UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

École doctorale IAEM (Informatique, Automatique, Électronique – Électrotechnique et Mathématiques)

**THÈSE**

Pour obtenir le titre de

Docteur de l'université de Lorraine

En « Automatique, Traitement du Signal et des Images, Génie Informatique »

Par **BENSMAÏNE Abderrahmane**

# Algorithmes évolutionnaires et méthodes approchées multicritères pour la génération des processus de fabrication dans un environnement reconfigurable

Soutenue le 27/11/2013 devant le jury composé de :

Rapporteurs

**DOLGUI Alexandre**, Professeur à l'École Nationale Supérieure des Mines de Saint-Etienne

**GENESTE Laurent**, Professeur à l'École Nationale d'Ingénieurs de Tarbes

Examinateurs

**ALLAOUI Hamid**, Professeur à l'Université d'Artois

**REZG Nidhal**, Professeur à l'Université de Lorraine

**YALAOUI Farouk**, Professeur à l'Université de Technologie de Troyes

Directeur de thèse

**BENYOUCEF Lyes**, Professeur à Aix-Marseille Université

Co-directeur de thèse

**DAHANE Mohammed**, Maître de Conférences à l'École Nationale d'Ingénieurs de Metz

# Remerciements

*Je tiens tout d'abord à exprimer mes remerciements les plus vifs à mes directeurs de thèse Professeur Lyes benyoucef et Dr. Mohammed Dahane, que j'ai particulièrement apprécié leur rigueur scientifique et leur disponibilité.*

*Je tien à remercier les Professeurs Dolgui et Laurent Geneste d'avoir accepté l'évaluation de ce travail en tant que rapporteurs et d'avoir apporté une lecture attentive à ce mémoire. Je remercie également les examinateurs Professeur Hamid Allaoui, Professeur Nidhal Rezg et Professeur Farouk Yalaoui d'avoir accepter d'examiner le présent travail. Qu'ils trouvent ici l'expression de ma profonde gratitude.*

*Je ne saurais oublier mes amis, Hamza et Mohammed Haoues pour leurs aides et leur soutien.*

*Enfin, je ne peux clore cette page de remerciements sans évoquer tous les membres du LGIPM, ainsi que tous ceux avec qui, de près ou de loin, j'ai passé ces dernières années ; je leur suis très redevable.*

*A mes parents, mon frère et ma sœur, et à Nardjes.*

# Table des matières

**Introduction générale** ..... **1**

**Chapitre 1. Systèmes manufacturiers reconfigurables : définitions et concepts de base** .. **5**

1.1. Introduction ..... 6

1.2. Évolution des systèmes manufacturiers ..... 6

1.2.1. Systèmes de production dédiés (DMS : Dedicated Manufacturing Systems) ..... 6

1.2.2. Systèmes manufacturiers flexibles (FMS : Flexible Manufacturing Systems) .... 7

1.2.3. Systèmes manufacturiers reconfigurables (RMS : Reconfigurable manufacturing systems) ..... 8

1.3. Principes et caractéristiques de base des RMSs ..... 11

1.3.1. Technologies clés des RMSs ..... 11

1.3.2. Caractéristiques de la reconfiguration ..... 11

1.4. Machine-outil reconfigurable (RMT : Reconfigurable Machine Tool) ..... 13

1.4.1. RMT modulaire ..... 15

1.4.2. RMT multi-outils ..... 16

1.4.3. RMT de type arc ..... 17

1.5. Technologies supplémentaires ..... 18

1.5.1. Dispositif de serrage reconfigurable ..... 18

1.5.2. Machine d'inspection reconfigurable (RIM) ..... 18

1.5.3. Contrôleur à architecture ouverte ..... 19

1.6. Conclusion ..... 20

**Chapitre 2. État de l'art et problématique de recherche** ..... **21**

2.1. Introduction ..... 22

2.2. Définitions et classification des approches de génération des processus de fabrication dans un RMS ..... 22

2.2.1. Classification par rapport au niveau de granularité ..... 24

2.2.2. Classification par rapport au degré d'adoption du CAPP ..... 24

2.3. Problématiques liées au développement des RMSs ..... 25

2.3.1. Problème de conception des RMSs ..... 26

2.3.2. Problème de génération des processus de fabrication dans le cadre des RMSs 27

2.4. Notre problématique de recherche ..... 28

2.5. Conclusion ..... 30

**Chapitre 3. Quelques concepts théoriques ..... 31**

3.1. Introduction ..... 32

3.2. Optimisation multicritère ..... 32

3.3. Méthodes de résolution multicritères traditionnelles ..... 34

3.3.1. Somme pondérée (Weighted Sum) ..... 34

3.3.2. ε-Constraint ..... 34

3.3.3. Méthode de distance à un objectif de référence ..... 35

3.4. Métaheuristiques ..... 36

3.4.1. Les algorithmes génétiques ..... 37

3.4.2. Recuit simulé ..... 44

3.5. Méthodes de sélection des solutions multicritère ..... 49

3.5.1. TOPSIS ..... 50

3.6. Conclusion ..... 52

**Chapitre 4. Problème 1 : Génération des processus de fabrication dans un RMS pour le cas unitaire ..... 53**

4.1. Description du problème ..... 54

4.2. Données d'entrée du problème ..... 55

4.3. Modélisation des données ..... 58

4.3.1. Structure de données du produit ..... 58

4.3.2. Structure de données des machines ..... 59

4.3.3. Structure de données des coûts : ..... 60

4.3.4. Structure de données des temps : ..... 60

4.4. Formulation mathématique du problème ..... 61

4.4.1. Coût total ..... 61

4.4.2. Temps total ..... 64

4.4.3. Contraintes sur le modèle ..... 66

4.5. Méthode de résolution ..... 67

4.5.1. Représentation du processus de fabrication ..... 67

4.5.2. Codage des solutions ..... 67

4.6. Expériences numériques et analyses ..... 75

4.6.1. Données d'entrée ..... 75

4.6.2. Résultats obtenus ..... 78

4.7. Extensions du problème 1 ..... 80

4.7.1. Extension 1 : utilisation de TOPSIS comme outil d'aide à la décision pour le choix du meilleur processus de fabrication ..... 80

4.7.2. Extension 2 : utilisation du process plan dans l'aide à la décision pour la conception des RMSs ..... 83

4.8. Conclusion ..... 86

**Chapitre 5. Problème 2 : Génération des processus de fabrication dans un RMS pour le cas multi-unité ..... 89**

5.1. Description du problème ..... 90

5.2. Approche proposée ..... 90

5.2.1. Environnement de simulation à événements discrets pour les RMSs ..... 92

5.2.2. Module d'optimisation ..... 94

5.3. Expériences numériques et analyses ..... 94

5.4. Conclusion ..... 96

**Chapitre 6. Problème 3 : Intégration des fonctions de génération des processus de fabrication et d'ordonnancement dans un RMS ..... 97**

6.1. Description du problème ..... 98

6.2. Etat de l'art de l'intégration de la fonction de génération des processus de fabrication et l'ordonnancement ..... 99

6.3. Heuristique proposée ..... 101

6.4. Expériences numériques et analyses ..... 105

6.5. Conclusion ..... 108

**Conclusion générale ..... 111**

# Liste des figures

Figure 1.1. Coût vs capacité (Koren 2010) ..... 9

Figure 1.2. Deux configurations d'une même RMT (Koren 2010) ..... 14

Figure 1.3. RMT modulaire (Koren 2010) ..... 15

Figure 1.4. RMT multi-outils (Koren 2010) ..... 16

Figure 1.5. Machine dédiée équivalente à la RMT précédente (Koren 2010) ..... 17

Figure 1.6. Une RMT de type arc avec quelques configurations possibles (Koren 2010) ..... 17

Figure 1.7. Une machine d'inspection reconfigurable (Koren 2010) ..... 19

Figure 2.1. Étapes du réalisation des produits ..... 22

Figure 2.2. Un exemple de processus de fabrication ..... 23

Figure 3.1.Solutions non dominées dans l'espace de solutions ..... 33

Figure 3.2. Décomposition des travaux utilisant les métaheuristiques pour des problèmes multicritères (Jones et al. 2002) ..... 37

Figure 3.3. Un croisement simple ..... 39

Figure 3.4.Une mutation pour un codage binaire ..... 39

Figure 3.5. Crowding distance ..... 43

Figure 3.6. Degré de domination entre deux solutions A et B ..... 47

Figure 4.1. Un graphe de précédence et séquencements possibles ..... 55

Figure 4.2. Exemple d'une pièce à manufacturer ..... 56

Figure 4.3. Graphe de précédence associé à la pièce présenté sur la Figure 4.2 ..... 56

Figure 4.4. Exemple d'une opération nécessitant un TAD (-Z) ..... 57

Figure 4.5. Exemple du problème de faisabilité ..... 68

Figure 4.6. Correction de solutions ..... 68

Figure 4.7. Exemple de décodage ..... 71

Figure 4.8. Exemple d'une sélection d'une caractéristique ..... 71

Figure 4.9. Les deux fronts de Pareto obtenus par AMOSA et NSGA-II ..... 79

Figure 4.10. Le passage d'un Front de Pareto (1) à une solution unique (2) en utilisant TOPSIS (Cas bi-objectifs). ..... 81

Figure 4.11. Représentation graphique des solutions obtenues ..... 83

Figure 5.1. Approche proposée pour le cas multi-unité ..... 91

Figure 5.2. Optimisation basée sur la simulation ..... 91

Figure 5.3. Diagramme de classes du simulateur ..... 92

Figure 5.4. Diagramme de classe de base pour le RMS ..... 93

Figure 5.5. Exemple d'un chromosome ..... 94

Figure 5.6. Approche proposée vs. Processus de fabrication redondant ..... 96

Figure 6.1. Organigramme résumant l'approche proposée ..... 104

Figure 6.2. Un génome correspondant à un ordonnancement ..... 106

Figure 6.3. Graphe de précédence généré pour un produit A ..... 108

Figure 6.4. Graphe de précédence généré pour un produit B ..... 108

# Liste des tableaux

Tableau 1.1. RMS combinant les caractéristiques des DMS et FMS (Koren 2010)..... 10

Tableau 1.2. Classes de machines et de systèmes manufacturiers (Koren 2006) ..... 11

Tableau 1.3. RMT combine les caractéristiques des machines dédiées et CNC..... 15

Tableau 4.1. Un processus de fabrication sous forme de tableau ..... 67

Tableau 4.2. Un processus de fabrication codé en nombre réel ..... 69

Tableau 4.3. TADs possibles sur la machine 1 ..... 75

Tableau 4.4. TADs possibles sur la machine 2 ..... 76

Tableau 4.5. TADs possibles sur la machine 3 ..... 76

Tableau 4.6. Outils disponibles avec les différentes machines ..... 76

Tableau 4.7. Relations de précédence pour la caractéristique 1..... 76

Tableau 4.8. Relations de précédence pour la caractéristique 2..... 77

Tableau 4.9. . Relations de précédence pour la caractéristique 3..... 77

Tableau 4.10. TADs requis par les opérations de la caractéristique 1 ..... 77

Tableau 4.11. TADs requis par les opérations de la caractéristique 2 ..... 77

Tableau 4.12. TADs requis par les opérations de la caractéristique 3 ..... 78

Tableau 4.13. Outils requis pour chaque opération..... 78

Tableau 4.14. Résultats obtenus par AMOSA et NSGA-II..... 79

Tableau 4.15. Résultats obtenus, triés avec TOPSIS ..... 82

Tableau 4.16. Données des machines..... 85

Tableau 4.17. Résultats obtenus pour le cas de sélection de machines..... 86

Tableau 5.1. Résultats de l’exemple numérique du cas multi-unité..... 95

Tableau 6.1. Processus de fabrication généré pour le produit A ..... 107

Tableau 6.2. Processus de fabrication généré pour le produit B ..... 107

Tableau 6.3. Résultats obtenus par l'approche classique et par notre heuristique..... 109

# Liste des abréviations

| | | |
|---|---|---|
| AMOSA | : | Archived Multi-objective Simulated Annealing |
| CAPP | : | Computer-Aided Process Planning |
| CNC | : | Computer Numerically Controlled machine |
| DML | : | Dedicated Manufacturing Line |
| DMS | : | Dedicated Manufacturing Systems |
| FMS | : | Flexible Manufacturing System |
| MOGA | : | Multi-Objective Genetic Algorithm |
| NSGA-II | : | Elitist Non-dominated Sorting Genetic Algorithm 2 |
| RIM | : | Reconfigurable Inspection Machine |
| RMS | : | Reconfigurable Manufacturing Systems |
| RMT | : | Reconfigurable Machine Tool (Machine-outil reconfigurable) |
| TAD | : | Tool Approach Direction |
| TOPSIS | : | Technique for Order Performance by Similarity to Ideal Solution |
| UML | : | Unified Modeling Language |

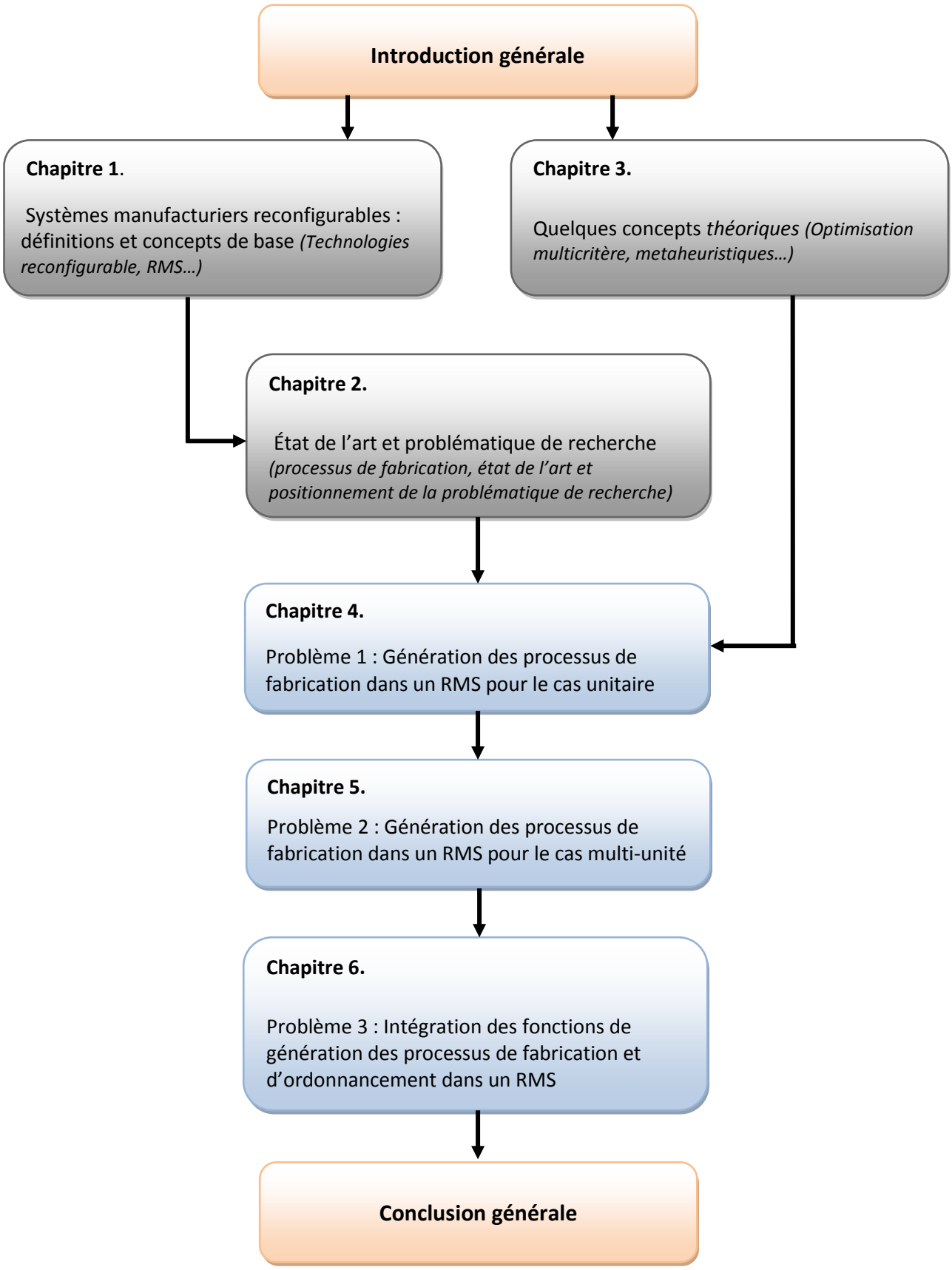


**Figure 0. Synoptique du manuscrit de thèse**

# Introduction générale

Dans le secteur industriel, la production est l'activité qui transforme des matières premières pour créer des produits finis. Les systèmes manufacturiers ont été développés afin d'assurer cette activité de production. Un système manufacturier regroupe l'ensemble des éléments matériels et immatériels nécessaires à la production, composé d'une collection de machines ou de stations en collaboration pour effectuer un ensemble d'opérations sur une matière première en vue d'obtenir la forme finale souhaitée du produit demandé et spécifié par le client.

Le contexte incertain des marchés actuels a contribué au développement des systèmes manufacturiers, donnant lieu à des systèmes ayant différents principes et fonctionnalités, et visant à répondre à des exigences en perpétuels changements. En effet, les systèmes manufacturiers sont immergés dans un environnement où les cycles de vie deviennent de plus en plus courts, les demandes très imprévisibles et les produits avec des caractéristiques personnalisables. Cet environnement a amené l'entreprise à toujours viser de hauts niveaux de réactivité et de flexibilité, lui garantissant la meilleure réponse aux besoins de ses clients.

A la fin du XX$^{e}$ siècle, on distinguait deux types principaux de systèmes manufacturiers : les Systèmes de Production Dédiés (Dedicated Manufacturing Systems – DMS) et les Systèmes de Production Flexibles (Flexible Manufacturing Systems – FMS), auxquels viennent de s'ajouter, ces dernières années, un nouveau type : les Systèmes Manufacturiers Reconfigurables (Reconfigurable Manufacturing Systems - RMS).

Un RMS est un système manufacturier où les machines, les composants des machines ainsi que le système de manutention peuvent être ajoutés, modifiés, supprimés ou échangés selon les besoins de la production. Un RMS combine le haut débit des DMSs avec la grande flexibilité des FMSs tout en garantissant un haut niveau de réactivité face aux changements.

Le principal composant d'un RMS est la machine reconfigurable (Reconfigurable Machine Tool – RMT). Une RMT est une machine possédant une structure qui permet sa

reconfiguration pour fournir une fonctionnalité alternative ou, dans d'autres cas, pour assurer une augmentation de la cadence de production.

Les RMSs constituent un domaine de recherche actif, stimulé par ses différences par rapport aux autres systèmes existants. En effet, les méthodes et les approches de gestion précédemment développées pour les systèmes manufacturiers classiques nécessitent d'être réexaminées, adaptées ou remplacées pour répondre au mieux aux nouveaux défis posés par les RMSs.

La conception des systèmes et la génération des processus de fabrication constituent les deux domaines les plus étudiés dans le cadre des RMSs, car ils dépendent fortement de la structure du système manufacturier concerné et le type de ses machines. Dans le cadre de cette thèse nous nous intéressons principalement au problème de la génération des plannings de fabrication dans le cadre des RMSs. Ce travail est motivé par le besoin de méthodologies qui prennent en compte la reconfigurabilité dans le cadre de la génération des processus de fabrication. En plus, la nécessité de représenter d'une manière générique les capacités des machines reconfigurables à effectuer les différentes opérations sur les différents produits.

Nous présentons dans le cadre de ce travail une contribution au CAPP (Computer-Aided Process Planning) dans le cadre de génération de processus de fabrication pour un RMS avec une représentation générique des machines reconfigurables. En effet, nous cherchons à exploiter au mieux les hauts degrés de reconfigurabilité des machines et des outils afin de proposer des processus de fabrication efficaces. Cette capacité de reconfigurabilité permet au processus de fabrication de s'adapter aux différentes contraintes des produits à réaliser et leurs futures évolutions.

Nous désignons par un processus de fabrication (*Process Plan* en anglais) la fonction qui consiste à décider quels procédés de fabrication et quelles machines doivent être utilisés pour effectuer les différentes opérations nécessaires à la fabrication d'un produit défini, ainsi que le séquencement des opérations que les processus doivent suivre.

Dans le cadre de cette thèse, nous avons abordé trois problématiques de recherche relatives aux RMSs :

1. La génération des processus de fabrication dans le cas d'une seule unité à produire. Nous proposons une approche multicritère qui génère un processus de fabrication optimal pour un type de produit dans un environnement reconfigurable.

2. La génération des processus de fabrication pour un RMS dans le cas de multi-unité, où plusieurs unités du même produit sont fabriquées en concurrence dans le RMS. Notre approche multicritère consiste à associer à chaque unité un processus de fabrication de tel sorte que plusieurs objectifs soient optimisés.

3. L'intégration des fonctions de génération des processus de fabrication avec l'ordonnancement. Dans ce cas nous proposons une heuristique permettant de réaliser cette intégration dans un cadre reconfigurable.

Le rapport de thèse est structuré en six chapitres :

Dans le chapitre 1, nous présentons un ensemble de définitions et de concepts de base liés aux systèmes manufacturiers reconfigurables. Nous nous intéressons aux limites des systèmes manufacturiers existants pour illustrer les besoins de mise en place des RMSs. Nous détaillerons les avantages, les caractéristiques et les technologies clés liées aux RMSs. Nous terminons le chapitre par une conclusion.

Le chapitre 2 est dédié à la présentation de notre état de l'art sur les RMSs et les problématiques associées, ainsi qu'au positionnement de notre problématique de recherche. Nous rappelons dans ce chapitre les notions de processus de fabrication et les différentes techniques existantes permettant d'avoir le/les processus de fabrication dans un RMS. Nous terminons ce chapitre par le positionnement de notre travail de recherche par rapport aux travaux rapportés et les objectifs tracés dans le cadre de cette thèse.

Dans le chapitre 3 nous décrivons quelques concepts de base liés à l'optimisation multicritère utilisés dans le cadre de cette thèse. Les méthodes de résolution classiques ainsi que les métaheuristiques les plus performantes (algorithmes génétiques et recuit simulé) sont présentées avec leurs adaptations multicritère (NSGA-II et AMOSA). Par ailleurs, TOPSIS (une méthode d'aide à la décision) est exposée, et servira comme outil d'aide à la sélection de la meilleure solution parmi les solutions obtenues avec les approches multicritère.

Le chapitre 4 expose l'approche adoptée pour la génération des processus de fabrication dans le cas unitaire. Un modèle mathématique est proposé permettant la minimisation du coût total et le temps total de réalisation d'une seule unité du produit. Une technique de codage en nombre réel est proposée pour éviter les solutions irréalisables. Un panel d'expériences numériques est présenté et analysé pour illustrer l'applicabilité de l'approche, où nous comparons les performances de NSGA-II et AMOSA.

Le chapitre 5 est consacré à l'approche proposée pour la génération des processus de fabrication dans le cas multi-unité. L'approche est composée de deux phases. La première phase détermine un ensemble de processus de fabrication optimaux. La seconde phase associe à chaque unité un processus de fabrication. L'identification du meilleur processus de fabrication, unité par unité, est effectuée via une optimisation basée sur la simulation. Le chapitre présente le cadre de simulation à évènements discrets développée pour un environnement reconfigurable. Pour illustrer l'applicabilité de l'approche, nous présentons un panel d'expériences numériques.

Dans le dernier chapitre, une approche heuristique pour l'intégration des fonctions de génération des processus de fabrication et d'ordonnancement dans le cas d'un RMS est développée. Cette heuristique repose sur la nature multi-configuration des différentes machines RMT. Pour illustrer l'applicabilité et l'efficacité de l'heuristique proposée, un exemple numérique est présenté où l'heuristique est comparée à une approche non intégrée, dissociant la génération des processus de fabrication et l'ordonnancement.

Enfin, nous terminons ce mémoire par quelques remarques et perspectives pour des travaux de recherche futurs.

# Chapitre 1.

## Systèmes manufacturiers reconfigurables : définitions et concepts de base

*Dans ce chapitre, nous rappelons l'ensemble des définitions et des concepts de base liés aux systèmes manufacturiers reconfigurables (RMS : Reconfigurable Manufacturing Systems). Dans un premier temps, nous nous intéressons aux limites des systèmes manufacturiers existants pour illustrer les besoins de mise en place des RMS. Ensuite, et afin d'avoir une vision générale et complète sur les systèmes manufacturiers reconfigurables, nous listons les avantages, les caractéristiques et les technologies clés liées aux RMSs. Nous terminons le chapitre par une conclusion.*

## 1.1. Introduction

Un système de manufacturier regroupe l'ensemble des éléments matériels et immatériels nécessaires à la production de biens par une entreprise. C'est une collection de machines ou de stations en collaboration pour assembler un produit fini ou pour effectuer un ensemble contrôlé et fini d'opérations sur une matière première en vue d'obtenir la forme finale souhaitée. Les besoins du marché ont contribué au développement des systèmes manufacturiers, donnant lieu à des systèmes ayant des principes et des fonctionnalités différentes, qui essayent de répondre aux exigences en évolution, passant des systèmes dédiés et lignes de production aux systèmes flexibles et reconfigurables.

## 1.2. Évolution des systèmes manufacturiers

A la fin du XX$^{e}$ siècle, on distinguait deux types principaux de systèmes manufacturiers : DMSs (Dedicated Manufacturing Systems) et les Systèmes flexibles de Production (Flexible Manufacturing Systems – FMS), auxquels viennent de s'ajouter les systèmes manufacturiers reconfigurables (Reconfigurable Manufacturing Systems - RMS) au début de ce nouveau millénaire.

### 1.2.1. Systèmes de production dédiés (DMS : Dedicated Manufacturing Systems)

Les DMSs, appelés aussi lignes de transfert ou lignes de production, sont conçues pour fabriquer un seul type de produit avec un très haut volume de production. Un DMS est basé sur une collection de machines relativement simples, arrangées séquentiellement le long d'une ligne sur laquelle la matière première avance d'une machine à l'autre d'une manière synchronisée. Un DMS est économiquement rentable lorsque de très grandes quantités du produit sont nécessaires.

Les DMSs constituaient un élément de base de la fabrication à haut volume pendant plusieurs décennies. Fidèles au paradigme de la production de masse, puisque leur volume de production est élevé, le coût par pièce est relativement faible. Une fois un DMS est entièrement mis en place, il fonctionne suivant un rythme constant, et peut produire des pièces identiques de haute qualité et à de grandes quantités. Cependant, avec l'accroissement de la concurrence mondiale et l'avènement des concepts de personnalisation, la demande de pièces

spécifiques peut varier considérablement en termes de volume de production et de caractéristiques des pièces à produire.

Ces nouvelles contraintes engendrent des situations dans lesquelles les DMSs ne fonctionnent pas à pleine capacité. Cette sous-exploitation d'un DMS provoque une augmentation considérable du coût par produit, ce qui fait perdre au DMS son économie d'échelle.

En effet, en raison de la volatilité accrue du marché et des pressions concurrentielles, les DMSs sont de plus en plus sous-utilisés. Un rapport publié en Italie en 1998 indique que le taux moyen d'utilisation des DMSs observé chez un constructeur automobile européen était seulement de 53% (Koren 2010). Ce faible taux s'explique par le fait que pendant la phase de lancement des nouveaux produits et la phase de fin du cycle de vie du produit, peu de produits sont nécessaires, soit nettement moins que les volumes optimaux. Ainsi, même les produits dans la phase de maturité n'atteignent pas toujours les volumes de production prévus lorsque la ligne a été conçue.

D’un autre côté, les DMSs sont très limités en matière de réactivité. Ceci présente un grand handicape lorsque la demande dépasse la capacité prévue à la conception. Si la popularité d'un produit dépasse les attentes et les prévisions de l'entreprise, ou lorsque de nouvelles utilisations sont mises en œuvre pour les produits existants, les DMSs ne peuvent réagir à ces fluctuations des quantités à produire, ce qui entraîne une perte de chiffre d'affaires de l'entreprise.

Mais le principal inconvénient des DMSs dans l'ère de la mondialisation est que ses lignes ne sont pas conçues pour un futur changement, et ne peuvent pas être facilement convertis pour réaliser de nouveaux produits, alors qu’à l'ère de la mondialisation, la durée de vie d'un produit commercialisable est de plus en plus courte et les produits concurrents sont introduits de plus en plus vite.

Toutes ces nouvelles contraintes rendent les DMSs non rentables, ce qui fait accentuer leur disparation dans de nombreuses industries manufacturières.

### 1.2.2. Systèmes manufacturiers flexibles (FMS : Flexible Manufacturing Systems)

Un FMS est défini comme un groupe intégré d'unités d’usinage, généralement des machines-outils à commande numérique (CNC : Computer Numerically Controlled), reliées par un

système de manutention automatisé, dont le fonctionnement est contrôlé par un ordinateur de supervision (Sethi & Sethi 1990).

Contrairement aux DMS qui couvrent la production avec un grand volume/une faible variété, les FMSs représentent un facteur majeur dans « la personnalisation de masse », car ils peuvent produire une variété de composants et de produits au sein de sa capacité initiale. Un FMS permet également la redéfinition de produits pour répondre aux nouvelles exigences du marché sans ajouter d'importants investissements dans le système de fabrication initial. La plupart des expériences acquises avec des installations FMS ont été réalisées avec des systèmes d'usinage CNC destinées à la production de faible volume/grande variété.

### 1.2.3. Systèmes manufacturiers reconfigurables (RMS : Reconfigurable manufacturing systems)

Le changement rapide dans les environnements manufacturiers a donné lieu à un nombre de concepts émergeants qui traitent des problèmes liés à la conception, le développement et l'amélioration des systèmes de production. Un système manufacturier est réactif s'il possède une capacité de production ajustable aux fluctuations des demandes, et ses fonctionnalités sont adaptables aux besoins des nouveaux produits (Koren 2010).

Historiquement, le besoin au développement dans le secteur manufacturier a toujours était lié à des objectifs économiques prédominants tel que le coût de production. Un tel objectif était suffisamment atteint par les systèmes de production de masse DMS. Cependant, ces systèmes classiques rencontrent des difficultés lorsqu'il s'agit des changements dynamiques sur un horizon de production. Par exemple, les systèmes dédiés sont conçus et ils fonctionnent sous l'hypothèse que des informations complètes sur le produit à manufacturer sont entièrement disponibles.

D'un autre côté, les FMSs, qui garantissent une grande variété de produits, sont confrontés à des difficultés compétitives du point de vue économique, puisqu'ils présentent des coûts d'installation ainsi que des coûts de production élevés (Koren & Ulsoy 1997).

Dans ce contexte, il existe un besoin pour un système manufacturier réactif, qui non seulement regroupe le haut débit garanti par les DMSs avec la variété offerte par les FMSs, mais qui offre de grande potentialité de réactivité aux changements du marché en proposant des possibilités d'une modification rapide et efficace du système manufacturier et de ses éléments.

Ce système manufacturier a été proposé par Koren et al. (1999), et appelé « Système Manufacturier Reconfigurable » (RMS : Reconfigurable Manufacturing System). Il est défini comme étant un système manufacturier où les machines, les composants des machines ainsi que le système de manutention peuvent être ajoutés, modifiés, supprimés ou échangés selon les besoins de la production, afin de répondre efficacement et rapidement aux changements continus des exigences du marché (Koren et al. 1999).

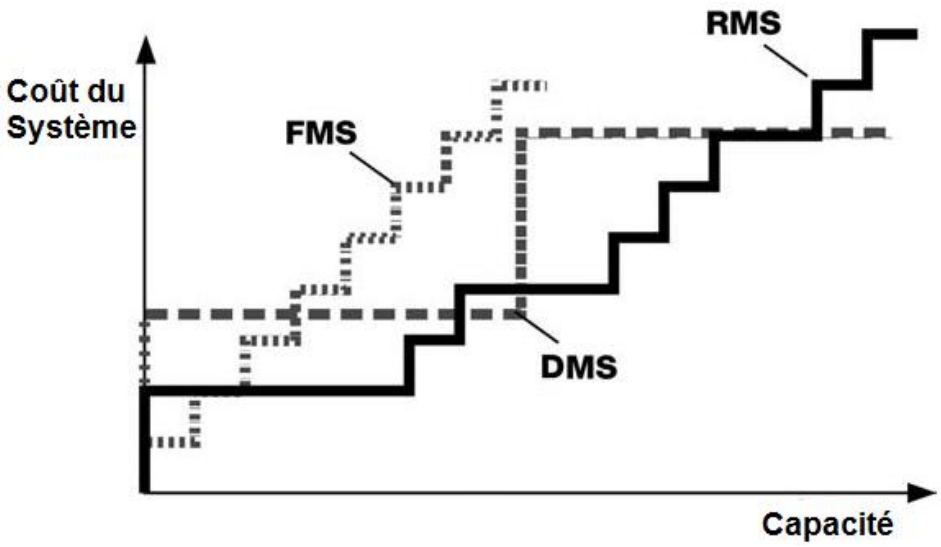


**Figure 1.1. Coût vs capacité (Koren 2010)**

Afin de de mieux répondre aux fluctuations des marche, et ainsi, faire face aux problèmes rencontrés par les autres systèmes manufacturiers, les RMSs ne doivent pas seulement combiner le haut débit des DMSs avec la flexibilité des FMSs (Figure 1.1), mais doivent également garantir un haut niveau de réactivité face aux changements (Koren 2006), et ceci en assurant les deux points suivants :

- La conception d'un système et de ses machines avec des structures ajustables qui permettent la modification et l'extensibilité aux niveaux système et machine.
- La conception du système avec une flexibilité personnalisable, suffisante pour manufacturer les produits ciblés, généralement de la même famille de produit.

A travers le Tableau 1.1 nous présentons une comparaison entre les différents types de systèmes définis ci-dessus, à savoir le DMS, FMS et RMS.

| | Dedicated Manufacturing System | Reconfigurable Manufacturing System | Flexible Manufacturing System |
|---|---|---|---|
| Structure système | Fixe | Modifiable | Modifiable |
| Structure machine | Fixe | Modifiable | Fixe |
| Focus du système | Produit | Famille de produits | Machine |
| Extensibilité | Non | Oui | Oui |
| Flexibilité | Non | Personnalisable | Générale |
| Outils parallèles | Oui | Oui | Non |
| Productivité | Très Haute | Haute | Faible |
| Coût | Bas | Moyen | Haut |

**Tableau 1.1. RMS combinant les caractéristiques des DMS et FMS (Koren 2010)**

Les systèmes reconfigurables présentent un ensemble d'avantages qui consistent en la réduction des coûts de fabrication par la diminution des dépenses de la planification ainsi que le raccourcissement des temps de changements, ce qui conduit à une réduction du coût de transformation.

Aussi, la valeur des composants adaptables augmentent s'ils peuvent être réutilisés. L'augmentation du nombre d'éléments standardisés réduit les dépenses pour les pièces mécaniques, alors que le coût d'assemblage augmente, mais en final, la somme des deux diminue (Heisel & Meitzner 2006).

Zhang et al. ont étudié les RMSs par rapport aux autres systèmes de production. La comparaison était faite analytiquement et empiriquement sur les critères suivants : le coût du cycle de vie, l'adaptabilité, la complexité, le taux de production, le temps de configuration et de temps de montage. Les résultats montrent clairement que les RMSs représentent le futur des systèmes de production (Zhang et al. 2006). Heisel et Meitzner présentent la reconfigurabilité comme la technologie de production principale en 2020 (Heisel & Meitzner 2006). Mehrabi et al. confirment l'avenir prometteur des RMSs et les proposent comme une alternative aux problèmes et à l'insatisfaction rencontrés avec les FMSs (Mehrabi et al. 2002).

## 1.3. Principes et caractéristiques de base des RMSs

### 1.3.1. Technologies clés des RMSs

Les dernières années ont vu l'émergence de deux nouvelles technologies qui ont constitué par la suite des éléments clés pour les RMSs. Ces deux technologies sont respectivement les **contrôleurs à architecture ouverte** sur le plan logiciel et les machines **modulaires** sur le plan matériel. Ces deux technologies offrent au client plus d'options fonctionnelles dans un système manufacturier (Tableau 1.2).

| | Hardware fixé | Hardware reconfigurable |
|---|---|---|
| Pas de software | Machine manuelle, DMS | ------ |
| Software fixé | FMS, machine CNC, Robot | Machines reconfigurable |
| Software reconfigurable | AGV[1], open architecture | RMS |

**Tableau 1.2. Classes de machines et de systèmes manufacturiers (Koren 2006)**

En outre, pour un RMS, du matériel et des logiciels reconfigurables sont nécessaires, mais ils ne sont pas suffisants pour garantir sa rentabilité. En effet, la conception des RMSs doit impérativement prévoir les fonctionnalités opérationnelles possibles durant la durée de vie du RMS avec leur modèle économique relatif. Avec un tel design, la capacité et la fonctionnalité peuvent être changées en réponse aux changements du marché. Nous remarquons ainsi qu'un RMS présente exactement les fonctionnalités requises et au moment où elles sont nécessaires.

### 1.3.2. Caractéristiques de la reconfiguration

Dès sa conception, un RMS doit comporter des modules hardware et software qui pourront par la suite être intégrés rapidement et d'une manière fiable. Dans le cas contraire, le processus de reconfiguration sera plus lent et non pratique. De ce fait, un RMS doit posséder six caractéristiques primordiales :

1) ***Personnalisation :*** cette caractéristique distingue nettement les RMSs des FMSs et des DMSs. En effet, la flexibilité personnalisable implique que le RMS doit s'adapter à la majorité dominante des caractéristiques de la famille de produit à manufacturer. La

[1] AGV (Automatic Guided Vehicle) : véhicule à guidage automatique, est un robot qui se déplace dans une usine de façon autonome sans l'intervention humaine, afin de transporter des produits ou d'autres entre des stations.

personnalisation permet l'existence de multiples outils sur la même machine, augmentant ainsi la productivité à moindre coût sans compromettre la flexibilité. Un RMS bien conçu garantira le bon équilibre entre la productivité et la flexibilité générale.

2) ***Convertibilité :*** il s'agit de la capacité du système à changer rapidement sa fonctionnalité pour produire tous les types de produits d'une famille. La convertibilité inclut la conversion des machines. Elle peut concerner par exemple le changement du degré de liberté de la machine, ou l'intégration, au niveau système, d'autres machines permettant d'élargir la fonctionnalité du RMS.

3) ***Extensibilité :*** il est de plus en plus difficile de prévoir les caractéristiques et les quantités des produits demandés, ce qui justifie la nécessité d'avoir des systèmes manufacturiers extensibles. L'extensibilité de la capacité de production désigne l'aptitude à changer le volume de production maximal en concordance avec les prévisions et les besoins de l'entreprise. L'extensibilité pourrait nécessiter l'ajout de nouveaux axes aux machines pour accroître leurs capacités. Au niveau système, l'extensibilité concerne la modification du routage des produits ou l'intégration de nouvelles machines.

4) ***Modularité :*** la nature des RMSs fait qu'ils nécessitent des structures modulaires. En effet au niveau système, chaque machine représente un module, qui à son tour, est constituée de plusieurs composants modulaires. En plus, le système de manutention est conçu d'une manière modulaire afin de faciliter les futures reconfigurations. En cas de besoin, les éléments modulaires sur chaque niveau, peuvent être remplacés pour mieux adapter le RMS aux nouvelles situations et contraintes du marché.

5) ***Intégrabilité :*** il s'agit, au niveau machine, de la possibilité à ajouter des axes et des degrés de liberté pour former de nouvelles machines. Au niveau système, l'intégrabilité concerne l'ajout de machines par l'intermédiaire des systèmes de manutention, afin d'avoir une configuration système intégrant de nouvelles fonctionnalités

6) ***Diagnosticabilité :*** Dans les RMSs, la diagnosticabilité a deux objectifs : la détection des pannes des machines et la détection des qualités des produits de niveau insatisfaisant par rapport aux exigences imposées. Ce deuxième aspect requiert une

attention particulière dans les RMSs. En effet, comme les systèmes de production sont reconfigurables et les dispositions de machines sont modifiées le plus souvent, il est essentiel de régler rapidement le système nouvellement reconfiguré de façon à produire rapidement des produits avec le niveau de qualité exigé. Dans ce contexte, les systèmes reconfigurables doivent comprendre des systèmes de contrôle de la qualité des produits. Ces systèmes sont basés sur les technologies de contrôle, des statistiques et des techniques de traitement du signal. Ils sont conçus pour identifier les problèmes de qualité des produits dans le système de production, de sorte qu'ils puissent être corrigés rapidement.

Ces six principales caractéristiques des RMSs réduisent le temps et les efforts de reconfiguration et améliorent ainsi la réactivité du système. Ces caractéristiques peuvent réduire de manière fiable le coût moyen du système par rapport à son cycle de vie en lui permettant d'évoluer constamment, pour répondre au mieux aux dynamiques des marchés, des demandes des consommateurs et des technologies.

## 1.4. Machine-outil reconfigurable (RMT : Reconfigurable Machine Tool)

Le principal composant d’un RMS est la machine reconfigurable (RMT : Reconfigurable Machine Tool). Il s'agit d'un nouveau type de machines possédant une structure modulaire permettant sa reconfiguration (Landers et al. 2001). Une RMT est une machine dont la structure peut être reconfigurée pour fournir soit une fonctionnalité alternative ou pour assurer une augmentation progressive de son taux de production, afin de répondre aux fluctuations de la demande. En fonction des besoins, un RMT peut toujours revenir à son état original ou modifié afin de fournir de nouvelles fonctionnalités ou une capacité de production la mieux adaptée.

La conception des RMTs se repose donc sur deux objectifs fondamentaux :

1. Adapter le fonctionnement de la machine pour répondre aux différents besoins en termes de fonctionnalités.
2. Augmenter le taux de production de la machine en ajoutant des ressources supplémentaires.

La Figure 1.2 présente un exemple d'un RMT. La pièce à manufacturer est située sur un support qui peut se déplacer simultanément sur les deux axes $X$ et $Y$. Les broches $Z_1$ et $Z_2$ peuvent se déplacer linéairement sur leurs axes respectifs, situés sur le même support. La broche $Z_3$ peut usiner la pièce sur un axe horizontal avec des différents angles. Les broches peuvent être rapidement ajoutées ou enlevées en fonction des besoins. Dans la deuxième configuration, on a déplacé le support horizontal du slot $B$ au slot $C$ et le support horizontal a également changé de position. Cette machine, avec ces différentes configurations possibles, a accès à plus de points d'usinage par rapport à une machine classique, et une productivité plus importante par rapport à une machine CNC ayant un outil unique.

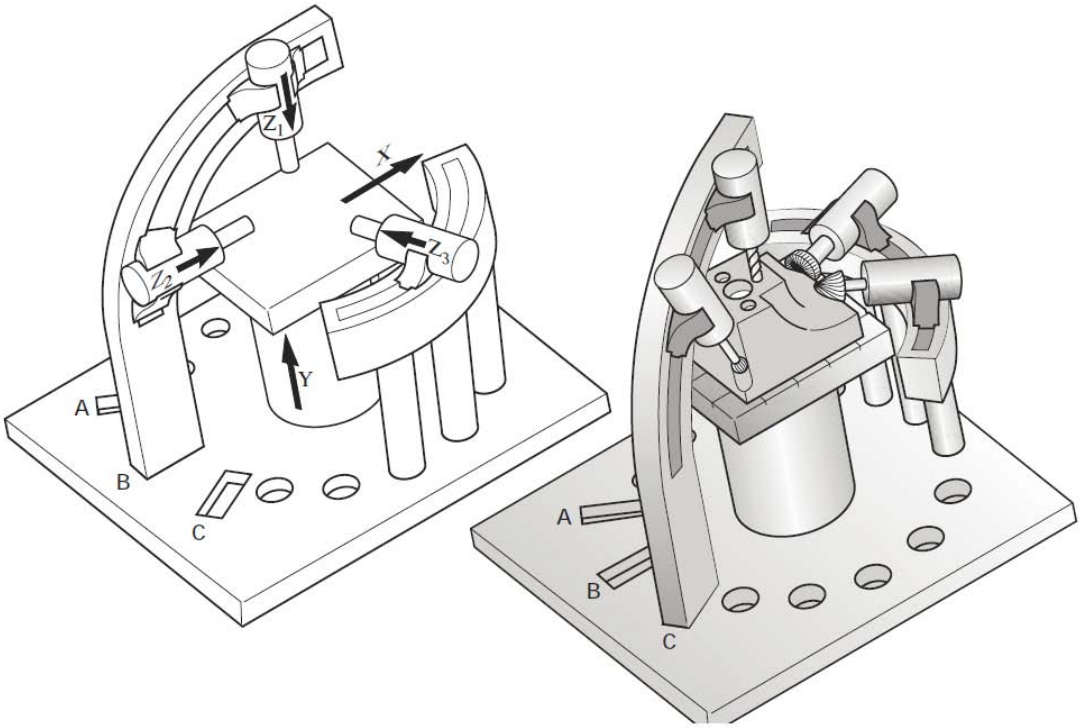


**Figure 1.2. Deux configurations d'une même RMT (Koren 2010)**

Ainsi, nous constatons que les RMTs sont plus rentables car elles sont conçues pour une gamme spécifique de contraintes (sur les opérations), et peuvent être économiquement converties d'une gamme à une autre. Les avantages des RMTs par rapport aux autre machines sont mis en évidence à travers le tableau 3.

| | Machine dédiée | RMT | Machine CNC |
|---|---|---|---|
| Structure de la machine | Fixe | Ajustable | Fixe |
| Conception centrée sur | Produit | Famille de produit | Machine |
| Extensibilité | Non | Oui | Oui |
| Flexibilité | Non | Oui | Oui |
| Outils opérationnels en même temps | Oui | Oui | Non |

**Tableau 1.3. RMT combine les caractéristiques des machines dédiées et CNC**

Nous trouvons dans la littérature trois types de RMT. Ils sont définis comme suit :

### 1.4.1. RMT modulaire

Le terme « reconfigurable » est naturellement associé à des machines modulaires. Bien qu'il existe des machines reconfigurables non modulaires (*cf.* sections 1.4.2 et 1.4.3), les RMTs modulaires représentent une classe importante des RMTs.

Une RMT modulaire est présentée dans la Figure 1.3, avec trois configurations différentes.

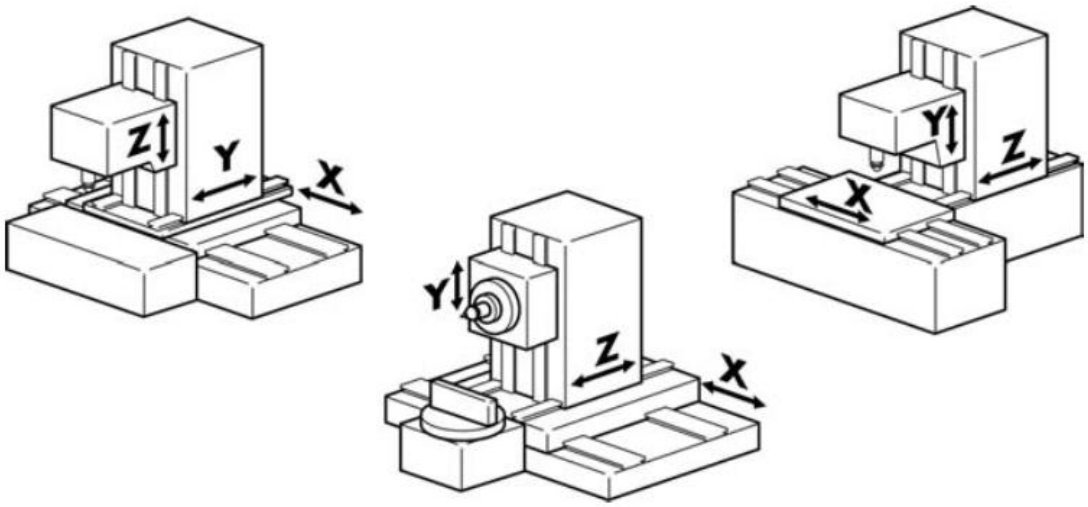


**Figure 1.3. RMT modulaire (Koren 2010)**

Le défi majeur de cette classe de machines et de concevoir des interfaces entre les modules garantissant le bon fonctionnement de la machine. Ces interfaces concernent généralement l'alimentation électrique, la connexion mécanique et hydraulique, ainsi que les interfaces d'information pour le contrôleur de la machine. En revanche, des standard d'interfaçage pour ce type de machines n'ont pas encore été établis (Koren 2010). Une autre barrière à l'utilisation des RMTs modulaire réside dans la difficulté à calibrer les différents composants pour avoir un ensemble qui fonctionne avec précision.

### 1.4.2. RMT multi-outils

Une RMT multi-outils est une machine que l'on peut reconfigurer de sorte que plusieurs outils peuvent opérer simultanément à chaque configuration. La multitude d'outils est adaptée à chaque type d'opération effectuée par la machine. Un exemple de RMT multi-outils est donné sur la Figure 1.4.

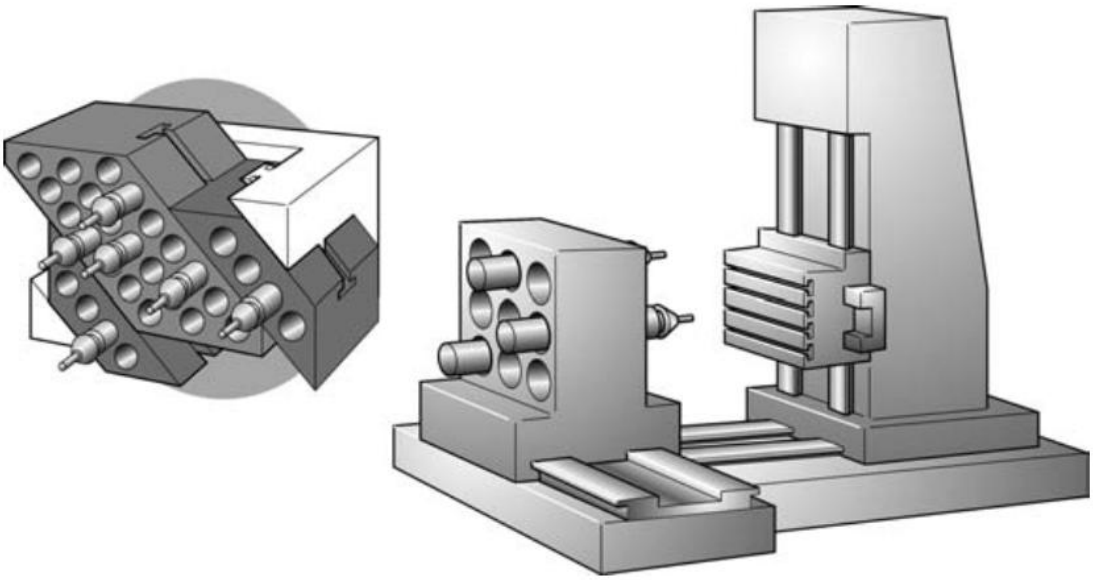

**Figure 1.4. RMT multi-outils (Koren 2010)**

Dans cet exemple, la RMT peut usiner une pièce avec plusieurs outils en un seul mouvement du support. En changeant le modèle de disposition des outils, la machine devient capable d'effectuer d'autres opérations, en terme de nombre de troues, évidement avec plusieurs outils à la fois. Cette machine peut être comparée à une perceuse à multiple têtes (Figure 1.5), sauf que cette dernière ne peut pas être reconfigurée pour d'autres types de troues ou des troues avec d'autres positionnements sur la pièce.

Une fois configurée, cette machine RMT peut être utilisée comme une machine dédiée avec un outil comme celui présenté sur la Figure 1.5, sauf qu'elle pourra être reconfigurée pour d'autre tâche en cas de besoin.

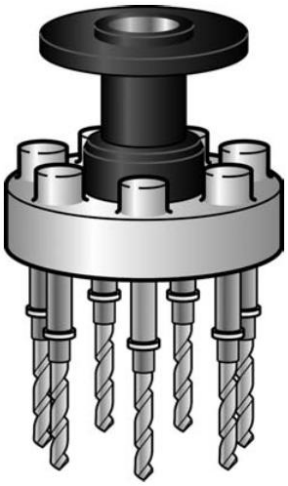

**Figure 1.5. Machine dédiée équivalente à la RMT précédente (Koren 2010)**

### 1.4.3. RMT de type arc

Ce type de machine est conçu pour des besoins d'une famille de produit particulière. La conception de ce type de machines est basée sur les résultats d'un logiciel de RMT-design développé par le *Engineering Research Center for Reconfigurable Manufacturing System* de l'université de Michigan, un des laboratoires pionniers dans les systèmes reconfigurables.

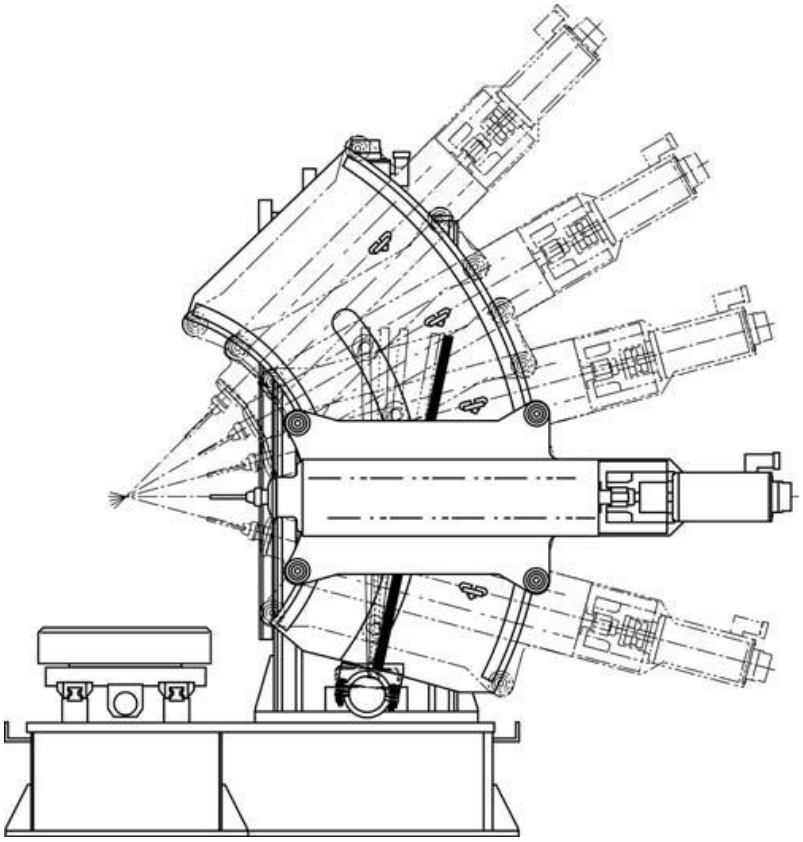

**Figure 1.6. Une RMT de type arc avec quelques configurations possibles (Koren 2010)**

Nous distinguons sur la Figure 1.6 les différentes possibilités de cette machine offertes par le déplacement de la broche support de l'outil sur l'arc de la machine, permettant ainsi d'atteindre des points d'usinage avec les différents angles possibles. En plus, et dans l'objectif de couvrir tout l'espace nécessaire, l'arc de la machine n'est pas fixe et donne la possibilité de le tourner autour d'un point de fixation en bas de l'arc.

## 1.5. Technologies supplémentaires

En plus des RMTs, le paysage des RMSs a été étoffé avec d'autres technologies reconfigurables. Nous décrivons dans ce qui suit les dispositifs de serrage (fixture) reconfigurables, les machines d'inspection reconfigurables et les contrôleurs à architecture ouverte.

### 1.5.1. Dispositif de serrage reconfigurable

L'usinage de produits complexes peut nécessiter le placement du produit sur des dispositifs de serrage conçus sur mesure de sorte que le produit soit déployé dans une position exacte lors d'une opération d'usinage. Puisque la majorité des produits possèdent plusieurs facettes à usiner, une solution conventionnelle consiste à utiliser de multiples dispositifs de serrage pour pouvoir la réalisation des opérations nécessaires en changeant le dispositif de serrage pour chaque surface usinée. Le coût par dispositif de serrage complexe est autour de 30000$ (Koren 2010), et lors du changement de produit, les dispositifs de serrage seraient également changés, ce qui peut engendrer d'importants investissements supplémentaires. Tandis qu'un dispositif de serrage reconfigurable peut accueillir le même produit à usiner dans des orientations différentes. Il offre dans d'autres cas la possibilité d'être utilisé pour plusieurs produits d'une même famille. Dans les deux cas de figure, il garantit d'énormes avantages économiques.

Malgré leurs rôle économique important, les dispositifs de serrage reconfigurables n'ont pas encore formé un domaine de recherche actif auprès des chercheurs du domaine (Shen et al. 2006).

### 1.5.2. Machine d'inspection reconfigurable (RIM)

Les Machines d'inspection reconfigurables (RIM : Reconfigurable Inspection Machine) représentent une nouvelle classe d'équipements d'inspection permettant d'effectuer des mesures « en ligne » sur des pièces usinées. La machine déploie un système d'inspection sans

contact basé sur des capteurs électro-optiques dont les emplacements et le nombre sont reconfigurables en fonction de la pièce à inspecter, contrairement aux machines d'inspection classiques utilisant un contact physique qui inspecte les endroits de contrôle où la précision est nécessaire. Les RIMs sont beaucoup plus rapides que les machines d'inspection classiques, qui peuvent mettre plus d'une heure pour inspecter une seule pièce complexe. Dans ce cas, après une période d'inspection relativement longue, si un problème est détecté, plusieurs pièces auraient été déjà usinées, ce qui génère une augmentation des coûts de rebuts.

Une RIM est composée d'un convoyeur de précision déplaçant la pièce suivant un axe de mouvement précis, au sein d'une séquence de dispositifs électro-optiques tels que des caméras numériques ou des capteurs à base de laser. L'emplacement de chaque capteur peut être modifié pour s'adapter à la géométrie de la pièce inspectée. Les capteurs électro-optiques peuvent être posés sur les deux côtés de la pièce, comme nous pouvons le constaté sur l'exemple illustré par la Figure 1.7.

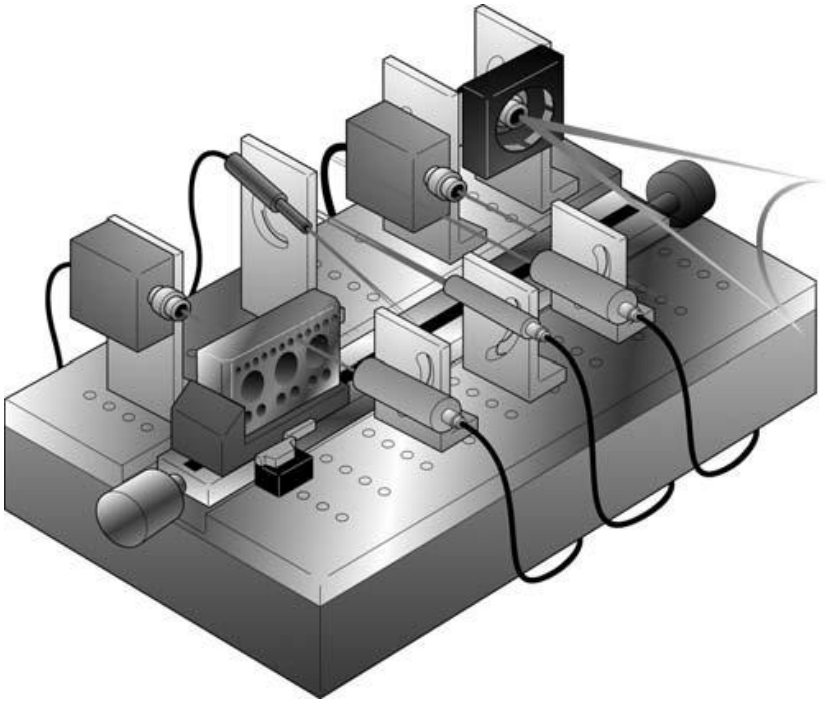

**Figure 1.7. Une machine d'inspection reconfigurable (Koren 2010)**

### 1.5.3. Contrôleur à architecture ouverte

Concevoir des RMTs crée le besoin de concevoir des contrôleurs facilitant la flexibilité de telles machines. Ces contrôleurs sont construits sur des principes d'architecture ouverte qui permet l'ajout et la modification des modules de contrôle pour s'adapter aux besoins.

Les systèmes à architecture ouverte incluent une bibliothèque d'algorithmes de contrôle ( e.g. le contrôle de température, les interpolateurs, etc.) qui peuvent être facilement intégrés dans le logiciel de la machine en cas de besoin (Koren et al. 1996). Les modules de contrôle souhaités sont sélectionnés et intégrés en utilisant un «configurateur de contrôle » pour la machine sélectionnée. Le configurateur vérifie également ses contraintes temps-réel. Après une exécution expérimentale de la procédure de configuration, le code de contrôle est finalement généré.

## 1.6. Conclusion

Dans ce chapitre, nous avons rappelé l'ensemble des définitions et des concepts de base liés aux systèmes manufacturiers reconfigurables (RMS : Reconfigurable Manufacturing Systems). Nous avons commencé par justifier le besoin aux RMS, qui présentent un nouveau paradigme de systèmes de production visant à faire face aux problèmes rencontrés avec les autres système de production classique tels que la rigidité physique et logique, la lenteur, les coûts élevés, et la non réactivité. Ensuite, nous avons introduit les notions de reconfigurabilité et les systèmes reconfigurables, où nous avons résumé les avantages, les caractéristiques et les technologies clés liées aux RMSs.

# Chapitre 2.

## État de l'art et problématique de recherche

*Ce chapitre est dédié principalement à la présentation de notre état de l'art sur les RMSs et les problématiques associées, ainsi qu'au positionnement de notre problématique de recherche. Plus particulièrement, nous nous intéressons aux problèmes abordés et aux approches de résolution proposées. Nous rappelons les notions de processus de fabrication et les différentes techniques existantes permettant d'avoir le/les processus de fabrication dans un RMS. Nous terminons le chapitre par le positionnement de notre travail de recherche par rapport aux travaux rapportés et les objectifs tracés dans le cadre de cette thèse.*

## 2.1. Introduction

Depuis leur apparition, les RMSs ont constitué un domaine de recherche actif sur plusieurs axes, stimulé par la multitude des différences par rapport aux autres systèmes existants. Ainsi, les méthodes et les approches de gestion précédemment développées pour les systèmes manufacturiers classiques nécessitent d'être réexaminées, adaptées ou remplacées pour répondre aux nouveaux défis posés par les RMSs.

## 2.2. Définitions et classification des approches de génération des processus de fabrication dans un RMS

Manufacturing planning, material processing, process engineering et machine routing, sont des expressions utilisées pour désigner la fonction de génération des processus de fabrication (Chang & Wysk 1985). La fonction de génération des processus de fabrication est une étape importante du processus de réalisation des produits (Figure 2.1). Elle est définie comme la fonction de décider quels procédés de fabrication et quelles machines doivent être utilisées pour effectuer les différentes opérations nécessaires à la production d'un composant, et la séquence que le processus doivent suivre (Marri et al. 1998).

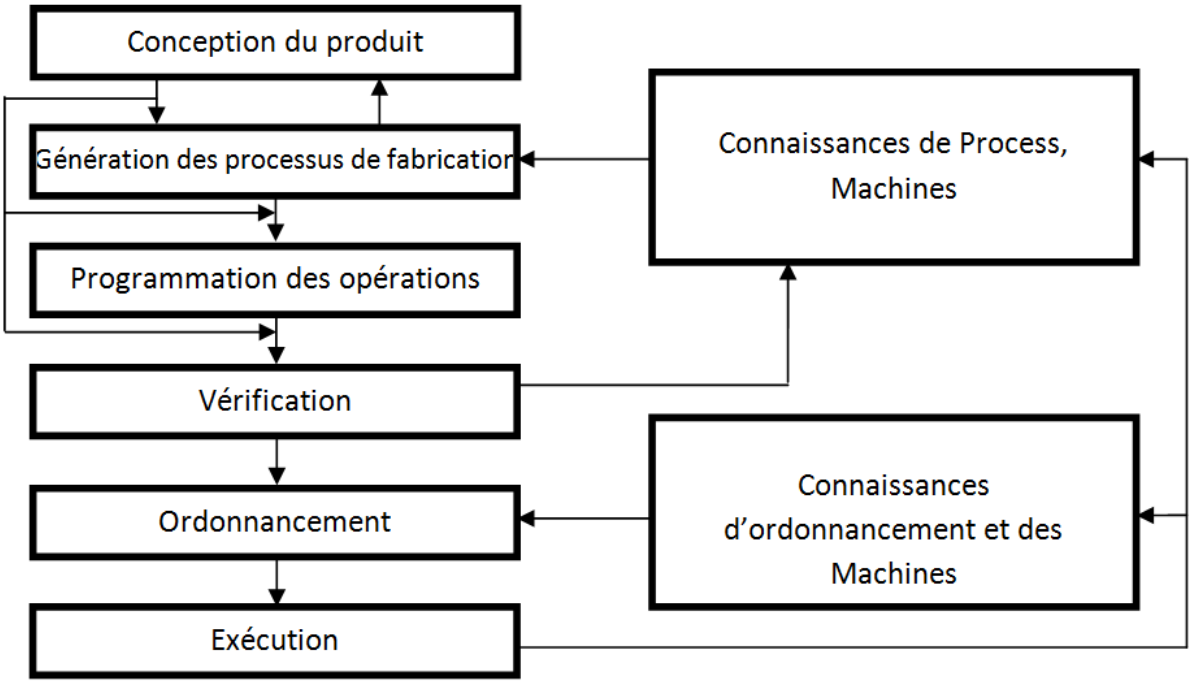


**Figure 2.1. Étapes de réalisation des produits**

Un processus de fabrication est présenté généralement sous forme d'un document (Figure 2.2) qui sera utilisé par l'équipe en charge de la fonction d'ordonnancement et par l'opérateur qui configure/reconfigure la machine.

PROCESS PLAN ACE Inc.

Part No. S0125-F Material: steel 4340Si
Part Name: Housing
Original: S.D.Smart Date: 1/1/89 Changes: _____ Date: _____
Checked: C.S.Good Date: 2/1/89 Approved: T.C. Chang Date: 2/14/89

| No. | Operation Description | Workstation | Setup | Tool | Time (Min) |
|---|---|---|---|---|---|
| 10 | Mill bottom surface 1 | MILL01 | see attach#1 for illustration | Face mill 6 teeth/4" dia | 3 setup 5 machining |
| 20 | Mill top surface | MILL01 | see attach#1 | Face mill 6 teeth/4" dia | 2 setup 6 machining |
| 30 | Drill 4 holes | DRL02 | set on surface1 | twist drill 1/2" dia 2" long | 2 setup 3 machining |

**Figure 2.2. Un exemple de processus de fabrication**

Avec le développement technologique et l'augmentation des capacités des outils informatique, beaucoup d'entreprises ont adopté le concept de génération assistée par ordinateur (CAPP : Computer-aided process planning). En effet, comme la génération des processus de fabrication est une fonction nécessitant beaucoup de temps et d'expériences, sa réalisation manuelle a rapidement basculée vers l'adoption du concept CAPP. Les premières tentatives d'automatiser cette fonction consistaient à construire des systèmes assistés par ordinateur pour la génération des rapports, le stockage et la récupération des plans. Même si ces premières automatisations ont permit de réduire considérablement les coûts de planification (jusqu'à 40% (Chang & Wysk 1985)), il ne s'agissait pas vraiment d'une automatisation des décisions, mais c'était des supports informatiques au responsable des processus de fabrication. Des travaux récents se sont concentrés sur l'élimination de cette intervention humaine sur tout le cycle décisionnel de la fonction de génération des processus de fabrication. Le CAPP présente les avantages suivants :

1) La réduction des compétences requises pour un planificateur,
2) La réduction du temps de planification,
3) La réduction du cout de la planification et de la production,

4) La création des plans cohérent et plus précis,
5) L'augmentation de la production.

### 2.2.1. Classification par rapport au niveau de granularité

ElMaraghy (Elmaraghy 2007) classe les concepts et les approches de génération des processus de fabrication en trois niveaux :

1) Approches multi-domaines : concerne les choix initiaux des domaines et des technologies industrielles les plus adaptées à la production (matériaux utilisés, type de jonction, etc.). Ce niveau n'a pas encore été automatisé vu sa complexité qui peut être mieux gérée avec l'expérience humaine que par des modèles mathématiques.
2) Approche macro : permet de choisir la meilleure séquence des étapes d'usinage et de changement ainsi que les machines effectuant les opérations. Contrairement au DMS, où il est effectué pendant la conception du système, l'approche macro dans les FMSs et RMSs doit être effectuée à chaque fois qu'un produit/une machine ou un de leurs composants change.
3) Approche micro: aborde l'optimisation des paramètres d'usinage (vitesse et profondeur de l'outil, etc.). Cette approche à niveau très détaillé transmit les instructions programmables aux machines.

L'approche macro est la plus concernée par la reconfigurabilité des RMSs, puisque la multitude de configurations du système et de ses machines étend l'espace des process potentiels.

### 2.2.2. Classification par rapport au degré d'adoption du CAPP

Alting et Zhang (Alting & Zhang 1989) ont classifié les systèmes du CAPP en trois classes : systèmes à approche variante, systèmes à approche générative et système à approches semi-générative.

#### *a) Approche variante*

L'approche variante marque le début de l'automatisation de la fonction de génération des processus de fabrication, elle est constituée principalement d'une base de données informatique. Le concept de base dans le fait que les produits ayant des caractéristiques de conception similaires sont groupés en familles de produits. Un processus de fabrication maître

est créé pour chaque famille de produits. Pour chaque nouveau membre de la famille, le plan maître est modifié pour générer le processus de fabrication correspondant.

#### *b) Approche semi générative*

L'approche semi générative représente le pas suivant vers l'automatisation. Le plan maître est soutenu par des algorithmes capables de prendre des décisions sur des produits spécifiques, en définissant des opérations avec leurs paramètres.

#### *c) Approche générative*

Une approche de génération des processus de fabrication entièrement générative génère - d'une façon complète - des processus de fabrication optimaux, en se basant sur des informations sur le produit à réaliser et le système de production. Des systèmes à base de connaissances et des systèmes d'intelligence artificielle fiables sont essentiels pour la réussite de cette approche, car l'intervention humaine est considérablement réduite par rapport aux autres approches, et donc une grande partie de la responsabilité est confiée à la partie automatisée du processus de génération.

## 2.3. Problématiques liées au développement des RMSs

Les RMSs constituent un domaine de recherche actif, visant à développer de nouvelles approches et méthodes qui soient adaptées à la nature reconfigurable des RMSs. Dans la littérature, on trouve des travaux traitant plusieurs aspects, tels que la conception, la génération des processus de fabrication, l'évaluation, l'équilibrage des charges, etc. Nous citons dans la suite les Les travaux récents ont touché l'évaluation des RMSs, tel que Singh et al. (Singh et al. 2007) qui ont utilisé la méthode AHP (Analytical Hierarchy Process) pour développer un système décisionnel multi-période et multicritère pour évaluer le passage d'un système de production classique vers un RMS. On trouve aussi le travail de Gumasta et al. (Gumasta et al. 2011) qui ont développé un index de mesurant la reconfigurabilité d'un RMS en se basant sur les caractéristiques de la reconfiguration (présentées dans la section 1.3.2). Cet index est transmis au décideur pour avoir une vision globale de son RMS.

Parmi les rares travaux étudiant la planification de la production dans les RMSs nous pouvons citer celui de Abbasi et al. (Abbasi & Houshmand 2010), qui ont proposé une méthode pour la planification de la production dans un RMS. Leur méthode génère la séquence de production optimale, la configuration système correspondante ainsi que la taille du lot.

Bratcu et al. (Bratcu et al. n.d.) ont considéré une ligne de transfert reconfigurable équipé de têtes multibroches (bloques). Ils ont considéré le problème d'association des bloques aux stations de travail en minimisant le coût et le temps total tout en respectant les contraintes technologiques pour une famille de produits. Le problème d'équilibrage des lignes de transfert reconfigurable avec des machines parallèles a été traité par Essafi et al. (Essafi et al. 2012), en considérant les temps de changement. L'objectif étant d'associer un ensemble d'opérations et de machines à une séquence de stations de travail en vue de minimiser le coût total de la ligne. Pour ce faire, une heuristique a été développée et validée par des exemples numériques.

D'autre travaux se sont concentrés sur les familles de produits (Tang 2005; Galan et al. 2007), la reconfiguration du système (Ashraf 2006; Dou et al. 2009; Goyal et al. 2011), et le système de contrôle (Zhang et al. 2002; Li et al. 2009). Mais les principales contributions ont concerné respectivement deux axes : la conception des RMSs, et la génération des processus de fabrication. L'état de l'art de ces deux domaines est présenté dans les sections suivantes.

### 2.3.1. Problème de conception des RMSs

La conception des RMSs est axée sur les familles de produits, plutôt que sur un éventail de produits limité uniquement par l'enveloppe géométrique et opérationnelle de la machine, comme c'est le cas pour les FMSs. Plus important encore, un RMS est conçu pour être « reconfiguré », pour grandir et changer. Donc il peut être réactif aux changements du marché. En d'autres termes, le RMS est conçu pour des changements dans sa capacité de production (le nombre de produits qu'il peut produire) et dans sa fonctionnalité (qui fournit la capacité de produire de nouveaux produits) d'une manière qui n'affecte pas à sa robustesse ou sa fiabilité.

Nourelfath et al. (Nourelfath et al. 2002) traitent le problème de conception d'un RMS capable de fonctionner même en cas de panne de l'un de ses composant, avec une certaine dégradation au niveau performance. En effet, deux sous problèmes sont traités par les auteurs : déterminer la configuration qui maximise le taux de production, puis trouver celle qui minimise les coûts de production.

Koren et Shpitalni (Koren & Shpitalni 2010) mettent l'accent sur les principes de base puis définissent les caractéristiques fondamentales et les principes de conception des RMSs. Dans leur travail les auteurs présentent une méthode mathématique pour la conception d'un RMS avec une structure recommandée.

Guan et al. (Guan et al. 2012) proposent une adaptation de la métaheuristiques « électromagnétisme » pour traiter le problème de l’agencement (Layout design) du RMS avec un système de manutention basé sur les AGVs (Automatic Guided Vehicle). Leur modèle pend en charge la circulation des AGVs chargés et vides.

Borisovsky et al. (Borisovsky et al. 2013) ont étudié le problème de conception des lignes de production reconfigurables. La ligne doit satisfaire l’ordre de précédence des opérations, les contraintes d’accessibilité ainsi que d’autres exigences concernant les opérations. La résolution de cette problématique a été réalisée avec les algorithmes génétiques.

### 2.3.2. Problème de génération des processus de fabrication dans le cadre des RMSs

La génération des processus de fabrication vise à déterminer les modalités, sur le plan économique, par lesquelles un produit doit être manufacturé (coût), tout en restant compétitif (délais, qualité, etc.).

Nous trouvons dans la littérature plusieurs études traitant le problème de la génération des processus de fabrication dans les RMSs. Nous citons dans ce qui suit quelques unes de ces études :

Takahashi et al. (Takahashi et al. 2006) ont proposé un modèle stochastique considérant simultanément la planification des commandes et la configuration système correspondante qui maximise les profits. Leur principale idée est d'attribuer plusieurs configurations système à un groupe de produits, en fonction du profit (utiliser des machines lentes et peu coûteuses lorsque il n'y a pas trop de demandes en attente, et vice-versa). Les auteurs proposent un algorithme, basé sur leur modèle stochastique pour définir un seuil de décision. Ce seuil est utilisé ensuite pour la prise de décision sur le choix entre les deux configurations (la configuration rapide ou la configuration lente).

Azab et ElMaraghy (Azab & ElMaraghy 2007) ont considéré un « processus de fabrication reconfigurable ». Ils considèrent que si une nouvelle caractéristique est ajoutée à une pièce existante, ce n’est pas nécessaire de générer un nouveau processus de fabrication qui incorpore cette modification. Ils proposent une méthode de reconfiguration du processus de fabrication, où l’ancien processus de fabrication est modifié pour répondre aux nouvelles exigences. Cela se fait en insérant l'opération qui vient d'être ajoutée quelque part dans le processus de fabrication existant.

Musharavati et al. (Musharavati et al. 2008) ont adapté la métaheuristique « recuit simulé » pour résoudre le problème de la génération de processus de fabrication dans les lignes de production reconfigurables. L'objectif était d'obtenir un processus de fabrication économique quasi-optimal.

Shabaka et al. (Shabaka & ElMaraghy 2008) ont proposé un nouveau modèle basé sur les algorithmes génétiques pour la génération des processus de fabrication avec la prise en compte des coûts de production. Le modèle considère plusieurs paramètres tels que les configurations des machines, l'affectation des opérations aux machines, etc.

Chaube et al. (Chaube et al. 2010) ont adapté les algorithmes génétiques multicritères (NSGA-II) pour générer un processus de fabrication dans le RMS. Le problème est de programmer les différentes parties sur un ensemble de machines, en fonction du type de pièce, le type d'opération, les machines, les configurations des machines, les outils, etc.

Oke et al. (Oke et al. 2011) ont proposé une combinaison de différentes techniques issues de la littérature pour former des processus de fabrication pour l'usinage des moules. Leur technique est basée sur le poids des facteurs de précédence pour former un ordre de priorité d'usinage. Ils ont considéré trois principaux facteurs pour déterminer l'ordre des opérations : le facteur technologique, le facteur géométrique et le facteur économique.

(Musharavati & Hamouda 2012) ont combiné le recuit simulé avec l'exploitation des connaissances et l'architecture parallèle pour améliorer les performances, ceci afin de générer des processus de fabrication dans un RMS. Les auteurs ont réussi à obtenir une amélioration satisfaisante des performances de la métaheuristique en termes de résultats obtenus et de l'effort de calcul.

## 2.4. Notre problématique de recherche

Le système manufacturier RMS représente une classe récente des systèmes manufacturiers. Il nécessite des approches et des méthodes qui prennent en compte de ses spécificités qui le distinguent des autres systèmes de production et qui justifient le développement de nouvelles méthodes de gestion, qui incluent nécessairement la reconfigurabilité dans ses différents aspects.

Le processus de fabrication dépend fortement du système manufacturier concerné, or un RMS est caractérisé par la capacité de ses machines à changer de configuration. Dans cette

perspective, il existe un besoin à des méthodologies qui prennent en compte la reconfigurabilité pour une génération concurrente des processus de fabrication avec la sélection des machines. En outre, il y a la nécessité de représenter d’une manière générique les capacités des machines reconfigurables à effectuer les différentes opérations sur les différents produits.

La génération des processus de fabrication en particulier est fortement liée aux capacités du système concerné, et vu que la capacité des machines change avec chaque configuration dans un RMS, nous présentons dans ce travail une méthode de CAPP (Computer-Aided Process Planning) qui remplisse ce rôle de génération de processus de fabrication dans un RMS, avec une représentation générique de la machine reconfigurable.

L’objectif de cette thèse est d’explorer la capacité des algorithmes évolutionnaires et méthodes approchées multicritères dans la génération des processus de fabrication dans un environnement reconfigurable sans et avec intégration de la fonction d’ordonnancement. En effet, nous cherchons à exploiter au mieux les hauts degrés de reconfigurabilité des machines et des outils pour proposer des processus de fabrication efficaces. Cette capacité de reconfigurabilité permet au processus de fabrication de s’adapter aux contraintes des produits à réaliser et aux changements des marchés.

Pour ce faire, dans le cadre de cette thèse, nous avons abordé trois problèmes de recherche, respectivement

1. La génération des processus de fabrication dans un cas unitaire, où nous proposons une approche multicritère qui génère un processus de fabrication optimal pour un type de produit.

2. La génération des processus de fabrication dans un RMS dans le cas multi-unité, ou plusieurs unités du même produit sont fabriquées en concurrent dans un RMS. Notre approche multicritère consiste à associer à chaque unité un processus de fabrication de tel sorte que les objectifs soient optimisés.

3. L’intégration des fonctions de génération des processus de fabrication avec l’ordonnancement, où nous proposons une heuristique afin de réaliser cette intégration dans un cadre reconfigurable.

## 2.5. Conclusion

Dans ce chapitre, nous avons donné un cadre générale des systèmes manufacturiers reconfigurables, où les différents concepts liés aux RMSs ont été expliqués. En outre, nous avons abordé une comparaison entre les RMSs et les autres systèmes manufacturiers, démontrons ainsi le besoin au RMS comme un système manufacturier prometteur grâce à ça réactivité et sa nature personnalisable d'un coté, et d'un autre coté son coût relativement raisonnable et son rendement comparable à celui des systèmes.

Des technologies telles que les machines reconfigurables (RMT) et les machines d'inspection reconfigurables (RIM) ont été présentées. Ces avancements technologiques et mécaniques constituent un facteur important dans le développement des RMSs, et sont toujours en cours de développement et d'amélioration.

Étant un système manufacturier récent, la littérature à vu l'apparition de plusieurs travaux consacrés aux problématiques liées aux RMSs, notamment la conception et la génération des processus de fabrication, dont les principales contributions ont été résumées dans ce chapitre.

# Chapitre 3.

## Quelques concepts théoriques

*Dans ce chapitre, nous décrivons quelques concepts de base liés à l'optimisation multicritère utilisés dans le cadre de cette thèse. Les méthodes de résolution classiques ainsi que les métaheuristiques les plus performantes (algorithmes génétiques et recuit simulé) sont présentées. Les adaptations des deux métaheuristiques, respectivement NSGA-II et AMOSA, sont détaillées avec leurs principes de base et les différents algorithmes associés. En outre, une méthode d'aide à la décision (TOPSIS) est exposée, qui servira comme outil d'aide à la sélection de la meilleure solution parmi les solutions obtenues par les approches multicritères.*

## 3.1. Introduction

L'optimisation désigne la recherche des valeurs d'un ensemble de variables, qui correspondent au maximum ou au minimum d'une ou de plusieurs fonctions objectifs. L'optimisation trouve de nombreuses applications dans l'ingénierie, la science, le commerce, l'économie, etc., en utilisant des modèles et des méthodes mathématiques et informatiques. Sans l'optimisation de la conception et de l'exploitation, les activités de production et d'ingénierie ne seront pas aussi efficaces telles qu'elles le sont aujourd'hui.

Généralement, l'optimisation consiste à minimiser un seul objectif, et dans ce cas on l'appelle optimisation mono-objectif. Mais dans certains cas, plusieurs objectifs, souvent en concurrence, doivent être optimisés simultanément. Dans ce cas l'optimisation est dite multicritère.

Dans le cadre de cette thèse, nous utilisons des approches multicritère pour la résolution des problématiques traitées, c'est pourquoi nous présentons les prochaines sections quelque concepts de l'optimisation multicritère nécessaire à la compréhension de la suite de manuscrit.

## 3.2. Optimisation multicritère

L'optimisation multicritère, également connue sous le nom « optimisation multi-objectif », se réfère à la recherche de valeurs de variables (de décision) qui correspondent à l'optimum de plus d'une équation objectif

$$F(X) = \{f_1(X), f_2(X), \dots, f_n(X)\} \tag{3.1}$$

Avec les contraintes

$$H(x) = 0$$

$$G(X) \leq 0$$

Dans ce cas, les fonctions qui doivent être optimisées (minimisées ou maximisées) sont l'ensemble des fonctions $F(X)$, où le vecteur $X$ est un ensemble de variables indépendantes. Les fonctions $H(X)$ et $G(X)$ sont les contraintes du modèle.

Les objectifs dans une optimisation multicritère sont souvent contradictoires, ce qui signifie que l'atteinte de l'optimum pour un objectif exige un compromis sur un ou plusieurs autres objectifs. En d'autres termes, lorsqu'une fonction est optimisée, les résultats des autres

fonctions pourraient se dégrader. En outre, contrairement à l'optimisation mono-objectif qui donne une solution unique, il y aura beaucoup de solutions optimales (front de Pareto optimal) pour un problème multicritère ; l'exception est lorsque les objectifs ne sont pas contradictoires. Dans ce cas seulement, une solution unique est attendue.

Les solutions Pareto optimales sont aussi appelées « solutions non dominées ». Dans un problème de minimisation, une solution $x^*$ est dite non dominée si quelque soit $x \in S$ (l'espace des solutions possibles), $f_i(x^*) \leq f_i(x)$, et il existe au moins un $j \in n$ tel que $f_j(x^*) < f_j(x)$ (voir Figure 3.1).

Par conséquent, l'optimisation multicritère implique des méthodes spéciales qui considèrent plus d'un objectif et analysent les résultats obtenus.

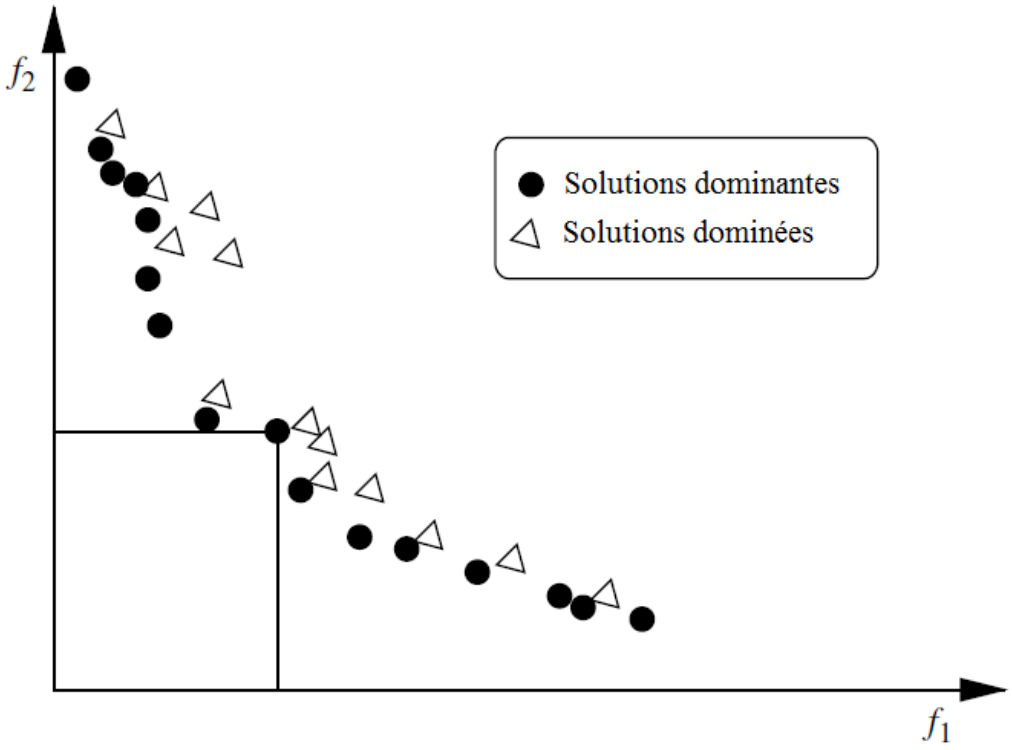


**Figure 3.1.Solutions non dominées dans l'espace de solutions**

La résolution d'un problème multicritère est très complexe, car cette résolution exige que les "optimiseurs" fournissent comme résultat final un ensemble de solutions optimales au lieu de fournir une solution unique et optimale, alors que ces "optimiseurs" n'existent pas (Donoso & Fabregat 2007). Ainsi, les méthodes proposées sont basées sur le contexte mono-objectif connu par les spécialistes, telles que la méthode de Benson, ε-Constraint, la somme pondérée et métriques pondérées.

## 3.3. Méthodes de résolution multicritères traditionnelles

### 3.3.1. Somme pondérée (Weighted Sum)

Cette méthode consiste à créer un modèle mono-objectif en pondérant les *n* fonctions objectifs par l'association de poids à chaque fonction. En sommant les fonctions pondérées, le modèle multicritère est établi comme suit :

$$max\ (ou\ min)\ F`(X) = \sum_{i=1}^{n} w_i f_i(X)$$

**( 3.2 )**

Avec comme contrainte supplémentaire $\sum_{i=1}^{n} w_i = 1$.

Pour déterminer le front Pareto optimal, les poids $w_i$ sont variés et chaque nouvelle solution trouvée appartient à l'ensemble des solutions optimales. Dans le cas où l'espace de recherche est convexe, chaque solution optimale possède une probabilité non nulle d'être trouvée par cette technique. Dans le cas de non convexité de l'espace de recherche, cette technique risque de manquer des solutions optimales. Un autre inconvénient de cette méthode concerne la plage des valeurs qui peuvent être obtenues par les fonctions optimisées. Le problème se pose lorsque les gammes de fonctions ($a \leq f_1 \leq b$, $c \leq f_2 \leq d$) ont des grandeurs différentes, et en particulier lorsque ces différences sont très grandes. Si cela se produit, la fonction ayant les valeurs les plus élevées va prédominer dans le résultat. Par exemple, si $0 < f_1 < 1$ et $100 < f_2 < 1000$, toutes les solutions seraient influencés par la fonction $f_2$. Une normalisation des fonctions $f_i$ semble une solution à cet inconvénient, puisqu'elle réduit l'écart entre leurs valeurs, mais ceci rajoute une modification supplémentaire au modèle mathématique (Donoso & Fabregat 2007).

### 3.3.2. ε-Constraint

Cette méthode consiste à créer un modèle mono-objectif dans lequel seulement une des fonctions sera optimisée, et les fonctions restantes deviendront des contraintes pour le modèle. Le modèle multicritère présenté dans l'équation ( 3.2 ) peut être réécrit comme suit :

$$min(ou\ max) f_i(X)$$

**( 3.3 )**

Avec la contrainte supplémentaire suivante :

$$f_k(X) \leq \varepsilon_k \,, \qquad \text{avec } k = 1, \dots, n \text{ et } k \neq i$$

Dans ce cas, seule la fonction $f_i(X)$ est optimisée, et les autres $(n-1)$ fonctions deviendront des contraintes. L'idée principale de cette technique d'optimisation est de varier les valeurs des $\varepsilon$ de chaque fonction pour obtenir plusieurs point optimaux pour la fonction $f_i(X)$.

On peut aussi changer la fonction à optimiser $f_i(X)$ par une fonction $f_j(X)$, tout en incluant $f_i(X)$ dans les contraintes, et cela pour trouver un nouveau point sur le front de Pareto optimal. Grâce à cette méthode, on peut résoudre des problèmes avec des espaces de solutions convexes et non convexes. L'inconvénient de cette méthode est l'obligation de bien connaître et bien maîtriser le problème et les valeurs de ε pour que les résultats obtenus soient de véritables solutions.

### 3.3.3. Méthode de distance à un objectif de référence

Cette méthode, comme la méthode de la somme pondérée, permet de transformer un problème d'optimisation multicritère en un problème mono-objectif. Cette technique utilise la distance à un point de référence pour réécrire le modèle multicritère ( 3.2 ) comme suit:

$$F`(X) = \left[ \sum_{i=1}^{n} |Z_i - f_i(X)|^r \right]^{\frac{1}{r}} \tag{3.4}$$

Avec une contrainte supplémentaire $r > 1$.

Dans cette méthode, la valeur de $Z_i$ doit être définie pour chaque fonction. Un autre paramètre à définir est $r$, qui spécifie quelle fonction de distance est utilisée. Par exemple si $r = 1$ alors le problème sera équivalent à la méthode de la somme pondérée, et si $r = 2$ alors c'est la distance euclidienne qui sera utilisée.

Il peut être impossible de trouver des solutions optimales pour certaines valeurs de $r$ lorsque l'espace des solutions est non convexe. Aussi, il est nécessaire de donner une bonne valeur à $Z$ le point de référence. Un mauvais point $Z$ peut entraîner des divergences au lieu de converger vers l'optimale (Donoso & Fabregat 2007). Pour cette raison, la connaissance de l'espace de solution est nécessaire pour appliquer cette technique.

D’autres méthodes sont dérivées de la « méthode de distance » telles que la méthode des métriques pondérées et la méthode de Benson, mais qui héritent cependant les mêmes inconvénients.

## 3.4. Métaheuristiques

Les Métaheuristiques sont des stratégies algorithmiques de haut niveau utilisées pour guider les autres heuristiques ou des algorithmes dans leur recherche dans l'espace des solutions possibles de la valeur optimale (dans le cas mono-objectif) ou de l'ensemble des valeurs optimales (dans le cas multicritère).

Traditionnellement, les métaheuristiques représentent de bonnes techniques pour la résolution des problèmes d'optimisation où la convergence vers un optimum doit passer par une analyse combinatoire des solutions. Généralement, la recherche de nouvelles solutions encore inconnues se fait à travers des solutions déjà connues.

Par rapport aux techniques classiques, les métaheuristiques présentent plusieurs avantages :

1. Toute solution appartenant à l’espace de solutions réalisables a une probabilité supérieure à 0 d’être trouvée. Dans ce cas il n’est pas nécessaire que l’espace de recherche soit convexe ;
2. Les métaheuristiques arrivent à éviter le piège du minimum local grâce à leurs stratégies qui n’élimine pas automatiquement les solutions jugées comme étant « mauvaises ». Les « mauvaises » solutions pourront par la suite mener la recherche vers un minimum global.
3. Dans le cas des algorithmes d’optimisation avec des temps de calcul polynomial, les métaheuristiques arrivent à donner des solutions satisfaisantes dans un temps polynomial, et ce grâce a leur nature probabiliste.

Durant les dernières années, l’utilisation des métaheuristiques pour résoudre des problèmes multicritères est devenue un domaine de recherche très actif. La majorité des études à propos des métaheuristiques pour l’optimisation multicritère se concentrent sur les métaheuristiques basées sur la population (Population-Based Metaheuristics), tels que les algorithmes génétiques (Deb 2001). Cependant, des adaptations récentes des métaheuristiques, comme dans les travaux de Bandyopadhyay et al. (Bandyopadhyay et al. 2008), ont prouvé une efficacité remarquable dans la résolution des problèmes multicritères.

Jones et al. (Jones et al. 2002), dans une étude sur l'utilisation des métaheuristiques multicritères, ont démontré que trois métaheuristiques dominent la littérature. Elles sont respectivement les algorithmes génétiques, le recuit simulé et la recherche tabou, avec une nette popularité des algorithmes génétiques (Figure 3.2). Les deux premières métaheuristiques sont présentées en détails dans les sections suivantes.

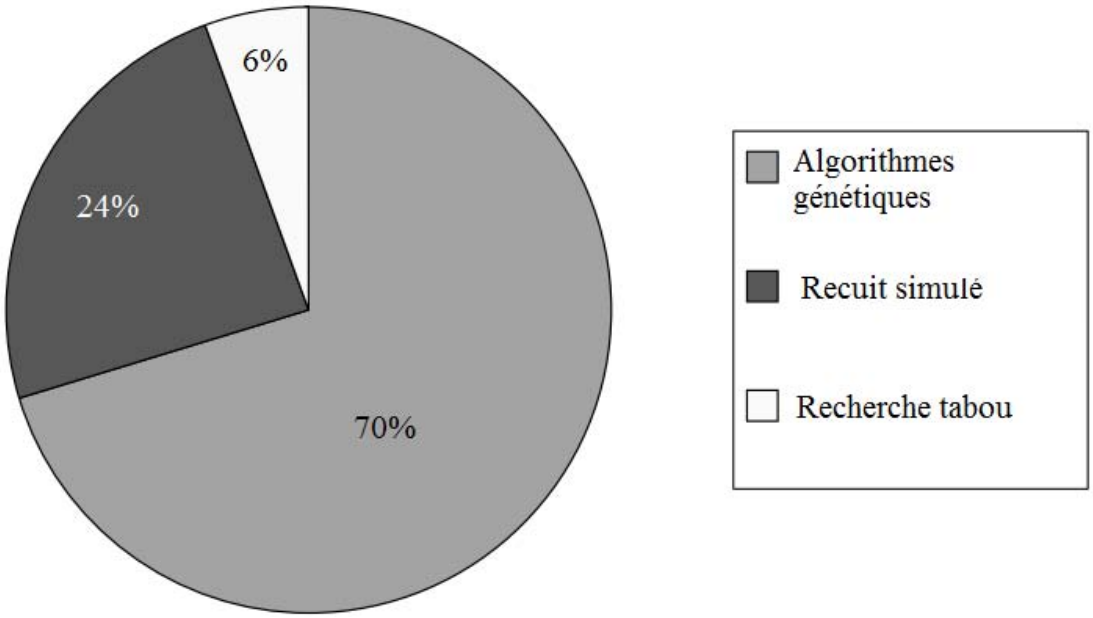


**Figure 3.2. Décomposition des travaux utilisant les métaheuristiques pour des problèmes multicritères (Jones et al. 2002)**

### 3.4.1. Les algorithmes génétiques

Les algorithmes génétiques constituent une technique de recherche et d'optimisation basée sur les principes génétiques et la théorie de la sélection naturelle développée par Charles Darwin. Cette technique a été développée pour la première fois par John Holland (Holland 1975) et popularisée par David Goldberg (David 1989).

Les algorithmes génétiques sont un des algorithmes à base de population évoluant d'une génération à une autre tout en cherchant des solutions au problème donné. Ce principe permet d'atteindre plusieurs solutions optimales ou proches des solutions optimales grâce à l'évolution de tout un ensemble d'individus.

Les algorithmes génétiques ont une popularité de plus en plus grandissante grâce aux avantages qu'ils offrent. Sans être exhaustif, nous citons les principaux avantages résumés dans (Haupt & Haupt 2004) :

- L'optimisation des variables discrètes et continues,
- La possibilité de traiter un grand nombre de variables,
- L'adaptation parfaite au calcul parallèle,
- Le fonctionnement avec des données générées numériquement, des données expérimentales ou des fonctions analytiques.

Les algorithmes génétiques sont basés sur des principes biologiques, tels que les chromosomes, les gènes et le phénomène naturel de l'évolution, etc.. Selon la théorie de Charles Darwin, les gènes conservés au fil de temps au sein d'une population donnée sont ceux qui sont le plus adaptés aux besoins de l'espèce vis à vis de son environnement. Ce même principe est utilisé pour faire évoluer les solutions les plus adaptées en termes d'optimalité, en utilisant les notions de sélection naturelle jusqu'à l'obtention d'une solution optimale ou au moins satisfaisante.

### *a) Principe des algorithmes génétiques*

Afin de permettre la résolution de problèmes, les algorithmes génétiques se basent sur les différents principes décrits ci-dessus. De manière globale, L'algorithme génétique entame sa recherche avec une population de base (population initiale) qui se compose de chaînes de caractères correspondant chacune à un chromosome, qui représente une solution du problème. Le contenu de cette population initiale – qui dépend du codage utilisé – est en générale généré aléatoirement. A chacune des solutions, une valeur fitness est attribuée. Elle correspond à son adaptation au problème étudié. Ensuite, on effectue une **sélection** au sein de cette population afin d'effectuer des opérations génétiques de **croisement** et de **mutation** pour ainsi faire évoluer la population en construisant la population suivante. Les termes « sélection », « croisement » et « mutation » sont expliqués dans ci-dessous :

- La sélection : pour faire évoluer un algorithme génétique, les individus d'une population, sur lesquels des opérations génétiques (le croisement et la mutation) seront effectuées, doivent être sélectionnés. Il existe plusieurs techniques de sélection, les principales utilisées sont (Deb 1999) : la **sélection par rang** qui consiste à toujours choisir les individus possédant les meilleurs scores d'adaptation (fitness), le hasard n'entre donc pas dans ce mode de sélection. La technique de la **roulette de fortune**, où la probabilité de chaque individu à être sélectionné est proportionnelle à sa fitness. La **sélection par tournoi** utilise la sélection proportionnelle sur des paires d'individus, puis choisit parmi ces paires l'individu qui a le meilleur score d'adaptation. La

sélection **uniforme** se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation. Chaque individu a donc une probabilité $1/P$ d'être sélectionné, où $P$ est le nombre total d'individus de la population.

- Le croisement : Lors de cette opération, deux chromosomes s'échangent des parties de leurs chaînes, pour donner de nouveaux chromosomes (Figure 3.3). Ces croisements peuvent être simples (sur un seul point de croisement) ou multiples (sur plusieurs points de croisement).

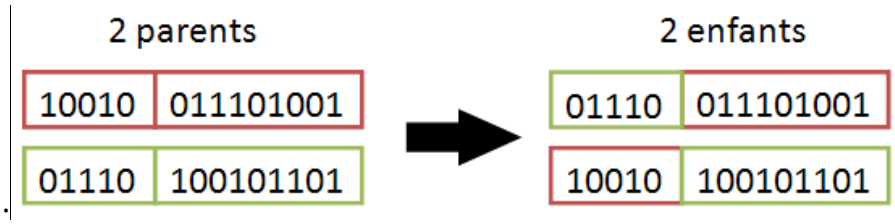


**Figure 3.3. Un croisement simple**

- La mutation : il s'agit de substituer un gène avec un autre au sein d'un chromosome donné, et cela d'une manière aléatoire (Figure 3.4). De la même façon que pour le croisement, on définit ici un taux de mutation lors des changements de population. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à conserver le principe de sélection et d'évolution naturelle et ne pas transformer l'algorithme en une recherche aléatoire. La mutation sert à éviter une convergence prématurée de l'algorithme.

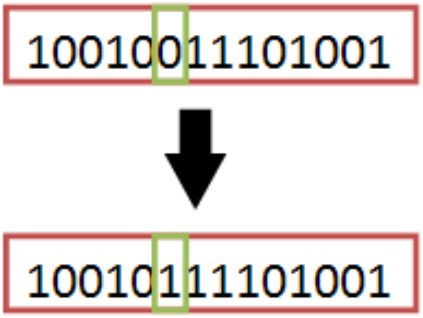


**Figure 3.4.Une mutation pour un codage binaire**

L'Algorithme 1 présente le fonctionnement général d'un algorithme génétique. Le processus des opérations génétiques (sélection, croisement, mutation) est réitéré un grand nombre de fois de manière à imiter le principe d'évolution, qui ne prend son sens que sur un nombre important de générations. On peut arrêter le processus au bout d'un nombre arbitraire de générations ou lorsqu'une solution possède une valeure suffisamment satisfaisante, ou dans

d'autres cas, si l'évolution n'introduit aucune amélioration pendant un nombre important d'itérations. Ces trois possibilités sont appelées critères d'arrêt de l'algorithme génétique.

**Algorithme 1. Algorithmes génétiques**

```
Générer la population initiale
repeter nombre_iterations fois
    pour chaque individu dans population
        calculer_Fitness(individu)
    EndPour
    Selectionner(individus)
    croisement (individus)
    Mutation(individus)
finRepeter
retourner meilleur individu de population
```

Plusieurs techniques ont été développées pour optimiser le fonctionnement de base des algorithmes génétiques. Parmi lesquelles nous pouvons citer - à titre d'exemple - la technique insérant à chaque génération un certain nombre d'individus générés aléatoirement (non issus de la descendance de la génération précédente mais générés aléatoirement). Cette technique vise à éviter une convergence vers des optima locaux.

### *b) Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II)*

La littérature dans le domaine des algorithmes génétiques multicritères (MOGA) est très riche. Nous nous contentons par la suite de présenter le NSGA-II (Deb et al. 2002), un algorithme devenu un vrai standard des MOGA durant les dernières années.

Le NSGA-II est une amélioration du NSGA (Srinivas & Deb 1995) dotée de nouveaux mécanismes qui ont prouvé leur efficacité sur des benchmarks de la littérature (Deb et al. 2002).

L'Algorithme 2 présente le fonctionnement du NSGA-II, qui peut être expliqué comme suit :

- Initialement, une population parente $\mathbf{parentPop_0}$ de taille $N$ est aléatoirement créée. La population est triée en fonction de la non-domination. Chaque solution se voit attribuée une fitness égale à son niveau de non-domination (1 étant le meilleur niveau). Des opérateurs de sélection, croisement et mutation sont utilisés pour créer une population enfante $\mathbf{childPop_0}$ de taille $N$. Ensuite, l'algorithme passe à sa boucle principale

**Algorithme 2. Algorithme du NSGA-II**

```
Générer (parentPop)
enfantPop = créer_nouvelle_population(parentPop)
tant que (t < nombreIteration)
    /*combiner population parent et enfant */
    newPop = parentPop_t ∪ enfantPop_t
    T= fast_nondominated_sort (newPop)
    parentPop_t+1 = ∅
        /*préparer la population pour la prochaine itération*/
        tant que (|parentPop_t+1| < N)
            crowding_distance_assignment(T)
            parentPop_t+1 = parentPop_t+1 + T
        fin tant que
    trier(parentPop)
    parentPop_t+1 = premiers N éléments de parentPop_t+1
     /* utiliser la sélection, croisement et mutation pour créer
     enfantPop */
    enfantPop_t+1 = créer_nouvelle_population(parentPop_t+1)
    t = t + 1
fin tant que
```

- Une population combinée **newPop = parentPop$_t$ ∪ childPop$_t$** est formée. La population **newPop** sera de taille $2N$.

- La population **newPop** est triée en fonction de la non-domination en utilisant la fonction **fast_nondominated_sort()**. Cette fonction est définie dans Algorithme 3.

- Une nouvelle population parente **parentPop$_{t+1}$** est formée en ajoutant des solutions du premier front jusqu'à ce que sa taille dépasse $N$

- les solutions du dernier front accepté sont triées en fonction de l'opérateur de densité $\geq_n$ (expliqué dans les paragraphes suivants), et les N premiers points sont pris. Ainsi est construite la population **parentPop$_{t+1}$** de taille $N$. Cette population est ensuite utilisée pour la sélection, croisement et mutation pour créer une nouvelle population **childPop$_{t+1}$** de taille $N$ également.

**Algorithme 3. la fonction fast_nondominated_sort()**

```
Pour chaque p ∈ Population
    Pour chaque q ∈ Population
          Si (p domine q) Alors
                S_p = S_p ∪ {q}
          Sinon
                Si (q domine p) Alors n_p = n_p + 1
        FinSi
    FinPour
    Si n_p = 0 Alors F_1 = F_1 ∪ {p}
FinPour
i = 1
Tant que F_i ≠ ∅
     H = ∅
     Pour chaque p ∈ F_i
          Pour chaque h q ∈ S_p
                n_q = n_q - 1
                Si n_q = 0 Alors H = H ∪ {q}
        FinPour
    FinPour
     i = i + 1
     F_i = H
FinTantQue
```

L'opérateur de densité (*crowded comparison operator*) « $\geq_n$ » guide le processus de sélection dans l'algorithme vers une répartition plus ou moins uniforme des solutions sur le front Pareto optimal. Pour cette raison, chaque individu $i$ de la population possède deux attributs :

1) Classement de non-domination ($i_{rank}$)

2) Crowding-distance locale ($i_{distance}$)

L'opérateur $\geq_n$ est défini comme suit :

$$i \geq_n j \text{ si et seulement si } (i_{rank} < j_{rank}) \vee \big((i_{rank} = j_{rank}) \wedge (i_{distance} < j_{distance})\big)$$

Cela signifie que, entre deux solutions avec des niveaux de non-domination, la solution avec un $i_{rank}$ bas est préférée. Dans le cas où les deux solutions sont du même rang, alors la solution qui possède le voisinage le moins dense est préférée.

Pour obtenir une estimation de la densité des solutions entourant un point particulier $i$ de la population, on calcule la distance moyenne entre deux points de chaque côté de $i$, avec chacun des objectifs. Cette quantité $i_{distance}$ représente une estimation de la taille du plus grand rectangle enfermant le point $i$ sans y inclure d'autres points de la population (Figure 3.5).

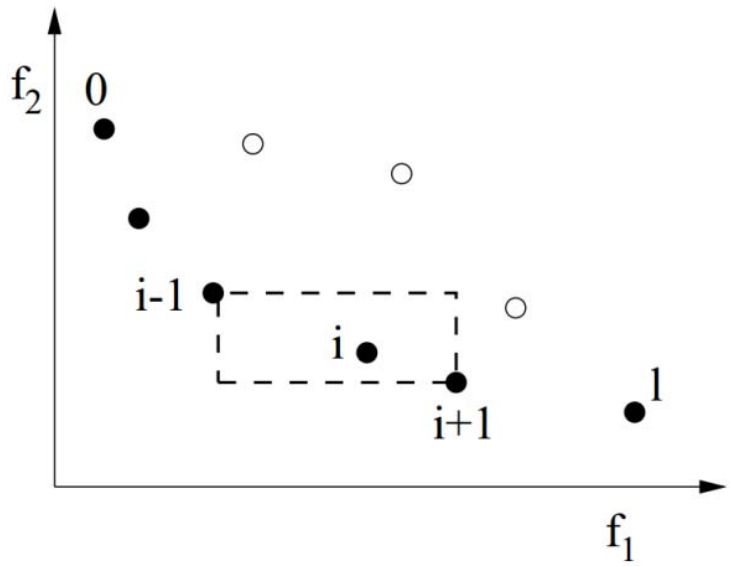


**Figure 3.5. Crowding distance**

Le Crowding-Distance de la $i^{\text{ième}}$ solution sur le front (indiqué par des cercles pleins sur la Figure 3.5) est le côté moyen du cuboïde (représenté par un cadre en pointillés). L'algorithme suivant est utilisé pour calculer le Crowding-Distance de chaque point de front :

**Algorithme 4. la fonction crowding-distance-assignment(I)**

```
nb = |I| /*nombre de solutions sur le front I */
Pour chaque i∈I
    I[i]distance=0
FinPour
Pour chaque objectif m
    I = trier(I,m) /*trie selon les valeurs m*/
    I[1]distance=∞
    I[nb]distance=∞/*forcer la prise des extrémités du front*/
    Pour i = 2 to (nb-1)
        I[i]distance=I[i]distance + (I[i+1].m - I[i-1].m)
    FinPour
FinPour
```

Ici, **I[*i*].m** réfère à la $m^{\text{ième}}$ valeur de la fonction objectif du $i^{\text{ème}}$ individu de l'ensemble **I**. `I[i+1].m` et `I[i-1].m` sont les voisin de `I[i].m`.

### 3.4.2. Recuit simulé

L'application du recuit simulé (*Simulated Annealing*) à des problèmes d'optimisation émerge des travaux de S. Kirkpatrick et al. (Kirkpatrick et al. 1983). Dans les années 1980, le recuit simulé a eu un impact majeur sur le domaine de la recherche heuristique pour sa simplicité et son efficacité dans la résolution de problèmes d'optimisation combinatoire.

Le recuit simulé est inspirée de la métallurgie. Le principe dans la métallurgie est d'alterner des cycles de refroidissement lent et de réchauffage (recuit) qui ont pour effet la minimisation de l'énergie du matériau. La méthode vient du constat que le refroidissement naturel de certains métaux ne permet pas aux atomes de se placer dans la configuration la plus solide. La configuration la plus stable est atteinte en contrôlant le refroidissement et en le ralentissant par un apport de chaleur externe.

L'algorithme du recuit simulé est stochastique, ce qui permet, sous certaines conditions, la dégradation d'une solution. Cette tolérance à la dégradation des solutions existe d'une manière ou d'une autre dans toutes les métaheuristiques, son objectif est d'éviter les optima locaux et ainsi de retarder la convergence.

#### *a) Principe du recuit simulé*

À partir d'une solution initiale, le recuit simulé se déroule en plusieurs itérations. A chaque itération, un voisin aléatoire est généré. Les déplacements de la solution qui améliorent la fonction de coût sont toujours acceptés. Dans le cas contraire, le voisin est accepté avec une probabilité donnée qui dépend de la température actuelle et la quantité de dégradation $\Delta E$ de la fonction objectif. Ces deux notions sont définies comme suit :

- $\Delta E$ représente la différence de la valeur de la fonction objectif entre la solution actuelle et la solution voisine générée. Comme l'algorithme progresse, la probabilité qu'un tel déplacement est accepté diminue. Cette probabilité est représentée généralement par une distribution de Boltzmann :

$$P(\Delta E, T) = e^{-\frac{\Delta E}{T}} \tag{3.5}$$

- Le paramètre de contrôle $T$, appelé température, est utilisé afin de déterminer la probabilité d'accepter des solutions dégradées. À un certain niveau de température, de nombreux essais sont explorés. Une fois que l'état d'équilibre est atteint, la température diminue

progressivement selon un planning de refroidissement de sorte que peu de solutions dégradées sont acceptées à la fin de la recherche. L'Algorithme 5 décrit un pseudo code du recuit simulé.

**Algorithme 5. Pseudo code de du recuit simulé**

```
s = s0 ; /* solution initiale*/
T = Tmax ; /* temperature initiale*/
Répéter
     Répéter /* à une température fixe */
          A partir de s, Générer un voisin aléatoire s' ;
          ΔE = f (s') − f (s) ;
          Si ΔE ≤ 0 Alors
            s = s' /*Accepter la nouvelle solution */
          Sinon
            Accepter s'avec une probabilité e^(−ΔE/T);
          FinSi
     Jusqu'à condition d'équilibre
T = g(T) ; /* mise à jour de la température (refroidissement) */
Jusqu'à Critère d'arrêt
Output: s
```

Nous trouvons dans la littérature une récente adaptation du recuit simulé, appelée AMOSA (Archived Multi-objective Simulated Annealing), montrant de bonnes performances par rapport aux autres méthodes existantes (Bandyopadhyay et al. 2008)..

### c) *Archived Multi-objective Simulated Annealing (AMOSA)*

La métaheuristique AMOSA a été développée par Bandyopadhyay et al. (Bandyopadhyay et al. 2008). Elle intègre la notion d'**Archive** où les solutions non-dominées trouvées jusqu'ici sont stockées. La taille de l'archive est gardée bornée car seul un nombre limité de solutions Pareto optimales plus ou moins régulièrement distribuées est nécessaire.

L'algorithme commence par l'initialisation d'un nombre de solutions initiales dans l'archive. Une solution est acceptée dans l'archive si et seulement si elle n'est dominée par aucune autre solution de l'archive.

AMOSA utilise le concept de degré de domination (*amount of domination*) dans le calcul de la probabilité d'acceptation d'une solution dégradante. Ce degré de domination est défini comme suit : soient deux solutions $a$ et $b$, le degré de domination $\Delta dom_{a,b}$ est calculé par la fonction suivante :

$$\Delta dom_{a,b} = \prod_{i=1, f_i(a) \neq f_i(b)}^{M} \frac{|f_i(a) - f_i(b)|}{R_i} \quad (3.6)$$

Où $M$ est le nombre d'objectifs et $R_i$ représente la plage de variation du $i^e$ objectif. Le concept du $\Delta dom_{a,b}$ est illustré sur la Figure 3.6 pour un cas à deux objectifs.

**Algorithme 6. AMOSA**

```
Initialiser(archive, variables, current_sol)
temp=temperature_max
Tant que (température > temperature_min)
   pour (i=0; i< nombre_iterations; i++)
    new_sol=perturber(current_sol).
    Si (current_sol domine new_sol) Alors /* Cas 1*/
        new_sol = current_sol avec une probabilité
    FinSi /*fin du cas 1*/
    Si (current_sol & new_sol sont non dominées l'une vis-à-vis
l'autre)/*Cas 2*/
        Si (new_sol est dominée par by k solutions de l'Archive)
            new_sol = current_sol avec une probabilité
        EndIF
        Si (no point in the archive dominates new_sol)
            new_sol = current_sol
            ajouter new_sol à l'Archive
            enlever de l'Archive les points dominés par new_sol
        FinSI
    FinSi /*fin du cas 2*/
    Si (new_sol domine current_sol) /* Cas 3 */
        Si (new_sol est dominée par k solutions de l'Archive)
            current_sol = sélectionner de l'Archive avec une
                     probabilité
        Sinon new_sol = current_sol.
        FinSi
        Si (aucune solution de l'Archive ne domine new_pt)
            new_sol = current_sol
            ajouter new_sol à l'Archive.
        FinSi
        Si (new_sol domine k solutions de l'Archive )
            Enlever les k solutions dominées de l'Archive.
        FinSi
    FinSi /*Fin du cas 3*/
   FinPour
   temp = alpha * temp /* mise à jour de la température*/
FinTantQue
Retourner Archive
```

Au début de l'algorithme, un des points appelé current_sol est aléatoirement sélectionné de l'archive initial, à une température **temp=temperature_max**. la solution courante

**current_sol** est ensuite perturbée pour générer une nouvelle solution **new_sol**. L'état de domination de la solution **new_sol** est vérifié par rapport à la solution **current_sol** et les solutions de l'archive. En se basant sur l'état de domination entre **current_sol** et **new_sol**, trois cas peuvent être distingués :

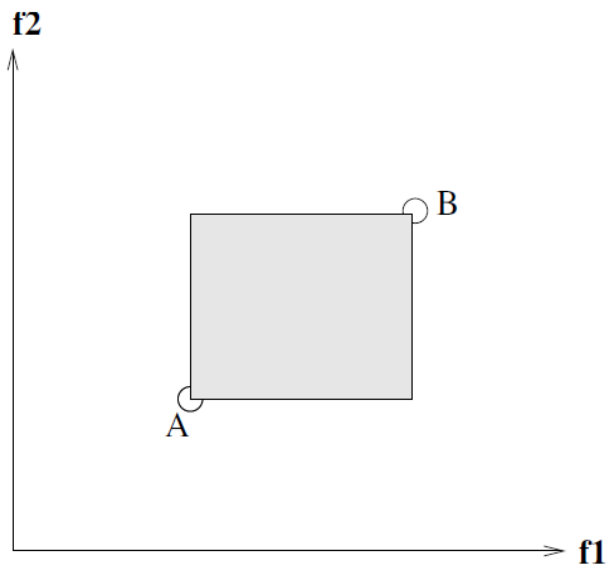


**Figure 3.6. Degré de domination entre deux solutions A et B**

1) **current_sol** domine **new_sol**, et $k$ solutions dans l'archive domine **new_sol** ($\boldsymbol{k \geq 0}$) :

   Dans ce cas, **new_sol** sera acceptée avec une probabilité égale à :

$$Probabilité = \frac{1}{1 + e^{(\Delta dom_{avg} \times temp)}} \tag{3.7}$$

   Avec $\Delta dom_{avg} = \frac{\left(\left(\sum_{i=1}^{k} \Delta dom_{i,new_pt}\right) + \Delta dom_{current_sol,new_sol}\right)}{k+1}$

   $\Delta dom_{avg}$ représente le degré de domination moyen du point **new_sol** par $k+1$ solutions qui sont respectivement la solution **current_sol** et les $k$ solutions dans l'archive.

   D'après l'équation ( 3.7 ), les deux paramètres qui contrôlent la valeur de la probabilité sont $\Delta dom_{avg}$ et la température $temp$. Plus ces deux valeurs sont importantes, moins est la probabilité d'accepter la solution dominée **new_sol**. Ceci est expliqué par le principe du paramètre $temp$ et sa diminution tout au long du déroulement des itérations de l'algorithme. Ainsi, plus on avance dans la recherche, la

probabilité d'accepter des solutions dominées diminue également. Par ailleurs, le paramètre $\Delta dom_{avg}$ prend une valeur importante si la qualité de solution dominée **new_sol** est très mauvaise par rapport aux solutions non dominées de l'archive, et par conséquent la probabilité d'accepter cette solution diminue.

2) **current_sol** et **new_sol** sont mutuellement non dominées :

   Dans ce cas, si **new_sol** n'est pas dominée également par aucune autre solution de l'archive, alors **new_sol** appartient au front des solutions. Par conséquent, **new_sol** est ajoutée à l'archive, et affectée à **current_sol**. Les solutions de l'archive dominées par **new_sol** seront évidemment éliminées.

   Si, par contre **new_sol** est dominée par k solutions de l'archive ($k > 0$), alors on est dans un cas qui ressemble au cas (1), new_sol sera acceptée avec une probabilité calculée selon l'équation ( 3.7 ) précédemment présentée, sauf que, dans ce cas, $\Delta dom_{avg}$ est calculée différemment :

$$\Delta dom_{avg} = \frac{\left(\sum_{i=1}^{k} \Delta dom_{i,new_{sol}}\right)}{k} \tag{3.8}$$

3) **new_sol** domine **current_sol** : dans ce cas, deux possibilités se présentent :

   a. **new_sol** est dominée par $k$ solutions de l'archive ($k \geq 1$). Ce cas n'est possible que si une solution dominée a été acceptée et affecté à **current_sol**. Ici, on prend le minimum des $\Delta dom_{i,new_sol}$, où $i = 1..k$. c'est-à-dire la solution $i^*$ dans l'archive qui possède le minimum de degré de dominance par rapport à **new_sol**. La solution $i^*$ de l'archive est sélectionnée en tant que **current_sol** avec une probabilité $= \frac{1}{1=e^{(-\Delta dom_{i^*})}}$.

   b. **new_sol** n'est dominée par aucune solution de l'archive. Dans cette situation **new_sol** appartient au front de l'archive elle est y ajoutée, et **current_sol** = **new_sol**. Bien évidemment, les solutions de l'archive dominées par **new_sol** seront évidemment éliminées.

Le processus est répété **number_iterations** fois pour chaque température **temp**. La température est réduite à **α × temp**, en utilisant la vitesse de refroidissement de **α** jusqu'à ce que la température minimale **temperature_min** soit atteinte et le processus de recherche s'arrête. Enfin, l'archive contient les solutions non dominées finales, ou le front Pareto optimal.

Une des caractéristiques majeures d'AMOSA est qu'une nouvelle solution dégradée (c'est-à-dire dominée une des solution de l'archive) peut être sélectionnée. Dans la plupart des autres métaheuristiques multicritère, si un choix doit être fait entre deux solutions $x$ et $y$, et si $x$ domine $y$, alors $x$ est toujours sélectionnée. En revanche, AMOSA reproduit le comportement de base des métaheuristiques mono-objectif, qui consiste à avancer même si la solution se dégrade, et ceci en vue d'éviter le piège du minimum local.

## 3.5. Méthodes de sélection des solutions multicritère

Une des questions fondamentales de l'optimisation multicritère est l'interaction entre la résolution du problème et les décideurs. En effet, l'optimisation multicritère fournit un Pareto de solutions au décideur, et c'est à lui de sélectionner, parmi celles-ci, la solution à adopter. Cette sélection est basée sur des informations supplémentaires que le décideur doit spécifier. L'interaction entre le décideur et le solveur peut prendre trois formes : a priori, a posteriori, et forme interactive (Hwang & Masud 1979) :

- ***Forme à priori***

Dans cette forme d'interaction, le décideur spécifie ses préférences avant l'optimisation. Ceci revient à réduire le problème d'optimisation multicritère en un problème mono-objectif. Mais comme cette forme nécessite des connaissances préalables de son problème en termes de possibilité et limitations, il n'est pas pertinent de trouver une seule solution en cas où les décideurs ne sont pas sûrs de leurs préférences entre des objectifs incompatibles. Un exemple de forme à priori utilisant par exemple une somme des méthodes présentées dans la section 3.3, où des poids sont données au différents objectifs afin de passer à une représentation mono-objectif.

- ***Forme à posteriori***

Dans cette forme, le processus de recherche détermine un ensemble de solutions Pareto optimales. Cet ensemble donne une vision au décideur sur le front Pareto de son problème. En quelque sorte, le front Pareto représente des connaissances acquises sur le problème traité. Avec l'aide de ces connaissances nouvellement acquises, le décideur pourra choisir une des solutions proposées par le solveur. On peut citer à titre d'exemple la méthode AHP, la méthode TOPSIS, et la méthode.

- ***Forme interactive***

Dans cette forme, une interactivité progressive existe entre le décideur et le solveur. À partir des connaissances acquises au fur et à mesure, le décideur peut définir et raffiner ses préférences, qui seront alors prises en compte par le solveur. Ce processus est itératif et dure plusieurs étapes. Puisque l'interactivité se déroule au moment de la recherche, le solveur ne doit pas prendre beaucoup de temps pour converger. Si le décideur connaît bien le problème, et que celui-ci est bien formulé, alors la convergence vers un Pareto optimal est très probable.

Les méthode à priori et interactive nécessitent des connaissances préalables de la problématique. Ce handicap est moins important avec les méthodes à postériori car les résultats de l'optimisation représentent des informations supplémentaires qui aideront davantage le décideur à effectuer son choix. Zanakis et al. (Zanakis et al. 1998) ont réalisé une étude comparative entre les méthodes d'aide à postériori. Les résultats ont montré des performances très satisfaisantes TOPSIS. Selon leur études, TOPSIS a démontré une supériorité par rapport à d'autres méthodes connues dans la littérature telles que AHP (Analytic Hierarchy Process, (Saaty 1980)) et ELECTRE (Roy 1968).

### 3.5.1. **TOPSIS**

Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) est une technique d'aide à la décision multicritère pour le classement et la sélection parmi un certain nombre d'alternatives via la distance euclidienne, développée par Hwang et Yoon (Hwang & Masud 1979).

Le but des méthodes d'aide à la décision n'est pas seulement de faire du profit autant que possible, mais aussi d'éviter autant que possible les risques. Partant de ce principe, le grand apport de TOPSIS est l'introduction des notions d'idéal et d'anti-idéal.

Le principe de TOPSIS est basé sur la distance des alternatives par rapport à l'idéal positif $V^+$ et l'idéal négatif $V^-$ (appelé aussi anti-idéal). Le positif idéal $V^+$ représente une solution fictive prenant la meilleure valeur (parmi les valeurs des alternatives) de chaque critère. Inversement, l'idéal négatif $V^-$ prend les pires valeurs. TOPSIS peut être utilisée comme une forme a posteriori pour aider le décideur à faire son choix parmi les solutions Pareto optimales. TOPSIS est capable de trier les solutions du front Pareto en considérant les solutions comme des alternatives et les valeurs des fonctions objectif de chaque solution comme des critères.

La méthode TOPSIS se définie en sept étapes :

1) Établir une matrice X, où un ensemble d'alternatives $a_1, a_2, \dots a_j \dots, a_k$ sont comparés sur des critères $c_1, c_2, \dots c_i \dots, c_n$.

2) Normaliser la matrice de décision pour obtenir une nouvelle matrice R d'élément $r_{ij}$ tel que :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{k} x_{ij}^2}} \tag{3.9}$$

3) Calculer la matrice normalisée pondérée. Les poids $w_{ij}$ sont donnés par les décideurs pour représenter leurs préférences entre les critères, avec $\sum_{i=1}^{n} w_i$

$$v_{ij} = w_i \times r_{ij} \tag{3.10}$$

4) Définir l'idéal positif $V^+$et l'idéal négatif $V^-$ :

$$V^+ = \left(best_j(v_{1j}), best_j(v_{2j}), \dots, best_j(v_{nj})\right)$$

$$V^- = \left(worst_j(v_{1j}), worst_j(v_{2j}), \dots, worst_j(v_{nj})\right)$$

Avec $best_j(v_{1j}) = \begin{cases} \max_j(v_{ij}) \; si \; c_i \; est \; bénéficial \\ \min_j(v_{ij}) \; si \; c_i n'est \; pas \; bénéfical \end{cases}$

et $worst_j(v_{1j}) = \begin{cases} \min_j(v_{ij}) \; si \; c_i \; est \; bénéficial \\ \max_j(v_{ij}) \; si \; c_i n'est \; pas \; bénéfical \end{cases}$

un critère est bénéficial si augmentation constitue un gain,

5) Calculer pour chaque alternative, la distance euclidienne entre l'idéal positif et l'idéal négatif, notées $d_j^+$ et $d_j^-$ respectivement :

$$d_j^+ = \sqrt{\sum_{i=1}^{n} \left(v_{ij} - v_i^+\right)^2}$$

$$d_j^- = \sqrt{\sum_{i=1}^{n} \left(v_{ij} - v_i^-\right)^2}$$

**( 3.11 )**

6) Calculer le degré de proximité au positif idéal $D_j^+$. Plus $D_j^+$ est important, plus l'alternative $j$ est proche de l'idéal positif et loin de l'idéal négatif :

$$D_j^+ = \frac{d_j^-}{d_j^- + d_j^+}$$

**( 3.12 )**

7) Finalement, trier les solutions par rapport à $D_j^+$. Les alternatives seront alors classées par ordre de préférence.

## 3.6. Conclusion

Dans ce chapitre, nous avons présenté les des concepts de base de l'optimisation tel que la domination et le front Pareto optimal. Nous avons aussi donné une introduction aux méthodes de résolution traditionnelles, avec l'inconvénient majeur qu'elles présentent, qui se résume en la modification de l'espace de recherche, ce qui dégrade la qualité des solutions obtenues. Les deux métaheuristiques multicritère NSGA-II et AMOSA, présentés en détail dans ce chapitre, ont été proposées comme des alternatives très performants. La méthode TOPSIS a été détaillée avec son principe à sept étapes.

# Chapitre 4.

## Problème 1 : Génération des processus de fabrication dans un RMS pour le cas unitaire

*Ce chapitre donne une vue détaillée de l'approche adoptée pour la génération des processus de fabrication dans le cas unitaire. Un modèle mathématique est proposé permettant la minimisation du coût total incluant (coût d'utilisation des machines/outils, coût de changement d'outils, coût de reconfiguration et le coût changement de machine) et le temps total de réalisation d'une unité du produit. L'approche adoptée utilise séparément NSGA-II et AMOSA pour la génération de processus de fabrication. Une technique de codage en nombre réel est proposée pour éviter les solutions irréalisables. Pour illustrer l'applicabilité de l'approche, nous présentons un panel d'expériences numériques et analyses, où nous comparons les performances de NSGA-II et AMOSA. Le chapitre se termine par une conclusion.*

## 4.1.Description du problème

La génération d'un macro-processus de fabrication pour un type de produit consiste en deux principales tâches : séquencer les opérations et les affecter aux machines. Comme nous l'avons présenté précédemment, un RMT offrent des fonctionnalités différentes qui dépendent de la configuration adoptée. En plus, cette configuration n'est disponible qu'avec des TADs (Tool Approach Direction) bien précis et des outils bien spécifiques. Un TAD représente généralement un des six axes géométriques $\pm x, \pm y$ , et $\pm z$.

### *1) Affectation des opérations aux machines*

L'affectation des opérations aux machines est effectuée suivant trois étapes :

- ***Identification des besoins de l'opération***

Avant d'affecter une opération à une machine, les besoins de cette opération doivent être clairement identifiés pour servir en tant que critères de choix.

Une opération est caractérisée par trois données principales : la matrice de précédence, les TADs requis et les outils candidats. En se basant sur ces trois données il est possible par la suite de garantir la faisabilité de l'opération, la capacité technique de la machine à effectuer cette opération, ainsi que la disponibilité de l'outil nécessaire.

- ***Identification des machines candidates***

Une fois les besoins de l'opération sont identifiés, il est nécessaire ensuite d'identifier parmi les machines disponibles, et leurs configurations respectives, l'ensemble des machines ayant la capacité d'effectuer l'opération concernée. Ceci est effectué par l'identification des TADs et des outils disponibles pour chaque machine/configuration, et la comparaison de ces TADs/Outils avec ceux requis par l'opération.

- ***Affectation de l'opération à la machine***

L'ensemble des machines candidates étant identifié, l'opération pourra donc être affectée à une de ces machines. Quelle machine choisir parmi les machines de cet ensemble ? Avec quelle configuration ? Et quel outil ? Les réponses à ces questions dépendent de la nature de l'objectif à atteindre.

### *2) Séquencement des opérations*

Le séquencement des opérations consiste à ordonner les opérations dans une séquence faisable, c'est-à-dire qui respecte les contraintes de précédence. Plusieurs séquences faisables

sont possibles à partir d'un graphe de précédence (Figure 4.1), et le choix parmi ces séquences dépend aussi de l'objectif recherché et la technique d'optimisation implémentée.

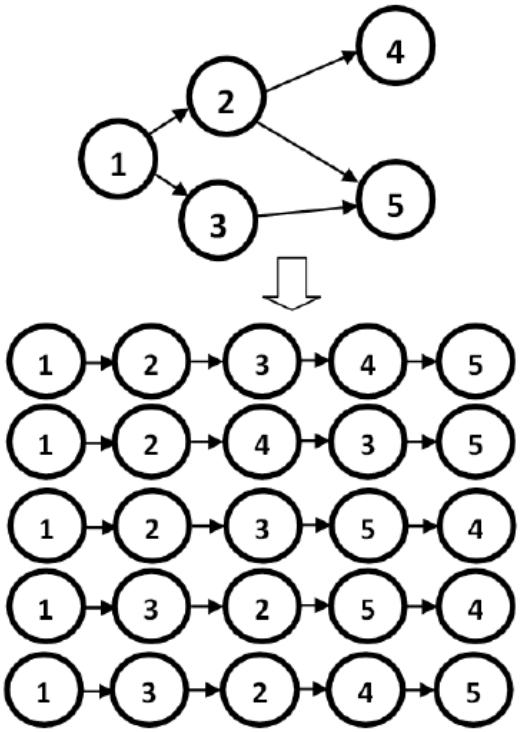


**Figure 4.1. Un graphe de précédence et séquencements possibles**

## 4.2. Données d'entrée du problème

Cette section regroupe l'ensemble des données d'entrée nécessaires à la génération des processus de fabrication. Les données peuvent être regroupées en trois types : informations sur le produit, informations sur les machines et informations sur les coûts.

### *1) Informations sur le produit*

- ***Opérations et graphe de précédence***

Le graphe de précédence résume les contraintes de précédence, qui définissent l'ordre de succession des opérations. Un produit est caractérisé par un ensemble de caractéristiques (*Feature* en anglais), dont chacune nécessite une ou plusieurs opérations (Figure 4.2).

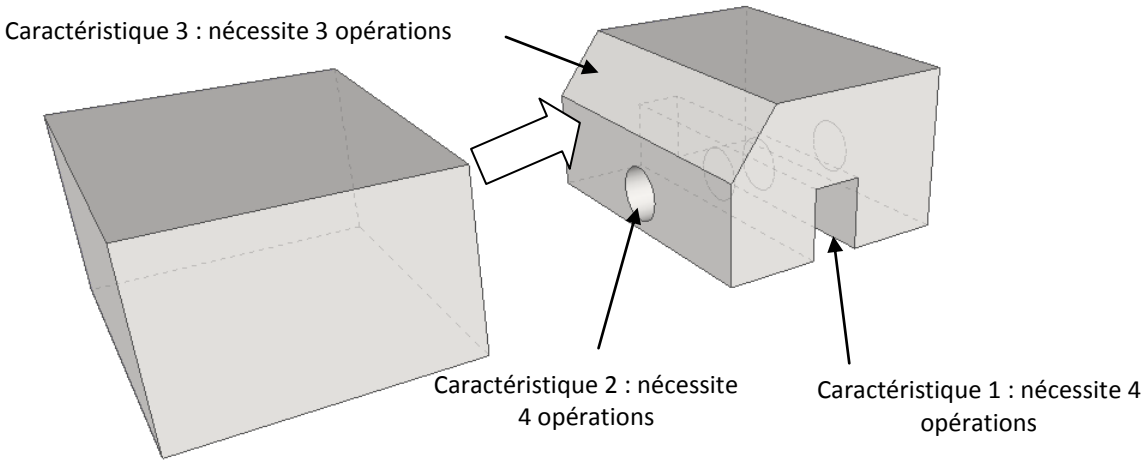


**Figure 4.2. Exemple d'une pièce à manufacturer**

Chaque nœud dans le graphe représente une opération nommée F*x_y*, ou *x* représente le numéro de la caractéristique, et *y* représente le numéro de l'opération de la caractéristique *x*.

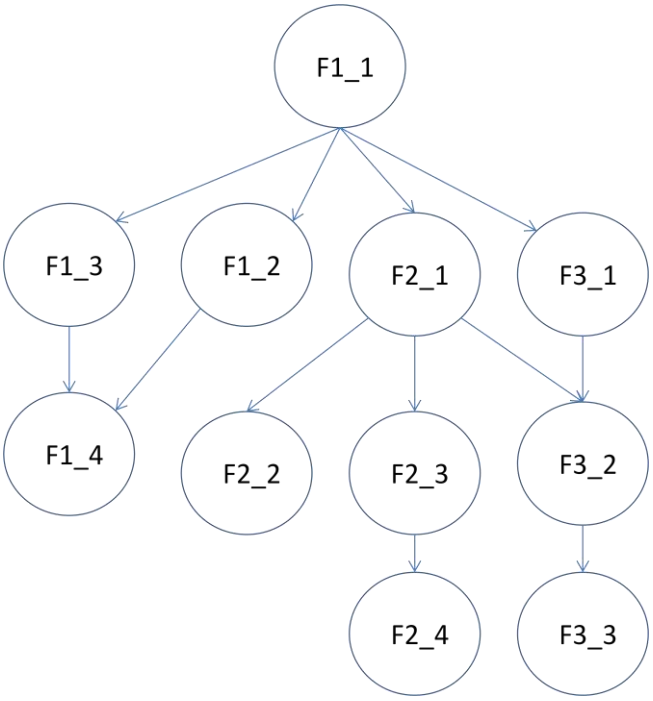


**Figure 4.3. Graphe de précédence associé à la pièce présenté sur la Figure 4.2**

Les arcs dans le graphe présentent les contraintes. Une opération avec un arc entrant ne peut être lancée que si l'opération du départ de cet arc est achevée. Ces contraintes de précédence doivent être respectées lors de la génération des processus de fabrication. Les relations de précédence entre les opérations sont dues aux considérations technologiques ou géométriques servant à effectuer la production avec une meilleure précision. Plusieurs raisons peuvent être à l'origine d'une contrainte de précédence entre deux opérations, par exemple : les points qui

nécessitent qu'une surface soit manufacturée pour servir en tant qu'un repère (géométrique) pour d'autres opérations, ou l'accessibilité à une partie du produit nécessite l'achèvement d'une opération particulière pour pouvoir passer aux suivantes, ou simplement des contraintes logiques comme le fait que la peinture doit succéder aux opérations d'usinage.

- ***Tool approach directions (TAD) requis***

Pour un produit, un TAD définit la direction avec laquelle une opération doit être effectuée. Un TAD est défini sur le repère tridimensionnel X Y Z (Figure 4.4).

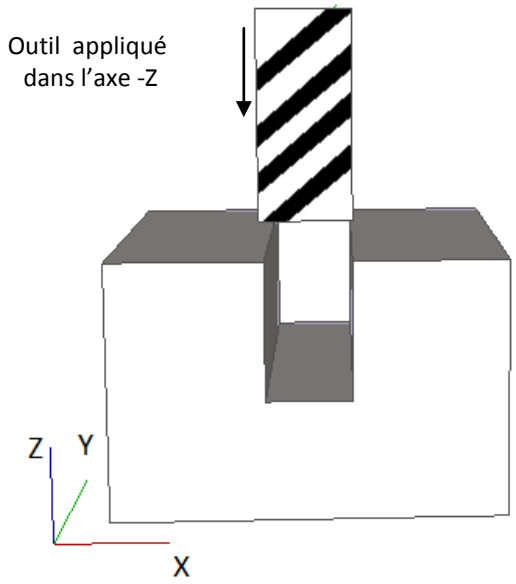


**Figure 4.4. Exemple d'une opération nécessitant un TAD (-Z)**

- ***Les outils candidats***

Cette information spécifie les outils pouvant effectuer une opération particulière. Chaque outil à ses propres caractéristiques en termes de précision et coût d'utilisation.

### *2) Informations sur les machines*

Les informations à propos des machines concernent essentiellement leurs configurations possibles et les capacités dont elles disposent dans chacune de leurs configurations, ainsi que les outils nécessaires pour la réalisation de leurs opérations. Chaque configuration d'une machine à ses propres caractéristiques, puisqu'elle est définie par son espace atteignable, et les TADs possibles.

#### *3) Information sur les coûts*

L'information sur les coûts se rapporte au coût d'utilisation d'une machine/un outil, le coût de changement d'une reconfiguration ou d'un outil, et le coût de déplacement (transfert) du produit entre deux machines.

#### *4) Informations sur les temps*

L'information sur les temps concerne le temps nécessaire pour les différentes activités pour accomplir toutes les opérations et donnant lieu au produit fini. Il se compose du temps opératoire et les temps de changement de configuration ou d'outil ainsi que le temps de déplacement du produit.

## 4.3. Modélisation des données

### 4.3.1. Structure de données du produit

Un produit est défini par ses dimensions, les opérations, les TAD ainsi que les outils nécessaires pour manufacturer ce produit :

#### *a) Dimensions*

$L, W$ et $H$ représente les dimensions du produit. Cette donnée est nécessaire pour définir l'enveloppe de travail (espace atteignable).

#### *b) Opérations*

- T$NOP$ : nombre d'opérations requises pour produire un produit.

- $OPP[1..TNOP][1..TNOP]$ : matrice représentant le graphe de précédence,

Avec $OPP[x][y]\begin{cases}1\ si\ OP_x \text{ doit être effectuée avant } OP_y \\ 0\ sinon\end{cases}$

$x, y$ Indice d'opérations, $x, y = 1..NOP$

#### *c) TADs*

- $OPTAD[1..TNOP][1..6]$ : matrice contenant les TADs requis par les opérations

Avec $OPTAD(x, d) = \begin{cases}1\ Si\ OP_x \text{ à } besoin\ du\ TAD\ d \\ 0\ sinon\end{cases}$

$x$ : index de l'opération, $x = 1..TNOP$

$d$ : index représentant le TAD, $d = 1..6$, où la valeur de $d$ indique les TADs suivants :

$d = 1$ : TAD dans la direction $+X$

$d = 2$ : TAD dans la direction $-X$

$d = 3$ : TAD dans la direction $+Y$

$d = 4$ : TAD dans la direction $-Y$

$d = 5$ : TAD dans la direction $+Z$

$d = 6$ : TAD dans la direction $-Z$

***d) Outils candidats***

- NT : nombre d'outils.

- $OPT[1..TNOP][1..NT]$ : Matrice des outils pouvant effectuer une opération

Avec $OPT[x][t] = \begin{cases} 1 \; si\ l'outil\ t\ est\ capable\ d'effectuer l'opération\ x \\ 0\ sinon \end{cases}$

$x = 1..OP$ index de l'opération,

$t = 1..NT$ index de l'outil

### 4.3.2. Structure de données des machines

***a) Machines :***

- $M$ : Nombre de machines

***b) Configurations***

- $NC[1..NM]$ : nombre de configurations disponibles pour la machine.

- $MTAvail[1..NM][1..NT]$ : outils disponibles pour la machine.

Avec $MTAvail[m][t] = \begin{cases} 1\ si\ l'outil\ d\ t\ est\ disponible\ sur\ la\ machine\ m \\ 0\ sinon \end{cases}$

$m = 1..NM$ : index de la machine.

$t = 1..NT$ : index de l'outil.

- $CTAD[1..NM][1..MAX(NC[m])][6]$ : matrice des TAD disponibles par configuration de machine, avec :

$$CTAD[m][c][d] = \begin{cases} 1 \; si \; la \; configuration \; c \; de \; la \; machine \; m \; offre \; le \; TAD \; d \\ 0 \; sinon \end{cases}$$

$m = 1..NM$ : index de la machine

$c = 1..NC(m)$ : index de la configuration

$d = 1..6$ : index du TAD

### 4.3.3. Structure de données des coûts :

Dans le modèle proposé, le coût total se compose des coûts suivants :

#### *a) Coût d'utilisation des machines*

$CM[1..NM]$ : coût d'utilisation des machines. C'est un coût unitaire qui représente le coût d'utilisation d'une machine particulière par unité de temps.

#### *b) Coût de changement de machines*

$TCost[1..NM][1..NM]$ : il s'agit du coût encouru par le déplacement du produit entre deux machines, c'est le cas quand deux opérations adjacentes dans le processus de fabrication nécessitent deux machines différentes.

#### *c) Coût de reconfiguration d'une machine*

$CCCost[1..NM][1..MAX(NC[m])][1..MAX(NC[m])]$ : c'est le coût généré par le passage d'une configuration à une autre, sur la même machine. Ce cas se présente lorsque deux opérations successives sur la même machine requièrent deux configurations différentes.

#### *d) Coût d'utilisation des outils*

$CT[1..NT]$ : c'est le coût d'utilisation de l'outil $t = 1..NT$.

#### *e) Coût de changement des outils*

$TCCost[1..NT][1..NT]$: c'est le coût du passage d'un outil un autre.

### 4.3.4. Structure de données des temps :

Les temps composant le temps total de notre modèle peuvent être décrits comme suit :

#### *a) Temps de changement de configurations*

$CCTime[1..NM][1..MAX(NC[m])][MAX(NC[m]]$ : c'est le temps de passage d'une machine d'une configuration à une autre.

#### *b) Temps de changement de machines*

$MCTime[1..NM][1..NM]$ : c'est le temps nécessaire pour un produit pour passer d'une machine à une autre.

#### *c) Temps de changement d'outils*

$TCTime[1..NT][1..NT]$ : il s'agit du temps de changement d'outils sur une machine pour effectuer une opération.

#### *d) Temps opératoire*

$PrTime[1..NM][1..MAX(NC(m))][1..NT]$ : c'est le temps nécessaire pour l'accomplissement d'une opération. Notons qu'à ce stade là, on n'a pas encore les détails du micro processus de fabrication (Zhang et al. 1997), pour cette raison on utilise un temps approximatif qui dépend de la configuration adoptée et l'outil utilisé

## 4.4. Formulation mathématique du problème

Plusieurs objectifs font traditionnellement l'objet des approches de génération des processus de fabrication, tels que le coût, le temps et la qualité des produits. Cependant, rares sont les cas où ces objectif ont été abordés simultanément dans le domaine des RMSs. Ceci peut être expliqué par des limites techniques (e.g. la puissance de calcul) mais surtout par l'émergence toute récente des RMSs, qui n'ont pas encore atteint la maturité industrielle.

Dans notre travail nous optimisons conjointement deux objectifs : le coût total et le temps total en seul modèle multicritère qui sera par la suite résolu en se basant sur les techniques de métaheuristiques :

Objectif 1 : minimiser le coût total

Objectif 2 : minimiser le temps total

### 4.4.1. Coût total

Le coût total regroupe l'ensemble des coûts générés par la production. Il est principalement composé de cinq grandes parties :

### a) Coût d'utilisation des machines (MUC : Machine Using Cost)

Il s'agit du coût d'utilisation des machines pour la réalisation des opérations. Il dépend du temps opératoire des opérations et le coût d'utilisation unitaire. Le MUC est exprimé comme suit :

$$MUC = \sum_{u=1}^{TNOP} CM[M_j(u)] \times PrTime\big[M_j(u)\big]\big[C_l^j(u)\big]\big[T_j(u)\big] \tag{4.1}$$

Avec

- $M_j(u)$ : la machine $M_j$ effectuant l'opération d'index $u$ .
- $CM\big[M_j(u)\big]$ : coût unitaire de l'utilisation de la machine $M_j$.
- $C_l^j(u)$ : la configuration $l$ de la machine $j$ utilisée pour effectuer l'opération d'index $u$.
- $T_j(u)$ : l'outil utilisé pour effectuer l'opération d'index $u$.
- $PrTime\big[M_j(u)\big]\big[C_l^j(u)\big]\big[T_j(u)\big]$ : temps requis pour achever l'opération.

### b) Coût de changement de machine (MCC : Machine Changing Cost)

Il s'agit des coûts générés lorsque deux opérations successives nécessitent deux machines différentes (il s'agit ainsi, d'un coût de transport). L'expression du MCC est donnée par l'expression suivante :

$$MCC = \sum_{u=1}^{TNOP} \emptyset\big(M_j(u), M_j(pos)\big) \times TCost\big[M_j(u)\big]\big[M_j(pos)\big] \tag{4.2}$$

Avec :

- $M_j(u)$ : La machine $M_j$ effectuant l'opération d'index $u$
- $pos = \min(OP_{u`})\ tel\ que\ \ OP_u < OP_{u`} < TNOP$ , c'est-à-dire, dans la séquence adoptée, l'opération $u`$ suit immédiatement l'opération $u$.
- $\emptyset(x, y) = \begin{cases} 0\ si\ x = y \\ 1\ sinon \end{cases}$

### c) Coût de changement de configuration (CCC : Configuration Changing Cost)

Il s'agit des coûts engendrés par le changement de configurations des machines. Son expression est donnée comme suit :

$$CCC = \sum_{u=1}^{TNOP} \emptyset\left(C_i^j(u), C_i^j(pos)\right) \times CCCost_j\left[C_i^j(u)\right]\left[C_l^j(pos)\right] \quad (4.3)$$

Avec

- $C_j^l(u)$ : la configuration d'index $l$ de la machine d'index $j$ effectuant l'opération d'index $u$.
- $pos = \min(OP_{u`})$ $tel\ que$ $OP_u < OP_{u`} < TNOP$ , $\forall\ M(u) = M(u`)$ c'est-à-dire l'opération $u`$ est la suivante sur la même machine qui a effectué $u$.
- $\emptyset(x, y) = \begin{cases} 0\ si\ x = y \\ 1\ sinon \end{cases}$

### d) Coût d'utilisation des outils (TUC : Tool Using Cost)

Le coût d'utilisation des outils dépend du type de l'outil utilisé et du temps d'utilisation. L'expression du TUC est la suivante :

$$TUC = \sum_{u=0}^{TNOP} CT[T(u)] \times PrTime\left[M_j(u)\right]\left[C_l^j(u)\right][T(u)] \quad (4.4)$$

Avec

- $T(u)$ : l'outil utilisé pour effectuer l'opération d'index $u$.
- $M_j(u)$ : la machine $M_j$ effectuant l'opération d'index $u$.
- $C_j^l(u)$ : la configuration d'index $l$ de la machine d'index $j$ effectuant l'opération d'index $u$.
- $PrTime\left[M_j(u)\right]\left[C_l^j(u)\right]\left[T_j(u)\right]$ : temps requis pour achever l'opération $u$.

### e) Coût de changement d'outils (TCC : Tool Changing Cost)

C'est le coût encouru par le changement d'outils, puisque les différentes opérations peuvent exiger des outils de différents types. Notons que ce coût varie d'une machine à l'autre. Il est donné par l'expression suivante :

$$TCC = \sum_{u=1}^{TNOP} \emptyset(T(u), T(pos)) \times TCCost[T(u)][T(pos)] \quad (4.5)$$

Avec

- $pos = \min(OP_{u`})$ $tel\ que$ $OP_u < OP_{u`} < TNOP$ , $\forall\ M(u) = M(u`)$ c'est-à-dire l'opération $u`$ est la suivante sur la même machine qui a effectué l'opération $u$.
- $\emptyset(x, y) = \begin{cases} 0\ si\ x = y \\ 1\ sinon \end{cases}$

Le premier objectif sera la minimisation de la somme des équations ( 4.1 ), ( 4.2 ), ( 4.3 ), ( 4.4 ) et ( 4.5 ). Ainsi, le premier objectif relatif au coût est exprimé sous la forme suivante :

$OBJECTIF\ 1 :\ Minimiser\ (TotalCost)\ avec$

$$\boxed{\boldsymbol{TotalCost = MUC + MCC + CCC + TUC + TCC}} \quad (4.6)$$

### 4.4.2. Temps total

Les RMSs sont conçus pour être des systèmes réactifs. Cette réactivité est basée sur des facteurs de coût, afin de garantir une réponse aux changements la plus économique. La réactivité des RMSs se doit aussi d'accompagner la dynamique des contraintes et des besoins avec les meilleurs délais. Ces délais sont exprimés sous la forme d'un temps total, englobant les différents temps nécessaires pour la réalisation de l'ensemble des activités et des opérations donnant lieu au produit fini. En effet, le temps total donne une information sur le temps requis pour qu'une unité de produit soit terminée en se basant sur le processus de fabrication généré dans le modèle proposé. Les différentes parties composant le temps total sont les suivantes :

#### *a) Temps opératoire (OPT : Operating Time)*

Le temps opératoire est le temps requis pour qu'une machine termine l'opération associée. Il dépend de la machine, la configuration adoptée ainsi que l'outil d'usinage utilisé. Les temps peuvent être obtenus via un historique de données ou avec des tests pilotes. L'expression du temps opératoire est donnée sous la forme suivante :

$$OPT = \sum_{u=1}^{TNOP} PrTime[Mj(u)][C_j^l(u)][T(u)] \quad (4.7)$$

Avec

- $M_j(u)$ : la machine$M_j$ effectuant l'opération d'index $u$.
- $C_j^l(u)$ : la configuration d'index $l$ de la machine d'index $j$ effectuant l'opération d'index $u$.
- $T(u)$ : l'outil utilisé pour effectuer l'opération d'index $u$.

$PrTime[M_j(u)][C_l^j(u)][T_j(u)]$ : temps requis pour achever l'opération $u$.

### b) Temps de changement d'outils (TCT : Tool Changing Time)

Le temps de changement d'outils est généré si deux opérations successives sur la même machine nécessitent deux outils différents. Son expression est la suivante :

$$TCT = \sum_{u=1}^{TNOP} \emptyset(T(u), T(pos)) \times TCTime\,[T(u)][T(pos)] \quad (4.8)$$

Avec

- $pos = \min(OP_{u`})$ $tel\ que$ $OP_u < OP_{u`} < TNOP$ , $\forall\, M(u) = M(u`)$ c'est-à-dire l'opération $u`$ est la suivante sur la même machine qui a effectué l'opération $u$.
- $\emptyset(x, y) = \begin{cases} 0\ si\ x = y \\ 1\ sinon \end{cases}$
- $TCTime[x][y]$: le temps de passage de l'outil $x$ vers l'outil $y$.

### c) Temps de changement de configuration (CCT : Configuration Changing Time)

Le temps de changement de configuration est le temps nécessaire pour qu'une machine passe d'une configuration à une autre. Il dépend de la machine et les deux configurations : la configuration courante (actuelle) et la configuration suivante (cible). Le CCT est exprimé avec l'équation :

$$CCT = \sum_{u=1}^{TNOP} \emptyset\big(C_l^j(u), C_{l`}^j(pos)\big) \times CCTime\,[C_l^j(u)][C_{l`}^j(pos)] \quad (4.9)$$

Avec

- $pos = \min(OP_{u`})\ tel\ que\ \ OP_u < OP_{u`} < TNOP$ , $\forall\ M(u) = M(u`)$ c'est-à-dire l'opération $u`$ est la suivante sur la même machine qui a effectué l'opération $u$.
- $\emptyset(x, y) = \begin{cases} 0\ si\ x = y \\ 1\ sinon \end{cases}$
- $CCTime[x][y]$: le temps de passage de la configuration $x$ vers la configuration $y$.

Le second objectif est égal à la somme des équations ( 4.7 ), ( 4.8 ) et ( 4.9 ). Ainsi, le deuxième objectif relatif au temps est exprimé sous la forme suivante :

$OBJECTIF\ 2 :\ Minimiser\ (TotalTime)\ avec$

$$\boldsymbol{TotalTime = OPT + TCT + CCT}$$

**( 4.10 )**

### 4.4.3. Contraintes sur le modèle

L'optimisation du modèle proposé est soumise à plusieurs contraintes résumées à travers les points suivants :

- Une machine affectée à une opération doit être capable à effectuer cette opération en termes de degré de liberté. Si $M_j(u)$ est la machine effectuant l'opération $u$ avec une configuration $C_l^j(u)$ alors :

$$CTAD_j\left[C_l^j(u)\right][tad] \geq\ OPTAD[\,u][tad] \qquad \forall\ tad \in \{1..6\}$$

**( 4.11 )**

- Les contraintes de précédence doivent être satisfaites

$$OPP[u][u`] = 0 \qquad \forall\, u < u`$$

**( 4.12 )**

- L'outil associé à une opération doit être capable de l'effectuer :

$$OPT[u][T(u)] = 1\ \ \forall\, u\ \in 1..TNOP$$

**( 4.13 )**

## 4.5. Méthode de résolution

### 4.5.1. Représentation du processus de fabrication

Afin de pouvoir manipuler le processus de fabrication par les métaheuristiques, il est nécessaire de passer d'un processus de fabrication sous forme de document (Figure 2.2) à un processus de fabrication sous forme matricielle. Ainsi, nous représentons un processus de fabrication en tant qu'une matrice $M \times N$, sous la forme illustrée dans le Tableau 4.1.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Caractéristique | F1 | F2 | F1 | F3 | F2 | F1 | F3 |
| Opération | O1 | O1 | O3 | O1 | O2 | O2 | O2 |
| Machine | M1 | M1 | M2 | M1 | M2 | M3 | M2 |
| Configuration | C1 | C1 | C3 | C1 | C2 | C1 | C2 |
| Outil | T1 | T2 | T1 | T2 | T3 | T2 | T4 |

**Tableau 4.1. Un processus de fabrication sous forme de tableau**

Le processus de fabrication sur le Tableau 4.1 est interprété de gauche à droite, colonne par colonne. Par exemple, la première colonne doit être lue comme suite : l'opération $O1$ de la caractéristique $F1$ est effectuée sur la machine $M1$, avec la configuration $C1$, en utilisant l'outil $T1$. Les colonnes suivantes sont interprétées de la même manière.

### 4.5.2. Codage des solutions

Comme nous l'avons indiqué précédemment, le codage des solutions est une étape clé dans l'implémentation des métaheuristiques. Rappelons que le codage consiste à passer de la représentation réelle des solutions vers une représentation codée. Il a pour objectif de rendre les solutions plus maniables par l'algorithme de recherche, et ainsi l'alléger des détails non nécessaires, en lui permettant de parcourir plus aisément l'espace de recherche. Durant l'évolution du processus de recherche la manipulation des solutions codées pourrait entraîner des modifications d'un certain nombre valeurs. Cette modification peut générer des solutions non faisables si le codage/décodage n'a pas été soigneusement sélectionné.

Prenons l'exemple suivant, où un ensemble d'opérations avec des contraintes de précédence sont séquencées (Figure 4.5).

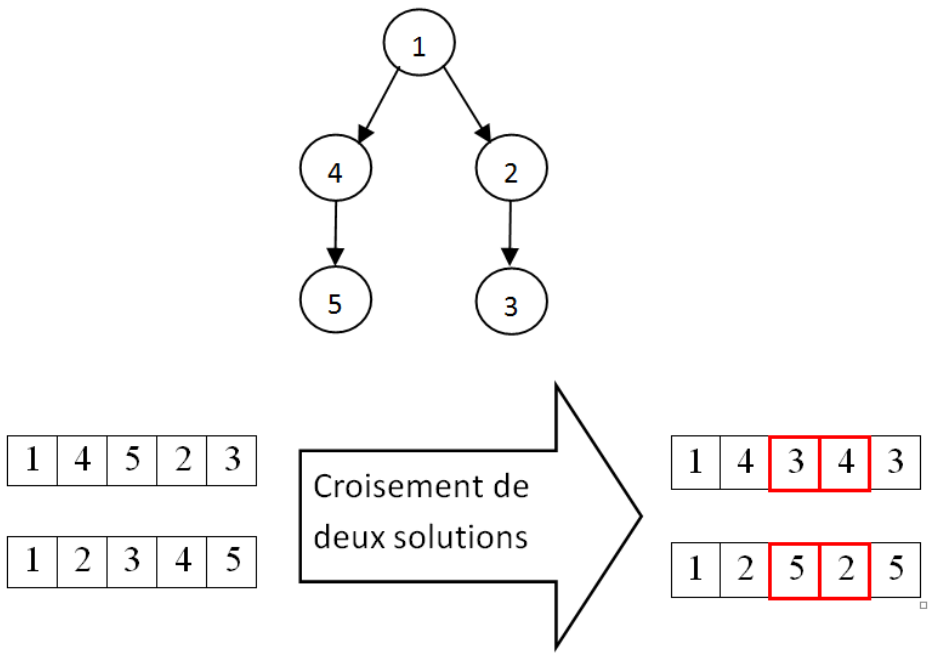


**Figure 4.5. Exemple du problème de faisabilité**

Nous remarquons que les deux solutions sont faisables et respectent les contraintes de faisabilité. En appliquant l'opérateur de croisement (sur un seul point) nous obtenons deux nouvelles solutions. Nous remarquons que ces deux solutions ne sont pas faisables, car il y a une duplication d'opérations (la première solution ayant deux opérations « 2 » et « 4 » et aucune opération « 3 » et « 5 ». le même problème existe avec la deuxième solution).

Cependant, même si ces solutions seront corrigées en substituant l'opération dupliquée par une autre manquante, le risque de générer de nouveau des solutions non faisables demeure toujours : sur le même exemple, nous procédons à la correction de la première solution 1 4 3 4 3 avec le parcours, la détection des gènes erronés (dans ce cas les opérations « 4 » et « 3 ») puis leur substitution par des opérations manquantes (qui ne figurent pas dans la solution, dans ce cas les opérations « 2 » et « 5 ».

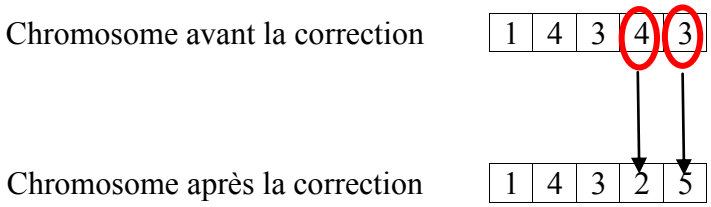


**Figure 4.6. Correction de solutions**

Ainsi, on aurait corrigé le problème de redondance, mais la faisabilité est encore loin d'être garantie. Si on observe les chromosomes corrigés, on remarque que les contraintes de

précédence ne sont pas respectées vue que l'opération « 3 » est séquencée avant l'opération « 2 ».

Il est possible de procéder à une nouvelle correction pour éliminer cette violation de contraintes, cependant, trop de modifications sur les solutions générées par le processus de recherche transformeront l'évolution de l'algorithme en une recherche aléatoire plus qu'une recherche intelligente basée sur une métaheuristique. En effet, les métaheuristiques orientent leurs recherches dans l'espace de solutions en se basant principalement sur la qualité des solutions nouvellement générées. Pour cette raison, les modifications de ces nouvelles solutions ont tendance de perturber l'évolution des algorithmes implémentant des métaheuristiques, ce qui entrainera une convergence tardive, ou dans les pires des cas, une divergence de la solution optimale

Pour éviter ce type de problème, nous proposons dans le cadre de cette thèse un codage dit en nombres réels (Tableau 4.2). La solution codée a toujours la même forme de celle de la structure de donnée du processus de fabrication présenté précédemment à travers le Tableau 4.1, c'est-à-dire une matrice $M \times N$. La chaine codée contient cinq groupes de variables, dont chacune occupe une ligne : caractéristique, opération, machine, configuration et outils. Le nombre de colonnes est égal au nombre total des opérations.

Chaque élément de la solution codée est un nombre réel, compris entre 0 et 1, le 1 étant non inclus (Tableau 4.2). Bien évidemment cette solution codée en nombre réel doit être décodée en un processus de fabrication pour pouvoir être évaluée. Ce décodage est effectué de gauche à droite, ligne par ligne en utilisant un principe proche de celui de « la sélection par roue de fortune » dans les algorithmes génétiques.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0,94 | 0,53 | 0,26 | 0,74 | 0,40 | 0,35 | 0,50 |
| 0,23 | 0,42 | 0,44 | 0,90 | 0,57 | 0,01 | 0,91 |
| 0,31 | 0,51 | 0,09 | 0,21 | 0,54 | 0,23 | 0,90 |
| 0,25 | 0,53 | 0,03 | 0,37 | 0,76 | 0,96 | 0,93 |
| 0,80 | 0,98 | 0,17 | 0,44 | 0,59 | 0,43 | 0,18 |

**Tableau 4.2. Un processus de fabrication codé en nombre réel**

Les procédures de décodage des différents composants du processus de fabrication sont présentées dans ce qui suit.

### *a) Décodage des caractéristiques*

La procédure de décodage des caractéristiques suit les étapes suivantes :

1) Identifier parmi les caractéristiques non encore terminées, celles qui peuvent être placées dans la cellule du tableau qu'on souhaite décoder.

2) Créer une liste de taille $n$, tel que $n$ est somme des nombres d'opérations restantes pour l'ensemble des caractéristiques candidates. Par exemple, si deux caractéristiques $F1$ et $F2$ sont candidates, avec 4 opérations restantes (c'est-à-dire pas encore effectuées) pour $F1$ et 3 opérations pour $F2$, alors la liste sera de taille $n = 4 + 3 = 7$.

3) Placer dans cette liste les caractéristiques candidates. Le nombre d'apparitions de chaque caractéristique dans la liste est égale au nombre d'opérations restantes. Ainsi, la liste sera :

$$F1 - F1 - F1 - F1 - F2 - F2 - F2$$

Par hypothèse, nous commençons par classer dans la liste les caractéristiques ayant index inférieur ($F1$ avant $F2$).

4) Multiplier $n$ par le nombre réel se trouvant dans la solution codée, dans la cellule correspondante à la cellule à décoder. Par exemple, pour décoder la deuxième cellule (de la première ligne puisqu'on est dans les caractéristiques), le nombre réel se trouvant dans la première ligne/deuxième colonne est multiplié par $n$. Cette multiplication donne un nombre réel $\boldsymbol{p}$ ( $0 \leq p < n$).

5) Arrondir $p$ au nombre entier immédiatement supérieur. Ainsi $p$ sera entre 1 et $n$ ($1 \leq p \leq n$).

6) Dans la liste précédemment établie, la caractéristique se trouvant à la position $p$ de la liste est sélectionnée pour être placée dans la cellule concernée.

Pour plus de clarté, nous donnons l'exemple suivant :

| Caractéristique | F1 | F2 | F2 | ? | | ... |
|---|---|---|---|---|---|---|
| Opération | | | | | | |
| ... | | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0,94 | 0,53 | 0,26 | 0,74 | 0,40 | 0,35 | 0,50 |
| 0,23 | 0,42 | 0,44 | 0,90 | 0,57 | 0,01 | 0,91 |
| 0,25 | 0,53 | 0,03 | 0,37 | 0,76 | 0,96 | 0,93 |
| 0,80 | 0,98 | 0,17 | 0,44 | 0,59 | 0,43 | 0,18 |
| 0,32 | 0,51 | 0,11 | 0,21 | 0,54 | 0,24 | 0,85 |

**Figure 4.7. Exemple de décodage**

Appliquons la démarche de décodage du chromosome codé en nombre réel sur la partie concernant les caractéristiques. L'objectif ici est de déterminer la caractéristique de la quatrième cellule. On remarque que sur les tableaux, le nombre réel correspondant à la cellule de la solution codée a une valeur égale à 0,74 (Figure 4.7).

Si deux caractéristiques $F1$ et $F2$ sont candidates, avec 4 opérations restantes (pas encore réalisées) pour $F1$, et 3 opérations pour $F2$. La liste sera alors de taille $n = 4 + 3 = 7$, et elle est construite est la suivante :

$$F1 - F1 - F1 - F1 - F2 - F2 - F2.$$

Ainsi $p$ :

$$p = 0{,}74 \times 7 = 5{,}18$$

En appliquant la 6e étape, $p$ est arrondi vers l'entier supérieur, il sera ainsi égal à 6.

Finalement, on choisit dans la liste précédemment établie la caractéristique se trouvant au rang n°6 (Figure 3.1). Il s'agit dans notre exemple de la caractéristique $F2$. Par ce fait, $F2$ sera placée dans le processus de fabrication.

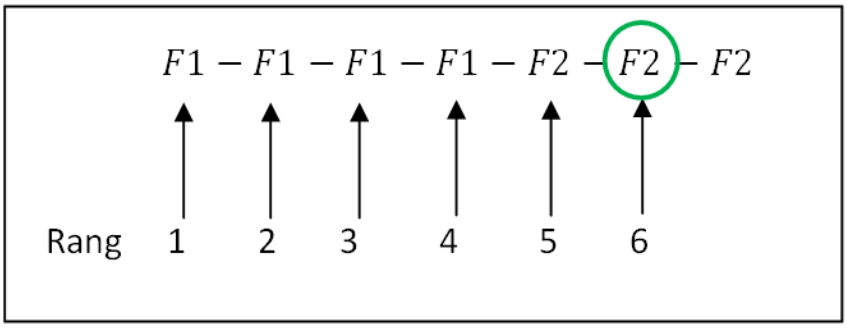


**Figure 4.8. Exemple d'une sélection d'une caractéristique**

### *b) Décodage des opérations*

Cette procédure repose sur le même principe que la procédure du décodage des caractéristiques. Le décodage des opérations est effectué après celui des caractéristiques, par conséquent dans chaque colonne la caractéristique correspondante existe déjà sur la première ligne (suite à la procédure du décodage des caractéristiques) :

1) Identifier parmi les opérations de la caractéristique de la même colonne, celles qui ne sont pas encore affectées.

2) Créer une liste de taille $n =$ nombre d'opérations restantes pour la caractéristique concernée. Par exemple, si la caractéristiques $F1$ nécessite quatre opérations : $OP1, OP2, OP3$ et $OP4$, et que $OP1$ et $OP3$ ont été déjà affectées, alors la liste sera constituée de $OP2$ et $OP4$, ainsi $n = 2$. Notons que les opérations prises en compte sont celles qui ne violent pas les contraintes de précédence.

   Par hypothèse, nous commençons par classer dans la liste les opérations ayant un index inférieur (dans cet exemple $OP2$ avant $OP4$).

3) Multiplier $n$ par le nombre réel se trouvant dans la solution codée, dans la cellule correspondante à la cellule à décoder. Par exemple, pour décoder l'opération sur la deuxième colonne (de la deuxième ligne puisqu'on est dans les opérations), le nombre réel se trouvant dans la première ligne, deuxième colonne est multiplié par $n$. Cette multiplication donne un nombre réel $\boldsymbol{p}$ ( $0 \leq p < n$).

4) Arrondir $p$ au nombre entier immédiatement supérieur. Ainsi $p$ sera entre 1 et $n$ ($1 \leq p \leq n$).

5) Dans la liste précédemment établie, l'opération se trouvant à la position $p$ de la liste est sélectionnée pour être placée dans la cellule concernée.

### *c) Décodage des machines*

Le décodage des machines est effectué après ceux des caractéristiques et des opérations. La procédure peut être résumée comme suite :

1) Identifier parmi les machines reconfigurables, celles qui possèdent les TAD et les outils requis pour effectuer l'opération de la même colonne.

2) Créer une liste de taille $n =$ nombre de machines capables de réaliser l'opération concernée. Par exemple, si les machines M1, M3 et M4 ont la capacité de d'effectuer l'opération concernée, alors la liste $M1 - M3 - M4$ , avec $n = 3$.

Par hypothèse, nous commençons par classer les machines avec un index inférieur (M1 avant $M3$ etc.).

3) Multiplier $n$ par le nombre réel se trouvant dans la solution codée, dans la cellule correspondante à la cellule à décoder. Par exemple, pour décoder la machine de la deuxième colonne (de la troisième ligne puisqu'il s'agit du décodage des machines), le nombre réel se trouvant dans la troisième ligne, deuxième colonne est multiplié par $n$. Cette multiplication donne un nombre réel $\boldsymbol{p}$ entre 0 et $n$.

4) Arrondir $p$ au nombre entier immédiatement supérieur. Ainsi $p$ est compris entre 1 et $n$ ($1 \leq p \leq n$).

5) Dans la liste précédemment établie, la machine se trouvant dans la position $p$ est sélectionnée pour être placée dans la cellule concernée.

### *d) Décodage des configurations*

Le décodage des configurations est effectué après celui des machines. En effet, pour prendre des décisions en matière de configuration des machines, il est nécessaire connaître la machine faisant l'objet de ces décisions.

La procédure de décodage des configurations peut être résumée de la façon suivante :

1) Identifier parmi les configurations de la machine déjà décodée (décodée (résultat de la procédure de décodage des machines), les configurations pouvant effectuer l'opération concernée en termes de TADs et d'outil.

2) Créer une liste de taille $n =$ nombre de configurations de la machine, capables de réaliser l'opération concernée. Par exemple, si parmi 5 configurations de la machine, les configurations C1 et C2 sont capables d'effectuer l'opération concernée, alors la liste $C1 - C2$ , avec $n = 2$.

Par hypothèse, nous commençons par classer les configurations avec un index inférieur ($C1$ avant $C2$).

3) Multiplier $n$ par le nombre réel se trouvant dans la solution codée, dans la cellule correspondante à la cellule à décoder ($RealV$). Par exemple, pour décoder la configuration sur la première colonne (de la quatrième ligne puisqu'on est dans les configurations), le nombre réel se trouvant dans la quatrième ligne/première colonne est multiplié par $n$. Cette multiplication donne un nombre réel $\boldsymbol{p}$ compris entre 0 et $n$.

4) Arrondir $p$ au nombre entier immédiatement supérieur. Ainsi, $p$ est compris entre 1 et $n$ ($1 \leq p \leq n$).

5) Dans la liste précédemment établie, la configuration se trouvant dans la position $p$ est sélectionnée pour être placée dans la cellule concernée.

#### *e) Décodage des outils*

Le décodage des outils représente la dernière étape du décodage du chromosome codé en nombres réels.

Les outils utilisés pour les différentes opérations du processus de fabrication sont décodés selon la procédure décrite par les points suivants :

1) Identifier parmi les outils disponibles pour la machine concernée, ceux qui ont la capacité d'effectuer l'opération à réaliser.

2) Créer une liste de taille $n =$ nombre d'outils capables de réaliser l'opération concernée. Par hypothèse, nous commençons par classer dans la liste les outils ayant un index inférieur ($T1$ avant $T2$ etc.).

3) Multiplier $n$ par le nombre réel se trouvant dans la solution codée, dans la cellule correspondante à la cellule à décoder. Par exemple, pour décoder l'outil de la quatrième colonne (et cinquième ligne, puisqu'il s'agit de décoder les outils), le nombre réel se trouvant dans la cinquième ligne, quatrième colonne sera multiplié par $n$. Cette multiplication donne lieu à un nombre réel $\boldsymbol{p}$ compris entre 0 et $n$.

4) Arrondir $p$ au nombre entier immédiatement supérieur. Ainsi, le nombre $p$ sera compris entre 1 et $n$ ($1 \leq p \leq n$).

5) Dans la liste précédemment établie, l'outil se trouvant dans la position $p$ sera sélectionné pour être placé dans la cellule correspondante.

## 4.6. Expériences numériques et analyses

Dans cette section nous présentons un cas d'étude pour la génération des processus de fabrication dans un RMS. A travers ce cas d'étude, nous utilisons, puis comparons les résultats de deux métaheuristiques multicritères qui ont montré le plus grand succès dans la littérature, à savoir les algorithmes génétique multi objectif (NSGA-II) et le recuit simulé multicritère (AMOSA).

### 4.6.1. Données d'entrée

Les données du problème du présent cas d'étude comprennent le graphe de précédence entre les différentes opérations ainsi que les besoins des opérations en termes d'outils et TADs.

Nous étudions un RMS comportant trois machines reconfigurables, possédant chacune de multiples configurations. Ce RMS est conçu pour un produit caractérisé par trois caractéristiques. La première caractéristique nécessite 3 opérations, la seconde nécessite 5 opérations et la troisième caractéristique nécessite 4 opérations.

Nous spécifions à travers les trois tableaux ci-dessous les détails des trois machines, à savoir leurs configurations respectives avec leurs capacités (TADs).

| Machine1 (3 configurations) | +X | -X | +Y | -Y | +Z | -Z |
|---|---|---|---|---|---|---|
| Configuration 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Configuration 2 | 1 | 0 | 1 | 0 | 1 | 0 |
| Configuration 3 | 1 | 1 | 0 | 1 | 0 | 1 |

**Tableau 4.3. TADs possibles sur la machine 1**

| Machine2 (4 configurations) | +X | -X | +Y | -Y | +Z | -Z |
|---|---|---|---|---|---|---|
| Configuration 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| Configuration 2 | 0 | 1 | 1 | 1 | 0 | 0 |
| Configuration 3 | 0 | 1 | 0 | 1 | 0 | 1 |
| Configuration 4 | 0 | 0 | 1 | 1 | 1 | 1 |

**Tableau 4.4. TADs possibles sur la machine 2**

| Machine3 (2 configurations) | +X | -X | +Y | -Y | +Z | -Z |
|---|---|---|---|---|---|---|
| Configuration 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| Configuration 2 | 1 | 1 | 0 | 0 | 0 | 0 |

**Tableau 4.5. TADs possibles sur la machine 3**

Le Tableau 4.6 présente la répartition des outils sur les différentes machines (le nombre total des outils = 7)

| | Outil 1 | Outil 2 | Outil 3 | Outil 4 | Outil 5 | Outil 6 | Outil 7 |
|---|---|---|---|---|---|---|---|
| Machine 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Machine 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| Machine 3 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

**Tableau 4.6. Outils disponibles avec les différentes machines**

Les tableaux Tableau 4.7Tableau 4.8 Tableau 4.9 résument, sous forme de matrices d'adjacence, les contraintes de précédence entre les différentes opérations nécessaires pour chacune des trois caractéristiques :

| **Caractéristique 1** | Opération 1 | Opération 2 | Opération 3 |
|---|---|---|---|
| Opération 1 | 0 | 0 | 0 |
| Opération 2 | 0 | 0 | 1 |
| Opération 3 | 1 | 0 | 0 |

**Tableau 4.7. Relations de précédence pour la caractéristique 1**

| **Caractéristique 2** | Opération 1 | Opération 2 | Opération 3 | Opération 4 | Opération 5 |
|---|---|---|---|---|---|
| Opération 1 | 0 | 0 | 0 | 0 | 0 |
| Opération 2 | 1 | 0 | 0 | 0 | 0 |
| Opération 3 | 1 | 0 | 0 | 0 | 0 |
| Opération 4 | 1 | 0 | 1 | 0 | 0 |
| Opération 5 | 1 | 0 | 1 | 0 | 0 |

**Tableau 4.8. Relations de précédence pour la caractéristique 2**

| **Caractéristique 3** | Opération 1 | Opération 2 | Opération 3 | Opération 4 |
|---|---|---|---|---|
| Opération 1 | 0 | 0 | 0 | 0 |
| Opération 2 | 1 | 0 | 0 | 0 |
| Opération 3 | 1 | 0 | 0 | 0 |
| Opération 4 | 1 | 1 | 1 | 0 |

**Tableau 4.9. . Relations de précédence pour la caractéristique 3**

A travers les trois tableaux suivants, nous présentons les TADs requis par les opérations de chaque caractéristique.

| **Caractéristique 1** | +X | -X | +Y | -Y | +Z | -Z |
|---|---|---|---|---|---|---|
| Opération 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Opération 2 | 0 | 0 | 0 | 0 | 1 | 1 |
| Opération 3 | 1 | 0 | 0 | 0 | 0 | 0 |

**Tableau 4.10. TADs requis par les opérations de la caractéristique 1**

| **Caractéristique 2** | +X | -X | +Y | -Y | +Z | -Z |
|---|---|---|---|---|---|---|
| Opération 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| Opération 2 | 0 | 0 | 0 | 0 | 1 | 1 |
| Opération 3 | 0 | 1 | 0 | 0 | 0 | 1 |
| Opération 4 | 0 | 0 | 0 | 0 | 1 | 0 |
| Opération 5 | 0 | 0 | 1 | 1 | 0 | 0 |

**Tableau 4.11. TADs requis par les opérations de la caractéristique 2**

| **Caractéristique 3** | +X | -X | +Y | -Y | +Z | -Z |
|---|---|---|---|---|---|---|
| Opération 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Opération 2 | 0 | 1 | 0 | 0 | 0 | 0 |
| Opération 3 | 0 | 0 | 0 | 1 | 0 | 0 |
| Opération 4 | 1 | 0 | 1 | 0 | 0 | 0 |

**Tableau 4.12. TADs requis par les opérations de la caractéristique 3**

A travers le tableau 16 nous présentons les outils requis pour chacune des opérations.

| | Op. 1 | Op. 2 | Op. 3 | Op. 4 | Op. 5 |
|---|---|---|---|---|---|
| Caractéristique 1 | 1 | 2 | 1 | | |
| Caractéristique 2 | 3 | 5 | 4 | 1 | 1 |
| Caractéristique 3 | 3 | 4 | 4 | 2 | |

**Tableau 4.13. Outils requis pour chaque opération**

### 4.6.2. **Résultats obtenus**

Les métaheuristiques AMOSA et NSGA-II ont été implémentées sous le langage JAVA, sur un ordinateur fonctionnant sous le système d'exploitation Windows 7. AMOSA a été initialisée avec une température maximum égale à 5000° et un paramètre de refroidissement égal à 0.90, en fixant le nombre d'itération à 1000 à chaque température. Le NSGA-II a été lancé avec une population de 500 individus, un taux de croisement 0.8 et un taux de mutation 0.2. la condition d'arrêt pour les deux métaheuristique était fixée 5 minutes de calcul. En effet, suite à plusieurs expériences, il s'est avéré que, après 5 minutes, la recherche progresse d'une manière très lente. En plus, après ce temps, l'amélioration du front de Pareto obtenu est jugée insignifiante. Les deux algorithmes ont donné deux fronts de Pareto différents (Figure 4.9).

Chaque point du front de Pareto représente un processus de fabrication entier, c'est-à-dire une matrice $M \times N$ (Tableau 4.1), possédant un temps total et un coût total de production. Les coûts et le temps des différents processus de fabrication obtenus sont présentés à travers le Tableau 4.14.

La représentation graphique de ces résultats sous forme de fronts Pareto montre une capacité de convergences plus importante avec AMOSA par rapport au NSGA-II (Figure 4.9). En effet, AMOSA a réussi à trouver un front plus optimal que celui de NSGA-II. La simplicité d'AMOSA permet d'effectuer plusieurs itérations de recherche avec un faible effort de calcul.

| AMOSA | | NSGA-II | |
|---|---|---|---|
| Coût total | Temps total | Coût total | Temps total |
| 49965 | 857 | 51004 | 907 |
| 50002 | 840 | 51287 | 882 |
| 50955 | 806 | 51314 | 880 |
| 51925 | 771 | 52620 | 832 |
| 52350 | 706 | 54718 | 726 |
| 57790 | 685 | 58725 | 712 |
| 66303 | 672 | 59753 | 711 |
| | | 60833 | 701 |

**Tableau 4.14. Résultats obtenus par AMOSA et NSGA-II**

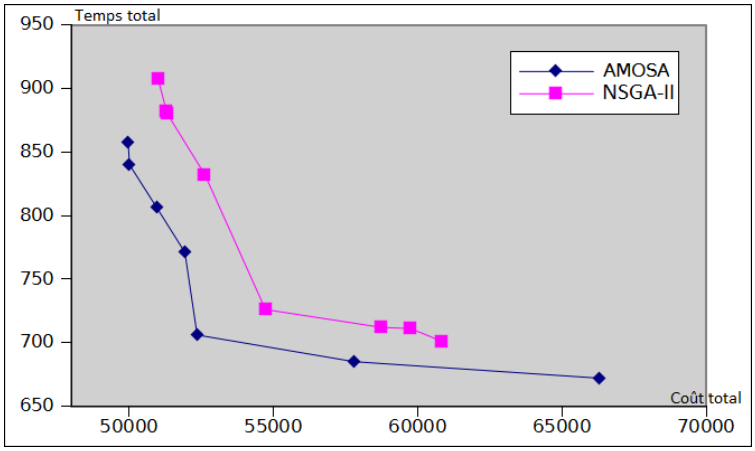


**Figure 4.9. Les deux fronts de Pareto obtenus par AMOSA et NSGA-II**

L'exemple numérique démontre l'applicabilité de l'approche proposée. Le modèle développé a été résolu avec deux métaheuristiques NSGA-II et AMOSA. Cette résolution peut être effectuée avec d'autres métaheuristiques (recherche avec tabous, essaim de particules etc.).

Cependant, nous remarquons que l'exploitation du codage en nombres réels s'est avérée très précieuse pour les performances de la recherche. En effet, la multitude des contraintes du modèle (i.e. opération/configuration, opération/opération, outil/opération, etc..) rend très peu probable l'obtention, dès le premier essai, d'une solution faisable en perturbant une solution existante. Sans le codage proposé, le processus de recherche perdra beaucoup temps à corriger les solutions générées plutôt que d'explorer l'espace de recherche et améliorer ainsi le front de Pareto.

En plus, les résultats obtenus montrent que, dans notre cas d'étude, la métaheuristique AMOSA possède une capacité de convergence plus importante que celle de NSGA-II. Ceci s'explique par deux raisons : d'un côté, sa faible utilisation des ressources de l'ordinateur (mémoire et processeur), et de l'autre côté, la probabilité d'acceptation des solutions dégradées (ce qui constitue un moyen pour éviter des minimums locaux).

Notons que le front de Pareto obtenu fournit au décideur plusieurs solutions, constituant plusieurs alternatives par rapport à chaque fonction objectif. C'est à lui ensuite, que revient de prendre la décision finale en adoptant parmi toutes ces alternatives, celle qui lui convient au mieux, généralement, en se basant sur son expérience et son vécu.

## 4.7. Extensions du problème 1

### 4.7.1. Extension 1 : utilisation de TOPSIS comme outil d'aide à la décision pour le choix du meilleur processus de fabrication

Dans le cas d'une optimisation bi-objectifs, le front de Pareto est plus ou moins visible et évaluable par les décideurs. En revanche, à partir d'une représentation sous forme de graphe, un décideur aura la possibilité d'observer visuellement la qualité du compromis entre les deux objectifs. Cependant, à partir de trois objectifs, l'évaluation intuitive ou visuelle n'est plus évidente. Dans ce cas, un mécanisme d'évaluation doit être donc mis en point.

Pour ce cas à plusieurs objectifs, nous proposons l'utilisation de TOPSIS (Technique for Order Performance by Similarity to Ideal Solution) comme support de décision. L'utilisation de cette technique est détaillée à travers les sections suivantes.

.

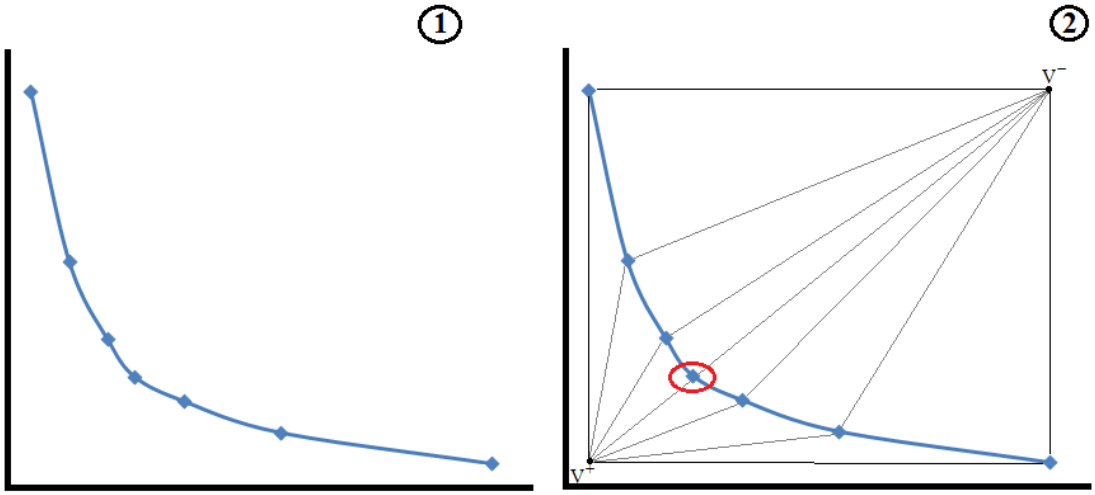


**Figure 4.10. Le passage d'un Front de Pareto (1) à une solution unique (2) en utilisant TOPSIS (Cas bi-objectifs).**

### *a) Modélisation*

Nous avons étudié précédemment des problématiques à deux objectifs, le coût total et le temps total. Pour la génération des processus de fabrication, d'autres objectifs peuvent être considérés tels que l'équilibrage des charges des machines, la maximisation du nombre des opérations automatiques dans le cas où les êtres humain interviennent dans la production, etc.

Dans un RMS, une machine peut être utilisée sous plusieurs configurations. Cette technologie de machines reconfigurables est en constant développement, mais remarquons que la reconfiguration n'est pas une activité à valeur ajoutée. Par conséquent, il est toujours préférable de la minimiser autant que possible. En outre, la reconfiguration répétitive des machines peut affecter la fiabilité des machines et causer des pannes ainsi que des pertes de précision. Pour ces raisons, nous proposons un nouvel objectif particulier aux RMS, en plus des objectifs ( 4.6 ) et ( 4.10 ). Cet objectif consiste à minimiser le nombre de reconfigurations dans un processus de fabrication. Il s'agit de la minimisation de l'«effort total de reconfiguration», noté CCE (Configuration Change Effort) :

$OBJECTIF\ 3 :\ Minimise\ (CCE)\ avec$

$$\boxed{CCE = \sum_{u=1}^{TNOP} \emptyset\left(C_l^j(u), C_{l`}^j(u`)\right)} \quad \textbf{(4.14)}$$

Avec

- $\emptyset(x,y) = \begin{cases} 0\ si\ x = y \\ 1\ sinon \end{cases}$
- $u` = \min(u_k)$ tel que $u < u_k < TNOP\ \ et\ M(u) = M(u`)$ (c'est-à-dire les deux opérations sont effectuées sur la même machine).

### b) *Expériences numériques et analyses de l'extension 1*

Pour ce cas d'étude, nous nous basons sur les mêmes données que celles de l'exemple précédent. Nous implémentons la métaheuristique AMOSA avec 3 objectifs : le coût total, le temps total et l'effort de reconfiguration, avec respectivement les poids 0,45, 0,40 et 0,15. Les résultats obtenus sont présentés dans le Tableau 4.15.

Il est clair que sans l'utilisation de TOPSIS, il est très difficile voire impossible de prendre une décision précise en se basant sur les résultats obtenus, tout en garantissant le meilleur compromis entre les différents objectifs, et cela même avec une représentation graphique (Figure 4.11).

Cependant, il est nécessaire de spécifier un poids à chaque objectif pour que TOPSIS puisse classer es différentes solutions.

Ainsi, dans ce cas d'étude, la technique de TOPSIS a classé la solution se trouvant sur la quatrième ligne comme étant la solution préférée, et cela en se basant sur les poids associés aux trois objectifs.

| AMOSA | | | TOPSIS |
|---|---|---|---|
| Coût total | Temps total | Nombre de reconfiguration | |
| 49942 | 904 | 7 | 7 |
| 50289 | 863 | 8 | 5 |
| 51012 | 832 | 8 | 4 |
| **51269** | **801** | **5** | **1** |
| 52580 | 785 | 7 | 3 |
| 52597 | 756 | 10 | 6 |
| 54392 | 727 | 6 | 2 |
| 58736 | 726 | 10 | 10 |
| 58750 | 707 | 7 | 8 |
| 61170 | 661 | 8 | 9 |

**Tableau 4.15. Résultats obtenus, triés avec TOPSIS**

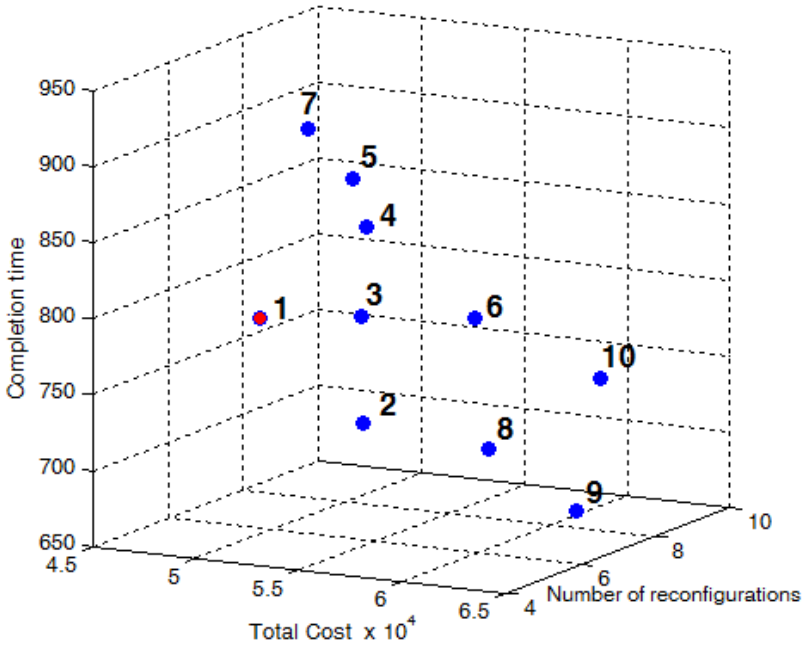


**Figure 4.11. Représentation graphique des solutions obtenues**

### 4.7.2. Extension 2 : utilisation du process plan dans l'aide à la décision pour la conception des RMSs

#### *a) Présentation du problème*

La phase de conception d'un RMS vise à sélectionner, parmi un ensemble de RMTs (machines) candidates, les RMTs ayant la capacité d'effectuer les opérations requises. Nous considérons que plusieurs machines reconfigurables sont candidates pour s'intégrer au nouveau RMS, où le décideur doit sélectionner parmi ces machines, lesquelles choisir pour faire partie de ce nouveau RMS.

Chaque machine candidate $m$ a un coût d'acquisition $MacCost[m]$ , $m = 1..NM$ où $NM$ est le nombre de machines).Ce coût d'acquisition sera pris en compte lors du calcul du coût total du produit. Nous introduisant dans la formule du coût total ( 4.6 ), un nouveau coût qui regroupe les coûts d'acquisition des machines ( 4.15 ). Chaque machine candidate $m$ possède un coût d'acquisition $MacCost[m]$, qui sera considéré dans le calcul du coût total sous la forme de ($MAC$ : Machine Acquiring Cost).

$$MAC = \sum_{m=1}^{NM} use(m) \times MacCost[m] \qquad (4.15)$$

Avec $use(m) = \begin{cases} 1 \; si \; la \; machine \; m \; est \; utilisée \; dans \; le \; process \; plan \\ 0 \; sinon \end{cases}$

Par conséquent, en incluant ce coût, la fonction objectif du coût total ( 4.6 ) sera modifiée et aura la forme suivante :

$OBJECTIF\ 1 :\ Minimise\ (TotalCost)\ avec$

$$\boxed{TotalCost = MUC + CCC + TUC + TCC + \boldsymbol{MAC}} \quad \textbf{( 4.16 )}$$

Étant encore dans la phase de sélection des machines, et n'ayant aucune information sur la disposition que ces machines prendront dans l'atelier, le $MCC$ (coût de transport) ne peut pas être considéré à ce niveau, car la disposition des machines (layout) n'est pas encore défini. De ce fait, il ne figure pas dans la nouvelle fonction objectif.

Le deuxième objectif (équation ( 4.10 )) est gardé tel qu'il est. L'approche consiste à lancer AMOSA, comme dans le cas du chapitre précédent, en considérant la nouvelle formulation du $TotalCost$. Toute machine figurant dans le processus de fabrication est considérée « sélectionnée ». Les machines ne figurant pas dans le processus de fabrication ne feront pas partie donc du nouveau RMS.

L'inclusion du $MAC$ dans le premier objectif permet de guider la recherche du processus de fabrication vers l'utilisation des machines relativement économiques, ce qui fait en sorte que les machines figurant dans le processus de fabrication sont celles qui garantissent un coût de production compétitif.

### *b) Exemple numérique de l'extension 2*

Pour ce cas d'étude, nous gardons les mêmes données que celles du cas d'étude précédent, tout en augmentant le nombre de machines. En effet, dix machines sont candidates pour former notre RMS, chacune possédant un ensemble de configurations décrites à travers le Tableau 4.16.

Les résultats obtenus sont illustrés dans le Tableau 4.17. Nous remarquons que les solutions proposant d'utiliser un grand nombre de machines sont coûteuses, mais elles garantissent un temps total relativement inférieur. Ce constat est logique car plus le nombre de machines octroyées est important plus le coût $MAC$ augmente, en provoquant une augmentation du coût total. Il est évident aussi qu'avec plusieurs machines, l'opération possède une grande plage de sélection et par conséquent plus de chance de trouver une paire machine/configuration relativement rapide.

| Machines | Configurations | TADs disponibles | | | | | |
|---|---|---|---|---|---|---|---|
| | | +X | +Y | +Z | -X | -Y | -Z |
| Machine 1 | C1 | X | | | | X | X |
| | C2 | X | X | | X | | X |
| | C3 | X | X | | X | | X |
| Machine 2 | C1 | X | X | | X | X | X |
| | C2 | X | X | X | X | | |
| | C3 | | X | | X | | X |
| | C4 | | | X | X | | X |
| Machine 3 | C1 | | | | | X | X |
| | C2 | X | X | | | | |
| Machine 4 | C1 | X | X | | | | |
| Machine 5 | C1 | | | | | X | X |
| Machine 6 | C1 | X | X | | | | |
| | C2 | | | | X | | |
| Machine 7 | C1 | X | X | | | | |
| | C2 | | | | X | | |
| Machine 8 | C1 | X | | | | X | X |
| | C2 | X | X | X | | | |
| | C3 | | X | | X | | X |
| Machine 9 | C1 | X | X | | | X | |
| | C2 | | | | X | | |
| Machine 10 | C1 | X | X | | | | |
| | C2 | | X | | X | | |

**Tableau 4.16. Données des machines**

| Liste des machines sélectionnées | (coût total, temps total) |
|---|---|
| M1, M2, M5, M6, M7, M8, M9, M10 | ( 68391 , 622 ) |
| M1, M2, M3, M5, M6, M8, M9, M10 | ( 66304 , 654 ) |
| M2, M3, M5, M6, M8, M9, M10 | ( 64656 , 660 ) |
| M1, M2, M3, M5, M6, M7, M8, M9 | ( 63329 , 702 ) |
| M1, M3, M5, M6, M7, M8, M9 | ( 62600 , 730 ) |
| M2, M3, M5, M6, M8, M9 | ( 60480 , 794 ) |
| M1, M2, M4, M5, M8, M9 | ( 59446 , 860 ) |
| M4, M5, M6, M8, M9 | ( 49657 , 1218 ) |
| M3, M5, M6, M8, M9 | ( 47114 , 1270 ) |

**Tableau 4.17. Résultats obtenus pour le cas de sélection de machines**

## 4.8. Conclusion

Nous avons présenté dans ce chapitre le problème de génération des processus de fabrication dans le cadre des systèmes reconfigurables. Une modélisation mathématique a été mise en place, prenant en compte le coût total et le temps total. Deux métaheuristiques multicritère, NSGA-II et AMOSA, ont été adaptées puis utilisées pour résoudre le modèle développé. Les résultats ont démontré de meilleures performances pour AMOSA par rapport au NSGA-II dans la génération des processus de fabrication. Cet avantage d'AMOSA est du a son algorithme relativement moins gourmand en termes de ressources de calcul, ce qui permet le passage d'une itération à une autre.

Dans la première extension, nous avons combiné TOPSIS avec AMOSA pour sélectionner le meilleur processus de fabrication parmi les solutions proposées par AMOSA. En effet, trios objectifs on été optimisés simultanément donnant lieu à un compromis entre la réactivité, le coût et l'effort de reconfiguration. Les solutions non dominées données par AMOSA sont classées par TOPSIS dans un ordre préférentiel. De cette façon, une vision sur la qualité des solutions est explicitement donnée au décideur, lui facilitant la tâche de sélectionner la plus adéquate.

Dans la deuxième extension, nous avons modifié le modèle de génération des processus de fabrication pour l'utiliser dans le cadre de la conception des RMSs. Les résultats obtenus (Tableau 4.17) sous forme d'un ensemble de solutions non dominées présentent une large marge de sélection pour le décideur, allant d'une sélection de machines réactives mais couteuses jusqu'aux machines économiques mais relativement lentes. Le décideur pourra ainsi faire son choix selon ses besoins.

# Chapitre 5.

## Problème 2 : Génération des processus de fabrication dans un RMS pour le cas multi-unité

*Ce chapitre présente une vue détaillée de l'approche proposée pour la génération des processus de fabrication dans le cas où un nombre d'unités du même produit doivent d'être fabriquées sur le même RMS (multi-unité). L'approche comprend deux phases. La première phase consiste à déterminer un ensemble de processus de fabrication optimaux au sens de Pareto en utilisant la technique AMOSA. Dans la seconde phase, afin d'optimiser la productivité du RMS, chaque unité est associée à un processus de fabrication. L'identification du meilleur processus de fabrication, unité par unité, fait appel à l'optimisation basée sur la simulation. Le chapitre présente le cadre de simulation à évènements discrets développé pour un environnement reconfigurable. Pour illustrer l'applicabilité de l'approche, nous présentons un panel d'expériences numériques et analyses. Le chapitre termine par une conclusion.*

## 5.1. Description du problème

Dans le présent chapitre, et contrairement à ce qui a été considéré précédemment, nous étudions la problématique de fabrication de plusieurs unités du même produit. En effet, les caractéristiques des RMSs garantissent un haut degré de reconfigurabilité avec une activité de reconfiguration très efficace. Ainsi, afin de mieux profiter de cet aspect de reconfiguration, nous proposons dans ce deuxième cas d'exploiter toutes les solutions optimales du front de Pareto obtenues par l'algorithme de recherche.

En effet, dans la pratique, les ressources sont partagées entre les différents produits qui existent sur le même système de production, même si ces produits sont du même type. En outre, les machines reconfigurables sont capables d'effectuer des opérations différentes. De ce fait, des temps d'attentes peuvent être engendrés.

## 5.2. Approche proposée

Pour considérer le cas multi-unité, nous proposons une approche à deux phases :

- Dans la première phase, en utilisant  nous exécutons l'algorithme de recherche (comme pour le cas d'une unitaire) pour obtenir un front Pareto des solutions optimales de processus de fabrication.

- Dans la phase 2, nous supposons que nous avons $n$ unités du même produit à manufacturer, et $p$ processus de fabrication Pareto optimaux obtenus suite à la phase 1. Dans ce contexte, le problème consiste à déterminer le meilleur processus de fabrication à affecter à chaque unité du produit, tout en prenant en compte sa position dans le lot. L'affectation d'un processus de fabrication. L'affectation est effectuée en vue d'optimiser deux objectifs, respectivement le temps total et le coût total pour l'ensemble du lot.

Les données nécessaires pour cette approche sont les processus de fabrication générés dans la phase précédente et la taille du lot à manufacturer.

Grâce à l'avancement technologique des ordinateurs et l'amélioration des techniques d'optimisation, la majorité des outils de simulation commencent d'intégrer un module d'optimisation.  L'optimisation basée sur la simulation consiste en deux modules fonctionnant

en coordination (Figure 5.2) : un module d'optimisation qui guide la direction de recherche des solutions et un module de simulation évaluant les solutions fournies par l'optimisation.

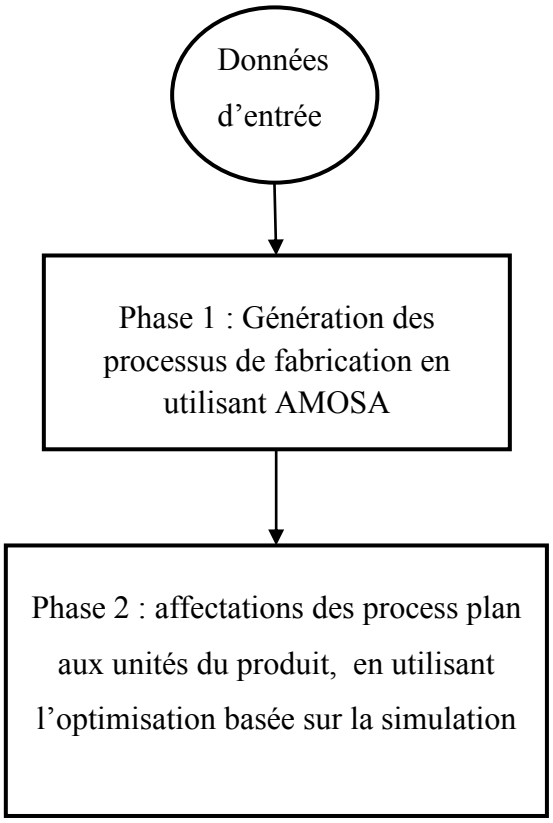


**Figure 5.1. Approche proposée pour le cas multi-unité**

Dans l'approche d'optimisation basée sur la simulation, comparée aux techniques de programmation mathématique, la fonction objectif est remplacée par un ou plusieurs modèles de simulation. En plus, les variables de décision sont les conditions sous lesquelles la simulation est exécutée. Les résultats de la simulation sont utilisés par l'optimisation d'une manière itérative pour lui fournir un retour permettant de guider la progression de la recherche de la solution optimale.

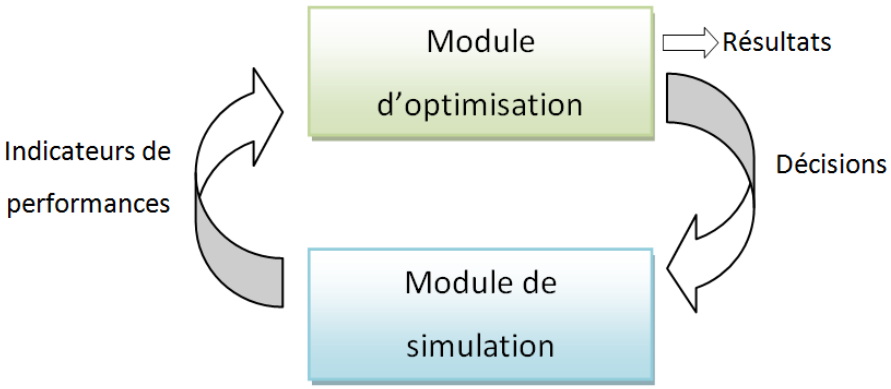


**Figure 5.2. Optimisation basée sur la simulation**

Le module d'optimisation est implémenté avec la métaheuristique NSGA-II. En effet, les algorithmes génétiques représentent la métaheuristique la plus adaptée à l'optimisation basée sur la simulation (Lacksonen 2001).

Le NSGA-II recherche une séquence optimale de $n$ processus de fabrication correspondant aux $n$ unités du lot des unités à manufacturer. Cette séquence est ensuite envoyée au simulateur pour l'évaluer en termes de coût et de temps total. Les résultats obtenus sont renvoyés à nouveau au module d'optimisation. Ce cycle est répété jusqu'à ce qu'une condition d'arrêt soit vérifiée.

### 5.2.1. Environnement de simulation à événements discrets pour les RMSs

Nous avons développé un modèle de simulation à événements discrets adapté aux RMSs. Le modèle est présenté en utilisant le langage de modélisation UML (Unified Modeling Language). Le modèle de simulation à événements discrets est construit sur un ensemble de classes réparties en deux packages en fonction de leur nature. Ces packages sont présentés sur les Figure 5.3 et Figure 5.4, avec un diagramme de classes UML.

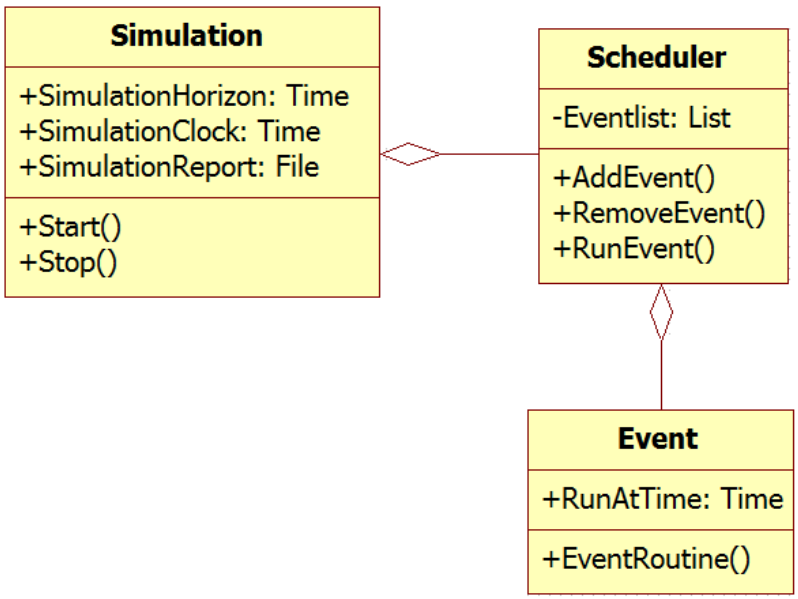


**Figure 5.3. Diagramme de classes du simulateur**

La Figure 5.3 définit le package du cadre général contenant les éléments nécessaires pour lancer la simulation.

La classe de ***Simulation*** est responsable du lancement et du suivi de l'ensemble d'activités de la simulation. Elle peut lancer une simulation, la mettre en pause, l'arrêter, générer le rapport des résultats et contrôler les données nécessaires pour le bon déroulement de la simulation, telles que l'avancement de l'horloge de simulation.

La classe ***Scheduler*** remplit la tâche de gestionnaire des événements de la simulation. Il est principalement composé d'une liste d'événements qui sont exécutés dans un ordre progressif en fonction de leurs temps d'exécution respectifs.

La classe ***Event*** est une classe abstraite, à partir de laquelle tous les événements de simulation sont dérivés. Chaque événement implémente la fonction ***EventRoutine*** en suivant ce qu'il est censé faire. Un évènement est exécuté lorsque l'horloge de la simulation atteint son ***RunAtTime***.

Le deuxième package, présenté dans la Figure 5.4, regroupe l'ensemble des classes nécessaires pour modéliser le RMS étudié. La classe **ReconfigurableMachine** est définie pour représenter un RMT dans le modèle, avec une classe **Link** qui définit une connexion physique utilisée par le produit pour se déplacer entre les machines. La classe **Entity** représente les produits dans le système.

Durant la simulation, des instances de ces différentes classes interagissent entre elles pour imiter le comportement réel d'un RMS.

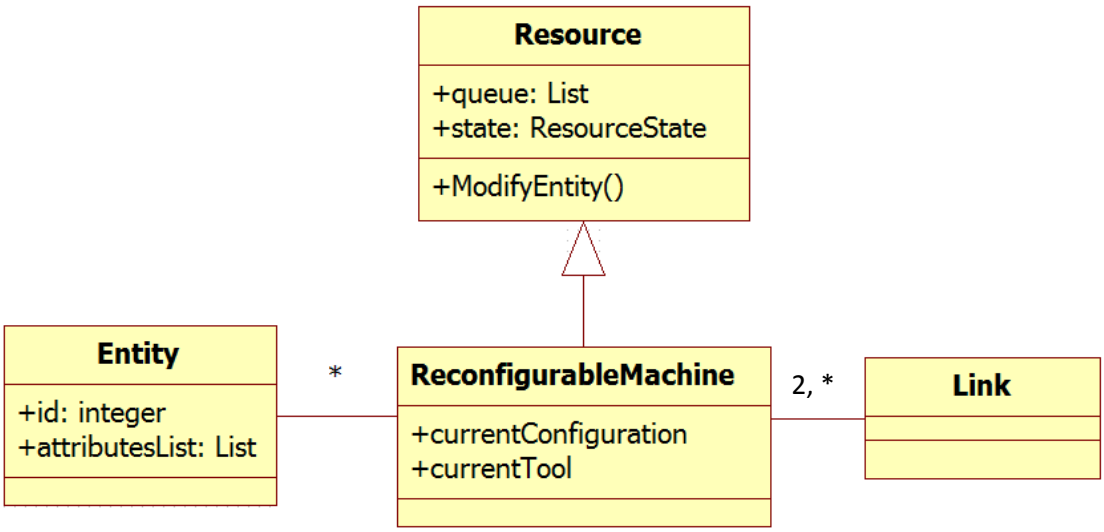


**Figure 5.4. Diagramme de classe de base pour le RMS**

### 5.2.2. Module d'optimisation

Pour ce module d'optimisation nous avons opté pour le NSGA-II avec un codage des solutions en nombres entiers. Chaque gène du chromosome représente une unité de produit et l'ensemble des unités représente ainsi le chromosome. A chaque gène nous associons une valeur entière représentant son rang dans la séquence (le lot).

Par exemple, si un lot d'un produit à manufacturer est composé de 6 unités et l'ensemble Pareto des processus de fabrication contient 8 solutions, alors la figure suivante illustre un exemple de solution possible.

| (1) | (2) | (3) | (4) | (5) | (6) |
|---|---|---|---|---|---|
| 5 | 2 | 8 | 1 | 7 | 7 |

**Figure 5.5. Exemple d'un chromosome**

Le chromosome est interprété de la façon suivante : la première unité du lot sera fabriquée en utilisant le processus de fabrication 5, la deuxième unité avec le processus de fabrication 2, la troisième avec le processus de fabrication 8 et ainsi de suite pour les unités suivantes.

Deux objectifs sont fixés, à savoir la minimisation du temps total et la minimisation du coût total. Le coût total représente simplement la somme des coûts des différents processus de fabrication utilisés pour manufacturer le lot. Le temps total par contre est calculé par la simulation. C'est le temps qui sépare l'entrée de la première unité du lot dans le système, et la sortie de la dernière unité du lot du système.

## 5.3. Expériences numériques et analyses

Nous utilisons les mêmes données que celles du cas d'étude précédent, en ajoutant une taille de lot de 8 unités. Notons que généralement, les lots dans un RMS sont de taille petite ou moyenne, pour cette raison une taille 8 semble raisonnable. Dix (10) processus de fabrication optimaux au sens de Pareto ont été générés par l'exécution d'AMOSA, notés $PPi, (0 \leq i \leq 9)$.

Après le lancement de l'optimisation basée sur la simulation, les résultats suivants ont été obtenus :

| Séquence de processus de fabrication | Coût total | Temps total |
|---|---|---|
| PP2_PP2_PP7_PP7_PP2_PP1_PP1_PP2 | 423889 | 5830 |
| PP2_PP2_PP7_PP7_PP7_PP0_PP2_PP2 | 431919 | 5686 |
| PP5_PP1_PP1_PP1_PP2_PP9_PP8_PP9 | 439482 | 5542 |
| PP4_PP1_PP2_PP2_PP2_PP2_PP8_PP9 | 447037 | 5398 |
| PP5_PP1_PP1_PP1_PP2_PP9_PP8_PP9 | 454585 | 5254 |
| PP4_PP1_PP1_PP1_PP2_PP9_PP8_PP9 | 462140 | 5110 |
| PP4_PP1_PP1_PP1_PP4_PP9_PP8_PP9 | 470146 | 4966 |

**Tableau 5.1. Résultats de l'exemple numérique du cas multi-unité**

Par exemple, la première ligne indique qu'il faut affecter le processus de fabrication $PP2$ à la première unité du lot, $PP2$ à la deuxième unité du lot, $PP7$ à la troisième unité du lot jusqu'à $PP1$ à la septième unité du lot et $PP2$ à la huitième unité. Cette séquence coûte $423889$ unités monétaires et la production dure $5830$ unités de temps.

Chaque ligne du tableau représente une solution optimale. On se retrouve une autre fois avec un front de Pareto car le problème a été traité en multicritère.

À partir du Tableau 5.1 et la Figure 5.6, on peut déduire que même si un processus de fabrication est optimal dans le cas unitaire, il n'est pas nécessairement optimal lorsque nous l'utilisons pour chaque unité d'un lot d'unités à manufacturer.

Nous présentons sur la Figure 5.6 les solutions du Tableau 5.1. Ainsi que des résultats obtenus avec un seul processus de fabrication répété pour chaque unité du lot (sur la figure en points indépendants). Un seul processus de fabrication répété signifie que toutes les unités dans le lot sont manufacturées en utilisant ce même processus de fabrication. Chaque processus de fabrication a été testé et tous les résultats montrent une amélioration considérable avec la nouvelle approche.

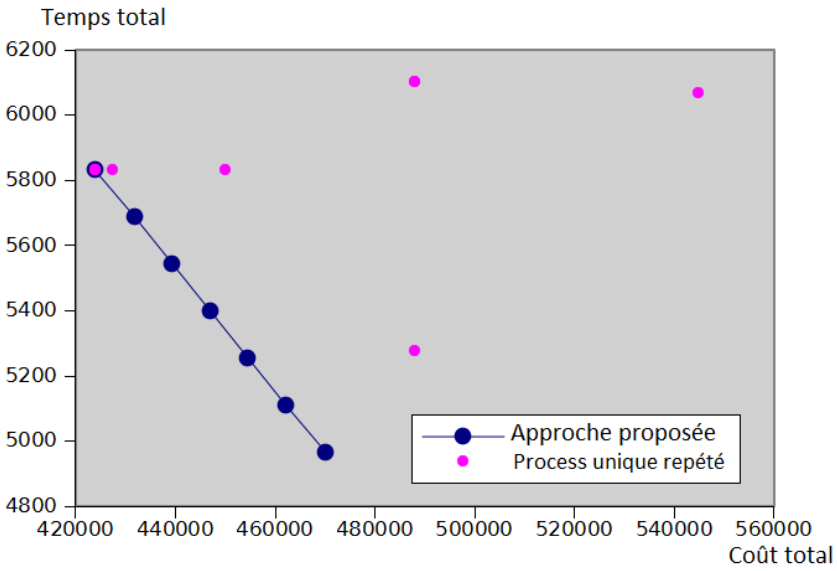


**Figure 5.6. Approche proposée vs. Processus de fabrication redondant**

## 5.4. Conclusion

Dans ce chapitre nous avons présenté une approche pour la génération des processus de fabrication dans un cas multi-unité où un nombre d'unités du même produit doivent d'être fabriquées sur le même RMS. En effet, les ressources sont partagées entre les différentes entités (du même produit) qui coexistent sur le même système de production.

Le problème a été résolu en utilisant une approche d'optimisation basée sur la simulation, où la fonction objectif est remplacée par un ou plusieurs modèles de simulation, et les variables de décision sont les conditions sous lesquelles la simulation est exécutée. Un environnement de simulation à évènements discrets à été développé, et sa modélisation illustrée sous forme de diagramme de classes UML.

Les résultats ont démontré que l'utilisation d'un processus de fabrication unique pour toutes les unités de produit réalisées dans le RMS ne garantit pas une production optimale en termes de temps total, même si le processus de fabrication est localement optimal. En effet, les ressources partagées de l'atelier créent une concurrence d'utilisation, ce qui engendre des retards dans la production.

Nous avons démontré que même avec des unités du même produit en concurrence sur un RMS, l'utilisation d'un seul unique processus de fabrication pour toutes les unités ne garanti pas l'optimalité de la production.

# Chapitre 6.

## Problème 3 : Intégration des fonctions de génération des processus de fabrication et d'ordonnancement dans un RMS

---

*Ce chapitre présente une approche heuristique pour l'intégration des fonctions de génération des processus de fabrication et d'ordonnancement dans le cas d'un RMS. L'heuristique proposée tient compte de la nature multi-configuration des différentes machines. Pour illustrer l'applicabilité et l'efficacité de l'heuristique proposée, un exemple numérique est présenté où l'heuristique est comparée à une approche séquentielle classique (non intégrée), et ceci en utilisant le cadre de simulation à événements discrets développé dans le chapitre précédent.*

---

## 6.1. Description du problème

La fonction d'ordonnancement consiste à affecter les ressources nécessaires aux opérations indiquées dans le processus de fabrication afin d'atteindre un objectif bien défini, tel que le respect des délais de production. Notons que l'ordonnancement est accompli généralement après la génération du processus de fabrication. En effet, en plus des restrictions imposées par les ressources, il doit satisfaire les contraintes imposées par le processus de fabrication.

En production, les deux fonctions de génération des processus de fabrication et d'ordonnancement se chargent de l'allocation et l'utilisation des ressources. Un processus de fabrication est généré sans prise en compte de la disponibilité des ressources, car lors de sa génération, les informations relatives à l'état du système ainsi que la disponibilité des ressources ne sont pas encore disponibles. Cette limitation a été clairement mise en évidence dans le cas multi-unité précédemment étudié. Nous avons démontré que même si un processus de fabrication est optimal, il est possible qu'il devienne inefficace lorsque la question de la disponibilité des ressources est posée.

En plus, le processus de fabrication et l'ordonnancement peuvent prendre en compte des objectifs différents durant leur recherche du meilleur processus de fabrication et meilleur ordonnancement respectivement. Ceci complique la recherche de solutions satisfaisantes avec une exécution séquentielle (en deux temps) de la génération des processus de fabrication et l'ordonnancement.

Par ailleurs, même si un processus de fabrication est généré d'une manière optimale, il n'est pas à l'abri de modifications afin qu'il soit conforme aux besoins de la phase d'ordonnancement. Ces modifications peuvent engendrer des processus de fabrication approximativement respectés, ce qui rend la collaboration entre la génération des processus de fabrication et l'ordonnancement inefficace et peu performante.

L'intégration de la fonction de génération des processus de fabrication avec l'ordonnancement en une seule fonction se propose comme une alternative efficace, car contrairement à l'approche séquentielle, l'approche intégrée relaxe la contrainte d'unicité du processus de fabrication, et l'association définitive des opérations aux machines. Elle repose sur le fait que les produits peuvent être réalisés en utilisant des processus de fabrication différents. Les processus de fabrication alternatifs sont possibles grâce au :

1) La flexibilité du process : il s'agit de la possibilité d'accomplir une opération sur des différentes des machines, configurations et avec des outils différents.

2) La flexibilité de la séquence des opérations : c'est la possibilité de modifier l'ordre dans lequel les opérations sont accomplies. En effet, à partir d'un graphe de précédence, plusieurs séquences faisables peuvent être adoptées (Figure 4.1)

Dans le cadre de ce travail, nous proposons une stratégie qui consiste à intégrer les deux activités de génération des processus de fabrication et ordonnancement. Cette intégration vise à garantir une meilleure utilisation des ressources, pour améliorer ainsi les coûts générés et les durées requises.

## 6.2. Etat de l'art de l'intégration de la fonction de génération des processus de fabrication et l'ordonnancement

L'idée d'intégration a été adressée la première fois par Khoshnevis et Chen (Khoshnevis & Chen 1991; Chen & Khoshnevis 1993). Ils ont présenté les principaux problèmes rencontrés lors de l'intégration des deux fonctions, et ont identifié les avantages potentiels de l'intégration. Saygin et Kilic (Saygin & Kilic 1999) ont élaboré un framework d'intégration pour des processus de fabrication flexibles et l'ordonnancement offline dans un FMS. Le framework est constitué de quatre niveaux d'intégration qui sont respectivement le module de sélection de la machine-outil, le module de sélection du processus de fabrication, et le module l'ordonnancement et ré-ordonnancement, afin de minimiser le temps total de production. Zhang et al. (Zhang et al. 2003) ont élaboré une nouvelle approche d'intégration du processus de fabrication et l'ordonnancement pour le cas de la production par lot de pièces prismatiques. Deux modules sont considérés, à savoir le module de génération des processus de fabrication et le module d'ordonnancement, liés par un « animateur intelligent ». Cet animateur fournit un feedback intuitif pour le module de génération des processus de fabrication sous la forme de contraintes supplémentaires basé sur les mesures de performance de l'ordonnancement. Plusieurs fonctions objectifs telles que le taux d'utilisation des machines et le délai de production sont utilisées pour l'évaluation du l'ordonnancement généré. Jain et al. (Jain et al. 2006) ont proposé un plan pour l'intégration qui peut être mis en œuvre dans une entreprise avec des départements de processus de fabrication et d'ordonnancement existantes, sans pour autant perturber leurs fonctionnements, ceci en utilisant de multiples processus de fabrication pour chaque type de produit. Le plan proposé tire partie de cette multiplicité de processus de

fabrication en se basant sur une stratégie d'ordonnancement en temps réel, adéquate aux ateliers dynamiques. Li et al. (Li et al. 2008) ont développé une approche basée sur les algorithmes génétiques pour l'intégration utilisant un schéma de codage et des opérateurs spécifiques. Ce schéma a été développé pour améliorer la qualité de la solution. Guo et al. (Guo et al. 2009) ont considéré l'intégration en tant qu'un problème combinatoire, et ont utilisé l'algorithme des essaims de particules (*particle swarm optimization*). Li et al. (Li et al. 2010) ont proposé un modèle mathématique pour l'intégration de la fonction de génération des processus de fabrication et l'ordonnancement. Une approche basée sur les algorithmes évolutionnaires a été développée pour le résoudre le modèle mathématique.

Plus récemment, Mohammadi et al. (Mohammadi et al. 2012) on développé un modèle mathématique en nombres entiers pour représenter le processus de fabrication et l'ordonnancement, avec une vision multicritère. Une adaptation hybride du recuit simulé a été utilisée pour la résolution du modèle développé. Li et al. (Li et al. 2012) se sont focalisés sur une approche multicritère basée sur la théorie des jeux pour l'intégration. Leurs objectifs étaient la minimisation du temps total, le taux d'utilisation des machines et l'équilibrage des charges entre machines. Lian et al (Lian et al. 2012) ont présenté un algorithme compétitif impérialiste pour résoudre le problème de génération des processus de fabrication et l'ordonnancement d'une manière intégrée avec comme objectif de minimiser le makespan. Mohapatra et al. (Mohapatra et al. 2013) ont utilisé une technique de recherche intelligente appelée « système immunitaire artificiel » pour l'intégration en donnant une attention particulière au planning de changement (setup plannings). Phanden et al. (Phanden et al. 2012) ont proposé une approche itérative cyclique pour intégrer la fonction de génération des processus de fabrication et l'ordonnancement. L'approche consiste en quatre modules: module de sélection de processus de fabrication, le module d'ordonnancement, le module d'analyse de l'ordonnancement et le module de modification du processus de fabrication.

Dans tous les travaux cités ci-dessus, l'intégration a été considérée plus avantageuse que la prise en compte des fonctions de génération des processus de fabrication et l'ordonnancement en tant que fonctions séparées. Ces travaux concernent principalement des job shops avec des machines classiques, et à notre connaissance, le problème de l'intégration n'a pas été clairement abordé dans un environnement reconfigurable. Par conséquent, nous proposons dans ce chapitre une heuristique pour l'intégration de la fonction de génération des processus de fabrication et l'ordonnancement dans un RMS, afin d'explorer les opportunités ainsi que les problèmes qui peuvent être rencontrés. L'heuristique proposée prend en compte les

caractéristiques des RMSs telles que la nature reconfigurable du système et des machines, sur le niveau logique et physique, ainsi que la capacité paramétrable des machines reconfigurables

## 6.3. Heuristique proposée

Les approches traditionnelles qui traitent la fonction de génération des processus de fabrication et l'ordonnancement en tant que deux activités dissociées peuvent donner lieu à des solutions optimales mais localement dans pour chacune des deux phases. L'intégration de ces deux activités en un seul problème d'optimisation avec la considération des contraintes de chacune d'elles augmente nettement la taille de l'espace de recherche, ce qui rend difficile l'utilisation des méthodes exactes.

Pour faire face à cette difficulté, nous proposons une nouvelle heuristique permettant une intégration efficace de la de la fonction de génération des processus de fabrication avec l'ordonnancement dans un environnement reconfigurable.

Dans une stratégie séquentielle, le principal critère utilisé pour affecter les opérations aux machines est le temps opératoire. Dans une stratégie intégrée, en plus du temps opératoire, l'affectation doit considérer la disponibilité des machines, puisqu'elles sont partagées entre toutes les unités circulant dans le système.

Pour inclure la disponibilité des machines, nous définissons un index de sélection $SI$ (Selection Index) pour décider quelle opération l'ordonnancer. L'index $SI$ est calculé pour chaque opération non encore ordonnancée, en utilisant les informations des machines reconfigurables pouvant effectuer cette opération. Ces informations sont représentées par un paramètre de disponibilité $AT$ (Availability Time). Ce deuxième paramètre donne le temps nécessaire pour qu'une machine donnée soit disponible, c'est-à-dire libre de toute opération.

En effet, à un instant donné de la production, une machine reconfigurable du système peut être occupée à réaliser une opération. Dans d'autres cas, elle peut avoir des produits en attente dans sa file d'attente. Par conséquent, la valeur du paramètre de disponibilité $AT$ représente la somme des temps opératoires, les temps de changements des configurations et des outils pour chaque opération dans la machine (cela concerne l'opération en cours et les opérations en attente). Ainsi, le paramètre $AT$ nous donne l'information : « à quel moment une machine particulière sera disponible avec l'état actuel des ressources ».

Le paramètre $AT$ est calculé pour chaque machine $m$ selon la formule ( 6.1 )

$$AT_m = \sum_{k=1}^{NQ_m} (PrTime[M_m(k)][C_m(k)][T(k)] + CCT[C_m(k)][C_m(k+1)] + TCT[T_m(k)][T_m(k+1)])$$

**( 6.1 )**

Avec

- $AT_m$ : temps de disponibilité de la machine $m$
- $NQ_m$ : nombre de produits en attente dans la file d'attente de la machine $m$
- $PrTime[M_m(k)][C_m(k)][T(k)]$ : temps opératoire nécessaire par l'opération du produit $k$
- $CCT[C_m(k)][C_m(k+1)]$ : temps de reconfiguration.
- $TCT[T_m(k)][T_m(k+1)]$ : temps de changement de l'outil.

Le paramètre $AT_m$ est en suite utilisé pour calculer un deuxième paramètre, le $OPTime_{op}$, qui indique le temps prévu de fin d'une opération particulière $op$ si elle est affectée à une machine particulière $m$ :

$$OPTime_{op}(m) = AT_m + PrTime[M_m(op)][C_m(op)][T(op)] + CCT[C_m(last)][C_m(op)] + TCT[T_m(last)][T_m(op)]$$

**( 6.2 )**

$OPTime_{op}(m)$ représente le temps nécessaire pour effectuer l'opération $op$ si elle est affectée à la machine $m$. Il est égal à la somme du temps opératoire, le temps de changement entre les besoins de l'opération $op$ et la configuration/outil utilisés par la dernière opération ($last$) à effectuer sur la machine $m$.

Finalement, nous calculons la moyenne $AverageAT_{op}$ des machines comme suite :

$$AverageAT_{op} = \frac{\sum_{i \in M(op)} OPTime_{op}(i)}{|M(op)|}$$

**( 6.3 )**

Avec

- $M(op)$ : l'ensemble des machines pouvant effectuer l'opération $op$.

Les paramètres $AverageAT_{op}$ et $OPTime_{op}$ sont utilisés pour calculer l'index $SI_{op}$ pour chaque opération non encore ordonnancée. Le $SI_{op}$ d'une opération $op$ est la différence entre la moyenne $AverageAT_{op}$ et le minimum des $OPTime_{op}(m), m \in M(op)$. Cette différence est divisée par $AverageAT_{op}$ pour avoir des valeurs normalisées des $SI_{op}$, de sorte que la comparaison entre les $SI_{op}$ ne soit pas biaisée. Ce calcul est exprimé comme suit :

$$SI_{op} = \frac{AverageAT_{op} - \min\left(OPTime_{op}(m)\right)}{AverageAT_{op}}, m \in M(op) \tag{6.4}$$

Avec

- $SI_{op}$ : index de sélection de l'opération $op$

Remarquons que le $SI$ d'une opération $op$ est plus élevé lorsque l'état actuel du système permet à $op$ d'être accomplie dans un temps court si la machine avec le minimum $OPTime_{op}(m)$ est utilisée ($m \in M(op)$). Quand un $SI$ d'une opération $op$ est élevé ceci implique qu'il existe une machine $m^*$ des machines pouvant effectuer l'opération $op$, c'est-à-dire $m^* \in M(p)$, telle que $m^*$ est relativement rapide par rapport aux autres machines. Notons que le mot « rapide » est utilisé pour indiquer temps d'achèvement de l'opération $OPTime_{op}(m)$, et non la vitesse d'usinage.

Dès que les $OPTime_{op}$ et les $SI$ sont calculés pour l'ensemble des opérations qui ne sont pas encore programmées, l'opération $op^*$ avec un $SI_{op^*}$ le plus élevé sera ordonnancée sur la machine qui donne le plus petit $OPTime_{op^*}(m)$, $m \in M(op)$. L'algorithme de l'heuristique est présenté sous forme d'organigramme sur la Figure 6.1.

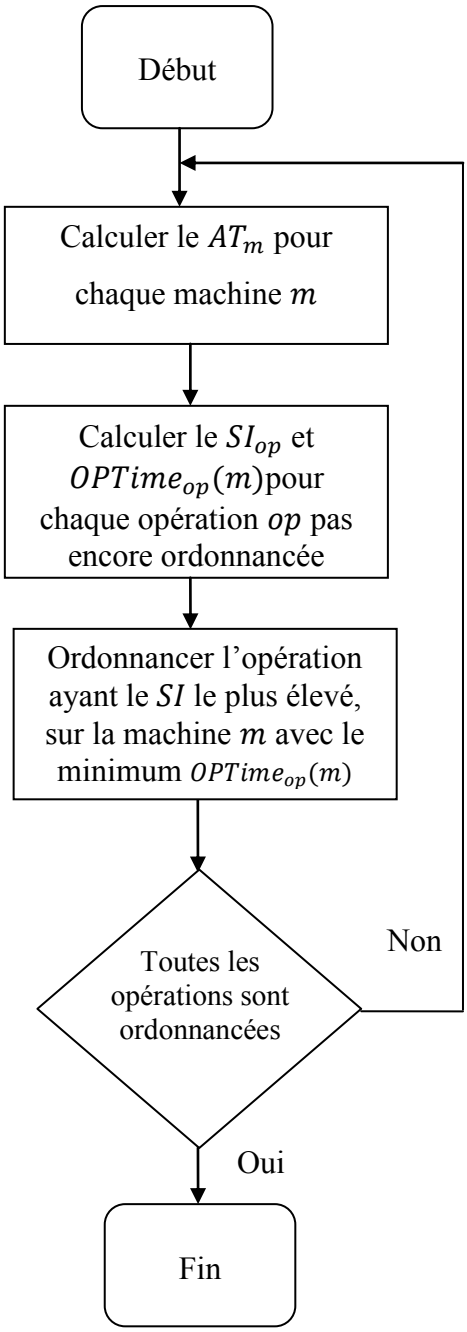


**Figure 6.1. Organigramme résumant l'approche proposée**

Le choix de l'opération avec le plus grand $SI$ se repose sur l'idée suivante : chaque opération $op$ peut être effectuée par au moins une machine reconfigurable dans le système. Dans le cas où plusieurs machines sont capables d'accomplir cette opération, une décision devrait être prise afin de choisir la machine la plus adéquate en termes de délai. Cette sélection dépend de l'état dynamique du RMS ainsi que les disponibilités des RMTs. Pour chaque opération parmi les opérations $op$ non ordonnancées, il existe une machine $m^*$ parmi les machine dans $M(op)$ capable de l'accomplir dans un temps minimal par rapport aux autres machines, cette machine

$m^*$ possède le plus petit $OPTime_{op}$. Ainsi, si on souhaite ordonnancer op, $m^*$ serait la meilleure candidate.

Maintenant que nous connaissons pour chaque opération $op$, la machine qui doit être utilisée si $op$ sera ordonnancée, la question principale qui se pose est quelle opération ordonnancer ? La sélection des opérations est basée sur le paramètre $SI$. Si une opération a un $SI$ plus élevé par rapport aux autres opérations candidates, elle sera alors ordonnancée immédiatement. En effet, le SI est élevé pour une opération particulière $op$ quand $OPTime_{op}(m^*)$ est relativement très petit par rapport à la moyenne $AverageAT_{op}$, c'est-à-dire que par rapport aux autres opérations, $op$ a la possibilité d'être accompli dans des délais relativement plus court puisqu'il existe une machine pouvant l'accomplir rapidement comparée aux autres machines.

Une opération $op$ avec un $SI$ faible signifie qu'elle peut être ordonnancée sur n'importe quelle machine candidate sans que le temps d'accomplissement ne soit significativement réduit. De ce fait, cette opération $op$ peut attendre un cycle supplémentaire sans qu'une opportunité de l'achever rapidement ne soit ratée.

## 6.4. Expériences numériques et analyses

Nous considérons un exemple numérique pour illustrer l'applicabilité de l'heuristique proposée. Les résultats sont comparés à une approche séquentielle classique. Avant d'entrer dans les détails de l'exemple numérique, nous présentons dans la suite une implémentation d'une approche séquentielle qui consiste en deux étapes principales, respectivement la génération des processus de fabrication et l'ordonnancement des opérations.

Initialement, pour chaque type de produit, l'approche séquentielle classique génère un plan de traitement. Nous utilisons l'approche proposée dans le chapitre 4 pour générer les processus de fabrication. Ensuite, sur la base des processus de fabrication générés, les opérations seront ordonnancées. L'ordonnancement est effectué par un algorithme génétique basé sur la simulation. Le modèle de simulation est construit en se basant sur l'environnement de simulation précédemment présenté (Section 5.2.1).

Mettre en œuvre un algorithme génétique, comme presque toutes les autres métaheuristiques, commence par la phase de codage. Le codage utilisé pour l'algorithme génétique est présenté sur la Figure 6.2, où le génome est défini comme une séquence d'opérations.

| | | | | | |
|---|---|---|---|---|---|
| O1P1 | O1P2 | O2P1 | O3P1 | O2P2 | O4P1 |

**Figure 6.2. Un génome correspondant à un ordonnancement**

Chaque gène contient l'identifiant de l'opération de son produit correspondant. Par exemple, le gène O1P1 représente l'opération numéro 1 du produit 1. En suite, la population évolue en utilisant les opérateurs génétiques qui sont le croisement et la mutation. La progéniture de cette combinaison est sélectionnée sur la base d'une fonction fitness de durée minimale donnée par la simulation.

Les données d'entrée sont constituées de deux types de produits, dont chacun possède des relations de précédence entre ses opérations, TADs, et outils nécessaires. Les RMSs sont conçus pour des familles de produit, par conséquent un RMS n'est pas dédié à une production où la variété des produits est importante. Selon Koren (Koren 2010), un ou deux produits peuvent exister simultanément dans un système de production RMS.

Pour explorer objectivement le potentiel de l'heuristique proposée, de nombreux cas sont générés aléatoirement, en générant le graphe de précédence des opérations ainsi que les TADs et les outils nécessaires. Les graphes de précédence sont générés selon les étapes suivantes :

1) Générer un nombre d'opérations entre 5 et 20 (selon la littérature, 5-20 est un intervalle assez représentatif) ;

2) Classer les opérations dans un certain nombre de niveaux allant de 4 à 15 niveaux, afin d'avoir une densité plus ou moins équilibrée sur les niveaux.

3) Des relations de précédence sont créées aléatoirement entre les opérations en tenant compte du niveau de chaque opération. Les liens de précédence sont établis à partir d'un niveau supérieur vers un niveau inférieur, afin d'éviter les cycles dans le graphe.

Les paires de produits générés sont ensuite ordonnancées en utilisant l'approche classique et l'heuristique proposée. Les résultats obtenus sont comparés.

Figure 6.3 et Figure 6.4 présentent les graphes de précédence générés pour les deux produit (Produit A et produit B) de la première instance du cas d'étude. Les processus de fabrication générés pour ces deux produits sont présentés sur Tableau 6.1et Tableau 6.2 respectivement.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Caractéristique | F1 | F1 | F1 | F1 | F2 | F1 | F3 | F2 | F3 | F3 | F3 | F3 |
| Operation | O1 | O5 | O2 | O4 | O1 | O3 | O1 | O2 | O2 | O3 | O4 | O5 |
| Machine | M3 | M2 | M2 | M3 | M1 | M3 | M1 | M3 | M3 | M3 | M2 | M2 |
| Configuration | C1 | C2 | C2 | C1 | C1 | C1 | C3 | C1 | C1 | C2 | C2 | C3 |
| Outil | T6 | T6 | T3 | T7 | T7 | T2 | T8 | T2 | T9 | T10 | T1 | T5 |

**Tableau 6.1. Processus de fabrication généré pour le produit A**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Caractéristique | F2 | F1 | F1 | F2 | F2 | F2 | F1 | F2 |
| Operation | O1 | O1 | O3 | O2 | O3 | O4 | O2 | O5 |
| Machine | M3 | M3 | M3 | M1 | M1 | M3 | M3 | M3 |
| Configuration | C1 | C1 | C1 | C1 | C1 | C1 | C1 | C2 |
| Outil | T6 | T6 | T2 | T7 | T8 | T9 | T7 | T2 |

**Tableau 6.2. Processus de fabrication généré pour le produit B**

Une fois les processus de fabrication sont générés, les opérations de produits sont alors prêtes à être ordonnancées. L'ordonnancement est obtenu en utilisant l'algorithme génétique présenté ci-dessus.

Notre heuristique est appliquée sur les mêmes instances, et les résultats globaux sont résumés dans le Tableau 6.3. Le gap entre le Makespan de l'approche classique et notre heuristique proposée est calculé selon la formule $\left(\frac{makespan\ approche\ classique - makespan\ heuristique}{makespan\ approche\ classique}\right)$. Le gap varie de 8,7% à 20,6%, en maintenant des valeurs positives pour toutes les instances, ce qui démontre une amélioration du Makespan global utilisant notre heuristique, et par conséquent, un délai de fabrication plus performant. Cette amélioration est due à l'exploitation de la reconfigurabilité des RMTs, via l'intégration entre la génération des processus de fabrication et l'ordonnancement, ce qui relaxe l'affectation des ressources dans le cadre du processus de fabrication, et permet une exploration plus libre de l'espace de recherche.

L'approche classique a l'avantage d'effectuer les fonctions de génération des processus de fabrication et l'ordonnancement des processus en mode hors ligne, avant le lancement de la production, ce qui libère le système de fabrication de toute activité de calcul lourd pendant la phase d'exploitation. Une approche intégrée généralement n'offre pas cette caractéristique, il est donc important dans ce cas de considérer l'effort de calcul. Le Temps de calcul de notre heuristique montre qu'elle ne nécessite pas beaucoup de temps par rapport au Makespan, ce qui est très nécessaire pour l'application dans un environnement de fabrication réaliste.

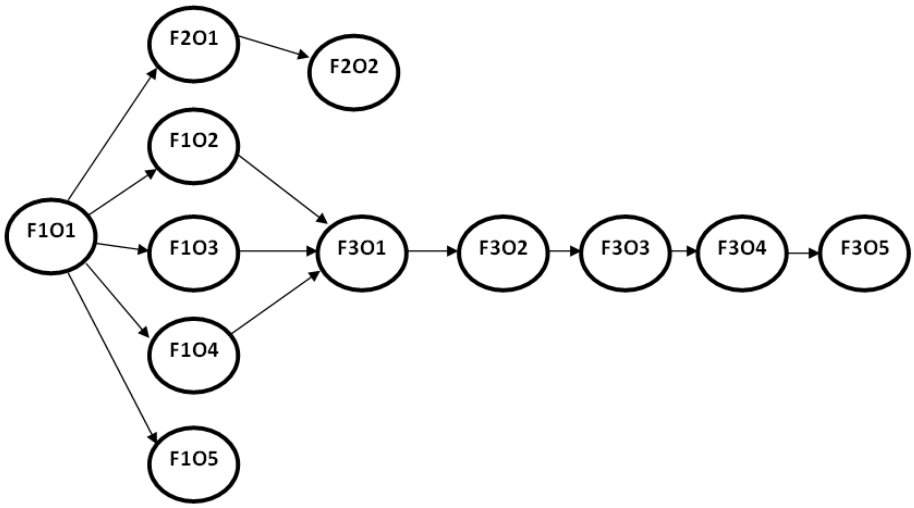


**Figure 6.3. Graphe de précédence généré pour un produit A**

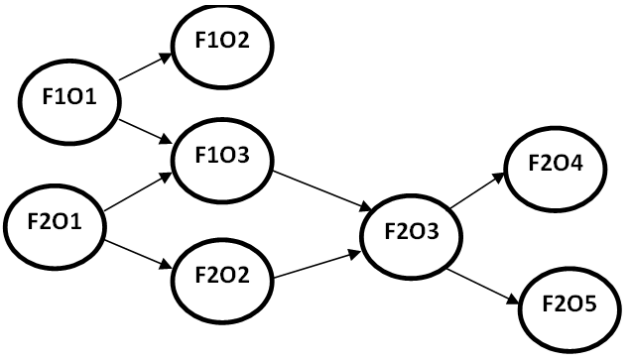


**Figure 6.4. Graphe de précédence généré pour un produit B**

## 6.5. Conclusion

Dans ce chapitre nous avons proposée une heuristique pour résoudre le problème d'intégration de la fonction de génération des processus de fabrication et l'ordonnancement dans le cadre d'un environnement reconfigurable. Les RMSs sont conçus pour être réactifs aux changements des marchés. Pour cela, il est nécessaire d'étendre au maximum l'exploitation des capacités des machines reconfigurables. Les performances de l'heuristique et d'une approche séquentielle classique ont été comparées, et les résultats ont démontré l'efficacité de l'heuristique sur l'ensemble des instances étudiées.

| | Makespan obtenu par l'approche classique séquentielle | Makespan obtenu par l'heuristique proposée | Temps de calcul de l'heuristique proposée | Gap $\left(\frac{classique - heuristique}{classique}\right)$ |
|---|---|---|---|---|
| Instance 1 | 3032 | 2602 | 2.3 | 16.5% |
| Instance 2 | 14290 | 11864 | 4.2 | 20.4% |
| Instance 3 | 7387 | 6481 | 3.7 | 14.0% |
| Instance 4 | 5521 | 4702 | 3.6 | 17.4% |
| Instance 5 | 10362 | 8754 | 4.0 | 18.4% |
| Instance 6 | 2941 | 2670 | 2.5 | 10.1% |
| Instance 7 | 3799 | 3149 | 2.6 | 20.6% |
| Instance 8 | 8814 | 8107 | 3.6 | 8.7% |
| Instance 9 | 5586 | 4852 | 3.0 | 15.1% |
| Instance 10 | 3546 | 3117 | 2.0 | 13.8% |

**Tableau 6.3. Résultats obtenus par l'approche classique et par notre heuristique**

Un autre avantage de l'heuristique proposée est le fait qu'elle s'applique en temps réel. En effet, dans un environnement de production réel, les machines sont vulnérables aux pannes, et parfois dans un moment imprévisible. L'heuristique proposée considère les machines disponibles à chaque moment de la production, donc une machine en panne pourra être tout simplement ignorée lors de l'affectation des opérations aux machines, et par conséquent cette panne ne remet pas en cause le fonctionnement de l'heuristique.

# Conclusion générale

Les travaux de recherche présentés dans cette thèse concernent les problèmes de génération de processus de fabrication dans le cadre des systèmes manufacturiers reconfigurables. Le système manufacturier RMS représente une classe récente des systèmes manufacturiers. Il nécessite des approches et des méthodes prenant en compte ses diverses spécificités qui le distinguent des autres systèmes de production et qui justifient le développement de nouvelles méthodes de gestion, incluant nécessairement la reconfigurabilité dans ses différents aspects.

Le processus de fabrication dépend du système manufacturier sur lequel il est généré. Or un RMS est caractérisé par la capacité de ses machines à changer de configuration. De ce fait, il existe un réel besoin à développer des approches qui prennent en compte cette reconfigurabilité des machines dans la génération des processus de fabrication.

Motivés par cet aspect, nous avons abordé dans ce travail trois problématiques liées à la génération des processus de fabrication :

1. La génération des processus de fabrication dans un cas unitaire, où nous avons développé une approche multicritère qui génère un processus de fabrication optimal pour un type de produit. Un modèle mathématique multi-objectif à été conçu, et deux métaheuristiques multicritère, NSGA-II et AMOSA, ont été utilisées pour résoudre ce modèle et trouver des solutions optimales. Les expériences numériques ont démontré de meilleures performances pour AMOSA. Nous avons proposé également deux extensions à ce premier problème, où nous avons utilisé dans la première extension TOPSIS (Technique for Order Performance by Similarity to Ideal Solution) afin de permettre aux décideurs de sélectionner parmi les solutions Pareto optimales, la solution qui convient au mieux à leurs besoins. Dans la deuxième extension, nous

avons proposé une adaptation de l'approche de génération des processus de fabrication pour qu'elle soit utilisée dans le cadre de la conception des RMSs, plus précisément lors du choix des machines constituant l'atelier du RMS.

2. La génération des processus de fabrication dans un RMS dans le cas multi-unité, ou plusieurs unités du même produit sont fabriquées dans un RMS. Un framework de simulation à événements discrets adapté aux systèmes reconfigurables à été développé et utilisé dans le cadre d'une optimisation basée sur la simulation, visant à résoudre le problème.

3. L'intégration des fonctions de génération des processus de fabrication avec l'ordonnancement, où nous avons proposé une heuristique afin de réaliser cette intégration dans un cadre reconfigurable.

Toutes les approches proposées ont été validées par des expériences numériques, démontrant leur applicabilité et efficacité.

Ce travail de recherche nous a permis de dégager plusieurs directions futures. Afin de refléter un contexte économique réel, nous souhaitons introduire des demandes avec des quantités et date dues. Dans la réalité, chaque entreprise de production reçoit des demandes provenant de ses clients d'une manière plus ou moins aléatoire. Pour répondre à ces conditions, un processus de fabrication doit être flexible pour pouvoir répondre à ces conditions, nous envisageons d'adapter notre approche de génération de processus de fabrication pour qu'elle considère les fluctuations des demandes en créant un processus de fabrication flexible facilement modifiable et adaptable en cas de nécessité.

Une autre direction de recherche consiste à inclure la vulnérabilité des machines aux pannes. En effet, les pannes des machines constituent un événement non rare dans les ateliers de fabrication. Une machine en panne peut remettre en cause un processus de fabrication s'il n'a pas été soigneusement généré en prenant en compte la disponibilité incertaine des machines reconfigurables.

Pour s'approcher encore plus de la réalité, nous planifions de prendre en compte d'autres objectifs lors de la génération des processus de fabrication. Même si le coût et le temps constituent des objectifs primordiaux pour le processus de fabrication, d'autres objectifs tels que l'équilibre des charges entre les machines, la qualité du produit réalisé, et l'impact environnemental peuvent être pris en compte.

Pour l'heuristique que nous avons proposée pour l'intégration de la fonction de générations des processus de fabrication avec l'ordonnancement dans un RMS, nous souhaitons approfondir l'étude de l'intégration dans le cadre des RMSs, en vue d'aboutir à un modèle mathématique qui permettra de valider l'heuristique. Enfin, notre souhait est de pouvoir valider nos algorithmes de calcul en utilisant des données réelles.

# Références bibliographies


Abbasi, M. & Houshmand, M., 2010. Production planning and performance optimization of reconfigurable manufacturing systems using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 54(1-4), pp.373–392.

Alting, L. & Zhang, H., 1989. Computer aided proces planning: the state-of-the-art survey. *International Journal of Production Research*, 27(4), pp.553–585.

Ashraf, A.M., 2006. *Optimal configuration selection for seconfigurable manufacturing systems*. Thesis.

Azab, A. & ElMaraghy, H.A., 2007. Mathematical modeling for reconfigurable process planning. *CIRP Annals - Manufacturing Technology*, 56(1), pp.467–472.

Bandyopadhyay, S. et al., 2008. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 12(3), pp.269–283.

Borisovsky, P.A., Delorme, X. & Dolgui, A., 2013. Genetic algorithm for balancing reconfigurable machining lines. *Computers & Industrial Engineering*, 66(3), pp 541–547.

Bratcu, A.I., Dolgui, A. & Belmokhtar, S., Reconfigurable transfer lines cost optimization - a linear programming approach. In *2005 IEEE Conference on Emerging Technologies and Factory Automation*. IEEE, pp. 625–632.

Chang, T.-C. & Wysk, R.A., 1985. *An introduction to automated process planning*, Prentice-Hall.

Chaube, A., Benyoucef, L. & Tiwari, M.K., 2012. An adapted NSGA-2 algorithm based dynamic process plan generation for a reconfigurable manufacturing system. *Journal of Intelligent Manufacturing*, 23(4), pp. 1141-1155.

Chen, Q. & Khoshnevis, B., 1993. Scheduling with flexible process plans. *Production Planning & Control : The Management of Operations*, 4(4), pp.333–343.

David, G., 1989. *Genetic algorithms in search, optimization, and machine learning.*, Addison-Wesley.

Deb, K. et al., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), pp.182–197.

Deb, K., 1999. An introduction to genetic algorithms. *Sadhana*, 24, pp.293–315.

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

Donoso, Y. & Fabregat, R., 2007. *Multi-objective optimization in computer networks using metaheuristics*, Auerbach publications.

Dou, J., Dai, X. & Meng, Z., 2011. A GA-based approach for optimizing single-part flow-line configurations of RMS. *Journal of Intelligent Manufacturing*, 22(2), pp.301–317.

Elmaraghy, H.A., 2007. Reconfigurable process plans for responsive manufacturing systems. In *Digital Enterprise Technology*. Springer US, pp. 35–44.

Essafi, M., Delorme, X. & Dolgui, A., 2012. A reactive GRASP and Path Relinking for balancing reconfigurable transfer lines. *International Journal of Production Research*, 50(18), pp.5213–5238.

Galan, R. et al., 2007. A systematic approach for product families formation in Reconfigurable Manufacturing Systems. *Robotics and Computer-Integrated Manufacturing*, 23(5), pp.489–502.

Goyal, K.K., Jain, P.K. & Jain, M., 2012, Optimal configuration selection for reconfigurable manufacturing system using NSGA II and TOPSIS. *International Journal of Production Research*, 50(15), pp. 4175-4191.

Guan, X. et al., 2012. A revised electromagnetism-like mechanism for layout design of reconfigurable manufacturing system. *Computers & Industrial Engineering*, 63(1), pp.98–108.

Gumasta, K. et al., 2011. Developing a reconfigurability index using multi-attribute utility theory. *International Journal of Production Research*, 49(6), pp.1669–1683.

Guo, Y.W. et al., 2009. Applications of particle swarm optimisation in integrated process planning and scheduling. *Robotics and Computer-Integrated Manufacturing*, 25(2), pp.280–288.

Haupt, R.L. & Haupt, S.E., 2004. *Practical genetic algorithms*, John Wiley & Sons.

Heisel, U. & Meitzner, M., 2006. Progress in reconfigurable manufacturing systems. In A. I. Dashchenko, ed. *Reconfigurable Manufacturing Systems and Transformable Factories*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 47–62.

Holland, J.H., 1975. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press.

Hwang, C.L. & Masud, A.S.M., 1979. Multiple objective decision making—methods and applications: A state-of-the-art survey. *Lectures Notes in Economics and Mathematical Systems*, 164.

Jain, A., Jain, P.K. & Singh, I.P., 2006. An integrated scheme for process planning and scheduling in FMS. *The International Journal of Advanced Manufacturing Technology*, 30(11-12), pp.1111–1118.

Jones, D., Mirrazavi, S. & Tamiz, M., 2002. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1), pp.1–9.

Khoshnevis, B. & Chen, Q.M., 1991. Integration of process planning and scheduling functions. *Journal of Intelligent Manufacturing*, 2(3), pp.165–175.

Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P., 1983. Optimization by simulated annealing. *Science (New York, N.Y.)*, 220(4598), pp.671–680.

Koren, Y., 2006. General RMS characteristics. comparison with dedicated and flexible systems. In A. I. Dashchenko, ed. *Reconfigurable Manufacturing systems and Transformable Factories*. Springer Berlin Heidelberg, pp. 7–45.

Koren, Y. et al., 1996. Real-time open-control architectures and system performance. *CIRP Annals - Manufacturing Technology*, 45(1), pp.377–380.

Koren, Y. et al., 1999. Reconfigurable manufacturing systems. *CIRP Annals - Manufacturing Technology*, 48(2), pp.527–540.

Koren, Y., 2010. *The global manufacturing revolution*, Wiley.

Koren, Y. & Shpitalni, M., 2010. Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems*, 29(4), pp.130–141.

Koren, Y. & Ulsoy, A.G., 1997. *Reconfigurable Manufacturing Systems, ERC/RMS Technical Report #5,*,

Lacksonen, T., 2001. Empirical comparison of search a algorithms for discrete event simulation. *Computer & Iindustrial Ingineering*, 40, pp.133–148.

Landers, R.G., Min, B.-K. & Koren, Y., 2001. Reconfigurable machine tools. *CIRP Annals - Manufacturing Technology*, 50(1), pp.269–274.

Li, J., Dai, X. & Meng, Z., 2009. Automatic reconfiguration of petri net controllers for reconfigurable manufacturing systems with an improved net rewriting system-based approach. *IEEE Transactions on Automation Science and Engineering*, 6(1), pp.156–167.

Li, X., Gao, L., Zhang G., Zhang, C. et Shao X., 2008. A genetic algorithm for integration of process and scheduling problem. In *Intelligent Robotics and Applications*. 5315, pp. 495–502.

Li, X. et al., 2010. Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling. *Computers & Operations Research*, 37(4), pp.656–667.

Li, X., Gao, L. & Li, W., 2012. Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling. *Expert Systems with Applications*, 39(1), pp.288–297.

Lian, K. et al., 2012. Integrated process planning and scheduling using an imperialist competitive algorithm. *International Journal of Production Research*, 50(15), pp.4326–4343.

Marri, H.B., Gunasekaran, A. & Grieve, R.J., 1998. Computer-aided process planning: A state of art. *The International Journal of Advanced Manufacturing Technology*, 14(4), pp.261–268.

Mehrabi, M.G. et al., 2002. Trends and perspectives in flexible and reconfigurable manufacturing systems. *Journal of Intelligent Manufacturing*, 13(2), pp.135–146.

Mohammadi, G., Karampourhaghghi, A. & Samaei, F., 2012. A multi-objective optimisation model to integrating flexible process planning and scheduling based on hybrid multi-objective simulated annealing. *International Journal of Production Research*, 50(18), pp.5063–5076.

Mohapatra, P., Benyoucef, L. & Tiwari, M.K., 2013. Realising process planning and scheduling integration through adaptive setup planning. *International Journal of Production Research*, 51(8), pp.2301–2323.

Musharavati, F. et al., 2008. A metaheuristic approach to manufacturing process planning in reconfigurable manufacturing systems. *Jurnal Teknologi*, 48, pp.55 – 70.

Musharavati, F. & Hamouda, A.S.M., 2012. Enhanced simulated-annealing-based algorithms and their applications to process planning in reconfigurable manufacturing systems. *Advances in Engineering Software*, 45(1), pp.80–90.

Nourelfath, M., Ait-Kadi, D. & Soro, I., 2002. Optimal design of reconfigurable manufacturing systems. In *International Conference on Systems, Man and Cybernetics, 2002 IEEE International Conference on.*

Oke, A.O., Abou-El-Hossein, K. & Theron, N.J., 2011. Process planning for reconfigurable manufacturing system for mould / die making. In *the 4th Robotics and Mechatronics Conference of South Africa*.

Phanden, R.K., Jain, A. & Verma, R., 2013. An approach for integration of process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 26(4) pp.284–302.

Roy, B., 1968. Classement et choix en présence de points de vue multiples. *Revue française d'informatique et de recherche opérationnelle*, 2(1), pp.57–75.

Saaty, T.L., 1980. *The Analytic Hierarchy Process*, McGrawHill.

Saygin, C. & Kilic, S.E., 1999. Integrating flexible process plans with scheduling in flexible manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 15(4), pp.268–280.

Sethi, A.K. & Sethi, S.P., 1990. Flexibility in manufacturing: A survey. *International Journal of Flexible Manufacturing Systems*, 2(4), pp.289–328.

Shabaka, A.I. & ElMaraghy, H.A., 2008. A model for generating optimal process plans in RMS. *International Journal of Computer Integrated Manufacturing*, 21(2), pp.180–194.

Shen, C.-H. et al., 2006. Reconfigurable fixtures for automotive engine machining and assembly applications. In A. Dashchenko, ed. *Reconfigurable Manufacturing systems and Transformable Factories*. pp. 155–194.

Singh, R.K., Khilwani, N. & Tiwari, M.K., 2007. Justification for the selection of a reconfigurable manufacturing system: a fuzzy analytical hierarchy based approach. *International Journal of Production Research*, 45(14), pp.3165–3190.

Srinivas, N. & Deb, K., 1995. Multi-Objective function optimizadon using non-dominated sorting genetic algorithms. *Evolutionary Computation*, 2(3), pp.221–248.

Takahashi, K. et al., 2006. Selecting an optimal production order and its corresponding configuration in a reconfigurable manufacturing system with multiple product. In *Reconfigurable Manufacturing Systems and Transformable Factories*. Springer Berlin Heidelberg, pp. 609–626.

Tang, L., 2005. *Design and reconfiguration of RMS for part family*. Thesis in Michigan University.

Zanakis, S.H. et al., 1998. Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research*, 107(3), pp.507–529.

Zhang, F., Zhang, Y.F. & Nee, A.Y.C., 1997. Using genetic algorithms in process planning for job shop machining. *IEEE Transactions on Evolutionary Computation*, 1(4), pp.278–289.

Zhang, G. et al., 2006. An analytical comparison on cost and performance among DMS , AMS , FMS and RMS. In *Reconfigurable Manufacturing Systems and Transformable Factories*. Springer Berlin Heidelberg, pp. 659–673.

Zhang, J. et al., 2002. Investigation of the reconfigurable control system for an agile manufacturing cell. *International Journal of Production Research*, 40(15), pp.3709–3723.

Zhang, Y.F., Saravanan, A.N. & Fuh, J.Y.H., 2003. Integration of process planning and scheduling by exploring the flexibility of process planning. *International Journal of Production Research*, 41(3), pp.611–628.

# Liste des articles publiés dans le cadre de la thèse

## Revues internationales avec comités de lecture

1. **Bensmaine A.**, Dahane M., Benyoucef L. (2013), A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems, article accepté dans *International Journal of Production Research*.

2. **Bensmaine A.**, Dahane M., Benyoucef L. (2013), A non dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment, *Computers & Industrial Engineering,* Volume 66, Issue 3, Pages 519-524.

## Conférences internationales avec actes et comités de lecture

1. **Bensmaine A.**, Dahane M., Benyoucef L (2013)., AMOSA based approach for process Plan generation In reconfigurable manufacturing system with machine load balancing, *IEEE international conference on systems, man and cybernetics (IEEE-SMC)* Manchester, UK.

2. **Bensmaine A.**, Dahane M., Benyoucef L. (2013), A simulation-based genetic algorithm approach for process plans selection in uncertain reconfigurable environment, *7th IFAC Conference on Manufacturing Modelling, Management, and Control* (IFAC MIM 2013), pp. 1961-1966, Saint Petersburg, Russie.

3. **Bensmaine A.**, Dahane M., Benyoucef L. (2012), Process plan generation in reconfigurable manufacturing systems using AMOSA and TOPSIS, *14th IFAC Symposium on Information Control Problems in Manufacturing* (IFAC INCOM 2012), pp. 560-565, Bucharest, Roumanie.

4. **Bensmaine A.**, Dahane M., Benyoucef L. (2011), Design of reconfigurable manufacturing systems: optimal machines selection using non-dominated sorting genetic algorithm (NSGA-II), *the 41st International Conference on Computers and Industrial Engineering* (CIE 2011), pp. 110-115, Los Angeles, USA.

5. **Bensmaine A.**, Dahane M., Benyoucef L. (2011), Simulation-based NSGA-II approach for multi-unit process plans generation in reconfigurable manufacturing system, *16th IEEE Conference on Emerging Technologies & Factory Automation* (IEEE ETFA 2011), pp. 1-8, Toulouse, France.

6. **Bensmaine A.**, Benyoucef L., Dahane M. (2011), Process plan generation in reconfigurable manufacturing systems using adapted NSGA-II and AMOSA, *9th IEEE Conference on Industrial Informatics* (IEEE INDIN 2011), pp. 863-868, Lisbon, Portugal.

## Conférences internationales sans actes avec comités de lecture

1. **Bensmaine A.**, Dahane M., Benyoucef L. (2012), An approach combining AMOSA and TOPSIS for process plan generation in reconfigurable manufacturing systems, *International Conference of Metaheuristics and Nature Inspired Computing (META 2012)*, Sousse, Tunisie.

**Résumé**

Le contexte incertain des marchés actuels a contribué au développement des systèmes manufacturiers, donnant lieu à des systèmes ayant des principes et des fonctionnalités multiples, qui essayent de répondre aux exigences changeantes.

Les systèmes manufacturiers reconfigurables (RMS) constituent une classe des systèmes où les machines, les composants des machines ainsi que le système de manutention peuvent être ajoutés, modifiés, supprimés ou échangés selon les besoins de la production. Un RMS combine le haut débit des systèmes manufacturiers dédiés avec la flexibilité des systèmes flexibles. Le principal composant d'un RMS est la machine reconfigurable (RMT) qui possède une structure qui permet sa reconfiguration.

Nous consacrons ce travail au problème de génération des plannings de fabrication dans un RMS, en exploitant au mieux les hauts degrés de reconfigurabilité des RMTs pour proposer des plannings efficaces. Trois problèmes liés à la fonction de génération des plannings de fabrication sont considérés : (1) la génération des process plans dans un cas unitaire où nous avons adaptés des techniques d'optimisation multicritère (NSGA-II et AMOSA), (2) la génération des process plans dans le cas multi-unité où une optimisation basée sur la simulation a été adaptée, et (3) l'intégration des fonctions de génération des process plans avec l'ordonnancement où nous avons développé une nouvelle heuristique permettant d'effectuer cette intégration

Toutes les approches proposées ont été validées par des expériences numériques, démontrant leur applicabilité et efficacité.

**Abstract**

The uncertain context of current markets has contributed to the development of manufacturing systems, resulting in systems with different features and principles, trying to meet the changing market demands.

Reconfigurable manufacturing system (RMS) is a recent class of systems where machines, machine components and the handling system can be added, modified, deleted or exchanged according to the production needs. RMS combines the high throughput of dedicated manufacturing systems with the customizability of flexible systems. The main component of an RMS is the reconfigurable machine (RMT: Reconfigurable Machine Tool) which has a structure that allows reconfiguration.

We focus in this work on CAPP (Computer-Aided Process Planning) to generate effective the process plans in an RMS that take advantage of the high degree of RMT reconfigurability Three problems associated with the function process planning are discussed: (1) the generation of process plans for a single-unit case, where we adapted multi-criteria optimization techniques (NSGA-II, AMOSA) (2) the generation of process plans in the multi-unit case, where we developed simulation based optimization technique, and (3) the integration of process planning and scheduling functions, where we developed a new heuristic aiming to achieve this integration.

All the proposed approaches have been validated by numerical experiments, demonstrating their applicability and effectiveness.