhicules sans critères entrent dans le stock. Cette probabilité est relativement importante, car le stock n'est pas du tout représentatif du volume de critères à fabriquer.

Guerre-Chaley [GUE 95] ne voit pas ce type de problème car il utilise une méthode à un pas : c'est l'exemple (7b) avec une longueur de séquence de 1. Cette solution est paradoxalement meilleure que celle obtenue avec une longueur de séquence de 4 si des véhicules sans critère entrent dans le stock.

Delaval [DEL 95] propose, pour contourner ce problème avec une méthode à base de Recuit Simulé, d'utiliser des poids pour favoriser les premiers véhicules de la séquence. En effet, ce sont les véhicules qui seront les prochains à être enlignés, et il y a de très fortes probabilités que ce choix ne soit pas remis en question : le premier sera effectivement enligné, le deuxième a une probabilité de 80% dans l'exemple étudié. Pour évaluer l'importances de ces

pois, on peut utiliser une matrice de préférence et un vecteur de stabilité que l'on retrouve également dans [CAS 95].

Ces deux approches sont adaptées pour le cas des autostockeurs. En revanche dans le cas des tiroirs, la complexité de la gestion nous amène à prévoir l'ensemble des mouvements potentiels dans le stock. Ce qui ne permet pas de tronquer la séquence prévisionnelle.

## **4. Algorithme de parcours d'arbre**

### **4.1. Introduction**

Si l'on considère uniquement la sortie, on a vu que la résolution du problème par un arbre style Branch and Bound avec recherche en profondeur d'abord peut donner de bons résultats [CAS 95]. La profondeur de recherche permet de déterminer une anticipation sur la liste future : c'est à dire que le choix d'un véhicule n'est pas lié uniquement aux véhicules qui ont été enlignés avant lui mais aussi aux véhicules que l'on peut enligner après lui. Dans le cas d'un autostockeur, [GUE 95] a montré que le choix du meilleur véhicule pouvait donner de bons résultats (stratégie à un pas). [DEL 95] a montré que le choix effectué à partir d'un recuit simulé donnait aussi de bons résultats (méthode globale). Les deux méthodes s'avèrent bonnes mais elles n'ont été testées que dans le cas des autostockeurs où le nombre de véhicules disponibles à chaque instant est très important (au moins une centaine, c'est-à-dire presque la totalité du stock). Ce nombre important de véhicules disponibles permet d'avoir la plupart du temps le véhicule le mieux adapté à la sortie à un instant donné. Le volume de véhicules va également permettre d'atténuer l'impact des deux phénomènes d'entrée et de sortie. L'entrée des véhicules dans l'autostockeur est transparente : un nouveau véhicule qui entre est un véhicule supplémentaire qui peut être cadencé en sortie.

Des algorithmes comme les Branch and Bound tronqués sont adaptés à de telles approches. Delaval [DEL 97] propose un Beam Search pour résoudre le problème de séquençage dynamique d'un autostockeur : c'est un parcours d'arbre en largeur d'abord. Le problème d'une telle approche est le fait que l'algorithme ne peut être interrompu. Il est paramétrable pour le temps de réponse (en réglant la largeur et la profondeur de l'arbre) mais la meilleure solution est obtenue après le parcours complet de l'arbre. Nous avons préféré utiliser une méthode de recherche en profondeur d'abord. Cette approche, parfois moins performante, permet d'obtenir très rapidement une solution de référence. Le parcours initial atteint directement la solution obtenue par une méthode gloutonne. Cela permet d'une part d'avoir une solution de référence

pour l'élagage de l'arbre de recherche et d'autre part d'interrompre le parcours d'arbre, si la nécessité apparaît, en obtenant néanmoins une solution au problème posé.

Dans le cas d'un stock à tiroirs, (FIFO en parallèles), le problème est beaucoup plus contraint. D'une part, le nombre de véhicules disponibles est limité en sortie et ne représente qu'une petite partie du stock. En effet, l'enlèvement d'un très bon véhicule, par rapport à ce que l'on a déjà enligné, peut être remis en cause dès l'enlèvement suivant si aucun véhicule n'ayant les critères attendus n'est disponible. D'autre part il y a le choix à effectuer en entrée (affectation dans un tiroir ou un autre).

Le choix retenu est un parcours d'arbre. Il a ses avantages et ses inconvénients. Il est déterministe, donne une bonne qualité de résultats mais il est gourmand en temps de calcul, ce qui nécessite de couper des branches de recherche pour pouvoir être certain de respecter le temps de cycle.

Dans le cas d'un stock à tiroirs, la difficulté la plus importante au niveau décisionnel va être la gestion de l'entrée des tiroirs. En effet, il est facile d'imaginer qu'un mauvais rangement des véhicules dans les tiroirs mènera à une situation mauvaise en sortie car on n'aura pas les véhicules souhaités au moment voulu. Par extension, il devient évident qu'une optimisation de la sortie nécessite un bon rangement des véhicules dans les tiroirs. Il faut donc un algorithme performant d'entrée dans les tiroirs.

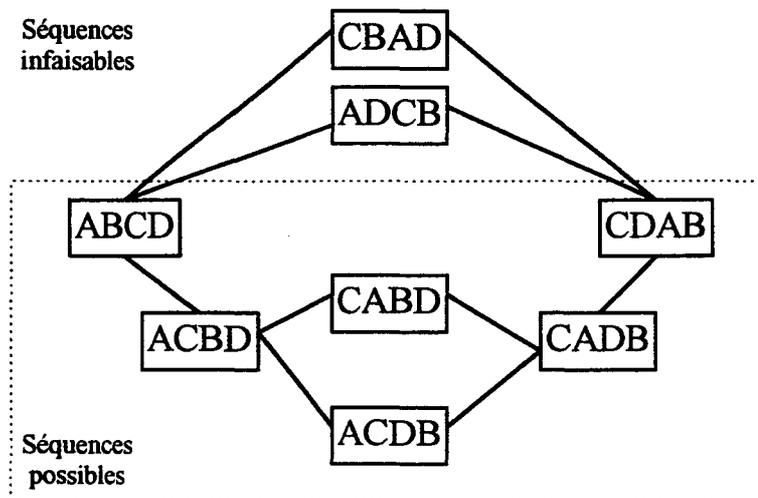
Pour éviter une explosion combinatoire et atteindre cet objectif, deux politiques de gestion du stock ont été abordées. La première consiste à avoir un algorithme d'entrée coopérant avec l'algorithme de sortie. Quand on entre un véhicule dans un tiroir, on sait dans quelle configuration il pourra sortir de celui-ci, et cette configuration aura été évaluée comme étant bonne. La deuxième solution consiste à découpler l'entrée de la sortie. Les deux algorithmes d'entrée et de sortie ne communiquent plus directement, mais le rangement des véhicules dans les tiroirs doit toujours être compatible avec leur sortie. Les deux approches, qui sont exposées ci-dessous, ont donné des résultats encourageants mais présentent chacune des avantages et des inconvénients.

Une autre approche envisagée peut être le recuit simulé qui nécessite d'établir un compromis entre qualité du résultat et temps de calcul. De plus, la détermination des voisinages possibles dans l'ensemble des solutions reste délicate. En effet dans le cas des autostockeurs, il est nécessaire de garantir une certaine stabilité de la liste prévisionnelle. Dans le cas des tiroirs, la topologie du stock ne permet pas toutes les permutations ou les insertions, il est alors nécessaire de vérifier si les transformations élémentaires sont possibles. Cette limitation du voisinage des solutions va augmenter la durée du recuit simulé car dans le cas des

autostockeurs l'espace de recherche est connexe et convexe alors que dans le cas des tiroirs, la topologie du stock rend cet espace fortement concave.

L'espace de recherche pour l'autostockeur est l'espace des séquences différentes au complet : l'espace est donc connexe et convexe. Pour les stocks à tiroirs, l'espace de recherche est un sous-ensemble de l'espace précédent qui peut être concave.

Prenons quatre véhicules dans deux tiroirs, "A" et "B" dans T1 et "C" et "D" dans T2. Sur les 24 séquences avec 4 véhicules (4!), seules 6 sont possibles avec cette configuration : ABCD, ACBD, CABD, ACDB, CADB et CDAB. La topologie de l'espace est donc la suivante :



Le voisinage de deux solutions modélisé par un trait correspond à une permutation entre deux véhicules de la séquence. Pour passer de la séquence ABCD à la séquence CDAB, il suffit de 2 permutations entre "A" et "C" et entre "B" et "D". Cela génère des séquences infaisables avec la configuration : CBAD et ADCB. Il faut donc passer par des solutions intermédiaires existant dans l'espace de recherche.

La notion de distance entre deux séquences est donc le nombre minimum de permutations entre deux véhicules pour passer de la séquence 1 à la séquence 2.

Cette concavité de l'espace des solutions pour l'architecture à tiroirs est la principale raison pour laquelle le recuit simulé n'a pas été retenu.

#### 4.2. Politiques de gestion

Une des hypothèses que nous avons retenues est la nécessité d'affecter les  $N_c$  véhicules pris en compte en entrée dans les tiroirs. En effet le séquençage que l'on propose doit pouvoir être obtenu quels que soient les événements qui se produisent. Si aucune sortie de véhicules du stock n'est faite alors que plusieurs entrées le sont, il faut que les affectations proposées aux pas précédents de l'algorithme puissent être effectives [BAR 97].

#### 4.2.1. Gestion simultanée de l'entrée et de la sortie des tiroirs

Dans cette première approche, l'algorithme d'entrée doit coopérer avec l'algorithme de sortie. Pour cela, il faut envisager de réordonnancer l'ensemble du stock lorsqu'une décision en entrée est nécessaire. En effet les difficultés que l'on va rencontrer pour l'algorithme d'entrée sont d'ordre combinatoire. Si l'on veut ordonnancer tout le stock pour prendre en compte les sorties possibles, il faut construire et explorer un arbre de profondeur  $N+N_e$ , qui permet de créer une liste prévisionnelle de sortie. Le nombre de combinaisons dépasse vite le temps de calcul imparti. Pour résoudre ce problème, on utilise tout d'abord une politique d'enveloppe réordonnançable (en prenant comme paramètre,  $N_e$ , le nombre de véhicules dans le poumon d'entrée).

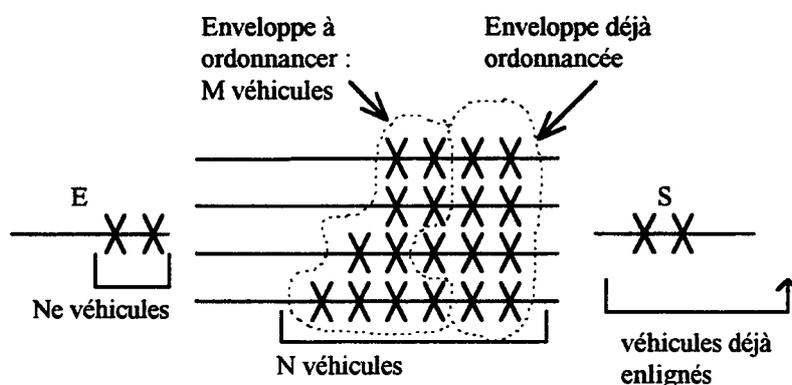


Figure IV.8 - Description des enveloppes à taille fixe.

Si l'on considère que les véhicules présents dans le stock sont déjà ordonnancés, on possède une liste de référence proposant un ordre de sortie pour ces véhicules. Pour limiter la combinatoire, on va supposer qu'une partie de ces véhicules sont déjà à la place définitive qu'ils doivent occuper dans la liste prévisionnelle de sortie. L'enveloppe des véhicules déjà ordonnancés est un choix que l'on ne remet pas en cause. La position des autres véhicules pourra être remise en cause en fonction des différents véhicules qui pourront entrer dans le stock.

Le principe de l'algorithme d'entrée sera donc de la forme:

1. Déterminer les 2 enveloppes " A ordonnancer " et " Déjà ordonnancée " à partir de la liste prévisionnelle.
2. Générer l'arbre de décision créant une liste avec les véhicules en entrée et à ordonnancer à la suite des véhicules déjà ordonnancés.
3. Entrer le véhicule dans le tiroir choisi de façon à permettre la construction de la liste calculée.

4. Etablir la nouvelle liste prévisionnelle.

L'algorithme de sortie est très simple:

1. Prendre le premier véhicule de la liste prévisionnelle.

Cette approche a un avantage non négligeable car on peut facilement régler les paramètres : le nombre de véhicules à prendre en compte lors de l'exécution de l'arbre d'entrée (les  $N_e$  véhicules pris en compte dans le poumon d'entrée) et les  $M$  véhicules de l'enveloppe à réordonner. On peut donc fixer ces valeurs et assurer un temps de réponse donné. On peut s'assurer du bon fonctionnement de l'algorithme en mode nominal : temps de réponse et qualité de la liste.

Le problème de cet algorithme apparaît lors d'un fonctionnement en mode non nominal. Par exemple, dans le cas d'un stock déséquilibré (un tiroir vide en attente d'un certain type de critères), l'algorithme va avoir des difficultés à gérer le fait qu'un véhicule dans un tiroir vide peut ressortir très rapidement. Or le fait d'avoir une enveloppe de véhicules déjà réordonnés limite la position à laquelle il pourrait se situer dans la liste prévisionnelle. Donc si on attend un véhicule avec un certain type de contrainte, il ne pourra pas passer tout de suite. Ce problème n'est pas critique en mode de fonctionnement nominal, mais il le devient quand des configurations particulières se présentent. Un flot de véhicules en entrée qui n'est pas du tout représentatif des volumes journaliers va déséquilibrer le stock. Si un véhicule qui entre a les critères qui permettent de rééquilibrer les ratios dans le stock, ce véhicule a de fortes probabilités de ressortir rapidement.

La deuxième approche qui a été testée, en particulier pour remédier à ce problème, consiste à avoir des enveloppes de tailles variables.

La limite de l'enveloppe à réordonner est définie par le véhicule situé en dernière position d'un des tiroirs et dont la position est la plus en aval dans la liste prévisionnelle. L'enveloppe des véhicules à réordonner est constituée de l'ensemble des véhicules situés après la limite dans la liste prévisionnelle.

Dans l'exemple ci-dessous, où la liste prévisionnelle est A-B-...-R-S, c'est le véhicule L. L'enveloppe à réordonner est la sous-liste M-N-...-R-S.

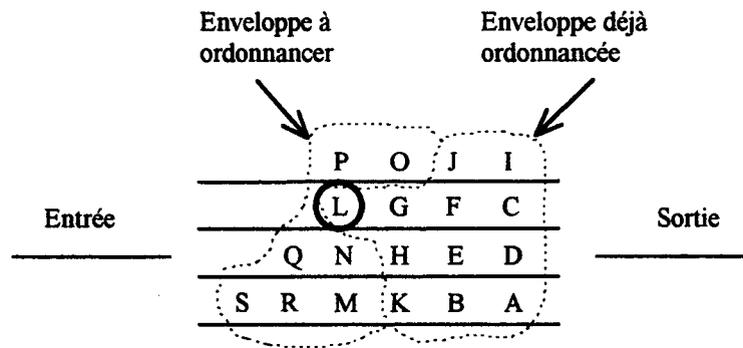


Figure IV.9 - Description des enveloppes à taille variable.

On suppose que l'influence d'un véhicule non ordonnancé sur la liste des véhicules déjà ordonnancés est limitée. L'intérêt de cette approche est l'apparition de la taille variable de l'enveloppe. Ce paramètre permet de prendre en compte le remplissage variable des tiroirs. Si un tiroir est vide, on va recadencer tout le stock, car un véhicule entrant pourrait ressortir quasiment aussi vite (en fonction de la vitesse de défilement des tiroirs). Cet avantage est également le défaut de l'approche. En effet, la taille de l'enveloppe étant variable, la taille de l'arbre de décision l'est aussi car il faut recadencer l'ensemble de l'enveloppe. Le temps de réponse de l'algorithme ne peut être donné.

Finalement une troisième approche consiste à réordonnancer l'ensemble du stock à l'aide de petits arbres de décision (recherche arborescente) qui permettent d'évaluer le choix d'un véhicule en vérifiant que ce choix ne mène pas à une situation de mauvaise qualité. Une fois un choix effectué sur un horizon assez large, on ne revient plus dessus. Cela revient à tirer un bout du fil hors de la pelote! L'avantage est de réduire la combinatoire car on effectue  $N+N_e$  arbres dont on connaît la taille  $T^p$  (où  $p$  est l'horizon de recherche).

Cette approche dans son comportement s'apparente au comportement de l'algorithme de l'autostockeur. On essaye de prévoir une séquence de sortie des  $N$  véhicules du stock et des  $N_e$  véhicules en entrée en simulant le fonctionnement sur  $N+N_e$  pas.

#### 4.2.2. Gestion de l'entrée indépendante de la sortie

Cette dernière approche est de considérer un algorithme d'entrée découplé de l'algorithme de sortie.

L'utilisation d'un des algorithmes (du chapitre 4.2.1), est relativement gourmande en temps de calcul, mais la méthode est exacte si l'on parcourt un seul arbre de taille  $(N+N_e)$ . L'idée sous-jacente de cette dernière approche est d'utiliser une heuristique qui va permettre de

ranger les véhicules dans les tiroirs de manière compatible avec la sortie mais sans connaître précisément l'ordre de sortie de ces véhicules. L'objectif de l'heuristique est de répartir les contraintes entre les différents tiroirs de manière à équilibrer la densité des types de véhicules (en fonction des contraintes qu'ils possèdent) dans chaque tiroir. L'impact sur l'algorithme de sortie permet d'avoir un choix des véhicules disponibles diversifié et homogène. Pour cela, on va essayer de recréer dans les tiroirs des sous-listes représentant la bonne densité de contraintes. On gère des cas particuliers pour les contraintes dont l'espacement est inférieur à 1 (exemple 2 véhicules sur 3). Dans ces cas là, on va privilégier le respect de l'espacement minimum (acceptable) pour éviter les engorgements si des rafales arrivent.

Initialement, le choix correspondait à la meilleure décision en entrée pour le premier véhicule à l'aide d'une métrique évaluant la répartition des véhicules dans les tiroirs. Cette première tentative a montré de bons résultats sauf dans un cas. Si les tiroirs sont trop remplis (une seule place disponible), le véhicule peut occuper alors la dernière place libre d'un tiroir. Il faudra diriger le véhicule vers un autre tiroir où il ne conviendra pas nécessairement. Finalement, avec un arbre prenant en compte les deux premiers véhicules, on s'assure que les deux premiers véhicules du poumon d'entrée peuvent entrer dans le stock. Le premier correspond au choix que l'on doit faire et le deuxième s'assure que le premier choix n'amène pas à des configurations critiques avec deux véhicules (en particulier lors de fonctionnement en saturation de stock).

L'intérêt de cette solution est le gain de temps en entrée. L'heuristique donne un résultat très rapidement. Cela permet de passer plus de temps à optimiser la sortie (le gain en temps correspond à une profondeur de  $N_e$  dans l'arbre de décision). En général, la perte de qualité en entrée est compensée par le gain de temps qui permet de trouver une meilleure solution. Par contre dans des situations très contraintes, on risque de ne pas faire le rangement qu'il faut et de ne pas trouver la configuration nécessaire à la création d'une liste correcte en sortie. En effet à la différence des trois premières approches, on fait un " pari " sur l'avenir. Or si une seule solution peut convenir il faut absolument faire le bon rangement, rangement que les trois premières méthodes sont susceptibles de découvrir car elles parcourent l'arbre de décision (et détectent la difficulté à ordonnancer correctement). En revanche l'heuristique va équilibrer les contraintes dans les tiroirs. Si on a un ordonnancement sur un type de contrainte pour laquelle la flexibilité est nulle, il faut assurer la bonne alternance des véhicules avec contraintes et sans contraintes. Tout dépassement d'un côté ou de l'autre est pénalisant, on n'a donc aucun droit à l'approximation pour ces contraintes et le rangement doit donc prévoir la sortie, en particulier si des contraintes croisées sont présentes.

Cette méthode peut donc être intéressante dans le cadre d'un ordonnancement où une certaine souplesse existe. Dans le cas d'un problème où certaines contraintes ont une marge de manoeuvre très réduite ou inexistante, cette solution s'avère expérimentalement moins bonne que celles décrites précédemment.

D'autres heuristiques peuvent être envisagées. On peut associer à un tiroir donné un type de contraintes particulières. On peut essayer d'équilibrer la charge des tiroirs, mais le résultat est moins bon.

### 4.3. Algorithme d'entrée

Dans le cas où l'entrée est couplée à la sortie, l'une des manières pour déterminer le meilleur rangement est d'effectuer tous les rangements possibles en entrée et de développer un arbre pour chacune de ces configurations.

Or le problème d'une telle approche est de prendre en compte plusieurs fois la même solution car on peut obtenir la même solution en rangeant un véhicule dans des tiroirs différents (cf. Figure IV.4). Cette première méthode développe  $T^{Ne}$  arbres de taille  $(N+Ne)$ .

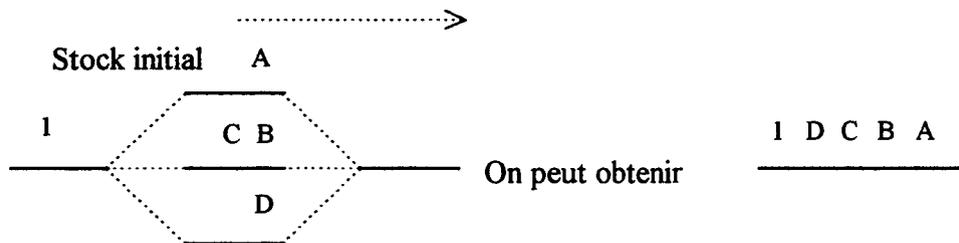


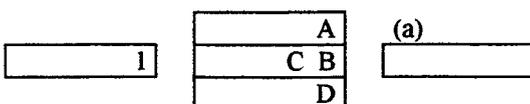
Figure IV.10 - Exemple pour l'algorithme d'entrée.

Cette solution peut être obtenue en rangeant le véhicule "1" dans n'importe lequel des trois tiroirs. Mais la deuxième méthode décrite ci-dessous élimine ces redondances.

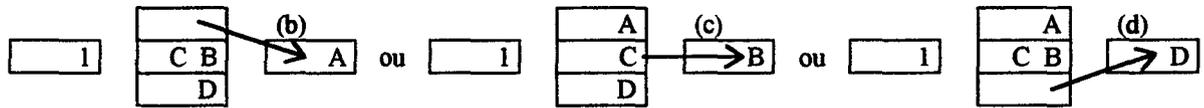
L'idée de l'algorithme est de supprimer toutes les configurations identiques et de ne proposer qu'une seule solution [BAR 98a].

Pour cela on utilise une approche directe. Pour obtenir une solution, on prend en compte dès le début l'ensemble des véhicules disponibles. Un véhicule est dit disponible s'il est, au pas P du parcours d'arbre, en tête de tiroir ou s'il appartient au poumon d'entrée et qu'un des tiroirs est vide (le véhicule a alors un chemin possible pour traverser le stock).

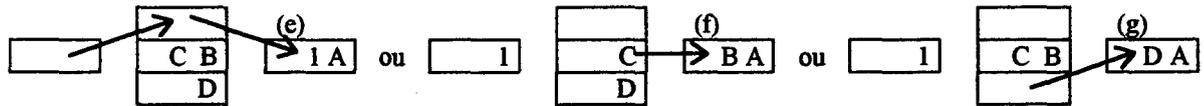
Prenons l'exemple suivant pour illustrer ces propos :



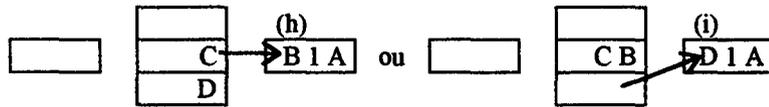
Pour le premier pas on a 3 solutions :



Si l'on fait le premier choix (b), on aura trois solutions par la suite



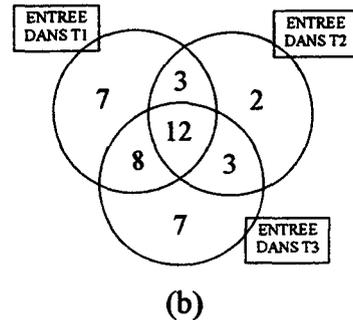
Si l'on fait le premier choix (e), on aura deux solutions pour le niveau suivant



Dans cet exemple, si l'on prend la première méthode, on a 3 possibilités de rangements pour le véhicule '1'. Cela donne pour la sortie:

entrée dans T1	30 solutions
entrée dans T2	20 solutions
entrée dans T3	30 solutions
total	80 solutions

(a)



(b)

Figure IV.11 - Affectation dans les tiroirs : comparaison des méthodes.

Pour la seconde méthode, on évite les configurations identiques et on accède directement aux 42 solutions différentes que l'on peut obtenir en faisant une entrée dans un ou plusieurs tiroirs comme le montre Figure IV.11 (b).

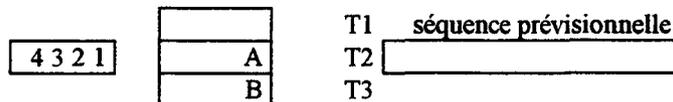
On peut remarquer que l'équilibrage des tiroirs permet d'obtenir un plus grand nombre d'ordonnements différents. Cela confirme les hypothèses faites pour l'heuristique de rangement dans les tiroirs.

La difficulté et la richesse de cet algorithme résident dans le cas où le nombre de véhicules en entrée est important. Une description de cet algorithme consiste à déterminer les choix possibles à chaque noeud du parcours d'arbre.

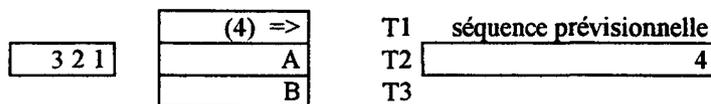
1. Il n'y a pas de passage direct à travers le stock (trou = 0) et le nombre de choix différents est nul (choix=0)
2. Pour chaque tiroir, on retient le véhicule en tête s'il existe (le nombre de choix est incrémenté  $\text{choix} = \text{choix} + 1$ ). Si aucun véhicule n'est là, un passage direct à travers le stock existe (trou = 1)
3. S'il n'y a pas de passage direct (trou = 0) alors on a tous les choix disponibles et l'algorithme de choix est terminé.  
 Sinon les véhicules du poumon d'entrée peuvent être choisis.  
 Mais il faut vérifier que le nouveau choix :
  - (test 1) ne remet pas en cause les choix précédents.
  - (test 2) n'amène pas à une solution inacceptable vis à vis des contraintes physiques (ordre des véhicules dans le poumon d'entrée).
 Dans le cas où le véhicule répond à ces critères, il peut être choisi :  $\text{choix} = \text{choix} + 1$ .

Le deuxième test est le plus difficile à mettre en oeuvre. En effet, il n'est pas utile dans le cadre d'un parcours d'arbre global. Si l'algorithme arrive à une solution impossible à appliquer, qu'il ne peut pas entièrement évaluer, il effectue alors un retour arrière standard et les solutions vérifiant le critère d'ordre seront trouvées. Le problème lié à une telle approche est la perte de temps liée à cette recherche inutile. L'algorithme que nous avons mis en place s'assure que le choix d'un véhicule à un niveau donné n'entraînera pas un blocage à un niveau suivant. Pour cela, il s'assure que le rangement des véhicules dans le poumon d'entrée non encore affectés à un tiroir est possible. La routine vérifie d'une part que les ordres entre véhicules sont vérifiés et d'autre part que la capacité des tiroirs est suffisante.

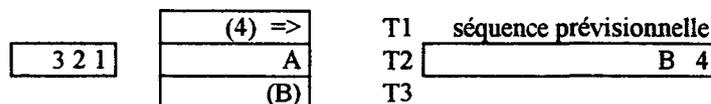
Prenons l'exemple suivant pour illustrer le fonctionnement de cet algorithme :



Le tiroir T1 est vide, donc les 6 véhicules "A", "B", "1", "2", "3" et "4" peuvent être choisis en première position de la séquence prévisionnelle. Nous supposons que le meilleur choix est le véhicule "4". Le véhicule "4" peut être alors affecté au tiroir T1 si les véhicules "3", "2" et "1" peuvent être rangés. Comme ils doivent être avant le véhicule "4" (règle d'ordre de succession ou de "précédence"), ils ne peuvent pas être orientés dans le tiroir T1. Il faut donc que les tiroirs T2 et T3 aient au total au moins trois places disponibles.

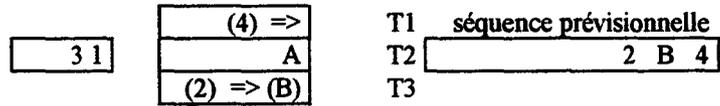


Au deuxième pas de l'algorithme, seuls les véhicules "A" et "B" peuvent être choisis. le meilleur choix est, par exemple, "A".



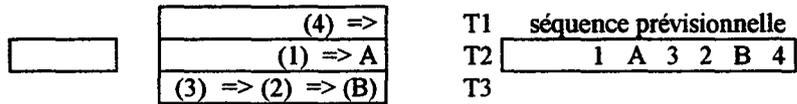
Au troisième pas de décision, les 4 véhicules "A", "1", "2" et "3" peuvent être choisis. Supposons que le meilleur choix est le véhicule "2". Le tiroir T3 est vide et le véhicule "2" peut y être affecté si les véhicules "3" et "1" peuvent être rangés dans le stock (en respectant les

règles d'ordre). Comme le véhicule "1" doit être avant le véhicule "4" et avant le véhicule "2" (règle de "précédence"), il ne peut pas être orienté vers les tiroirs T1 et T3. Il faut donc que le tiroir T2 ait au moins une place de libre. Ce n'est pas la peine de tester le cas pour le véhicule "3" car aucune nouvelle contrainte de rangement n'apparaît pour lui, il peut toujours être orienté vers T2 ou T3.



Si les conditions sont respectées, le choix d'affecter le véhicule "2" à T3 est retenu. Par la suite, le véhicule "1" ne pourra être dirigé que vers le tiroir T2. Le véhicule "3" pourra toujours être dirigé vers le tiroir T3.

Pour le pas suivant, les véhicules "A" et "3" sont possibles : l'affectation du véhicule "3" sera nécessairement le tiroir T3. Si le véhicule "3" est le meilleur choix, il est affecté au tiroir T3. Pour les deux pas suivants, les choix sont dans l'ordre, le véhicule "A" puis le véhicule "1" en l'affectant au tiroir T2.



L'algorithme utilisé fonctionne idéalement dans le cadre de tiroirs de taille infinie, mais nécessite plus de calcul dans le cas de tiroirs à taille limitée si l'on veut s'assurer que tous les véhicules peuvent être rangés dans les tiroirs. En effet l'algorithme a une complexité en  $O(N_e+T)$  et s'assure que les tiroirs ne sont pas saturés.

La Figure IV.12 illustre l'exemple de la Figure IV.10 en développant l'arbre en entier. Il montre la taille importante de l'arbre des solutions, même pour un petit cas. L'intérêt d'une telle méthode est d'une part le gain de temps par rapport à une politique cherchant à développer plusieurs arbres en parallèle. D'autre part une telle approche est plus facile à paramétrer pour obtenir le temps de réponse souhaité. En effet, il n'est pas nécessaire de parcourir  $T^{N_e}$  arbres différents qui correspondent aux diverses manières de ranger les véhicules en entrée du stock.

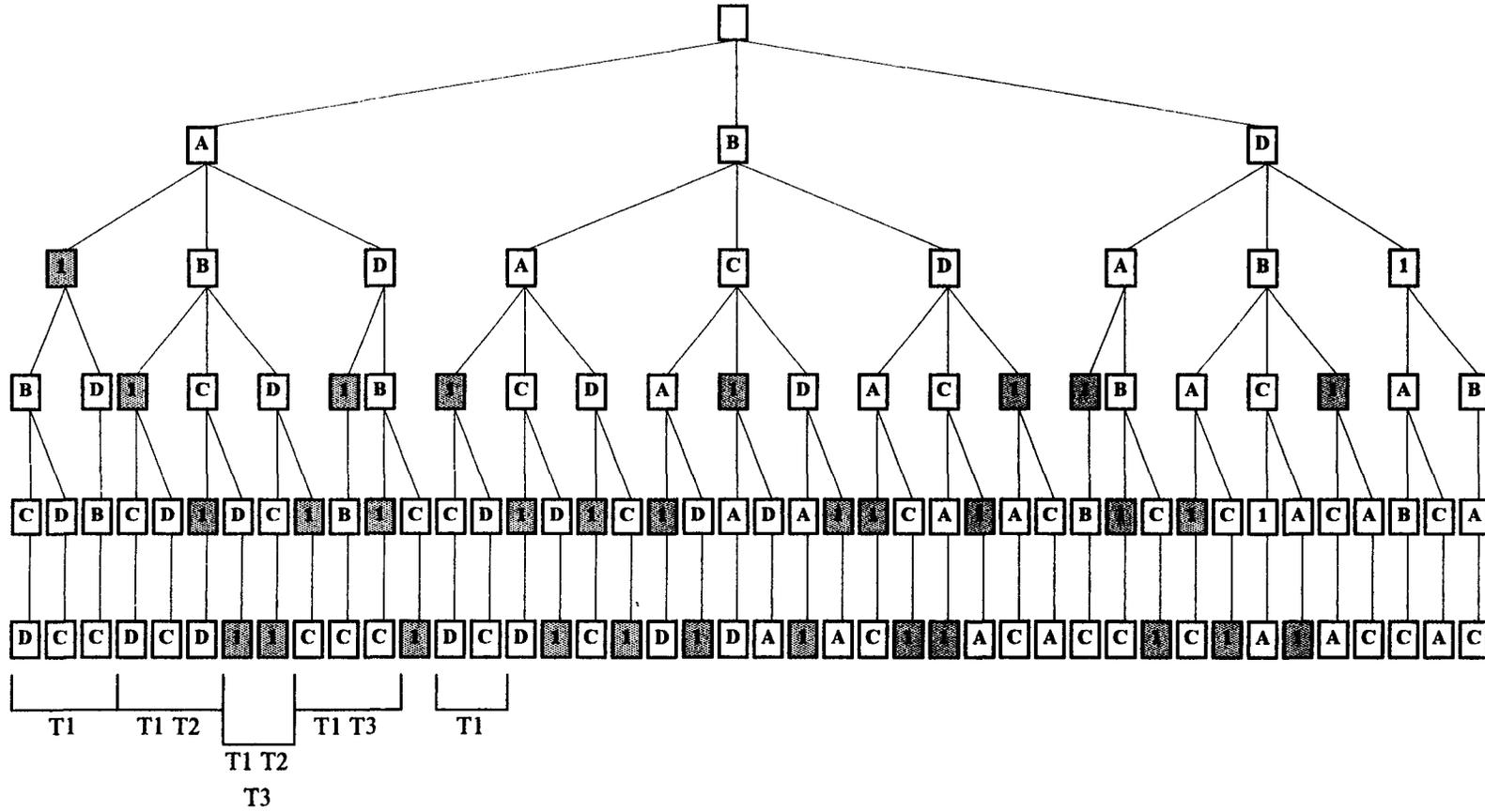


Figure IV.12 - Développement complet de l'arbre et obtention des 42 solutions avec l'orientation du véhicule "1" vers les tiroirs possibles.

## 5. Métriques de cadencement

Les métriques de type constructives doivent prendre en compte l'existant (au moins virtuellement) et on peut faire un comparatif avec le paragraphe 4.2.2 où l'approche est sensiblement la même.

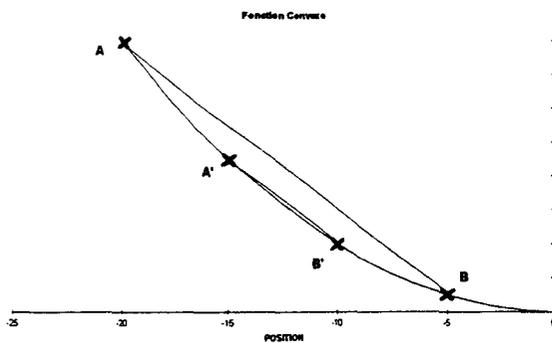
### 5.1. Convexité des métriques

Selon Kubiak et Sethi [KUB 94], le problème de séquencement se comporte comme un problème d'affectation lorsque la fonction de pénalisation ou métrique est convexe et non négative. Cette interprétation est d'autant plus justifiée que la fonction évaluant cette métrique est **strictement convexe**.

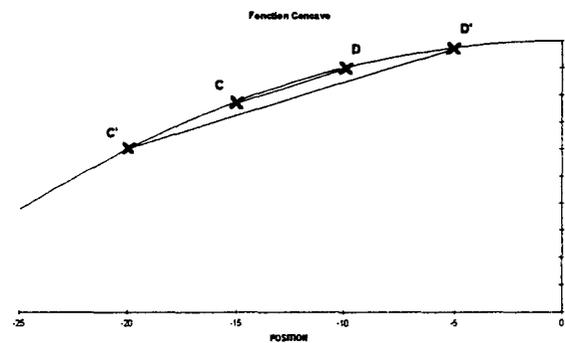
$$\forall x, y \text{ et } \lambda \in [0,1] \quad f(\lambda \cdot x + (1-\lambda) \cdot y) < \lambda \cdot f(x) + (1-\lambda) \cdot f(y)$$

Une fonction strictement convexe permet de réduire la dispersion des critères d'optimisation. En effet, la moyenne d'un indicateur est constante quelle que soit la solution obtenue. Il n'en est pas de même pour l'écart type. Notre objectif étant de réduire cette dispersion, il faut s'arranger pour que les objectifs soient le plus proche possible de la moyenne.

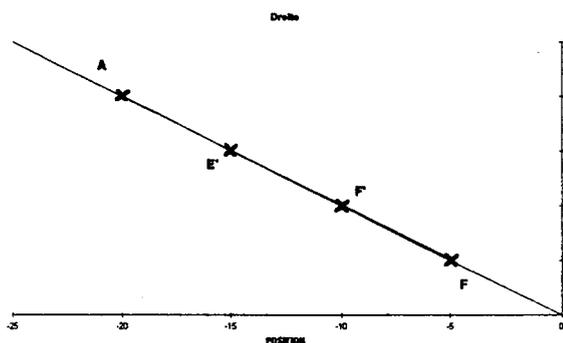
Prenons l'exemple suivant, lié à la date cible : deux véhicules ont les mêmes critères, mais le véhicule A devrait être positionné à la date 10 et le B à la date 20. Dans la séquence, que l'on peut créer, ils peuvent être positionnés en position 5 ou en position 25.



*fonction convexe*



*fonction concave*



fonction droite  
Figure IV.13 - Comparaison d'une métrique concave et d'une métrique convexe

Dans le cas de la fonction convexe, pour réduire le coût global de l'ensemble des produits, il faut réduire leur dispersion. Dans l'exemple ci dessus, le coût de A et B est supérieur à ce qu'il pourrait être, pour une même moyenne s'ils étaient positionnés en A' et B'. En revanche, dans le cas d'une métrique concave, la dispersion est encouragée en raison de la forme de la courbe : C' et D' sont une meilleure configuration (pour le coût) que C et D.

Le caractère strictement convexe apparaît dans le cadre où il est nécessaire de différencier deux cas proches mais différents dans le sens de la dispersion des données. Le troisième exemple de la figure illustre cette difficulté : il est impossible de différencier les deux possibilités, la moins bonne E'-F' et la meilleure E-F.

## 5.2. Répartition des produits

Pour la métrique de répartition des produits ou de lissage des critères, nous avons choisi de garder une vitesse de consommation constante. En effet dans le cadre d'un cadencement (séquencement **dynamique**), nous ne sommes pas à même de connaître avec exactitude l'espacement théorique que le flux doit respecter. Le volume prévisionnel de chaque critère est effectivement connu a priori. En revanche, le volume réel ne pourra être connu qu'a posteriori. Le volume prévisionnel peut en effet varier de quelques unités suite à des aléas de production. Il n'est donc pas judicieux de garder comme objectif un point précis (en l'occurrence  $(Q, N_j)$  dans la définition de Monden). Il est nécessaire de garder la notion de fenêtre glissante à long terme pour nos objectifs. D'où la volonté de garder la vitesse de consommation constante.

La métrique de pénalisation pour le lissage correspond à un non-respect de l'espacement moyen entre deux véhicules ayant le critère.

Pour calculer la valeur de pénalisation, nous utilisons une variable intermédiaire  $Ecourant_i$  qui est l'espacement actuel entre le dernier véhicule ayant le critère  $i$  et le véhicule courant  $k$ .

Cette variable est remise à 0 à chaque fois que le véhicule courant possède le critère, dans les autres cas, il est incrémenté.

Si le véhicule  $k$  possède le critère  $i$  et que  $Ecourant_i < Etheo_i$  alors le véhicule courant ne respecte pas l'espacement moyen, il faut donc le pénaliser de :

$$P = \left( \frac{Ecourant_i - Etheo_i}{Etheo_i} \right)^2$$

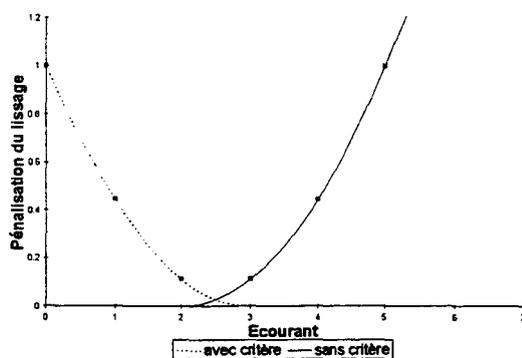
De même si le véhicule  $k$  ne possède le critère  $i$  et que  $Ecourant_i + 1 > Etheo_i$  alors le prochain véhicule ne respectera pas l'espacement, il faut donc le pénaliser de :

$$P = \left( \frac{Ecourant_i + 1 - Etheo_i}{Etheo_i} \right)^2$$

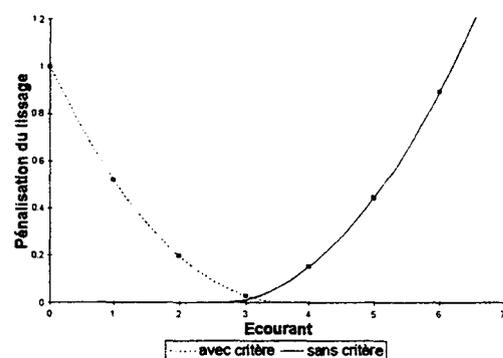
L'explication du ratio en  $1/Etheo_i$  correspond à la volonté de normaliser la métrique. Cette normalisation de la fonction permet par la suite de mieux comparer deux contraintes dont les ordres de grandeurs (espacements théoriques par exemple) sont très différentes. La fonction obtenue est convexe, ce qui correspond aux prérequis. En revanche, la puissance de 2 est choisie arbitrairement pour obtenir une fonction strictement convexe. Le choix de cette puissance est un paramètre difficile à régler. Il est également difficile d'en apprécier l'influence.

Cette approche est toujours valable dans le cas où l'espacement théorique est non entier. Dans ce cas, il n'existe pas de solution optimale de coût nul mais les solutions idéales oscilleront entre  $E(Etheo_i)$  et  $E(Etheo_i)+1$ .

Dans les deux exemples de la Figure IV.14, nous illustrons cette fonction de pénalisation  $P$ . Dans le cas d'un espacement entier, les courbes avec et sans critère sont symétriques. Dans le cas d'un espacement non entier (ici 3.6), les courbes sont décalées. Il n'existe pas de solution de coût nul car pour un espacement courant de 3, les deux choix sont pénalisés (mais celui sans critère moins que celui avec).



exemple de pénalisation pour  $Etheo=3$



exemple de pénalisation pour  $Etheo=3.6$

Figure IV.14 - Exemple de pénalisation pour le lissage des critères

Dans le cas où l'espacement théorique est inférieur à 1, il est préférable de prendre en compte des fenêtres et des ratios pour le lissage plutôt que la notion d'espacement.

### 5.3. Ratio

Pour les ratios, l'approche est sensiblement la même. L'objectif est de s'assurer du respect des ratios pour chaque véhicule de la séquence possédant le critère. Dans le cas où l'on souhaite respecter un espacement acceptable (ou son ratio associé), la fonction de pénalisation va être définie comme dans la partie III.3.2. Le ratio à respecter est  $N_i/D_i$ . Tout dépassement du nombre de véhicules ayant le critère dans une fenêtre de taille donnée est pénalisé.

Si le véhicule  $k$  ne possède pas le critère, il n'y a pas de pénalisation. En revanche si ce véhicule possède le critère, il est nécessaire de dénombrer le nombre «  $n$  » de véhicules ayant ce critère dans la fenêtre de taille  $D_i$ . Si ce nombre de véhicules est supérieur à  $N_i$ , il y a violation de l'acceptable et le véhicule est pénalisé de :

$$Pr = \left( \frac{n - N_i}{N_i} \right)^2 \text{ pour Pénalisation Ratio}$$

La puissance permet toujours d'avoir une fonction strictement convexe. Le dénominateur permet de normaliser l'importance de la pénalisation indépendamment de l'ordre de grandeur des ratios.

Dans le cas où le lissage est pris en compte sous forme de ratio (sous la forme  $NL_i/DL_i$ ), la métrique prend les mêmes paramètres (espacement courant, numérateur et dénominateur) et doit s'assurer que dans chaque fenêtre de taille  $DL_i$  le nombre de véhicules ayant le critère est le même que celui souhaité par le ratio (la fonction de pénalisation est la même que pour le lissage cf. chapitre 5.2).

Si le véhicule a le critère et si  $n$  est supérieur à  $N_i$  alors on pénalise de  $Pl$  (pour pénalisation lissage) :

$$Pl = \left( \frac{n - NL_i}{NL_i} \right)^2$$

Si le véhicule n'a pas le critère et si  $n$  est inférieur à  $N_i$  alors on pénalise de :

$$Pl = \left( \frac{n - NL_i}{NL_i} \right)^2$$

Nous allons illustrer maintenant les deux métriques de ratio et de répartition des produits.

Nous souhaitons avoir un flux qui respecte un ratio de 1 véhicule sur 2 ayant le critère (Espacement acceptable de 1,  $N_i=1$  et  $D_i=2$ ) et qui respecte un volume moyen de 1/3 (espacement théorique de 2,  $NL_i=1$ ,  $DL_i=3$ ). Les voitures blanches n'ont pas le critère, les autres l'ont. Nb est le nombre de véhicules ayant le critère dans la fenêtre de taille  $N_i$

	séquence de véhicule										
	fin										début
	11	10	9	8	7	6	5	4	3	2	1
											
Ecourant	0	1	0	3	2	1	0	2	1	0	
Pénalisation lissage	1	0.25	0	0	0.25	0	0	0	0	0	
Nb	2	1		1				1			
Pénalisation acceptable	1	0		0				0			

Dans cette séquence de véhicule, le septième ne respecte pas le lissage moyen du critère. En effet nous avons trois véhicules sans le critère entre deux véhicules qui l'ont. Le véhicule 7 est donc pénalisé pour mauvais lissage. Il aurait fallu un véhicule avec critère.

De même le véhicule 10 est pénalisé pour mauvais lissage. Dans ce cas le véhicule a le critère alors qu'il n'y a qu'un seul véhicule sans le critère avant. Il aurait fallu un véhicule sans critère.

Le onzième véhicule ne respecte ni le lissage moyen ni le ratio. Il est donc lourdement et doublement pénalisé.

#### 5.4. Position souhaitée et date

Une position souhaitée peut être considérée comme un cas particulier des dates au plus tôt et des dates au plus tard. Nous avons retenu une fonction de pénalisation du retard sous forme d'une exponentielle. En effet notre métrique constructive doit accélérer les véhicules les plus en retard pour réduire le délai et la dispersion de ce délai.

Si le véhicule  $k$  a une position souhaitée  $S_o$ , il est important de réduire l'écart entre  $k$  et  $S_o$ . Pour cela, il suffit de pénaliser les véhicules les plus en avance et de favoriser les véhicules en retard.

Si  $C$  est le coût du véhicule  $k$  sans la prise en compte de la position souhaitée, la pénalisation totale liée à ce véhicule sera :

$$P = C \cdot \exp \frac{S_o - k}{h}$$

où  $h$  est un paramètre à calibrer en fonction de l'importance du facteur retard.

Si  $k > S_0$ , le véhicule est en retard, la valeur de l'exponentielle est inférieure à 1. Le véhicule est donc plus facile à enligner. En revanche si  $k < S_0$ , le véhicule est en avance et  $P$  est supérieur à  $C$ , le véhicule est alors plus pénalisant à engager.

Le facteur  $h$  correspond au nombre de places d'avance pour lequel le coût de  $C$  est multiplié par la constante  $e$  (exponentielle de 1).

Une autre approche consiste à utiliser une fonction en escalier qui permet de définir des paliers de véhicules (en avance, en retard, très en avance, très en retard, à l'heure). Les risques liés à une telle approche sont les phénomènes liés à une fonction discontinue. Il faut s'assurer qu'il n'y a pas de dérives.

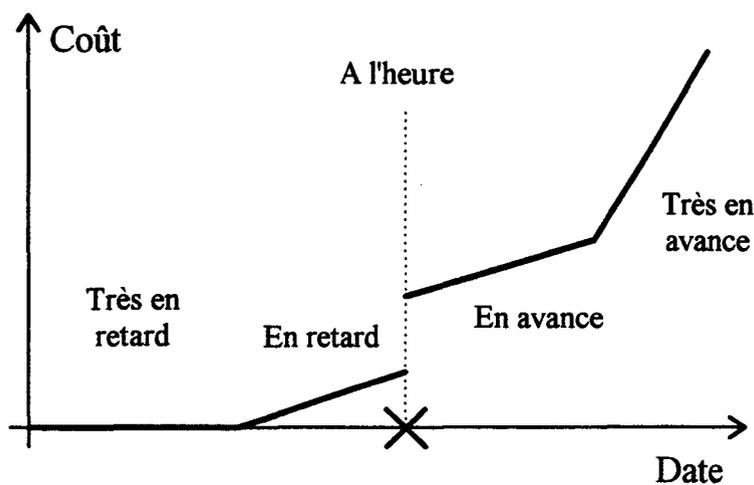


Figure IV.15 - Fonction de date souhaitée en escalier

Dans le chapitre 7, nous verrons les deux utilisations et leur différences pratiques.

### 5.5. Rafales

Pour les rafales nous utilisons la même métrique que dans la troisième partie (III.3.3).

### 5.6. Interaction entre les règles

Comme nous l'avons indiqué dans la partie II, il est important de s'assurer de la compatibilité entre les contraintes. Par exemple, il n'est pas possible d'optimiser à la fois une contrainte de rafale et une contrainte d'espacement pour un même critère.

En revanche il est possible d'optimiser les règles de lissage et de ratio en même temps. Il est néanmoins préférable de s'assurer que l'indicateur  $I_r$  (indicateur de difficulté à respecter les ratios cf. II.3.2.1) associé est positif sinon l'une des deux règles ne pourra pas être respectée.

Dans le cadre du cadencement entre les ateliers peinture et montage, nous prenons en compte le lissage des critères, le respect des ratios et dans certains cas la position souhaitée. La

métrique de cadencement que nous avons élaborée pour tenir compte de toutes ces règles est basée sur la formule suivante :

$$F(k) = \left( \sum_{i=1}^c \alpha_i \Pr(i, k) + \beta_i Pl(i, k) \right) \exp\left(\frac{So - k}{h}\right)$$

$F(k)$  est le surcoût lié au véhicule en position  $k$ .  $\Pr(i, k)$  et  $Pl(i, k)$  sont les coûts unitaires liés aux respects des ratios et du lissage des critères décrits précédemment.

Pour paramétrer la métrique au complet, il est important de régler les poids respectifs des contraintes  $\alpha_i$  et  $\beta_i$ .

Dans un premier temps, nous proposons une méthode pour régler les poids des ratios par rapport au lissage. Comme le respect des ratios est plus important que le lissage des critères, nous proposons qu'un dépassement de l'espacement acceptable soit plus pénalisé qu'un très mauvais lissage. La définition de  $Pl(i, k)$  nous a permis de voir que :

$$Ecourant \leq 2.Etheorique \Leftrightarrow Pl(i, k) \leq 1$$

Le cas où  $Ecourant$  est supérieur au double de  $Etheorique$ , est rare car alors, le volume journalier ne pourra être respecté.

Donc le cas nominal correspond à  $Pl(i, k) < 1$ , ce qui nous permet de poser :

$$\forall i, \alpha_i > \sum_{j=1}^c \beta_j$$

pour s'assurer de la prépondérance des pénalisations pour non respect des ratios (ou de l'espacement acceptable).

Cette approche peut être appliquée également à la classification des contraintes de respects des ratios entre elles.

Supposons 10 contraintes correspondant à 10 critères, classés

$$A > B > C > D > \dots > I > J$$

où  $A > B$  signifie que le respect de  $A$  est plus important que celui de  $B$ , etc.

Par la suite, on peut essayer de partitionner les contraintes en classes de contraintes exclusives :

classe I : il faut absolument garantir le ratio

classe II : il faut éviter de ne pas garantir le ratio

classe III : il faut si possible garantir le ratio

En utilisant la même méthode que pour l'équivalence des poids entre lissage et ratios acceptables, on peut poser comme condition :

$$\forall i \in \text{classelI}, \alpha_i > \sum \alpha_j (j \in \text{classelII})$$

$$\forall i \in \text{classelI}, \alpha_i > \sum \alpha_j (j \in \text{classelII} \text{ et } j \in \text{classelIII})$$

Une fois toutes ces contraintes respectées, nous sommes capables de donner une valeur à chacun des poids et de pouvoir contrôler l'impact de chaque règle que l'on souhaite utiliser.

Supposons dans notre exemple que ces dix contraintes ont été classées comme suit :

- classe I (garantir absolument le ratio) : A, B puis C
- classe II (éviter de ne pas garantir le ratio) : D, E, F puis G
- classe III (garantir si possible le ratio) : H, I puis J.

Pour garantir les ratios, les poids sont définis de la façon suivante :

	classe III			classe II				classe I		
	J	I	H	G	F	E	D	C	B	A
$\beta$	1	2	3	7	8	9	10	41	42	43
somme des $\beta_i$	pour la classe III		6	pour les classes II et III			40	pour les classes I, II et III		166
$\alpha$	167	168	169	671	672	673	674	3361	3362	3363
somme de tous $\beta_i$ et des $\alpha_i$ pour	pour la classe III		670	pour les classes II et III			3360			

Figure IV.16 - Exemple de calcul des poids entre règles

Cette classification permet de définir les poids reliant les différentes contraintes et les différents critères entre eux, pour permettre d'avoir une classification exclusive quand on compare deux véhicules pour une même place dans la séquence.

Dans le cas où l'on souhaite introduire des contraintes de nature différente, par exemple ratios et rafales ou ratios et position souhaitée, l'approche est similaire. En revanche, comme les fonctions d'évaluation représentent des grandeurs complètement différentes, il est nécessaire de s'assurer des bornes de chacune d'elle pour pouvoir les comparer entre elles.

## 6. Approche de résolution proposée

Dans ce chapitre, nous allons décrire l'approche modulaire retenue pour notre algorithme de décision et de gestion de stock. Cette approche doit prendre en compte les particularités physiques des stocks d'une part et d'autre part être capable de prendre en compte une ou plusieurs des métriques décrites dans le paragraphe 5.

### 6.1. Structure de l'algorithme

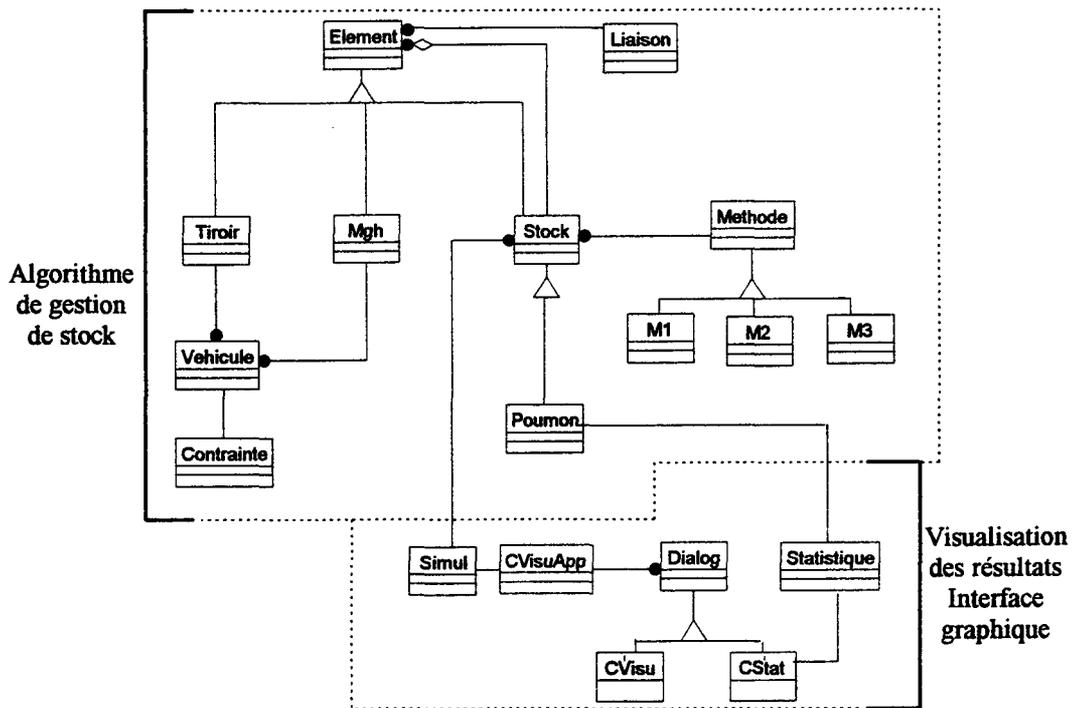


Figure IV.17 - Modélisation OMT de l'algorithme et de son interface.

Un algorithme gérant un stock doit posséder différentes fonctionnalités :

- cadencement des véhicules
- gestion virtuelle du stock
- métrique d'optimisation du séquençement.

Le cadencement des véhicules doit pouvoir être géré de manières différentes en fonction de la topologie du stock. En effet, nous avons vu que pour la sortie du stock, un arbre de décision est la meilleure approche. Dans le cas d'un autostockeur (ou MGH), seule la sortie est gérée par l'arbre de décision. Dans le cas d'un stock à tiroir ou d'un stock mixte, une décision doit toujours être prise en sortie, mais une autre doit être prise en entrée pour affecter les véhicules dans les tiroirs. Dans ce cas là deux approches sont possibles. Soit l'algorithme gère à la fois l'entrée et la sortie à l'aide d'un arbre de décision, soit l'entrée dans le stock est décidée à l'aide d'une heuristique et seule la sortie est optimisée à l'aide de l'arbre de décision.

Nous avons donc deux fonctions :

- Cadencement\_sortie
- Cadencement\_entrée

qui peuvent être paramétrées en fonction du type de stock et de la stratégie de rangement choisie (Sortie seulement, Entrée et Sortie simultanées ou Entrée et Sortie indépendantes).

L'algorithme de gestion du stock doit également avoir une vision parfaite du stock ainsi que des poutons d'entrée et de sortie. La connaissance du pouton de sortie permet d'évaluer le coût du véhicule enligné par rapport aux véhicules déjà enlignés. La connaissance du pouton d'entrée est nécessaire pour prendre des décisions intelligentes en entrée. L'algorithme doit également connaître à chaque niveau de l'arbre la composition exacte du stock ainsi que sa répartition. Pour cela nous avons besoin de trois fonctions :

- Détermine\_véhicules\_disponibles
- Enlève\_virtuellement\_véhicule\_stock
- Ajoute\_virtuellement\_véhicule\_stock

La première fonction (*Détermine\_véhicules\_disponibles*) permet à chaque noeud de l'arbre de connaître les véhicules qui sont susceptibles d'être enlignés. Les deux dernières correspondent à la gestion virtuelle du stock quand l'arbre de décision fait le choix d'enligner un véhicule. La gestion virtuelle du stock consiste à prévoir, lors du parcours d'arbre, l'évolution du contenu du stock. Quand un véhicule est enligné de manière prévisionnelle, il est nécessaire de répercuter ce choix sur la vue du stock que l'algorithme a. Cette fonctionnalité est relativement facile dans le cas d'un autostockeur, elle est plus compliquée dans le cas de stocks à tiroirs.

Ces trois fonctions dépendent évidemment du style de stock. Dans un autostockeur, les véhicules disponibles sont tous les véhicules présents dans le stock à l'instant virtuel donné, c'est à dire à un noeud donné du parcours d'arbre à l'instant considéré. Dans le cas d'un stock à tiroirs, ce ne sont que les véhicules positionnés en tête de tiroirs.

Finalement, l'algorithme doit avoir une métrique qui lui permet d'évaluer la qualité du séquençement qu'il propose de générer. Cette fonction est indépendante du style de stock et ne prend en compte que les règles de séquençement que l'on souhaite respecter en sortie de stock.

Ces fonctions sont le coeur de l'algorithme car elles doivent à la fois permettre la flexibilité en fonction du style de stock et garantir le temps de réponse de l'algorithme.

L'algorithme est donc un parcours d'arbre en profondeur d'abord (pour garantir de bonnes performances mémoire). Le principe de l'algorithme récursif est le suivant :

<p><b>Etape 1</b> : Détermine_véhicules_disponibles permet d'obtenir à un noeud donné ("k") les véhicules qui peuvent être enlignés <math>\{P_{1,k}, \dots, P_{P,k}\} \in D_k^P</math></p> <p><b>Etape 2</b> : On trie les véhicules de <math>D_k^P</math> en fonction de la métrique choisie.</p> <p><b>Etape 3</b> : Pour chacun des véhicules de <math>D_k^P</math>, on effectue un enlignement virtuel et réitère l'algorithme au noeud ("k+1")</p> <p><b>Etape 4</b> : Si une feuille est atteinte, la solution trouvée est évaluée par rapport à la meilleure trouvée jusque là</p> <p><b>Etape 5</b> : Lors du retour arrière, l'algorithme doit restituer l'état initial.</p>
---

Figure IV.18 - Parcours d'arbre

## 6.2. Heuristiques et paramètres pour le parcours d'arbre

Nous avons vu dans le cas des stocks à tiroirs que différentes stratégies peuvent être mises en oeuvre pour la gestion de l'entrée et de la sortie : gestion indépendante ou simultanée; politique d'enveloppe ou non. Le parcours d'arbre est également paramétrable : profondeur des arbres pour les cadencements en sortie; largeur des arbres ; temps de calcul des parcours.

Nous avons utilisé deux jeux de test.

Le premier possède 10 contraintes sur 15 jours de production avec un flux peu contraint : (Ir moyen : 0.45, minimum : 0.115 et maximum : 0.72). Le flux peu contraint permet de prendre en compte en entrée deux types d'heuristiques. Le deuxième jeu de test (10 contraintes sur 7 jours de production) est un flux très contraint : Ir moyen : 0.46, minimum : -0.027 et maximum : 0.921 (et deux contraintes sur les dix ont un Ir négatif).

Le premier jeu de test va principalement servir à illustrer les résultats utilisant les politiques d'enveloppes et les heuristiques. Pour le deuxième jeu de test, les résultats sont cohérents avec le premier mais montre des limites si nous utilisons des heuristiques en entrée. Nous nous servirons de ce deuxième jeu pour illustrer les résultats suivants.

### 6.2.1. Comparaisons des méthodes sur un jeu de test

Nous allons décrire les résultats obtenus avec le premier jeu de test pour lequel nous avons mis en place les différentes stratégies de gestion des entrées et des sorties du stock : méthodes avec enveloppes et méthodes avec heuristiques de rangement.

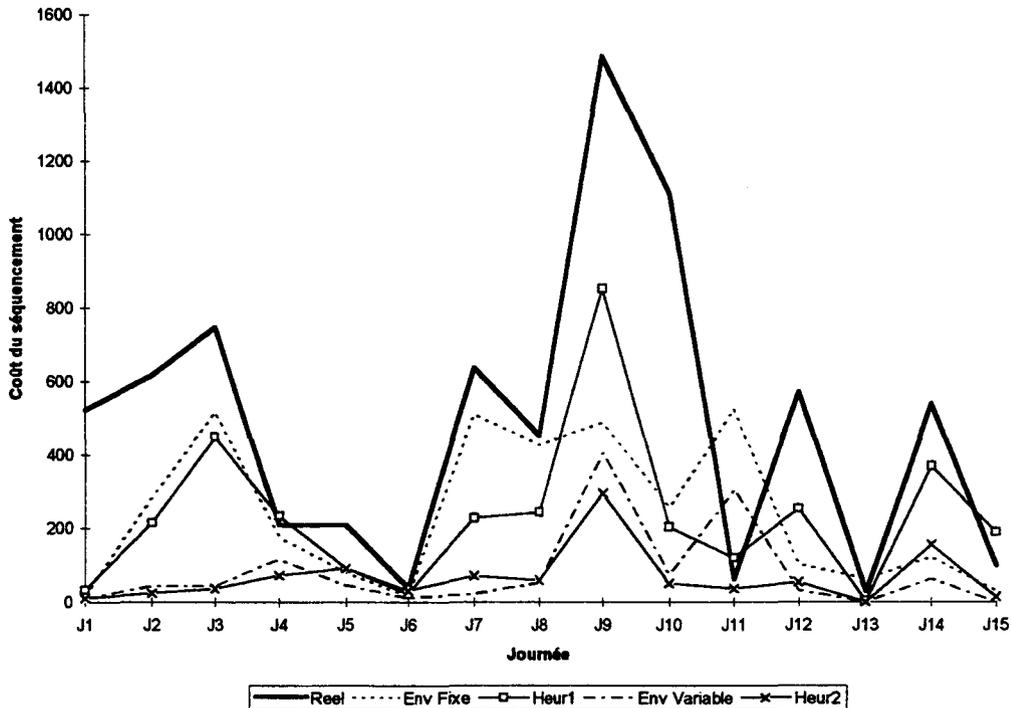


Figure IV.19 - Stratégies de séquençage entrée-sortie

Cette première étude montre les résultats comparés sur le premier jeu de test. La référence est l'enlèvement effectivement réalisé par les pilotes dans la réalité. Dans ce jeu de test, nous avons suivi le séquençage entrée-sortie du stock effectué par les pilotes du flux.

Pour chacune des méthodes le temps de calcul est similaire (5 secondes de calcul par temps de cycle sur un DX2/66).

La méthode avec enveloppe fixe prend en compte l'entrée dépendante de la sortie. La taille de l'enveloppe est fixée ici à 20 véhicules, l'anticipation en entrée (la valeur  $N_e$ ) est de trois véhicules.

La méthode avec enveloppe variable reprend les concepts développés précédemment en recadencant le stock, quand un véhicule entre. Cette approche permet de garantir une meilleure qualité de la séquence, en particulier lorsque le stock n'est pas en fonctionnement nominal (stock presque vide, tiroirs saturés ou vides).

L'heuristique 1 va répartir les volumes de critères entre tous les tiroirs. Cette heuristique permet d'offrir un choix varié de véhicules en sortie de stock.

L'heuristique 2 va répartir en entrée les critères pour que des sous-listes puissent être recrées dans les tiroirs. Cette approche permet de créer de bonnes sous-séquences qu'il est nécessaire de fusionner pour atteindre notre résultat. Cette approche fonctionne bien si le flux est peu dense et s'il n'y a pas de phénomènes de saturation dans les tiroirs.

Pour les deux heuristiques, l'algorithme de sortie est le même que pour les enveloppes : c'est un B&B. Mais la profondeur du parcours d'arbre est plus importante car l'algorithme dispose en plus du temps de calcul imparti à l'entrée dans le cas des deux autres algorithmes.

Les résultats globaux sur les quinze jours donnent :

Gain par rapport à	Enveloppe Fixe	Heuristique 1	Envelop. Variable	Heuristique 2
Enlignement Réel	50.82%	52.40%	83.33%	86.36%

Figure IV.20 - Gains apportés par les différentes stratégies

Les quatre méthodes décrites ci-dessus ont donné des résultats meilleurs que la solution réellement enlignée. La méthode à enveloppe fixe donne des résultats corrects mais sensiblement moins bons que celle à enveloppe variable. En effet l'ordonnement à l'intérieur du stock est plus figé et cela peut poser des problèmes quand le stock est rempli de manière déséquilibrée. Dans le cas où des tiroirs sont vides ou dans le cas où certains ont un taux de remplissage beaucoup plus important que d'autres, la méthode à enveloppe de taille variable permet de remettre en cause des choix sur une partie plus importante de la séquence. Dans certains cas limites, cette séquence prévisionnelle est entièrement remise en cause. Le temps de calcul est plus important, mais le gain au niveau de la qualité est très important. La robustesse de l'algorithme est renforcée vis-à-vis de fonctionnements non nominaux du stock.

D'autre part, l'heuristique 1 est moins bonne que l'heuristique 2 car dans le cas où le flux d'entrée est très perturbé (des rafales trop longues de véhicules ayant le même critère), l'heuristique n'est plus capable de traiter correctement cette perturbation.

A temps de calcul égal, l'heuristique 2 est un peu meilleure que la méthode à enveloppe variable car la profondeur d'arbre est plus importante. Cela est dû au temps de calcul gagné en entrée par l'heuristique. En revanche si des arbres de même taille sont utilisés pour la méthode à enveloppe variable, on obtient des résultats meilleurs mais dans un temps de calcul plus important (ce temps de calcul dépendant du nombre de véhicules en anticipation, c'est-à-dire de  $N_e$ ).

Par ailleurs, nous nous sommes aperçus que les heuristiques sont moins efficaces quand le flux de production est très contraint. C'est le cas lorsque les contraintes d'espacement minimum et moyen sont égales. En effet, pour effectuer un bon ordonnancement, il faut ranger les véhicules de manière parfaite, ce que les heuristiques ne peuvent pas toujours faire.

Pour des raisons de performance et de robustesse de l'algorithme, nous avons retenu la méthode à enveloppe variable en recadençant le stock à l'aide de petits arbres de décision.

Ce premier jeu de test permet de faire également une remarque importante. La simulation du fonctionnement du stock doit être faite sur plusieurs journées. En effet, la capacité de stockage correspond à une fraction non négligeable d'une journée de production (80 véhicules sur 500). Les résultats obtenus avec le fonctionnement d'une seule journée peuvent être faussés, si le stock retient des véhicules difficiles à enligner. Delaval [DEL 97] contourne ce problème en évaluant la charge restant dans le stock. Dans le cas d'un stock à tiroirs, cette évaluation est assez difficile. Les résultats de la journée 11 montrent un phénomène paradoxal : l'enlèvement réel est meilleur que celui des différents algorithmes. L'explication en est très simple. La journée 10 était une journée difficile comme le montre les très mauvais résultats de l'enlèvement réel. Les algorithmes développés ont retenu des véhicules difficiles pour réduire le coût global de l'enlèvement. Ces véhicules ont du être enlignés le lendemain (journée 11) et le coût de cette journée en a été d'autant plus important. De même les algorithmes ont anticipé, la veille, les difficultés de la journée 12.

Les simulations doivent donc être effectuées sur de longues périodes pour éviter deux phénomènes de bord : début ou fin difficile de simulation et capacité de rétention du stock.

### 6.2.2. Paramétrage de l'arbre

L'arbre de décision mis en oeuvre est un Branch&Bound tronqué que l'on parcourt en profondeur d'abord. Le parcours d'arbre est à la fois tronqué en profondeur et en largeur pour limiter le temps de réponse.

La profondeur est définie a priori et les largeurs de recherche sont définies pour chaque profondeur.

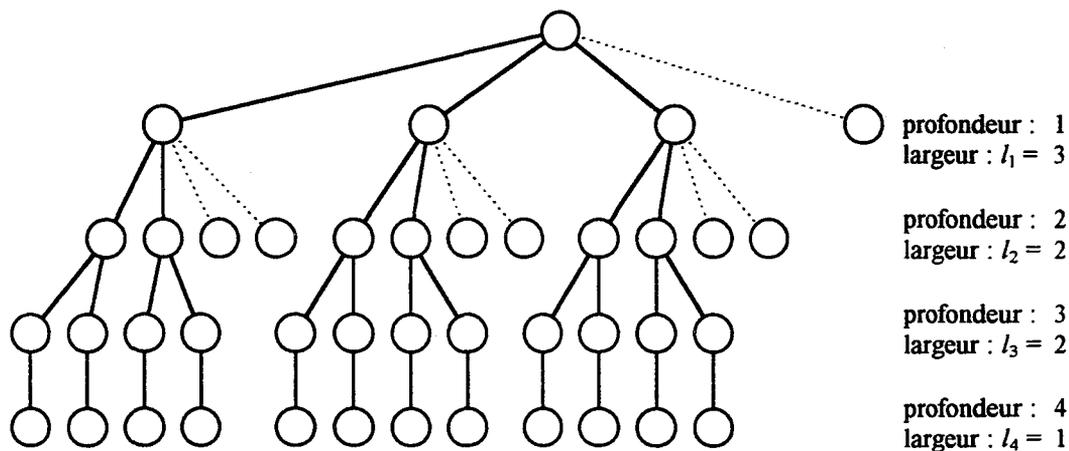


Figure IV.21 - Parcours d'arbre et largeurs.

Dans le parcours d'arbre de la Figure IV.21, les branches en trait continu sont celles qui sont effectivement explorées, celles en pointillés sont abandonnées car on suppose qu'elles ne permettent pas d'atteindre des bonnes solutions. Il est à noter également que la largeur pour la dernière profondeur de l'arbre est nécessairement 1 car on ne garde que la meilleure solution.

Castelain [CAS 95] et Delaval [DEL 97] proposent pour dimensionner l'arbre d'utiliser une matrice de préférence et un vecteur de stabilité de séquence.

Nous n'avons pas retenu cette approche car les paramètres sont alors susceptibles de changer en fonction de la densité du flux entrant dans le stock.

Nous avons préféré paramétrer l'arbre de décision en fonction du temps de calcul dont dispose l'algorithme. En effet, on est capable de déterminer la taille de l'arbre a priori. Pour l'exemple ci dessus, le nombre de noeuds  $N$  parcourus est :

$$N = l_1 + l_1 \cdot l_2 + l_1 \cdot l_2 \cdot l_3 + l_1 \cdot l_2 \cdot l_3 \cdot l_4 = 33$$

D'autre part le temps de calcul pour chaque noeud est constant (il ne dépend que du nombre de véhicules à évaluer pour le rang suivant).

Dans le cas des tiroirs, la valeur de  $l_1$  correspond au nombre maximum de véhicules disponibles en sortie de stock. Les valeurs de  $l_i$  suivantes décroissent pour atteindre 1.

Pour le calcul en entrée de stock à tiroirs, la taille de l'arbre est calculée automatiquement en fonction de l'enveloppe de réordonnement.

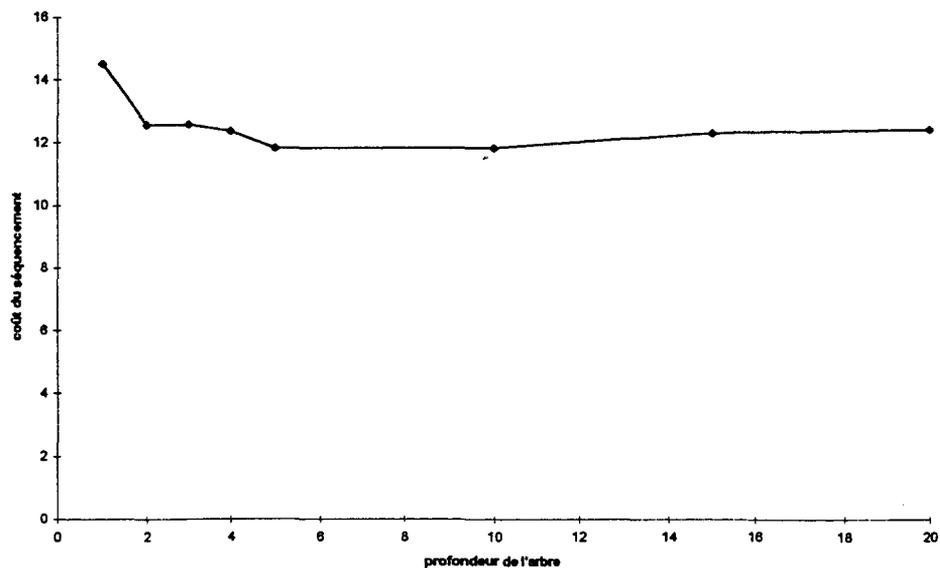


Figure IV.22 - Importance de la profondeur de l'arbre sur la qualité du séquençement (à temps de calcul égaux).

La taille de l'arbre de calcul en sortie du stock à tiroirs est relativement importante. Les gains peuvent être (à temps de calcul constant) de l'ordre de 20%.

Si la profondeur de l'arbre est trop faible, les choix effectués sont myopes. Le premier véhicule choisi permet une bonne qualité de séquençement par rapport aux véhicules déjà enlignés mais ne s'assure pas qu'il existe un séquençement potentiellement bon pour les véhicules suivants.

Si la profondeur de l'arbre est trop importante, l'algorithme passera beaucoup de temps à optimiser les derniers véhicules du séquençement prévisionnel sans remettre en cause le séquençement des premiers véhicules qui est le plus important. (La fonction de coût se dégrade légèrement quand la taille augmente) (cf. Figure IV.22).

Il existe donc une taille optimale de la séquence prévisionnelle qui permet de concilier à la fois temps de calcul et qualité de la séquence globale. Expérimentalement, nous nous sommes aperçus que cette taille optimale devait être supérieure à la plus grande des fenêtres utilisées pour les ratios.

$$prof > \max(D_i)$$

Cette remarque se justifie par le fait que le parcours d'arbre prend en compte au moins deux véhicules ayant le même critère : on peut ainsi s'assurer que le ratio est respecté ou non dans au moins une fenêtre pour chaque critère.

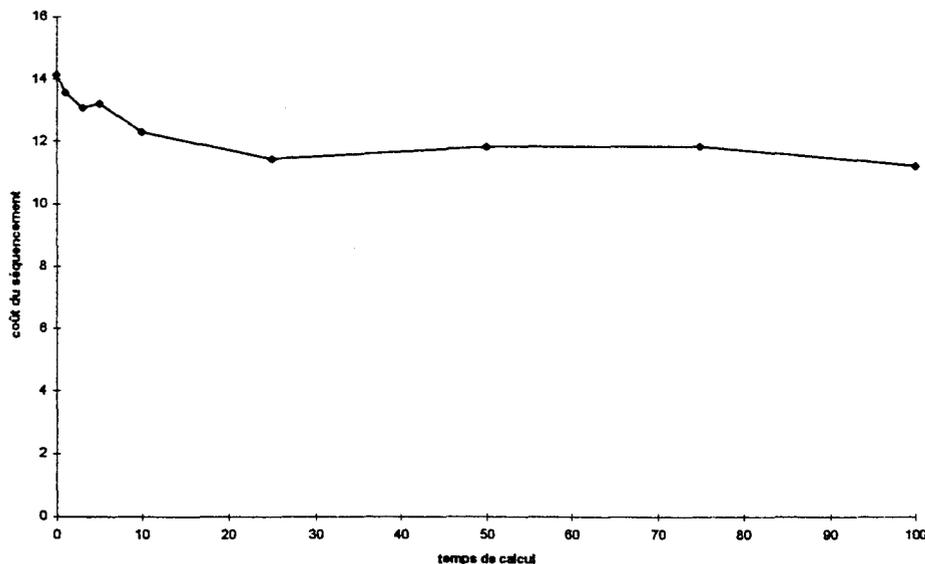


Figure IV.23 - Importance du temps de calcul sur la qualité du séquençement

Dans un deuxième temps, il est également important de dissocier la qualité du séquençement et le temps de calcul. La figure IV-23 montre que la qualité du séquençement est bonne et stable à partir d'un certain seuil qu'il est important d'établir expérimentalement. Ici, par

exemple, ce seuil vaut 30 unités de temps. (A titre indicatif, l'unité de temps correspond à une seconde de calcul sur PC DX2/66).

### 6.2.3. Paramétrage du stock

Nous allons maintenant décrire l'influence que peuvent avoir les différents paramètres du stock. Ces paramètres sont au nombre de deux. Le premier est dynamique : le taux de remplissage du stock. Le second est statique et son choix doit être fait à la conception du stock : c'est la topologie du stock (par exemple, c'est le nombre de tiroirs et leur capacité). L'influence du premier paramètre est intéressante à connaître dans le cadre d'un fonctionnement automatique du stock. L'influence du deuxième est à prendre en compte en phase de conception de l'installation pour dimensionner le stock en vue de la fabrication d'un certain type de véhicules.

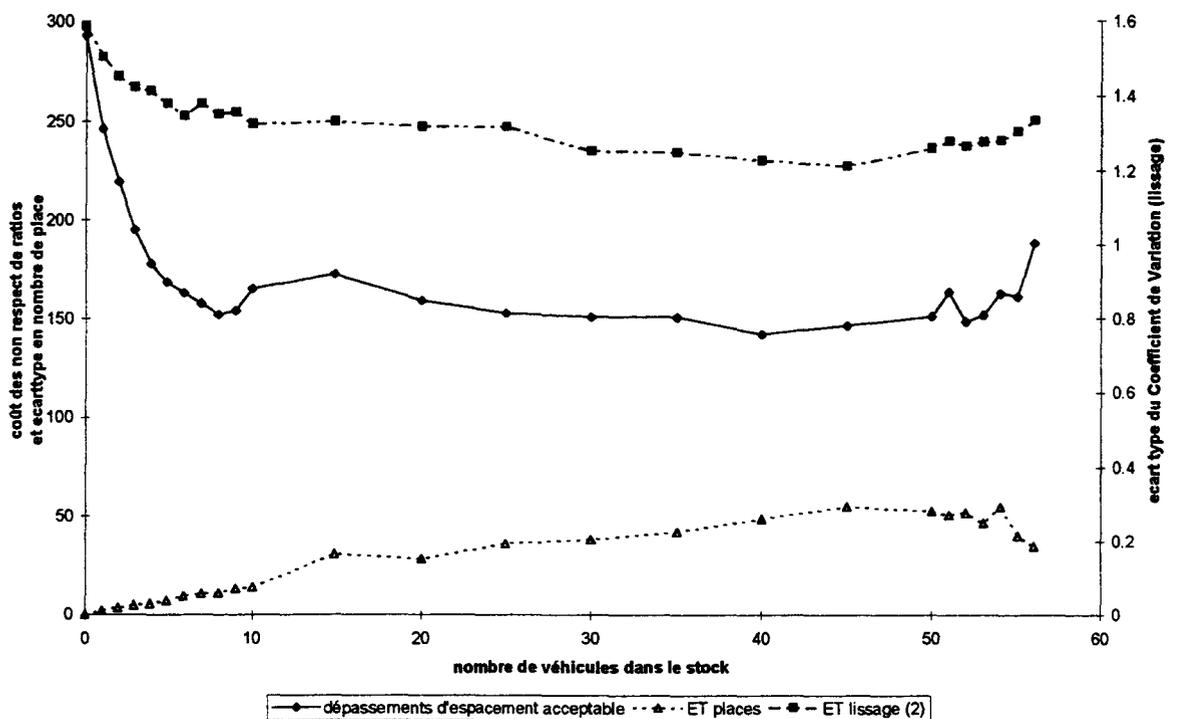


Figure IV.24 - Influence du taux de remplissage pour un stock de 4 tiroirs.

Les dépassements d'espace acceptable illustrent le respect des capacités de production, l'ET place (Ecart type du nombre de places par rapport à la position de référence du véhicule) illustre le respect des dates.

Le taux de remplissage du stock a une grande importance sur les performances de l'algorithme et celles du stock, comme le montre la simulation sur 7 jours de production et 10

critères de la Figure IV.24. Les indicateurs repérés sont une note pour le respect des ratios, une note pour le délai cumulé des véhicules et une note évaluant le lissage des critères.

Le respect des ratios est évalué par une moyenne pondérée des dépassements d'espace acceptable. Le délai des véhicules est évalué par l'écart-type du nombre de places d'avance ou de retard de chaque véhicule par rapport à sa date de référence. Le lissage des critères est évalué par l'écart type de la somme pondérée des Coefficients de Variation (cf. II.3.2.2).

Ces performances globales peuvent être décrites avec la figure ci-dessous :

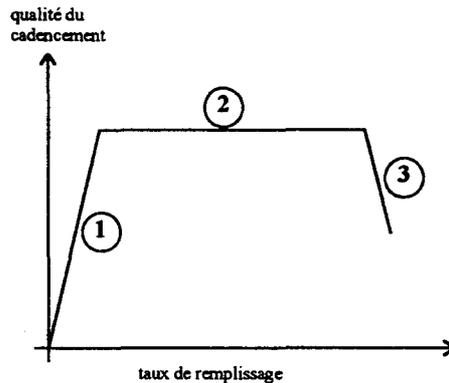


Figure IV.25 - Synthèse de l'influence du taux de remplissage.

La qualité de la liste dépend peu du taux de remplissage du stock si on évite les deux phénomènes de bord que sont la saturation (3) ou la famine (1) du stock. Le fonctionnement idéal du stock est donc le palier (2) qui permet un bon cadencement du flux avec une bonne qualité de séquence en sortie de stock.

La famine (1) s'explique par le faible choix des véhicules dans le stock. Dans l'exemple de la Figure IV.24, le seuil de 10 véhicules correspond à celui de la profondeur de l'arbre de recherche. L'explication de ce seuil est la même que pour la profondeur. En effet, il est nécessaire de respecter un ratio de 9/10 pour un critère avec un volume moyen de 9/10 également. Il faut donc au moins un véhicule sur 10 sans le critère. La probabilité d'avoir 10 véhicules ayant le critère parmi les 10 est de près de 35% (c'est  $(0.9)^{10}$ ). Il est donc intéressant d'avoir un nombre de véhicules important pour augmenter la probabilité d'avoir le véhicule avec les critères souhaités présent dans le stock. Dans le cas des tiroirs c'est une condition nécessaire mais pas suffisante.

La saturation du stock (3) est un phénomène moins important mais qu'il ne faut pas négliger. La qualité de la liste est moins dégradée que dans le cas de la famine. En revanche ce dysfonctionnement de l'ordonnancement est dû à l'incapacité à gérer correctement l'entrée du stock. Ce problème n'apparaît pas en effet dans le cas d'un autostockeur (ou MGH). Ce phénomène de saturation commence à apparaître quand des tiroirs sont pleins, le choix en

entrée en est automatiquement réduit et la possibilité de ne pas faire un bon rangement augmente. La pire configuration apparaît quand tous les tiroirs sauf un sont pleins, il ne reste alors qu'un seul choix possible en entrée et l'algorithme ne peut donc pas faire de proposition : il n'y a qu'une seule solution, le tiroir où il reste de la place.

Le point de fonctionnement idéal du stock se situe dans la partie gauche du palier (2). En effet, avec ce taux de remplissage du stock, l'algorithme est capable de générer un bon cadencement, proche de l'optimum et le nombre de véhicules immobilisés dans le stock est réduit (le temps de défilement moyen dans le stock en est réduit d'autant).

Il faut également noter que les résultats sont des moyennes et ne reflètent pas les écarts qu'il peut y avoir entre un flux régulier et un flux très perturbé : plus le stock est rempli, plus sa capacité à absorber les aléas de fabrication et les perturbations du flux est importante.

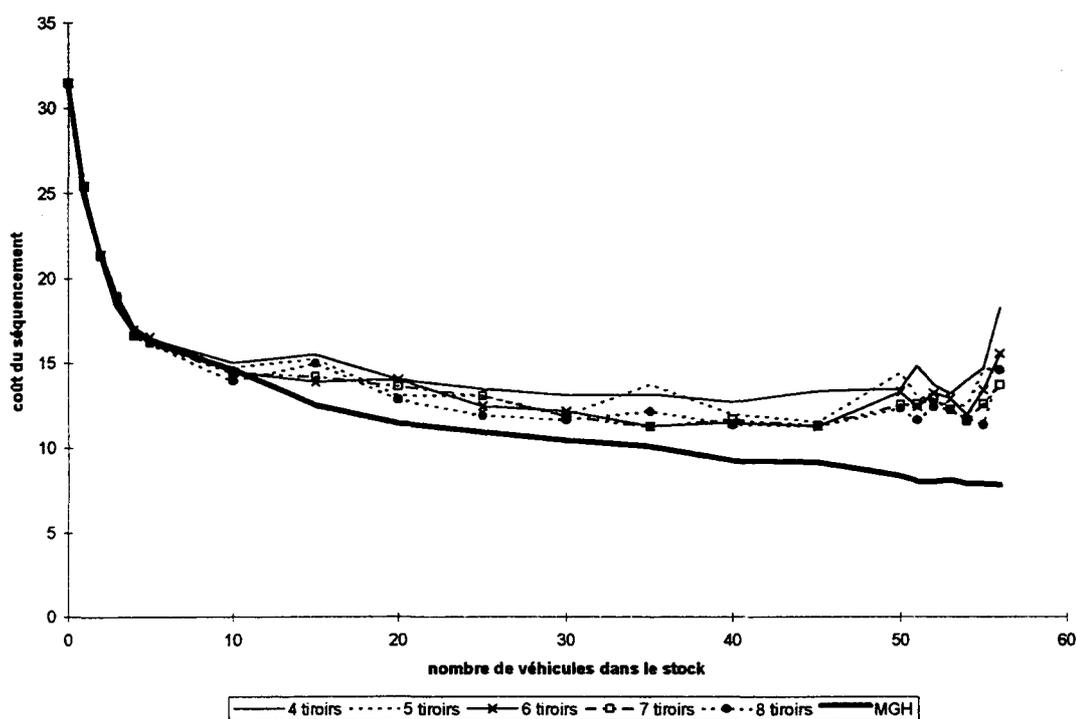


Figure IV.26 - Influence du taux de remplissage et de la topologie du stock.

Au niveau de la conception d'un stock, il peut être intéressant de comparer les performances moyennes de fonctionnement dans des conditions similaires. Les tests analysés à la Figure IV.26 avec toujours le même jeu de données (7 jours de production et 10 critères) donnent une idée de l'influence de la topologie du stock sur la qualité de la séquence générée par le cadencement.

L'autostockeur a un comportement semblable à celui des tiroirs quand ils sont vides (dans ce cas, les contraintes liées à la topologie du stock sont faibles). En revanche plus le stock est

rempli, plus la flexibilité de l'autostockeur se manifeste. Le grand choix de véhicules en sortie de stock permet de limiter de plus en plus le nombre de dépassements d'acceptables et de lisser de mieux en mieux les critères dans la séquence générée.

Le tableau de la Figure IV.27 montre la dégradation due à la topologie du stock de manière synthétique pour le jeu de données utilisé. L'autostockeur (ou MGH) est le type de stock le plus flexible. Son coût supérieur à celui d'un stock à tiroirs doit relativiser le gain effectif du coût de la séquence enlignée. La performance des tiroirs est proportionnelle au nombre de tiroirs comme on pouvait s'y attendre. Dans notre simulation, le coût de la séquence est proportionnel à  $N^{-0.67}$  où N est le nombre de tiroirs.

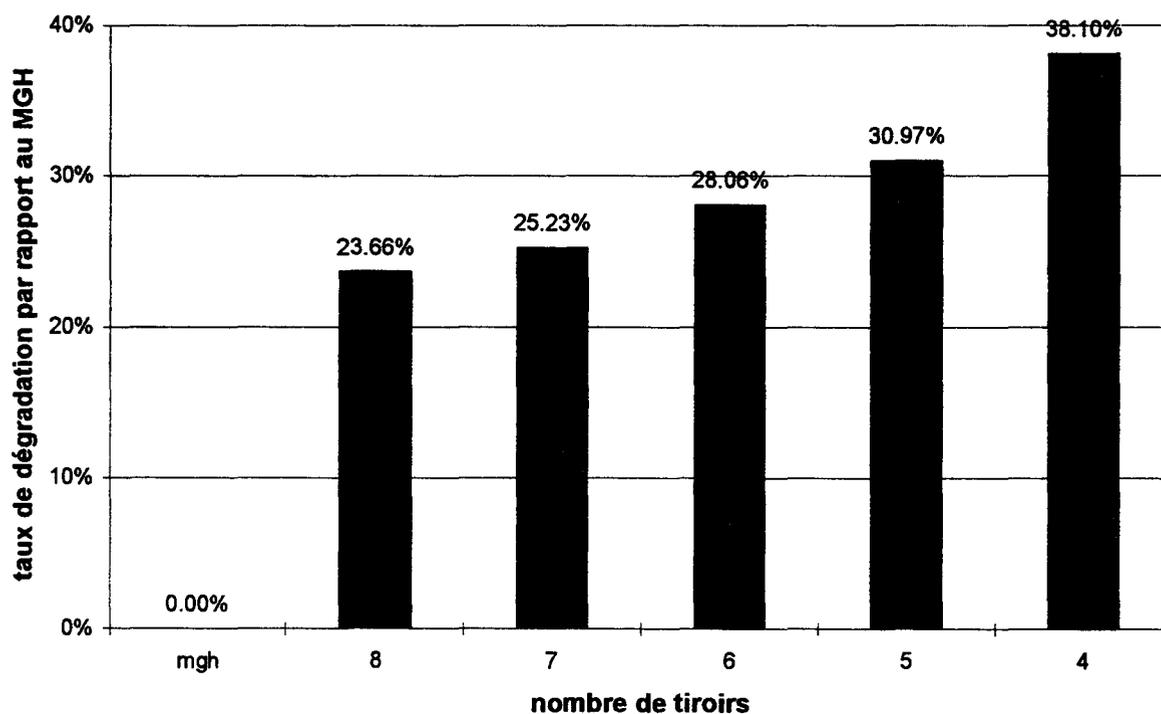


Figure IV.27 - Performance relative des tiroirs par rapport à un autostockeur

### 6.3. Conclusion

Nous avons décrit un algorithme capable de cadencer les véhicules entrant et sortant d'un stock intermédiaire de production. Nous avons proposé des règles qui permettent de choisir une stratégie pour gérer l'entrée du stock et de paramétrer l'arbre de décision et les stocks. Nous conseillons dans le cas d'un stock à tiroirs l'utilisation de la politique à enveloppe variable car elle est plus robuste.

## 7. Gestion amont : approche curative

Si on se reporte à la description des flux poussés et des flux tirés dans une usine (cf. partie I, chapitre 4), il existe deux approches concurrentielles pour gérer un stock. La première,

traditionnelle, consiste à réagir aux aléas de fabrication et à subir le flux de véhicules entrant dans le stock : c'est la gestion amont dont le rôle est "curatif". L'autre approche a une orientation prédictive : l'objectif est d'optimiser localement un flux de production dans un atelier sans perturber le flux global de l'usine.

Dans ce paragraphe, nous allons étudier les différentes stratégies que nous avons testées pour gérer l'approche curative de la gestion de stock.

Dans un premier temps nous allons décrire les algorithmes ne gérant qu'un seul type de critère : soit la qualité du flux (respecter des espacements acceptables ou des ratios et lissage des contraintes) soit le délai des véhicules (éviter les retards ou les diminuer). Les indicateurs a posteriori utilisés sont le nombre pondéré de dépassements de ratios acceptables et le retard maximum pour 95% des véhicules.

Dans un deuxième temps, nous allons essayer de combiner ces deux critères d'optimisation. La première approche va privilégier le respect des ratios en minimisant si possible le retard. A l'inverse, la seconde approche va privilégier la minimisation du retard en respectant au mieux les ratios.

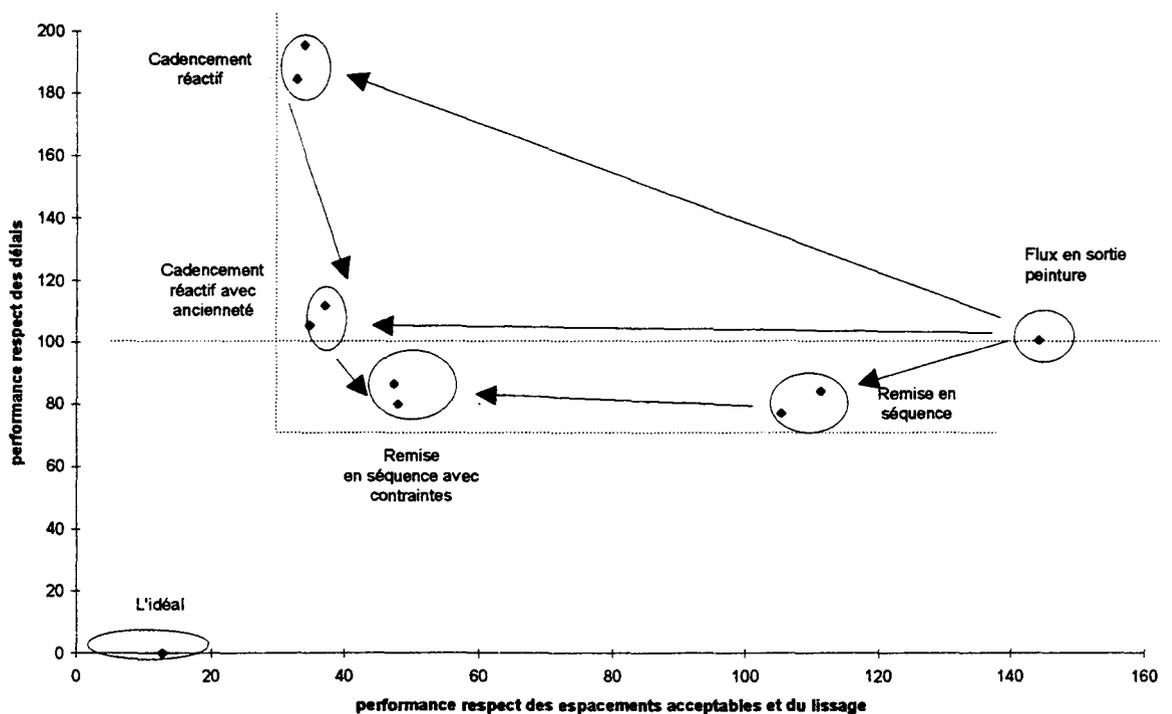


Figure IV.28 - Les différentes approches retenues et leurs niveaux de performance.

Sur la Figure IV.28, nous avons positionné les performances respectives de chacune des approches décrites par la suite. Nous avons pour cela utilisé le deuxième jeu de test qui est très contraint et très dense.

Comme référence, nous avons positionné l'idéal qui correspond à la liste de référence optimisée à partir du lot d'OF en statique grâce à la méthode combinée (cf. partie III.2.2), c'est la borne minimale qu'on ne pourra pas dépasser.

Le flux en sortie peinture est donné à titre d'information : c'est le flux perturbé qui entre dans le stock et qu'il faut cadencer.

Les deux approches monocritères donnent pour le cadencement réactif :

- une référence basse pour la qualité de la séquence vis à vis du respect des ratios et du lissage

- une limite basse pour le respect du délai pour la remise en séquence.

Les deux approches multicritères sont positionnées par rapport aux quatre références. Les flèches indiquent les tendances entre les méthodes. Par exemple entre le flux en sortie peinture et le cadencement réactif, nous avons une très nette amélioration pour le respect des ratios et le lissage au détriment d'une rétention de certains véhicules dans le stock (le délai est dégradé).

L'algorithme de base est celui décrit précédemment. Seule la métrique associée à l'algorithme change pour prendre en compte les contraintes spécifiques au problème.

### **7.1. Cadencement Réactif pur**

L'algorithme de cadencement réactif pur est l'algorithme représentant l'approche de base traditionnellement retenue pour optimiser un flux de production automobile. L'objectif de cet algorithme est de générer une séquence en sortie de stock qui respecte les contraintes de l'atelier suivant (au niveau des ratios acceptables et du lissage des critères) pour obtenir une charge de travail sur l'ensemble des postes de l'atelier compatible avec les contraintes matérielles et humaines.

Les différentes contraintes prises en compte sont le respect des ratios pour les différents critères retenus et le lissage de ces critères dans la mesure du possible. (Cf. 5.2 et 5.3)

Cette approche permet de définir les limites maximum qu'il est possible d'atteindre pour optimiser les contraintes physiques de la ligne.

### **7.2. Remise en séquence**

La deuxième approche décrite ici consiste, à partir d'un ordonnancement calculé en statique, à limiter au maximum le retard pris par les véhicules lors de leur fabrication. Le stock intermédiaire de production sert alors à garantir la date de livraison du véhicule, en accélérant ou en ralentissant les véhicules en fonction des aléas de production qui ont eu lieu dans les

ateliers amonts. Les contraintes étant respectées lors de la création d'un ordonnancement statique (cf. partie III), le respect de la date due des véhicules permet de respecter les contraintes fixées au préalable.

Cette approche permet d'apprécier la capacité à respecter les dates dues que le stock est capable d'atteindre.

Pour un flux donné, nous sommes capables alors de déterminer les capacités de réordonnement d'un stock : d'une part pour garantir le respect des contraintes physiques mais aussi pour garantir le respect de la date due des véhicules. A partir de ces deux données, nous sommes donc capables d'évaluer la qualité d'une optimisation multicritère prenant en compte ces deux aspects.

### 7.3. Cadencement Réactif avec ancienneté

Cette approche permet de prendre en compte à la fois une optimisation des contraintes liées aux critères des véhicules et le respect d'une date due du véhicule. Cependant cette méthode privilégie d'abord le respect des contraintes comme pour le cadencement réactif. La pénalisation liée à un véhicule s'exprime sous la forme :

$$\text{Penalisation} = \text{Note}(\text{Cadencement Reactif}) \times \exp \frac{\text{avance}}{k_b}$$

Le paramètre  $k_b$  est difficile à régler pour positionner de manière précise les deux critères d'optimisation entre eux. (voir 5.4).

En revanche les résultats sont très intéressants car ils permettent pour une très faible dégradation des contraintes de ratios de limiter le temps passé dans le stock.

Cette approche est adaptée quand le critère le plus important est la qualité du flux et quand le respect d'une date souhaitée sert à éviter que des véhicules restent bloqués dans le stock, car ils sont difficiles à enligner pour un algorithme comme le cadencement réactif.

### 7.4. Remise en séquence avec contraintes

L'approche de la remise en séquence avec contraintes est le pendant du cadencement réactif avec ancienneté. En effet le critère d'optimisation principal est le respect d'une date comme pour l'algorithme de remise en séquence.

Il est néanmoins très important de respecter un flux de bonne qualité (vis à vis des ratios) en sortie de stock. Pour atteindre ces deux objectifs, il faut être capable de positionner les deux critères les uns par rapport aux autres. Par exemple, en reprenant la notation du 5.6 :

- un véhicule très en retard est prioritaire par rapport à tous les véhicules en avance.
- un véhicule en retard est toujours prioritaire par rapport à un véhicule très en avance.

Il est important de connaître alors une borne supérieure (B) pour le respect des ratios; il est alors possible de reprendre le graphique de la Figure IV.26.

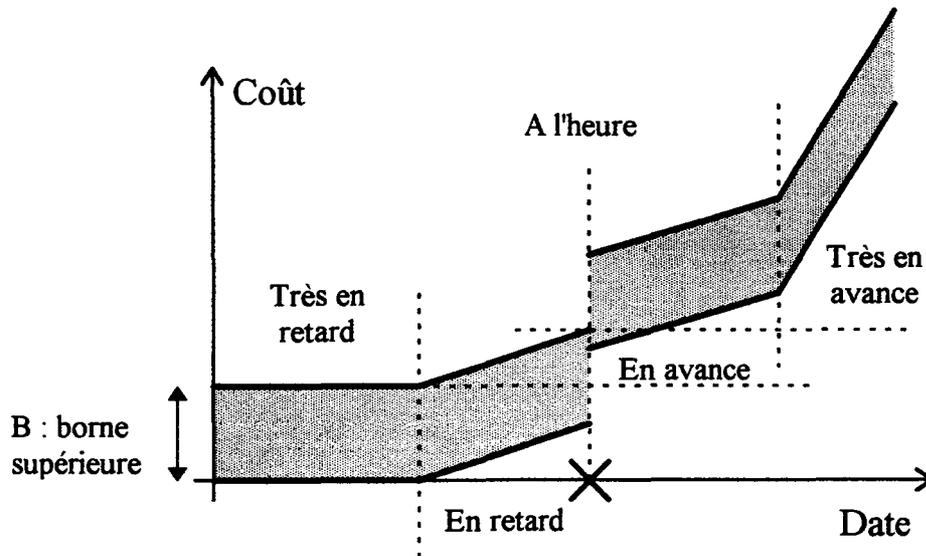


Figure IV.29 - Remise en séquence avec contraintes

En fonction de la date (ou de la position) souhaitée pour le véhicule, la pénalisation liée à un véhicule est comprise dans la zone grisée. Elle est proche de la limite inférieure si le coût lié aux dépassements de l'espacement acceptable est faible et au contraire si le véhicule est très mauvais vis-à-vis de ces critères, il se situera près de la limite supérieure.

La différence avec l'approche précédente apparaît sur l'ordre des importances prises entre le respect des contraintes et le respect des dates. Avec cette approche, on est capable d'utiliser des notions d'ordre strict entre les véhicules.

### 7.5. Résultats

Nous allons illustrer ces différentes approches à l'aide d'un jeu de test sur 8 journées de production réelles et 10 critères par véhicules pour lesquels on cherche à optimiser deux contraintes : les ratios et le lissage. Le flux est de nouveau très contraint. Il est important de diminuer le délai pris par les véhicules.

Sur le graphique ci-dessous, nous positionnons chacune des approches en fonction de l'écart type du délai (en nombre de places de retard ou d'avance) et en fonction d'une moyenne pondérée des dépassements d'acceptables. Les droites en pointillées représentent les deux seuils atteints pour le respect des ratios et la diminution du délai des véhicules. On s'aperçoit que le cadencement réactif avec ancienneté permet d'avoir un meilleur respect des contraintes

liées aux critères que la remise en séquence avec contrainte. En revanche cette dernière permet, pour une très faible dégradation du délai, des respects des ratios corrects sans être excellents.

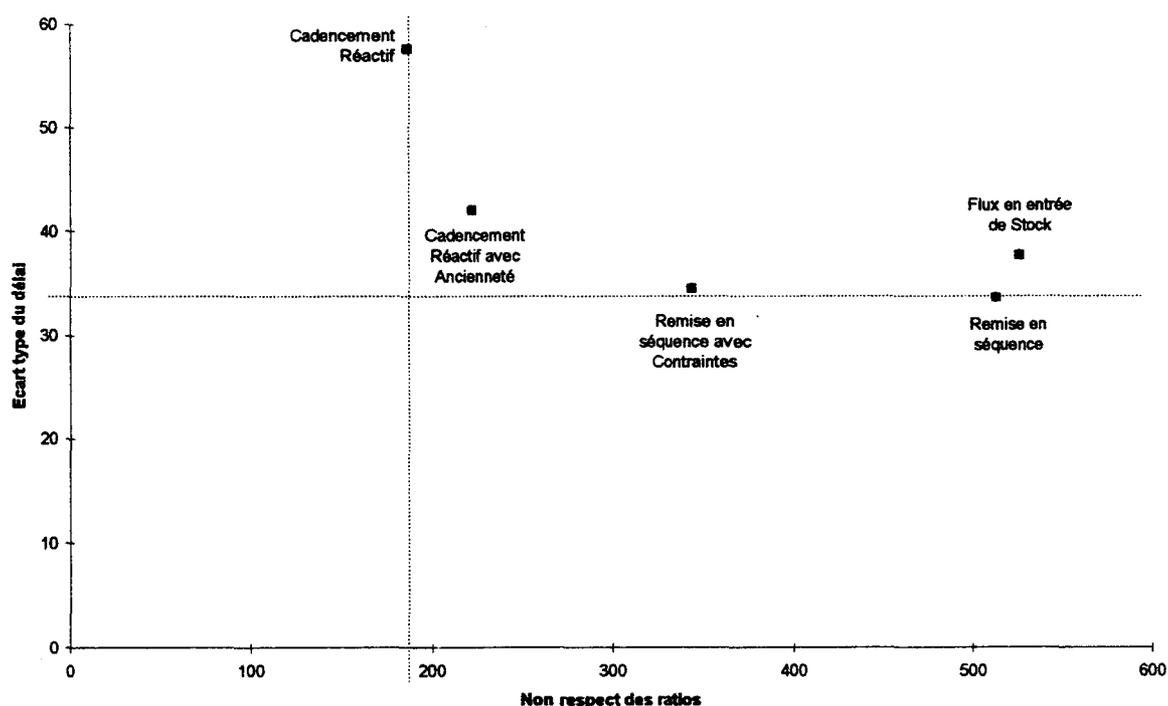


Figure IV.30 - Jeu de test pour les quatre approches

Le tableau de la Figure IV.31 illustre les dégradations pour chaque critère d'optimisation (le délai et le respect des ratios) en fonction de l'approche choisie par rapport à la meilleure approche pour le jeu de test précédent. Il confirme les remarques précédentes.

Le cadencement réactif avec ancienneté limite la dégradation du délai pour une faible perte au niveau des non respects de ratios. A l'inverse, la remise en séquence avec contraintes permet d'obtenir presque la même qualité pour le délai (que la remise en séquence). En revanche, elle permet d'obtenir une qualité de respect des ratios deux fois meilleure mais néanmoins moins bonne que le cadencement réactif avec ancienneté.

	entrée du stock	cadencement réactif	remise en séquence	cadencement réactif avec ancienneté	remise en séquence avec contraintes
ratios	+ 182%	+ 0%	+ 175%	+ 19%	+ 85%
lissage	+ 31%	+ 0%	+ 42%	+ 17%	+ 25%
délai	+ 12%	+ 71%	+ 0%	+ 25%	+ 3%

Figure IV.31 - Tableau comparatif des approches en fonction des critères

## 8. Gestion avale prédictive

### 8.1. Introduction

L'approche prédictive consiste à séquencer le flux de production localement dans un atelier sans perturber le flux global de l'usine.

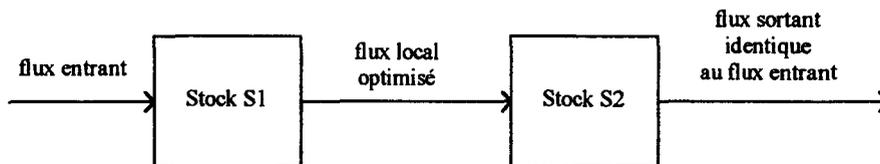
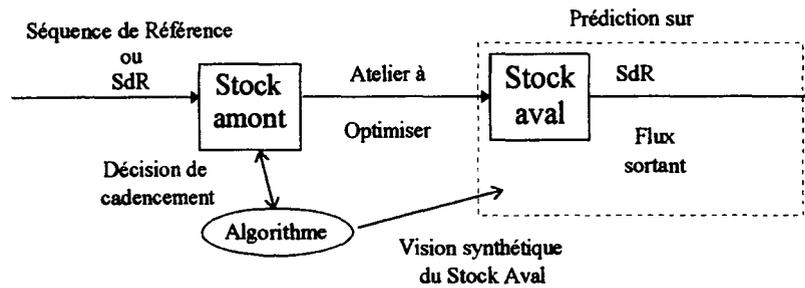


Figure IV.32 - Optimisation locale du flux.

Le problème consiste donc à générer une séquence localement dans un atelier de production, considéré comme une FIFO et situé entre deux stocks S1 et S2. Cette séquence de véhicule doit garantir deux types de contraintes. La plus forte des contraintes doit absolument être respectée. Elle consiste à retrouver en sortie du stock S2 la séquence qui est entrée dans le stock S1. Cette contrainte correspond au caractère local des modifications apportées au flux entre les stocks S1 et S2. Le deuxième type de contraintes consiste à optimiser le flux dans l'atelier. Cette optimisation du flux utilise les mêmes règles que celles définies dans les chapitres précédents pour l'optimisation "curative", mais elle est limitée par la capacité des deux stocks à remettre le flux en séquence.

Le problème supplémentaire qui apparaît pour cet algorithme consiste à déterminer la capacité de remise en séquence d'un atelier. En pratique, il faut évaluer la capacité du couple stock S1, stock S2 à modifier l'ordre des véhicules sur le tronçon compris entre eux deux et uniquement entre eux deux. Le but est donc de trouver un algorithme permettant de s'assurer que le cadencement en sortie de stock S1 est compatible avec une remise en séquence possible en sortie du stock S2, quelles que soient les optimisations que l'on souhaite faire entre les deux stocks pour satisfaire les contraintes de l'atelier.

L'intérêt d'une telle approche existe dans le cas où deux ateliers ont des contraintes à respecter très différentes, voire incompatibles. L'optimisation prévisionnelle effectuée pour une journée privilégie les contraintes de l'atelier dont le fonctionnement est le plus intéressant à améliorer. En pratique, c'est l'atelier montage. Mais nous avons vu qu'il est également très intéressant d'optimiser d'autres ateliers comme la peinture à l'aide de rafales de teintes. Cette optimisation n'est pas toujours possible au niveau prévisionnelle car des contraintes peuvent être incompatibles. En revanche il peut être bénéfique d'utiliser les flexibilités de l'usine pour optimiser dans chaque atelier les contraintes de cet atelier.



Cette méthode s'appelle le cadencement reséquençable.

## 8.2. Cadencement reséquençable

La flexibilité de l'atelier dépend fortement du stock aval, c'est-à-dire S2. Elle permet en effet de reprendre les modifications faites en amont. Il est également évident que le stock aval doit être au moins aussi flexible que le stock amont. Dans le cas contraire, la flexibilité du stock amont ne peut pas être utilisée à son maximum car les modifications apportées à la liste ne pourront pas être corrigées dans le stock aval.

Dans le cas, où le stock S2 est un MGH, dont tous les véhicules sont accessibles dans le temps cycle, nous allons montrer qu'il suffit d'une seule condition pour garantir la remise en séquence du flux.

Soit  $N$ , le nombre de véhicules dans le stock S2 (la valeur est inférieure à la capacité du stock).

Pour que la séquence soit remise en ordre à chaque pas, il faut et il suffit (Condition Nécessaire et Suffisante) que le véhicule à enligner soit dans le stock S2. Donc à chaque pas il faut s'assurer que le véhicule que l'on enligne puisse arriver dans les délais (en entrée de S2). Pour cela, la date de lancement (en sortie de S1) de tous les véhicules ne doit pas excéder de  $N$  pas celle prévue dans la liste de référence. Comme on a une bijection entre la référence du véhicule et sa date de lancement, on peut inverser cette proposition en : il faut qu'à une date donnée (pas  $P$ ) tous les véhicules devant être enlignés avant le pas  $(P - N)$  soient déjà sortis du stock S1. Comme la vérification se fait à chaque pas de l'algorithme, il suffit de vérifier que pour l'enlignement du pas  $P$ , le véhicule devant passer au pas  $(P - N)$  a déjà été enligné. En effet pour le véhicule du pas  $P - N - 1$ , la vérification a été faite au pas  $P - 1$ , et ainsi de suite.

**D'où la CNS pour s'assurer de la remise en séquence possible dans S2 :**

**Pour l'enlignement au pas  $P$ , il faut s'assurer que le véhicule prévu au pas  $(P - N)$  de la liste de référence a déjà été enligné ou va l'être au pas  $P$ .**

Dans le cas où le stock S2 est un stock à tiroirs, le problème est plus compliqué. En effet, il y a toujours le problème de délivrer le véhicule à la bonne date (celle de sortie de S2), et le problème de la configuration du stock. En effet dans cette configuration, il existe des contraintes d'ordre entre les véhicules qui ne peuvent pas se dépasser dans les tiroirs. De plus les tiroirs ont une contenance maximale qui ne peut pas être dépassée.

Dans un premier temps, nous allons faire des hypothèses. Si le stock S1 est plus flexible que le stock S2 (c'est contraire à la logique), il faut rajouter des contraintes pour que la séquence en sortie ne soit pas trop modifiée. Si le stock S2 est plus flexible que S1, on n'a pas de problème, mais cela ne permet pas de rattraper plus de perturbations. En revanche la capacité de stockage d'un tiroir pourra être artificiellement augmentée en utilisant les tiroirs non utilisés. Pour ces raisons, nous allons nous attacher au cas où les deux stocks S1 et S2 sont identiques : ils ont le même nombre de tiroirs avec les mêmes capacités.

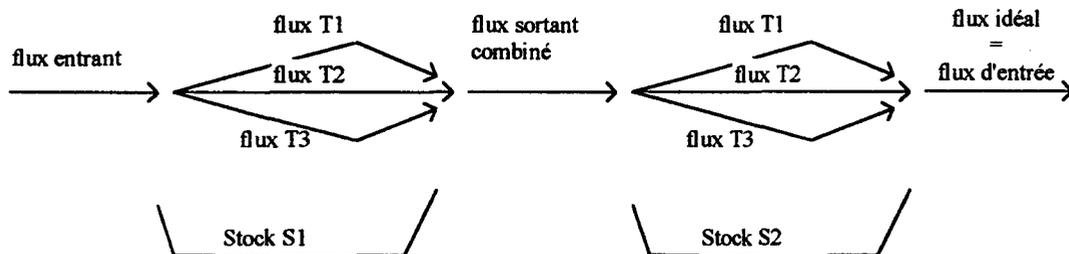


Figure IV.33 - Fonctionnement des stocks à tiroirs pour le cadencement reséquenceable.

Dans le cas des autostockeurs, la configuration du stock S2 n'influence que très peu la remise en séquence. En revanche, dans le cas de stocks à tiroirs, la configuration est très contraignante et la condition nécessaire décrite précédemment n'est plus suffisante car il faut prendre en compte la capacité de chaque tiroir.

L'idée de base provient du fait qu'un flux passant à travers T tiroirs est décomposé en T flux distincts qui sont recombinaés de manière plus ou moins aléatoire en sortie du stock pour générer la séquence traversant l'atelier intermédiaire (cf. Figure IV.33). Une contrainte suffisante pour remettre en séquence le flux intermédiaire est de redécouper ce flux en T flux identiques à ceux du premier stock (S1) et de recombinaer ces T flux de façon à obtenir le flux de référence.

C'est une Condition Suffisante car il faut gérer le problème du stockage dans les tiroirs. En effet, pour permettre de conserver le flux  $F_i$ , il faut s'assurer que le Tiroir  $i$  du stock S2 ne sera pas saturé.

**D'où la CN pour s'assurer de la remise en séquence possible dans S2 :**

**Pour l'enlèvement au pas P, il faut s'assurer que le véhicule prévu au pas (P-N) de la liste de référence a déjà été enligné ou va l'être au pas P.**

**D'où une CS pour s'assurer de la remise en séquence possible dans S2 :  
recréer les sous-flux du stock S1 dans le stock S2**

Cette approche est restrictive car nous ne proposons pas une condition nécessaire et suffisante. En revanche elle est facile à implémenter et n'augmente pas de manière significative la complexité de l'algorithme. Il suffit en effet de simuler le fonctionnement du stock (S2) pour s'assurer que la séquence, que l'on propose en sortie de S1, peut entrer dans le stock (S2).

A priori, les seuls cas non traités, quand on utilise cette condition nécessaire, sont ceux où il existe des phénomènes de saturation ou de famine dans les tiroirs. Dans ces cas, des tiroirs pourront être couplés pour augmenter la capacité.

Supposons que la séquence en entrée de S1 et souhaitée en sortie de S2 soit :

11-10-9-8-7-6-5-4-3-2-1



La séquence optimisée dans l'atelier entre les stocks est :

1-11-10-9-8-7-6-5-4-3-2



Pour le stock S1, tous les véhicules entrent dans le deuxième tiroir sauf le 1 qui entre dans le premier. Si les tiroirs ont une capacité de cinq véhicules, la séquence initiale ne pourra pas être recrée à l'aide du deuxième stock (S2) sauf si on utilise un troisième tiroir qui n'était pas utilisé dans le premier stock (S1).

Illustrons cet exemple :

1
11-10-9-8-7-6-5-4-3-2

1-11-10-9-8-7-6-5-4-3-2

1
6-5-4-3-2
11-10-9-8-7

En revanche, il est utile d'indiquer que nous aurions pu atteindre le même résultat en équilibrant dans le premier stock (S1) la charge dans les deuxième et troisième tiroirs, soit en créant des sous-séries :

1
6-5-4-3-2
11-10-9-8-7

1-11-10-9-8-7-6-5-4-3-2

1
6-5-4-3-2
11-10-9-8-7

soit en créant des alternances entre sous-séries :

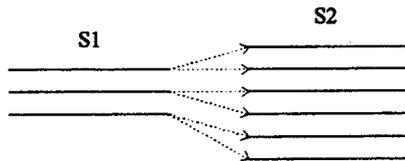
1
10-8-6-4-2
11-9-7-5-3

1-11-10-9-8-7-6-5-4-3-2

1
10-8-6-4-2
11-9-7-5-3

Cette approche permet d'obtenir les mêmes résultats.

Il reste alors à gérer les difficultés recensées au début de ce paragraphe. Dans le cas où le stock (S2) est plus flexible que le stock (S1), il suffit d'attribuer le ou les tiroirs supplémentaires de manière dynamique ou de manière statique pour augmenter la taille des tiroirs. Par exemple, si le stock (S1) a 3 tiroirs et (S2) en a 6, on peut attribuer statiquement deux tiroirs du stock (S2) pour un de (S1) :



Dans cet exemple cela permet d'assigner deux tiroirs du deuxième stock pour chaque tiroir du premier.

Si le stock (S1) est un autostockeur et (S2) est un stock à tiroirs, il faut s'assurer que le nombre des sous listes créées et leur taille sont compatibles avec le stock (S2) : l'autostockeur doit fonctionner comme un stock à tiroirs semblable à (S2).

Le problème des stocks à tiroirs revient donc à gérer la capacité de contenance de chacun des tiroirs.

### 8.3. Résultats et modes de fonctionnement

Dans le cadre d'un fonctionnement sans aléas, l'atelier se comporte comme une file d'attente. Les deux algorithmes pour les deux stocks (S1) et (S2) peuvent être découplés. L'architecture du système peut être décrite comme suit :

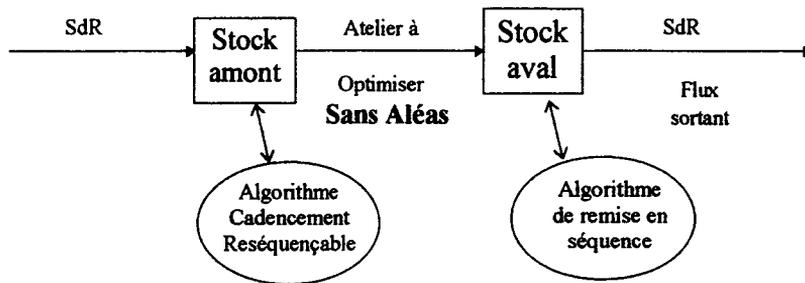


Figure IV.34 - Fonctionnement sans aléas

Illustrons les résultats sur un exemple :

Les deux stocks S1 et S2 sont identiques : ils ont 4 tiroirs de 10 places chacun.

Les stocks sont initialement remplis de 30 véhicules. L'objectif de l'atelier intermédiaire est d'optimiser la taille moyenne des rafales qui le traversent en sachant que en entrée la taille des rafales est de un.

Pour évaluer les performances du cadencement reséquenceable, nous allons comparer ses performances avec celles d'un algorithme optimisant uniquement la taille des rafales.

	Rafales	Cadencement reséquenceable
nombre de rafales	145	166
taille moyenne des rafales	6.4225	5.6125
écart type de la taille des rafales dans l'atelier	2.7275	1.64
délai minimum	-46	-22
délai moyen	0	0
délai maximum	29	24
écart type du délai	12.17	7
après le stock S2		
délai minimum	-42	0
délai moyen	0	0
délai maximum	18	0
écart type du délai	8.9	0

Figure IV.35 - Résultats cadencement reséquenceable

Le cadencement reséquenceable permet de garantir la séquence obtenue en sortie du stock S2. En contrepartie la longueur moyenne des rafales générées pour l'atelier intermédiaire est légèrement plus faible que dans le cas où l'on ne se préoccupe pas de la remise en séquence : 5.6 contre 6.4. La dégradation de 15% permet d'éviter un écart type de près de 10 places en sortie du stock (S2) : cet écart correspond à 1/3 de la capacité des stocks.

En mode de fonctionnement nominal, il est nécessaire de prendre en compte les aléas de fabrication de l'atelier intermédiaire (cf. Figure IV.36). Le flux arrivant en entrée du stock S2 (stock aval) n'est pas nécessairement le même que celui sortant de (S1). L'algorithme gérant le flux en sortie du stock aval doit donc prendre en compte ces perturbations et optimiser la séquence de sortie pour l'atelier aval. Il faut donc utiliser un algorithme comme décrit dans le chapitre IV.5 (cadencement avec ancienneté ou reséquenceur intermédiaire). L'objectif du deuxième stock étant de générer dynamiquement une séquence respectant au mieux les contraintes de l'atelier aval tout en essayant de recréer la séquence de référence SdR.

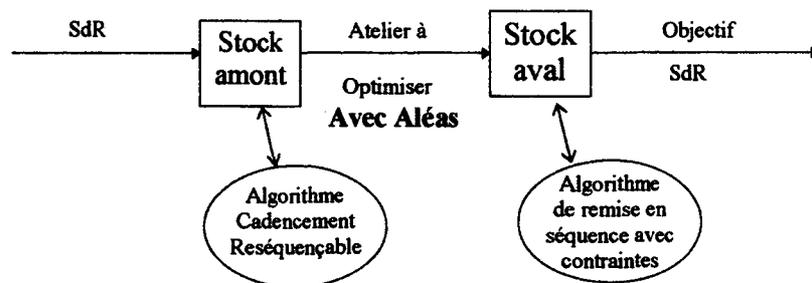


Figure IV.36 - Fonctionnement avec aléas

## 9. Conclusion

Dans cette partie nous avons présenté une méthode adaptée à l'aspect dynamique du cadencement dans des stocks intermédiaires de production. Elle est constituée à partir d'un parcours d'arbre en profondeur d'abord. L'aspect déterministe de cette méthode et sa réactivité vis-à-vis des changements lui permettent d'être adaptés à tous les styles de stocks.

Cette méthode est évolutive et adaptable. D'une part, nous sommes capables de prendre en compte les différents styles de stocks : autostockeurs, stocks mixtes et en particulier stocks à tiroirs. D'autre part cette méthode s'adapte à différentes stratégies de gestion du flux. La métrique est le coeur de celles-ci.

L'algorithme mis au point est robuste. En effet si le roulement de stock change le ratio du stock, les performances globales de son cadencement ne sont que faiblement altérées. De même si des aléas de production apparaissent, l'algorithme est capable d'en atténuer les perturbations sur le flux dans une certaine mesure (qui dépend de la configuration physique du stock).

La mise en place de cet algorithme est relativement facile car il y a peu de paramètres à régler. En revanche, il est prudent d'effectuer des simulations sur le comportement du flux afin de vérifier si la difficulté à ordonnancer le flux n'est pas trop grande : contraintes liées aux ratios (indices  $I_r$ ), nombre de contraintes à prendre en compte, croisements de ces contraintes les unes avec les autres. Deux implémentations ont été effectuées avec deux types de stratégies pour des stocks à tiroirs et utilisées dans les usines du groupe PSA.

Nous avons ensuite présenté différentes stratégies de gestion des stocks. Dans un premier temps, nous avons abordé l'approche curative comme politique de gestion. Elle consiste à optimiser localement la séquence de véhicules suivant une métrique donnée pour enligner un véhicule dans l'atelier aval. Cette succession d'optimisations locales permet d'obtenir une séquence globale sur une journée de production respectant les contraintes qui ont été fixées.

Dans un deuxième temps, nous avons proposé une approche prédictive qui permet d'optimiser localement un atelier situé entre deux stocks sans modifier la séquence amont en aval du deuxième stock. Elle s'apparente au flux hybride de Mertins [MER 96] ou de Chan [CHA 96] : on a une approche descendante du flux. Le flux est poussé d'atelier en atelier mais ce sont les requêtes de l'atelier aval qui sont prises en compte pour être optimisées.

*CONCLUSION*



## CONCLUSION

Nous avons présenté dans cette thèse des algorithmes de séquençement et de cadencement pour des lignes d'assemblages à modèles mélangés et plus particulièrement pour les usines automobiles. Ces algorithmes d'ordonnancement doivent être capables de prendre en compte différents types d'objectifs :

- respect de ratios ou respect d'espacement entre véhicules ayant un critère.
- lissage de la charge de travail pour chaque ressource.
- respect de convois de succession (rafales).
- respect de date ou de position souhaitées.

Le premier objectif utilise la notion de ratio maximum à ne pas dépasser (pas plus de  $P$  véhicules sur  $N$  doivent avoir le critère) ou d'espacement minimum à garantir entre deux véhicules ayant le critère (**espacement acceptable**). Le deuxième objectif peut être modélisé de deux façons différentes. La première approche consiste à modéliser de façon explicite la charge de chaque poste de travail [COM 96] en identifiant les tâches à effectuer sur le véhicule. La deuxième approche retenue consiste à lisser le taux de consommation de chaque critère du véhicule. L'objectif est alors de maintenir constant l'espacement entre deux véhicules ayant un critère donné (**espacement théorique**). Le troisième objectif peut être modélisé de deux façons différentes : minimiser le nombre de changement d'outils ou optimiser la taille des rafales en garantissant une longueur moyenne. Nous avons retenu la dernière approche. Le quatrième objectif consiste à garantir la présence d'un véhicule à une date ou à une position donnée dans la séquence. La pénalisation liée au véhicule dépend de l'avance ou du retard de celui-ci par rapport à la référence choisie.

Un premier algorithme a été proposé pour résoudre le problème statique : le séquençement prévisionnel des ordres de fabrication sur la ligne (en général une fois par jour). Cet algorithme est constitué de trois étapes. La première consiste à faire une partition des ordres de fabrication en un ensemble de groupes dont les caractéristiques sont représentatives de l'ensemble des OF. Les regroupements doivent être fait à ce niveau, les volumes moyens de critères pour le lissage ou le respect des ratios doivent être garantis au mieux et les OF ayant des dates souhaitées doivent également être assignés aux bons groupes. Une fois cette partition faite, une recherche par faisceaux dans chaque groupe permet d'obtenir une première solution de bonne qualité qui sert de solution initiale pour un recuit simulé (troisième étape). La méthode développée permet en limitant le temps de calcul d'obtenir une solution de très bonne qualité (par rapport aux contraintes fixées). Elle est également robuste vis-à-vis de la difficulté à respecter des

contraintes croisées qui rendent l'espace de recherche très chaotique. A travers des simulations, nous montrons les gains possibles par rapport à des méthodes plus traditionnelles comme le Goal Chasing ou le Recuit Simulé.

La deuxième approche étudiée dans cette thèse est le cadencement. C'est l'ordonnancement dynamique des véhicules dans les stocks intermédiaires de production. L'algorithme doit respecter en plus des contraintes gérées pour le séquençement des contraintes liées au caractère dynamique du problème :

- topologie du stock
- composition du stock
- véhicules déjà enlignés
- temps de réponse compatible avec le temps cycle de la ligne

La méthode retenue utilise une approche par construction progressive de séquence en profondeur d'abord pour limiter l'espace mémoire nécessaire. Cette méthode, déjà utilisée pour les architectures à autostockeurs [GUE 95] et [DEL 97], est adaptée dans ce mémoire pour les architectures à tiroirs. Pour les architectures à tiroirs, la méthode utilise une politique d'enveloppe à taille variable et une généralisation du parcours d'arbre pour affecter les véhicules dans les tiroirs. Testé sur des jeux d'essais réels, l'algorithme permet, en garantissant le temps de réponse, de prendre en compte plus de contraintes et de critères que l'homme n'est capable d'appréhender.

Cette méthode peut être appliquée à différentes stratégies de gestion de stock. Pour des approches traditionnelles, nous avons testé des stratégies d'optimisation monocritères (respect des ratios ou respect des dates). Puis nous avons utilisé la méthode pour des optimisations multicritères prenant en compte à la fois le respect des ratios et le respect des dates avec un minimum de dégradation des performances.

Nous avons appliqué cette méthode à une approche prédictive. Elle consiste à optimiser localement un atelier dont les contraintes ne sont pas compatibles avec les contraintes des autres ateliers sans perturber le flux dans son ensemble. Le stock aval est alors capable de générer une séquence identique à celle qui est entrée dans le stock amont. Cette approche peut être utilisée typiquement pour un atelier comme la peinture et permet de garantir une remise en séquence avec une faible dégradation des résultats pour les contraintes.

Une partie des travaux développés dans cette thèse a vu son application dans un cadre industriel. Dans un premier temps, un prototype, en bord de ligne, a permis de valider le concept de l'algorithme de construction progressive de séquence avec la métrique de cadencement réactif. Puis, dans un deuxième temps, ces résultats et des études

complémentaires ont permis de mettre au point la méthode d'optimisation retenue dans le cadre d'une application industrielle sur le site.

Les travaux présentés dans cette thèse offrent différentes perspectives de recherche. La première consiste à définir trois séquençements prévisionnels, un pour chacun des trois ateliers, et de garantir que les séquences ainsi définies pourront être générées en traversant les stocks intermédiaires de production. L'optimisation étant ciblée sur un seul atelier, le nombre de contraintes à garantir est plus faible. Les résultats devraient donc être plus prometteurs. Cette première perspective concerne le **séquençement**.

La deuxième orientation concerne les problèmes de **cadencement**. L'idée est de prendre en compte les perturbations des ateliers pour obtenir un cadencement reséquençable de meilleure qualité. Cette approche consisterait à identifier les critères qui posent chroniquement un problème et à s'arranger pour modifier l'ordonnancement.

Le troisième axe de recherche serait commun aux deux approches : séquençement et cadencement. Les études pourraient porter sur les métriques. En effet le lissage est abordé à un niveau macroscopique. Une étude serait nécessaire pour évaluer l'apport d'une modélisation plus fine de la charge de travail : quel gain peut on escompter et une telle approche est-elle compatible avec le caractère réactif du cadencement? En parallèle, il pourrait être intéressant de développer des indicateurs pour évaluer le croisement entre les critères : ce point est important pour déterminer la configuration chaotique de l'espace de recherche.



**REFERENCES BIBLIOGRAPHIQUES**



## REFERENCES BIBLIOGRAPHIQUES

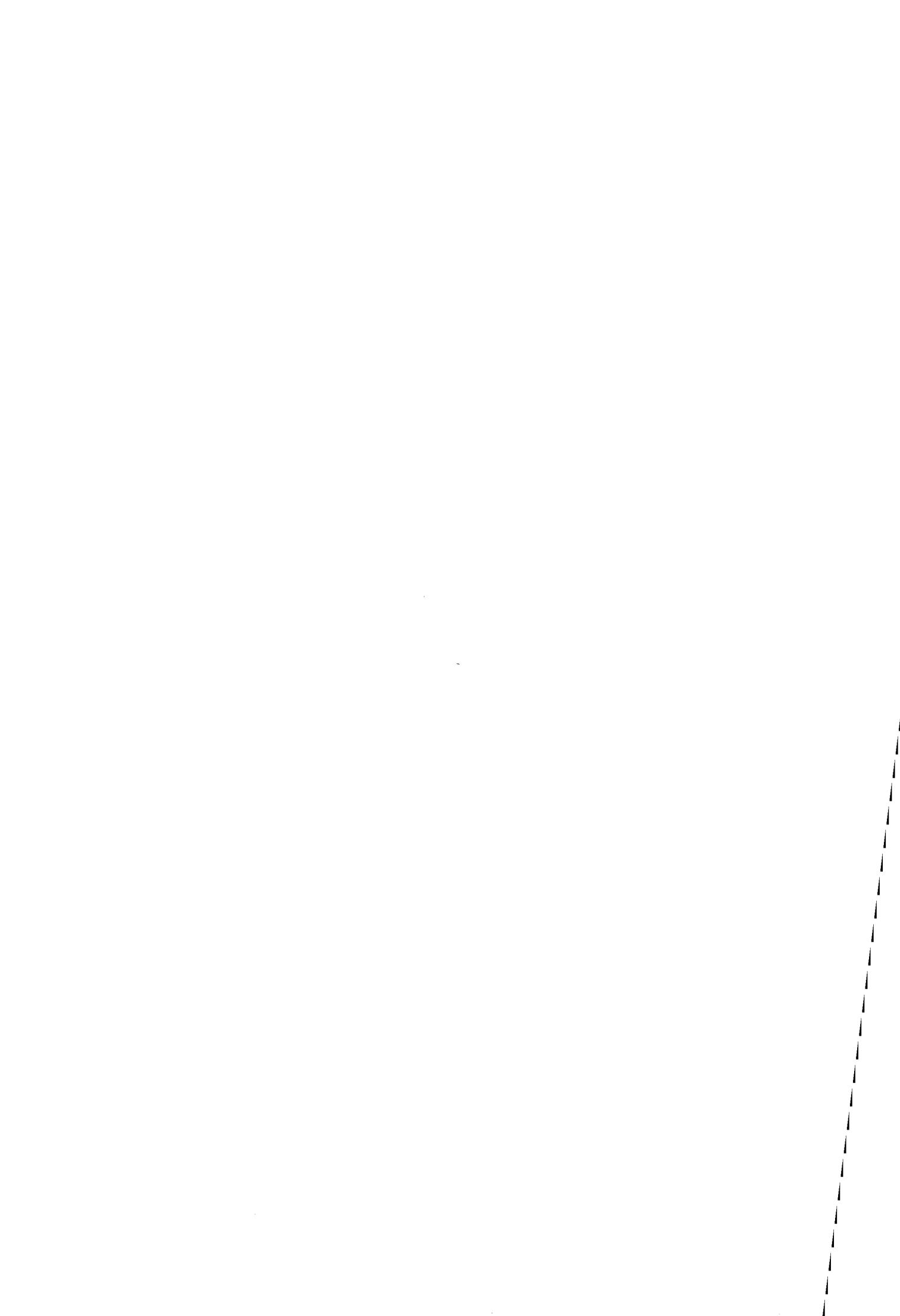
- [ARC 66] ARCUS A.L., "Comsoal a Computer Method of Sequencing Operations for Assembly Lines", *Int. J. Prod. Res.*, vol 4, n°4, p.259-277, 1966.
- [BAR 97] BARATOU P., CASTELAIN E., BONNEAU P., "Gestion Réactive de Stocks à Tiroirs dans le Domaine Automobile, Validation des Algorithmes par Simulation", *MOSIM'97, AFCET/Francosim/SCS, Rouen, p.333-340, juin 1997*.
- [BAR 98a] BARATOU P., CASTELAIN E., BONNEAU P., "Reactive Stock Management by Means of FIFO Stocks", *CESA'98 IMACS, Nabeul-Hammamet, avril 1998 (à paraître)*.
- [BAR 98b] BARATOU P., CASTELAIN E., BONNEAU P., "Quick Scheduling in Car Sequencing Problem", *CESA'98 IMACS, Nabeul-Hammamet, avril 1998 (à paraître)*.
- [BAR 92] BARD J.F., DAR-EL E., SHTUB A., "An Analytic Framework for Sequencing Mixed Model Assembly Lines", *Int. J. Prod. Res.*, vol 30, n°1, p.35-48, 1992.
- [BAR 94] BARD J.F., SHTUB E.S., JOSHIS S.B., "Sequencing Mixed Model Assembly Lines to Level Parts Usage and Minimize Line Length", *Int. J. Prod. Res, Vol.32, No 10, 1994*.
- [BAR 93] BARRANCO M.A.S., "Etude du Car Sequencing Problem", *Mémoire d'Ingénieur du Centro politecnico Superior, Universidad de Zaragoza, 21 juin 1993*.
- [BEL 96] BEL G., BENSANA E. "Intelligence Artificielle pour la Conception et la Conduite de Systèmes de Production en Mécanique", *Revue Française de Mécanique, 1996-4*.
- [BHA 93] BHATTACHARYA S.K., ROY R., LOW M.J., "A Computer Simulation System for the Evaluation of Man Assignments on Car Assembly Tracks", *Simulation 61:2, p;124-133, August 1993*.
- [BOL 94] BOLAT A., SAVSAR M., AL-FAWZAN M.A., "Algorithms for Real-Time Scheduling of Jobs on Mixed Model Assembly Lines", *Computers Ops Res., Vol 21, n°5, p.487-498, 1994*.
- [BON 89] BONOMI E., LUTTON J.-L., "Le Recuit Simulé", *Bibliothèque pour la science "le calcul intensif", Paris 1989*.
- [BUR 87] BURNS L.D., DAGANZO C.F., "Assembly Line Job Sequencing Principles", *Int. J. Prod. Res.*, vol 25, n°1, p.71-99, 1987.
- [CAL 95] CALATAYUD S., "Conception et Mise en Oeuvre d'une Modélisation Générique de Stock pour les Besoins de Cadencement Réactif du Groupe PSA Peugeot-Citroën", *Mémoire de DESS IA, Université Pierre et Marie Curie - Paris VI, septembre 1995*.
- [CAM 97] CAMUS H., "Conduite de Systèmes Flexibles de Production Manufacturière par Composition de Régimes Permanents Cycliques : Modélisation et Evaluation de Performances à l'Aide des Réseaux de Petri", *Thèse de Doctorat, Université de Lille I, 1997*.
- [CAS 94] CASTELAIN E. et al "Ordonnancement Réactif : Génération de la Séquence de Véhicules en Entrée de Ligne de Montage Automobile", *Groupement de*

- Recherche Automatique, Ordonnancement et Entreprise, Toulouse, p.305-315, juin 1994.*
- [CAS 95] CASTELAIN E., DELAVAL M., OHL H., GENTINA J.-C., RIAT J.-C., YVARS P.A., "Algorithmes de cadencement réactif des véhicules en entrée de ligne de montage", *RAPA, Vol.8, n°5, 1995.*
- [CAU 94] CAUX C., PIERREVAL H., PORTMANN M.C., "Les Algorithmes Génétiques et leur Application aux Problèmes d'Ordonnancement", *Groupement de Recherche Automatique, Ordonnancement et Entreprise, Toulouse, p.5-45, juin 1994.*
- [CHA 96] CHAN F.T.S., "Exploring the Scheduling Problems in a Flow Shop Environment via Simulation", *Int. J. of materials and Product Technology, Vol. 11. Nos1/2, 1996.*
- [CHE 91] CHEW T.L., DAVID J.M., NGUYEN A., TOURBIER Y., "Solving Constraint Satisfaction Problems with Simulated Annealing : The Car Sequencing Problem Revisited", *Twelfth International Conference. Artificial Intelligence, Expert systems, natural language, p 405-416, 1992.*
- [COM 96] COMBY G., "Aide au Séquencement des Produits sur une Ligne de Fabrication Multi-Modèles", *Thèse de Doctorat, INSA Lyon, 1996.*
- [DAR 75] DAR-EL E.M., Cothier R.F., "Assembly Line Sequencing for Model Mix", *Int. J. Prod. Res., vol 13, n°5, p.463-477, 1975.*
- [DAW 95] DAWN T., "Search Strategies Can Unearth Solutions to Complex Problems", *Scientific Computing World, 1995.*
- [DEL 95] DELAVAL M., CASTELAIN E., GENTINA J.C., "Dynamic Resequencing of Cars on Assembly Lines Using Simulated Annealing". *28th ISATA p.169-175, 1995.*
- [DEL 96] DELAVAL E., LEFORT A. "Car Sequencing Problem Two Approaches: Hypernets and Simulated Annealing", *CIMAT 96, Grenoble, mai 1996.*
- [DEL 97] DELAVAL M., "Séquencement des Lignes d'Assemblage à Modèles Mélangés", *Thèse de Doctorat, Université de Lille I, 1997.*
- [DIN 88] DINCBAS M., SIMONIS H., VAN HENTENRYCK P., "Solving the Car-Sequencing Problem in Constraint Logic Programming", *ECRC Munich, European Conference on AI, 1988.*
- [ELL 92] ELLEGARD K., JONSSON D., ENGSTRÖM T., JOHANSSON M.I., MEDBO L., JOHANSSON B., "Reflective Production in the Final Assembly of Motor Vehicles - An Emerging Swedish Challenge", *International Journal of Operations & Production Management, vol 12, n°7, p.117-133, 1992.*
- [GHO 89] GHOSH S., GAGNON R.J., "A Comprehensive Literature Review and Analysis of the Design, Balancing and Scheduling of Assembly Systems", *Int. J. Prod. Res., vol 27, n°4, p.637-670, 1989.*
- [GLO 86] GLOVER F., "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers And Operations Research, vol. 13, p. 533-549, 1986.*
- [GRO 89] GROEFLIN H., LUSS H., ROSENWEIN M.B., WAHLS E.T., "Final Assembly Sequencing for Just-In-Time Manufacturing", *Int. J. Prod. Res., vol 27, n°2, p.199-213, 1989.*

- 
- [GUE 95] GUERRE CHALEY F., "Sur les Séquencement en Unité de Production Automobile", *Thèse de Doctorat, Grenoble 95*.
- [INM 92] INMAN R.R., BULFIN R.L., "Quick and Dirty Sequencing for Mixed-Model Multi-Level JIT Systems", *Int. J. Prod. Res*, vol. 30, n°9, 2011-2018, 1992.
- [JOH 90] JOHANSSON M.I., JOHANSSON B., "High Automated Kitting System For Small Parts - A case of Study from the Volvo Uddevalla Plant", *Proceedings of the 23rd International Symposium on Automotive Technology and automation, Wien*, p.75-82, 1990.
- [KIR 83] KIRKPATRICK S., GELATT C.D., VECCHI M.P., "Optimization by Simulated Annealing", *Science*, Vol 220, N° 4598, p.671-680, 1983.
- [KUB 94] KUBIAK W, SETHI S.P., "Optimal Just-In-Time Schedules for Flexible Transfer Lines", *The International Journal of Flexible Manufacturing Systems*, vol. 6, p. 137-154, 1994.
- [MER 96] MERTINS K., ALBRECHT R., O. BAHNS, BECK S., "Pull Oriented Synchronization of Logistics and Production Flow in Automobile Industries", *J.Browne et al.(Eds.), IOS Press, 1996*.
- [MIL 89a] MILTENBURG J., "Level Schedules for Mixed-Model Assembly Lines in JIT Production Systems", *Management Science Vol. 35, n°2, Février 1989*.
- [MIL 89b] MILTENBURG J., SINNAMON G., "Scheduling Mixed-Model Multi-Level Just-In-Time Production Systems", *Int. J. Prod. Res, Vol.27, No 9, 1989*.
- [MON 83] MONDEN Y. "Toyota Production System", *Appendix 2 p 181-192, Institute of Industrial Engineers Press, 1983*.
- [OKA 79] OKAMURA K., YAMASHINA H., "A Heuristic Algorithm for the Assembly Line Model-Mix Sequencing Problem to Minimize the Risk of Stopping the Conveyor", *Int. J. Prod. Res.*, vol 17, n°3, p.233-247, 1979.
- [OW 88] OW P.S., MORTON T.E., "Filtered Beam Search in Scheduling", *Int. J. Prod. Res, Vol.26, No 1, 1988*.
- [PAR 86] PARRELLO B., KABAT W.C. ET WOS L., "Job-Shop Scheduling using Automated Reasoning : A Case of Study of the Car-Sequencing Problem" *Journal of Automated Reasoning*, n°2, 1986.
- [SET 90] SETHI A.K., SETHI S.P., "Flexibility in Manufacturing : A Survey", *International Journal of Flexible Manufacturing Systems, Kluwer Academic Publishers, Boston*, vol 2, N°4, p.289-328, july 1990.
- [SIA 87] SIARRY P., DREYFUS G., "La Méthode du Recuit Simulé : Théorie et Applications", *ESPCI-IDSET, Paris, 1987*.
- [SUM 92] SUMICHRAST R.T., RUSSELL R.S., TAYLOR B.W., III, "A Comparative Analysis of Sequencing Procedures for Mixed-Model Assembly Lines in a Just-In-Time Production System", *Int. J. Prod. Res.*, vol 30, n°1, p.199-214, 1992.
- [THO 67] THOMOPOULOS N.T., "Line Balancing-Sequencing for Mixed-Model Assembly", *Management Science Vol 14, No 2, 1967*.
- [WIN 84] WINSTON P.H., "Artificial Intelligence", *Addison-Wesley Publishing Company, Chap. 4, 1984*.
-

- [WOM 90] WOMACK J., JONES D., ROOS D., "The Machine that Changed the World", *Macmillan, New-York, 1990.*
- [YAN 89] YANO C.A., BOLAT A, "Survey, Development and Application of Algorithms for Sequencing Paced Assembly Lines", *Journal Manufacturing Oper. Mgt. 2,p. 172-198, 1989*
- [YAN 91] YANO C.A., RACHAMADUGU R., "Sequencing to Minimize Work Overload in Assembly Lines with Product Options", *Management Science, vol 37, n°5, p. 572-586, 1991*

# *LISTES DES FIGURES*



## LISTES DES FIGURES

---

Figure I.1 - Evolution de l'offre automobile au XX <sup>e</sup> siècle .....	18
Figure I.2 - Logistique chez Peugeot (Le Revenu Français) .....	19
Figure I.3 - Flux global d'une usine de production automobile .....	22
Figure I.4 - Coûts des changements de teintes en fonction de la longueur des rafales. ....	25
Figure I.5 - Tableaux d'activité (perte d'efficacité) .....	28
Figure I.6 - Les différentes architectures des stocks intermédiaires.....	29
Figure I.7 - Graphique de capacité de réordonnancement d'un stock à tiroir en fonction du nombre de tiroirs et de la taille de la liste. ....	30
Figure I.8 - Tableau comparatif des stocks en statique. ....	32
Figure I.9 - Classification des problèmes d'équilibrage [GHO 89].....	34
Figure I.10 - Technique de regroupement et d'espacement pour Burns .....	40
Figure I.11 - Systèmes de gestion de production en flux poussé .....	42
Figure I.12 - Systèmes de gestion de production en flux tiré .....	43

---

Figure II.1 - Rappel de la classification de la littérature sur l'équilibrage des lignes d'assemblage.....	48
Figure II.2 - Relation entre $X_{jk}$ et $kN/Q$ dans la formulation de Monden.....	51
Figure II.3 - Niveaux de production d'un système juste-à-temps [MIL 89b].....	54
Figure II.4 - Positions cibles proposées par Inman et Bulfin [INM 92].....	55
Figure II.5 - Approche de Monden avec des vitesses de consommation instantanée constantes	56
Figure II.6 - Position des fenêtres glissantes.....	57
Figure II.7 - Exemple d'évolution de $r(n,i)$ en fonction de $i$ avec $D_n = 11$ .....	58
Figure II.8 - Limites de la notation $r(n,i)$ .....	59
Figure II.9 - Métrique des rafales.....	62
Figure II.10 - Métrique temporelle .....	63
Figure II.11 - Tableau synthétique des articles sur le séquençement .....	70
Figure II.12 - B&B binaire .....	72
Figure II.13 - B&B non binaire en profondeur d'abord.....	72
Figure II.14 - Goal Chasing Method.....	75
Figure II.15 - Méthode à P pas.....	76
Figure II.16 - Méthode à un pas avec évaluation de la borne inférieure [YAN 89] .....	76
Figure II.17 - Recuit simulé.....	78
Figure II.18 - Méthode Tabou .....	80
Figure II.19 - Croisements et mutation pour un algorithme génétique .....	80
Figure II.20 - Algorithme génétique.....	81
Figure II.21 - Comparaison des différentes méthodes .....	82

---

Figure III.1 - Exemple d'insertion.....	87
Figure III.2 - Exemple de permutation.....	88
Figure III.3 - Exemple d'une modification dans une liste respectant un ratio de 1/3.....	89
Figure III.4 - Exemples de transformations élémentaires : insertion .....	90
Figure III.5 - Comparaison entre les diverses méthodes.....	93
Figure III.6 - Enchaînement des procédures de la méthode mixte .....	94

---

<i>Figure III.7 - Importance relative des contraintes.....</i>	<i>96</i>
<i>Figure III.8 - Influence de la taille de la partition sur la qualité de la séquence .....</i>	<i>97</i>
<i>Figure III.9 - Objectifs de la recherche par faisceaux.....</i>	<i>98</i>
<i>Figure III.10 - Influence de la largeur du Beamsearch sur la qualité de la séquence .....</i>	<i>99</i>
<i>Figure III.11 - Métrique des ratios .....</i>	<i>102</i>
<i>Figure III.12 - Limitation de la vitesse de consommation en fonction des ratios à respecter</i>	<i>103</i>
<i>Figure III.13 - Simulation 1 : respect des ratios.....</i>	<i>108</i>
<i>Figure III.14 - Simulation 1 : respect du lissage des critères .....</i>	<i>108</i>
<i>Figure III.15 - Simulation 2 : respect des ratios.....</i>	<i>111</i>
<i>Figure III.16 - Simulation 2 : respect du lissage des critères .....</i>	<i>111</i>
<hr/>	
<i>Figure IV.1 - Architecture de l'application. ....</i>	<i>118</i>
<i>Figure IV.2 - Pelote de laine.....</i>	<i>120</i>
<i>Figure IV.3 - Répartition des séquences en fonction des entrées dans les tiroirs. ....</i>	<i>122</i>
<i>Figure IV.4 - Points d'insertion possibles dans la séquence .....</i>	<i>122</i>
<i>Figure IV.5 - Capacité de réordonnancement des tiroirs.....</i>	<i>124</i>
<i>Figure IV.6 - Modélisation OMT des stocks.....</i>	<i>126</i>
<i>Figure IV.7 - Influence de la longueur de la séquence prévisionnelle.....</i>	<i>131</i>
<i>Figure IV.8 - Description des enveloppes à taille fixe. ....</i>	<i>135</i>
<i>Figure IV.9 - Description des enveloppes à taille variable.....</i>	<i>137</i>
<i>Figure IV.10 - Exemple pour l'algorithme d'entrée. ....</i>	<i>139</i>
<i>Figure IV.11 - Affectation dans les tiroirs : comparaison des méthodes.....</i>	<i>140</i>
<i>Figure IV.12 - Développement complet de l'arbre et obtention des 42 solutions avec l'orientation du véhicule "1" vers les tiroirs possibles. ....</i>	<i>143</i>
<i>Figure IV.13 - Comparaison d'une métrique concave et d'une métrique convexe.....</i>	<i>145</i>
<i>Figure IV.14 - Exemple de pénalisation pour le lissage des critères.....</i>	<i>146</i>
<i>Figure IV.15 - Fonction de date souhaitée en escalier .....</i>	<i>149</i>
<i>Figure IV.16 - Exemple de calcul des poids entre règles.....</i>	<i>151</i>
<i>Figure IV.17 - Modélisation OMT de l'algorithme et de son interface.....</i>	<i>152</i>
<i>Figure IV.18 - Parcours d'arbre .....</i>	<i>154</i>
<i>Figure IV.19 - Stratégies de séquençement entrée-sortie.....</i>	<i>155</i>
<i>Figure IV.20 - Gains apportés par les différentes stratégies.....</i>	<i>156</i>
<i>Figure IV.21 - Parcours d'arbre et largeurs.....</i>	<i>157</i>
<i>Figure IV.22 - Importance de la profondeur de l'arbre sur la qualité du séquençement (à temps de calcul égaux).....</i>	<i>158</i>
<i>Figure IV.23 - Importance du temps de calcul sur la qualité du séquençement .....</i>	<i>159</i>
<i>Figure IV.24 - Influence du taux de remplissage pour un stock de 4 tiroirs.....</i>	<i>160</i>
<i>Figure IV.25 - Synthèse de l'influence du taux de remplissage. ....</i>	<i>161</i>
<i>Figure IV.26 - Influence du taux de remplissage et de la topologie du stock. ....</i>	<i>162</i>
<i>Figure IV.27 - Performance relative des tiroirs par rapport à un autostockeur .....</i>	<i>163</i>
<i>Figure IV.28 - Les différentes approches retenues et leurs niveaux de performance.....</i>	<i>164</i>
<i>Figure IV.29 - Remise en séquence avec contraintes.....</i>	<i>167</i>
<i>Figure IV.30 - Jeu de test pour les quatre approches .....</i>	<i>168</i>
<i>Figure IV.31 - Tableau comparatif des approches en fonction des critères .....</i>	<i>168</i>
<i>Figure IV.32 - Optimisation locale du flux.....</i>	<i>169</i>
<i>Figure IV.33 - Fonctionnement des stocks à tiroirs pour le cadencement reséquençable. ....</i>	<i>171</i>
<i>Figure IV.34 - Fonctionnement sans aléas.....</i>	<i>173</i>
<i>Figure IV.35 - Résultats cadencement reséquençable.....</i>	<i>174</i>
<i>Figure IV.36 - Fonctionnement avec aléas.....</i>	<i>175</i>

---

**RESUME :**

---

Pour répondre aux besoins de la clientèle, les constructeurs automobiles ont été obligés de s'adapter à une diversité grandissante de leur fabrication. Cette fabrication sur les lignes de montage automobile s'apparente à des lignes d'assemblage à modèles mélangés qui permettent de produire différents produits à la fois.

Pour une utilisation efficace de ces lignes (faisabilité de la production et coût de production), il est nécessaire de résoudre deux problèmes : l'équilibrage de la ligne (affectation des opérations aux postes) et le séquençement (choix de l'ordre) des véhicules sur cette ligne.

Dans cette thèse, nous étudions les problèmes liés au séquençement des véhicules. Nous proposons des outils pour juger de la qualité des séquences (indicateurs et métriques évaluatives) prenant en compte les problèmes de répartition des produits, de ratio de production et de dates ou de positions. Puis nous étudions les différentes approches (locales ou globales) pour optimiser des séquences.

Pour le séquençement prédictif, nous proposons une nouvelle procédure combinant des optimisations partielles puis globales qui permet d'obtenir, en limitant le temps de calcul, un flux de bonne qualité et robuste.

Cet ordonnancement prédictif ne peut être maintenu au long du processus de fabrication à cause des différents aléas de production. Des stocks intermédiaires permettent alors de remettre en séquence le flux. Nous proposons pour ces stocks un algorithme de cadencement (séquençement dynamique des véhicules en respectant la cadence de la ligne) qui optimise la séquence, mais dans un environnement plus critique : une succession de décisions locales (enlèvement de chaque véhicule) à valider sur la journée complète, respect du temps cycle de la ligne et "vision" limitée de l'algorithme au voisinage direct du stock. L'algorithme s'applique à tout type de stocks et en particuliers aux stocks dits à tiroirs, mais également à différentes politiques d'ordonnancement.

---

**INTERMEDIATE STOCKS DYNAMIC SEQUENCING FOR AN AUTOMOBILE PRODUCTION FLOW**

---

**ABSTRACT :**

To answer customers' needs, automobile constructors must adapt to a greater variety in their production. This production looks like Mixed-Model Assembly Lines which make the production of different models simultaneously possible.

To use them efficiently (production feasibility and cost), we must solve two problems : line balancing (division of work between operators) and the sequencing of models on the line.

In our work, we deal with the second of these problems. We have tools to estimate the sequence quality (indicators and functions) with spacing constraints, part usage goal and due dates objectives. We then study different approaches (local and global) to optimize sequences.

A prospective sequencing calls for the development of a new method combining partial and global optimizations to create a good and robust sequence quickly.

This prospective sequencing cannot be kept along the whole factory line because of different production hazards. Intermediary stocks then allow a resequencing of the flow. We propose an algorithm for these stocks to sequence the flow of cars dynamically (for the input and the output of the stock) and to respect the line rate. It optimizes the sequence on the line but the environment is more critical : several local decisions must be validated on the whole production day, each of them must respect the line cycle time and the algorithm has only a local "view" of the line, around the stock itself. Our algorithm works for all stocks and specially for FIFO stocks but it works for different scheduling policies too.

---

**DISCIPLINE : Productique, Automatique et Informatique Industrielle**

---

**MOTS CLES :** productique, ordonnancement, séquençement, charge de travail, contrainte de production, optimisation du flux, stocks de production, industrie automobile.

