REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : SIOD24 /M2/2018

## Mémoire

Présenté pour obtenir le diplôme de master académique en

## Informatique

Parcours : système d'information optimisation et décision

---

# Conception et optimisation de performance d'un système de production reconfigurable utilisant les algorithmes évolutionnaires multi-objectif

---

**Par :**

**DERARDJA YASSINE**

Soutenu le 25 Juin 2018, devant le jury composé de :

| | | |
|---|---|---|
| **TIBERMACINE Ahmed** | M C A | Président |
| **TORKI Fatima Zohra** | M A A | Rapporteur |
| **CHAMI Djazia** | M A A | Examinateur |

# Remerciement

Grand merci à Allah, le tout puissant qui m'a donné la force et le courage d'arriver jusque-là.

Je tiens à remercier profondément mon encadreur : madame TORKI Fatima Zohra qui m'a encouragé à faire le maximum d'efforts dans ce travail ainsi que pour sa disponibilité.

Je remercie par la même occasion les membres de jury qui ont bien voulu accepter d'examiner et évaluer mon travail et participer à ma soutenance.

Je remercie aussi ma famille et tous les enseignants du département d'informatique pour leurs efforts afin de nous assurer la meilleure formation possible durant notre cycle d'étude.

Enfin, je remercie tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

# Dédicace

Louange a  Allah de m'avoir donné la patience d'aller jusqu'au bout de mes rêves et le bonheur de lever mes mains vers le ciel et de dire

"Al Hamdoulillah"

Je dédie ce modeste travail à ceux qui mon aider dans ma vie, aux personnes généreuses, vertueuses qui m'ont été d'un grand secours, chacun a sa manière, qui ont sacrifiés leur temps pour mon bonheur et pour  ma réussite, depuis l'école de mon enfance, et grâce a le éducation, mes parents, qui mon permis d'arriver à la fin de mon cycle d'étude

A l'esprit de mon père, et ainsi que ma mère rabi yarhamha, inchalla qu' il seront fière de moi.

Qu Allah les garde et les protèges ainsi que toute ma famille

Mes frères et mes sœurs,

A mes oncles, mes tantes, et leurs familles

A mes amis de la promotion

A tous ceux qui me sont chers.

Je dédie ce travail également a toutes les personnes chèrs a mon cœurs .Qu'elles trouvent ici l'expression de toute ma gratitude et mon amour.

## Le Résumé de mémoire

Le Système de Production Reconfigurable est une nouvelle approche dans les systèmes de production permettant à ces systèmes d'avoir des structures physiques changeable durant leurs exécutions. Ces systèmes sont, par conséquent, de bons candidats pour satisfaire de nouvelles exigences comme : multifonctions, la sociabilité, l'amélioration des qualités des produits, l'optimisation du temps et du coût de la production, et bien sur un temps de réponse raisonnable face `à d'urgentes demandes de modifications. L'objectif du présent mémoire est de traiter un problème célèbre et reconnu dans les RMS, celui de la sélection de la configuration la plus optimale pour produire un certain produit. Ceci est reconnu en tant que problème multi-objectifs car il s'agit de satisfaire probablement plusieurs objectifs souvent contradictoires. Dans cette ´étude, on considère la satisfaction en même temps « minimisation du temps de production » et aussi « minimisation du coût de production ». Pour traiter ce problème nous avons implémenté des techniques d'optimisation multicritère (Les Algorithme évolutionnaire élitistes pour l'optimisation multiobjectif NSGA-II et SPEA-II). Les algorithmes NSGA-II et SPEA-II prendront en entrée des données relatives au système de production, au produit à produire ainsi que les seuils de convergence souhaites. Ils s'arrête sur un ensemble de solutions optimales dites « front Pareto ». Afin d'aider l'utilisateur à choisir l'une des solutions optimales parmi toutes ces solution, une technique dite TOPSIS est exploitée dans ce mémoire.

# SOMMAIRE

SOMMAIRE

## **Chapitre 2:** Systèmes manufacturiers reconfigurables



## **Chapitre 3 :** Conception et modélisation

SOMMAIRE



## Chapitre 4: Implémentation

# Liste des figures

## Liste des tableaux

### **Chapitre 2:** Systèmes manufacturiers reconfigurables



### **Chapitre 3 :** Conception et modélisation



### **Chapitre 4: Implémentation**

# Introduction générale

Aujourd'hui, les besoins des personnes sont nombreux, donc un problème peut avoir plus d'un objectif et chaque objectif est modélisé en utilisant une fonction mathématique. Le processus de résolution de ce problème résulte d'un problème de prise de décision, car ils recherchent une solution optimale ou un ensemble de solutions approchées à chaque fonction objectif, et ces fonctions peuvent être en conflit. Pour prendre la décision optimale, certains compromis entre les objectifs contradictoires sont nécessaires. Ce type de problèmes est appelé problèmes d'optimisation multi-objectifs.

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.

Dans la fabrication, les chercheurs cherchent à minimiser le temps de production, en contrôlant coût de production, et améliorer le taux de production. Cela ressemble donc à un problème multi-objectif. Cette vision est conçue sur un système de fabrication reconfigurable qui contient des machines, des outils, la disposition du système, etc., et chaque machine a son propre comportement, ses caractéristiques physiques et ses caractéristiques logiques. Pour créer un produit sous ce type de systèmes caractérisés par la modularité, le processus de production ne nécessite pas de modifier l'ensemble du système pour fabriquer une partie d'un produit, mais il doit remplacer un composant modulaire ou le mettre à niveau. La structure du système (composants matériels ou logiciels). Par ce moyen, la capacité de production a augmenté à court terme et de nombreux coûts deviennent plus faibles que dans les systèmes de fabrication classiques.

Quand un produit est dans son processus de production, une machine passe par de nombreuses configurations, ce qui permet à une machine d'exécuter un ensemble de fonctionnalités en présence des outils nécessaires. Pour réaliser le produit final dans un temps réduit et avec le plus petit coût, les machines et les configurations de système qui influencent le cycle de vie du produit, devraient être bien choisi pour compromettre entre les deux objectifs contradictoires.

Les méthodes d'optimisation traditionnelles nécessitent plus de temps et de nombreuses lignes de code pour obtenir le processus de fabrication optimal.

Les algorithmes évolutionnaires semblent être une voie très prometteuse. L'objectif est d'implémentés de nouvelles solutions basées sur des algorithmes évolutionnaires pour résoudre le Problème de la sélection du processus de fabrication avec une meilleure optimisation en temps et en coût.

De ce fait , notre mémoire s'articule de quatre chapitres :

La partie I est divisée en deux chapitres. Chapitre 1 nous décrivons quelques concepts de base liés à l'optimisation multicritère utilisés dans le cadre de ce mémoir. Les Algorithme évolutionnaire élitistes pour l'optimisation multiobjectif (NSGA-II et SPEA-II). 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. Chapitre 2 nous présentons un ensemble de définitions et de concepts de base liés aux systèmes manufacturiers reconfigurables. Nous détaillerons les caractéristiques et les technologies clés liées aux RMSs. Nous terminons le chapitre par une conclusion.

La partie II se concentre sur le développement de notre solution, est divisée en deux chapitres. Chapitre 3 illustre la conception de notre application dans les deux niveaux (la conception globale et détaillée). Chapitre 4 présentes les résultats de l'implémentation de notre application où nous comparons les performances de NSGA-II et SPEA-II.

notre travail par une conclusion générale et en suite les perspectives entendus

# 1. Introduction

Différent secteur de l'industrie rencontre plusieurs problèmes au cours de leur parcours professionnel pour lesquels les décisions doivent être de façon optimale, c'est problèmes d'optimisation en général sont multi-objectif, des objectifs qui ne sont pas forcément parallèles (peuvent être contradictoires) qu'il faut satisfaire. [1]

Quelques problèmes qui ont été traité dans différent domaines de l'industrie :

- Désigne de systèmes dans les sciences d'ingénieurs (mécanique, aéronautique, chimie, etc.) : Ailes d'avions, moteurs d'automobiles, etc.

-Ordonnancement et affectation : ordonnancement en productique, localisation d'usines, planification de trajectoires de robots mobiles.

- Agronomie : programme de production agricole, etc.

- Transport : gestion de containers, design de réseaux de transport tracé autoroutier, etc.

- Environnement : gestion de la qualité de l'air, distribution de l'eau, etc.

- Télécommunications : design d'antennes, affectation de fréquences, radiotéléphonie mobile, etc. [1]

# 2. Définition

L'optimisation multi-objectif permet de modéliser des problèmes réels faisant intervenir de nombreux critères (souvent contradictoires) et contraintes. Ils sont traités parallèlement ce qui veut dire qu'aucun objectif ne sera privilégié sur un autre et donc la solution finale sera sous forme d'un ensemble de solution optimal, ces solutions ne peuvent être comparé entre elle car elles ont une même qualité. [2]

Toutes sociétés quel qu'en soit sa nature, son but principal est de géré/résoudre les problèmes rencontrer afin d'optimiser son rendu et atteindre ces objectifs.

Une formule générale d'une optimisation multi-objective peut être formulée comme suit :

Minimisation de la valeur (x) avec $F(x)= \{f_1(x), f_2(x), \ldots, f_L(x)\}$

Avec : $g_j(x) <= 0$, $j=1, \ldots, m$,

$h_k(x)=0$, $k=1, \ldots, p$,

F : représente l'ensemble de fonction à optimiser

L : le nombre de fonction d'objectif à traiter ($L \geq 2$)

X : l'ensemble de solution réalisable ou espace de décision, il peut être définie comme suit $\{x \in \Omega : g(x) \leq 0 \text{ et } h(x)=0\}$

m : nombre de contraintes d'inégalité

p : nombre de contraintes d'égalité

Dans le cas de l'optimisation multi-objectif tous les objectifs sont traités parallèl3

ement ce qui veut dire qu'aucun objectif ne sera privilégié sur un autre et donc la solution finale sera sous forme d'un ensemble de solution optimal, ces solutions ne peuvent pas être comparé entre elles, car elles ont une même qualité (optimales). [3]

# 3. Choix de la méthode d'aide à la décision

La résolution d'un problème multi-objectif menant à la détermination d'un ensemble de solutions Pareto, il est nécessaire de faire intervenir l'humain à travers un décideur, pour le choix final de la solution à garder [4].

On peut répartir les méthodes de résolution de problèmes multi-objectifs en trois familles, en fonction du moment où intervient le décideur.

## 3.1 Les méthodes d'optimisation a priori

Les préférences pour chacun des objectifs doivent être définies par le décideur et puis on lance la recherche des diverses solutions satisfaisant ces préférences. Cette approche nécessite une bonne connaissance a priori du problème[4].

### 3.2 Les méthodes d'optimisation progressives

Le décideur intervient pendant la recherche pour la guider vers les solutions prometteuses en ajustant les préférences dans le processus[4].

### 3.3 Les méthodes d'optimisation a posteriori

Les diverses solutions vont être tout d'abord trouvées puis alors le décideur choisit la plus appropriée[4].

## 4. Méthodes de résolution des problèmes combinatoire

### 4.1 Les méthodes exactes

L'objectif des méthodes exactes consiste à trouver une solution exacte a un problème donné, par le biais d'un algorithme naïf qui vas tester sur l'espace de recherche chaque solution possible pour qu'à la fin il sélectionne la meilleure d'entre elle. Cette manière de résolution prend un temps considérable afin de trouver cette solution (solution exacte). Parmi les méthodes exactes, on trouve la plupart des méthodes telles les techniques de séparation et évaluation (Branch & Bound) ou la programmation dynamique. [1]

### 4.2 Les méthodes approchées (métaheuristique)

Ces méthodes se sont des méthodes de recherche générale utilisées pour résoudre des problèmes d'optimisation difficile (NP-hard). Il existe deux types de méthode métaheuristique, la 1ere à base d'une solution et la seconde à base d'une population de solutions. On va développer uniquement les métaheuristiques à base d'une population de solution (algorithme évolutionnaire). [1]

Les approches heuristiques peuvent être classées en trois catégories :

**a) Approches transformant le problème en un ou plusieurs problème(s) mono-objectif(s)**

Ces approches consistent à transformer un problème donné en un ou plusieurs problèmes mono-objectifs mais aussi elles nécessitent une connaissance du problème et ne donne au final qu'une seule solution. Parmi ces méthodes on trouve les méthodes d'agrégations, les méthodes ∈-contrainte et aussi les méthodes qui utilisant un vecteur cible. On peut les classer dans les méthodes d'optimisation a priori qu'on a déjà présentée un peu plus tôt. [5]

**b) Approches Non Pareto**

Ces approches transforment le problème initial. Elles traitent les objectifs indépendamment tout le long de la recherche. Elles ont du mal à trouver les solutions de compromis parce qu'elle traite les objectifs indépendamment. Ces approches sont de type postériori. [5]

**c) Approches Pareto**

Ces approches utilisent la notion de dominance pour comparer les solutions entre elles par rapport à tous les objectifs en même temps. Une seule résolution permet d'approximer l'ensemble de la frontière Pareto. [5]

Définition : la relation de dominance [6]

On dit que le vecteur $\vec{x}_1$ domine le vecteur $\vec{x}_2$ si : $\vec{x}_1$ est au moins aussi bon que $\vec{x}_2$ dans tous les objectifs, et, $\vec{x}_1$ est strictement meilleur que $\vec{x}_2$ dans au moins un objectif.

Le front qui domine tous les autres fronts, il contient des solutions de même qualité incomparable entre elles, ces solutions sont appelées solutions **optimales** au sens de **Pareto** (ou **solutions non dominées**).

## 4.3 Les algorithmes évolutionnaires

Ce sont des algorithmes heuristiques qui s'inspire de l'évolution naturelle pour résoudre différents problèmes, ils permettent de faire évoluer un ensemble d'individus (population) de façon aléatoire, chaque individu sera évalué grâce un mécanisme

d'évaluation (fitness) afin de les mesuré la qualité de chaque individu, ensuite de faire évolué cette population et sa en assurant la descendance avec les parents qui se sont le mieux adaptés (répondent à des critères bien définie) [1]. Il existe trois types d'algorithmes. [7]

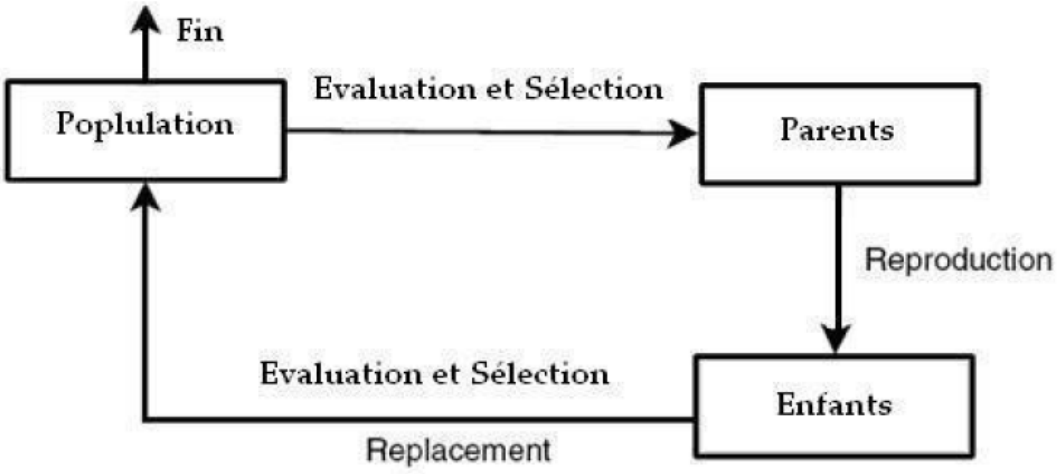


**Figure 1.1** : Organigramme de fonctionnement d'un algorithme évolutionnaire [8]

### 4.3.1. Les algorithmes génétiques

Ils sont les plus répondus. Les individus (solution) sont codés le plus souvent en binaire, ces individus seront évalués selon leur capacité à résoudre le problème. Pour chaque génération les on va obtenir de nouveaux individus issus de la génération précédente et ainsi de suite.

#### 4.3.1.1 Codage de la solution

Pour chaque problème il faut choisir le codage approprié, le choix de cette représentation aura une influence considérable sur la façon dont les solutions seront manipulées [11].

Quelques types de codages [9]

- **Le codage binaire** : chaque gène sera codé (représenté) par l'alphabet binaire {0,1} .

- **Le codage réel** : le gène est représenté par un nombre réel. Ce codage consiste simplement à la concaténation des variable « x » d'in individu « X ».

#### 4.3.1.2 Initialisation de la population

L'initialisation de la population dans NSGA-II ou SPEA-II (qu'on va voir par la suite) à un rôle très important dans qualité des résultats obtenus. Il faut avoir une population avec des individus divers afin de garantir la diversité génétique mais aussi, il faut que ces solutions soient de bonne qualité. [10]

#### 4.3.1.3 L'évaluation

La fonction d'adaptation, elle donne à chaque individu une valeur qui va servir à évaluer les individus selon leur niveau d'adaptation à leur environnement. Ce qui signifie qu'elle quantifie la réponse fournit au problème pour une solution potentielle donnée. Ainsi on peut comparer les individus entre eux. [10]

La comparaison des individus va se faire selon le principe de Pareto.

#### 4.3.1.4 Les opérateurs génétiques

**a) Sélection :** Cette étape consiste à enrichir la population en croisant des individus. C'est la combinaison être des morceaux de solutions de certains individus avec d'autre moreaux d'autres individus pour créer des nouveaux individus qui seront peut-être une solution meilleure au problème.

Quelques méthodes : [11]

- La sélection par rang : il s'agis d'une implémentation d'une roulette, les secteurs de la roue seront proportionnels au rang de l'individu dans la population, cette population est triée en fonction de la qualité des individus.

- La sélection par tournoi : cette méthode de sélection permet aux individus de piètre qualité de participer à l'amélioration de la population.

- La sélection par l'élitisme : elle permet de mettre en avant les meilleurs individus de la population. Ce sont les individus les plus prometteurs qui vont participer à l'amélioration de la population.

**b) Croisement :**

Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants. [12]

Le principe est de créer deux nouveaux individus à partir des deux parents.

**c) Mutation :**

Cet opérateur génétique consiste à modifier le génotype d'un individu. La mutation a pour but de garantir l'exploration de l'espace d'états. L'opérateur de mutation consiste généralement à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire. [12]

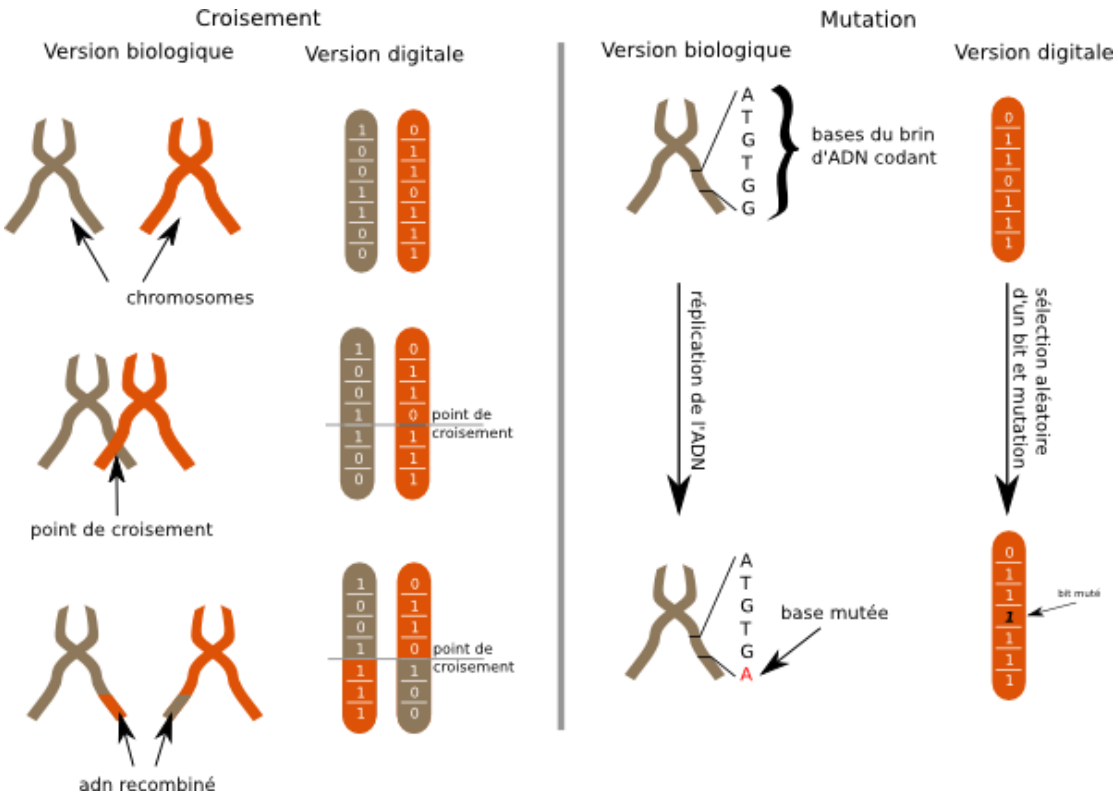


**Figure 1.2** : Illustration des opérateurs de mutation et de croisement (codage binaire) [13].

### 4.3.2 Les stratégies d'évolution

Le même principe que pour les algorithmes génétiques sauf que cette fois si seuls les meilleurs individus seront choisis parmi l'ancienne génération (parents) et la nouvelle génération (enfant).

### 4.3.3 La programmation évolutionnaire

Elle fut mise au point initialement pour la découverte d'automates à états finis pour l'approximation de séries temporelles, puis a rapidement été généralisée à des espaces de recherche très variés. Elle ne travaille que sur le comportement des individus, n'utilise que des opérateurs de mutation et de remplacement agissant sur le phénotype, et pas d'opérateur de croisement, avec toutefois l'utilisation fréquente d'un remplacement plus stochastique que déterministe (tournoi) dans lequel les plus mauvais ont tout de même une petite chance de survie. [14]

## 4.4 Algorithme Génétique Élitiste de Tri Non- Dominé (NSGA-I I)

Les algorithmes génétiques sont des algorithmes évolutionnaires, aussi, il en existe plusieurs types qui varient selon l'approche (Pareto/ non-Pareto) mais aussi certain sont une amélioration a d'autre plus ancien tel que le NSGA en NSGA-II. Pour certain, la différence réside dans l'approche pour le traitement des individus, que ce soit dans la phase de dominance ou bien dans la phase de sélection tel que NSGA et NPGA. [6**]**

L'algorithme NSGA-II « Non dominated Sorting Genetic Algorithm », est une méthode évolutionnaire. Il consiste a ordonné toutes les solutions générées, cet ordonnancement se fait sur des critères qui peuvent être contradictoire. [15]

La procédure NSGA-II a été introduite par [9], qui est une amélioration de la procédure NSGA nommé dans l'article [16], cette amélioration réside dans les points suivants :

- Il utilise une approche Elitiste qui lui permet de sauvegardés les meilleures solutions.

- Se base sur le principe de Pareto pour le trie de la non-dominance.
- Utilisation de la distance de Crowding qui va permettre de garantir une diversité génétique[17].

### 4.4.1 Fonctionnement de l'algorithme (NSGA-II)

On commence par créer une population initial (parents) ($\mathbf{P_t}$) de taille **N**, en suite une 2eme population ($\mathbf{Q_t}$) de la même taille sera générer à partir de la population ($\mathbf{P_t}$) en utilisant les opérateurs de croisement et de mutation. Une population ($\mathbf{R_t}=\mathbf{P_t}\cup\mathbf{Q_t}$) est formée par l'union des deux populations $\mathbf{Q_t}$ et Pt, elle aura une taille de (2**N**) individu. Ces individus seront triés selon la non-dominance de Pareto en Différents fronts ($\mathbf{F_1}$, $\mathbf{F_2}$,..., $\mathbf{F_g}$). Les individus du front $\mathbf{F_1}$ domine ceux du front $\mathbf{F_2}$ celui-ci qui dominera le front $\mathbf{F_3}$ ainsi de suite. Et vus que cet algorithme est élitiste donc il consiste à préserver les meilleures solutions d'une itération à l'autre. C'est-à-dire que les individus du premier front sont tous copiés dans la prochaine population $\mathbf{P_{t+1}}$ si leur nombre ne dépasse pas **N**. Tant que la taille de la population $\mathbf{P_{t+1}}$ est inférieure à **N**, des solutions sont ajoutées pour compléter la population. Ainsi les solutions du front $\mathbf{F_2}$ sont ajoutées, celles de $\mathbf{F_3}$ et ainsi de suite. Si les solutions d'un front ne peuvent pas rentrer toutes dans la population $\mathbf{P_{t+1}}$, alors les solutions de ce front sont classées suivant une mesure de diversité (la distance de Crowding). Les solutions les plus dispersées (de plus grande distance) sont ajoutées jusqu'à compléter la population $\mathbf{P_{t+1}}$.

Une fois les individus de la population (**Pt+1**) sont identifiés, une nouvelle population enfants (**Qt+1**) sera créée par la sélection, croisement, mutation.

Le processus continue, d'une génération a la suivante, jusqu'à atteindre un critère d'arrêt .

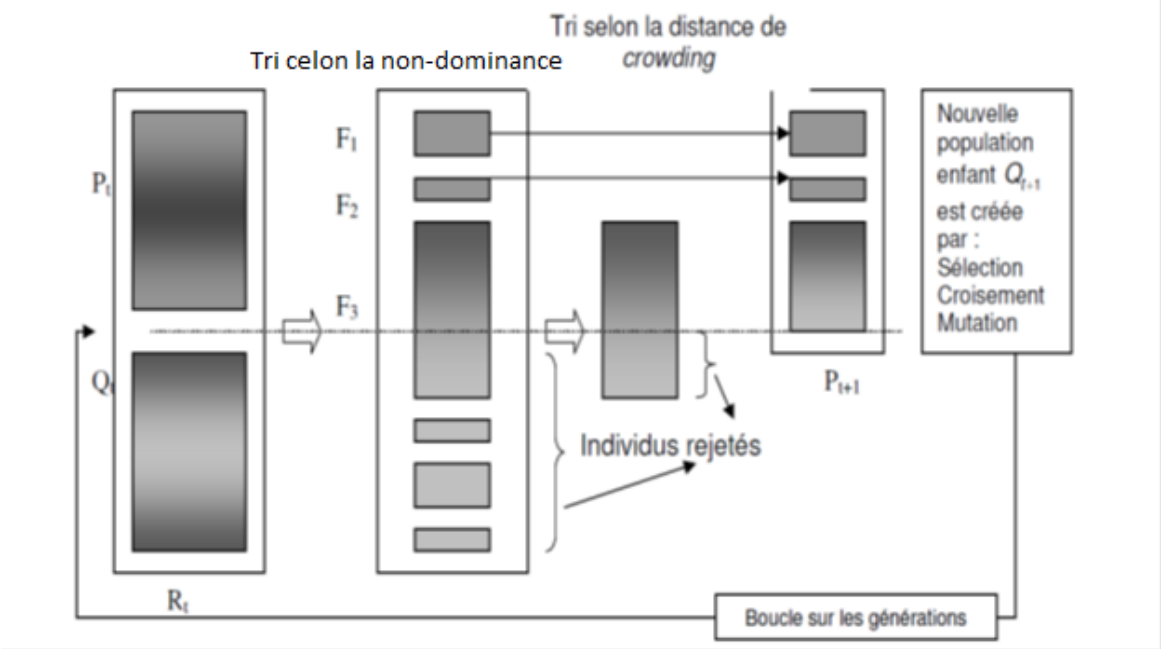


**Figure 1.3** : Principe de l'algorithme NSGA-II [9]

### 4.4.2 La distance de Crowding

L'attribution de la distance de Crowding pour chaque solution se fait comme suit :

- Trie des solutions selon chaque objectif, dans un ordre descendant.
- Pour chaque objectif attribué la distance infinie aux solutions possédant la valeur limite (minimal et maximal).
- Pour les autres solutions le calcul de la distance de Crowding égale à la différence normalisée des valeurs de fonctions objectives de deux solutions adjacentes. Ce calcul est réalisé pour chaque fonction objective. La distance de Crowding d'une solution est calculée en sommant les distances correspondantes à chaque objectif. [18]

Assignement-distance-crowding (I)

$L = |I|$

Pour chaque i, atrribué $I[i]_{distance} = 0$

Pour chaque objectif m

$I = trie(I,m)$

$I[1]_{distance} = [L]_{distance} = \infty$

Pour i=2 a ($L$-1) faire

$I[i]_{distance} = I[i]_{distance} + (I[i+1].m - I[i-1].m)/(f^{max}_m - f^{min}_m)$

**Figure 1.4** : pseudo code distance de Crowding [18]

## 4.5 Algorithme Génétique Élitiste (SPEA-II) :

L'algorithme SPEA-II (Strength Pareto EvolutionaryAlgorithm-2) est un algorithme proposé par [19]. C'est un algorithme qui implante trois techniques communes à d'autres approches telles que :

- l'utilisation de la dominance au sens de Pareto pour évaluer et sélectionner les individus
- l'utilisation d'une population secondaire (archive externe) pour stocker les solutions non-dominées
- l'utilisation du clustering pour maintenir la diversité basée sur la notion de niche

### 4.5.1 Fonctionnement de l'algorithme (SPEA-II) :

- Pour commencer, SPEA2 crée aléatoirement une population initiale P0 à partir de laquelle il rempli l'archive externe P par les individus non dominés dans P0.

- A chaque itération, il calcul la valeur fitness des individus des deux populations.

- Selon les valeurs de fitness obtenues, les individus vont être sélectionnés pour les opérations de croisement et mutation afin de générer de nouveaux individus.

- Avant d'entamer une nouvelle génération, l'archive est mis à jour par les individus non dominés nouvellement produits. Si après cette mis à jour, des individus se révèlent dominés, ils seront automatiquement supprimés de l'archive.

- Si la taille de l'archive après insertion, excède la taille permise, une réduction par clustering est alors effectuée.

### 4.5.2 Calcul de la valeur de fitness (SPEA-II)

Le calcul de la valeur de fitness s'effectue en deux étapes de la manière suivante :

La première étape consiste à affecter une valeur appelée valeur de force (S appartient [0,1]) aux individus de l'archive P cette valeur est déduite à partir du nombre d'individus qu'un élément i appartient P domine faiblement dans la population courante Pt :

S(i) = (nombre des domines par i / |Pt| + 1) (1)

Pour un individus j de la population courante Pt', la valeur de fitness est calculée en sommant les valeurs de force des individus de P dominant j plus 1.

Le chiffre 1 est ajouté au total de la somme pour éviter que les individus de P aient une valeur de fitness plus grande que certains individus de Pt.

### 4.5.3 Technique de clustering:

La technique de clustering est utilisée pour réduire la taille de l'archive. Principe : Au début de la procédure, chaque individu constitue son propre cluster (classe), puis on fusionne deux à deux les clusters les plus proches en terme de distance. Cette étape est itérée jusqu'à l'obtention du nombre désiré de clusters. Une fois les clusters identifiés, il ne reste plus qu'à choisir un représentant par cluster. Ce représentant peut être déterminé de plusieurs façons, par exemple en prenant le barycentre du cluster. C'est ce représentant qui sera gardé, les autres éléments étant tout simplement supprimés.

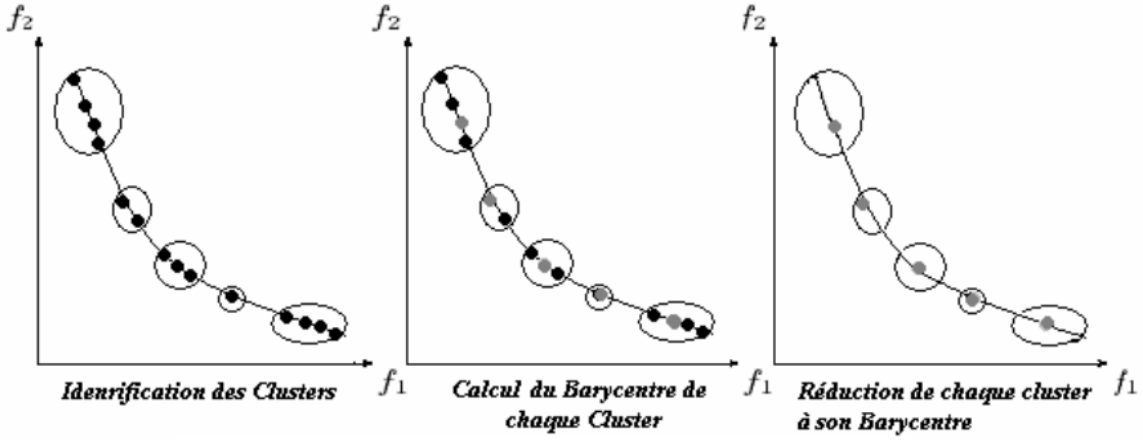


**Figure 1.5** : La technique de clustering

## 5. 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

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 $V^+$ positif 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. [20]

La méthode TOPSIS se définie en sept étapes :

1) Établir une matrice X, où un ensemble d'alternatives $a_1, a_2 \ldots a_{j\ldots}, a_k$ sont comparés sur des critères $c_1, c_{2,\ldots} c_{j\ldots,} c_n$
2) Normaliser la matrice de décision pour obtenir une nouvelle matrice R
   D'élément $r_{ij}$ tel que :

$$x = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{k} {x_{ij}}^2}} \qquad (1.1)$$

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

$$v_{ij} = w_i \times r_{ij} \qquad (1.2)$$

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

$V^+ = (best_j(v_{1j}), best_j(v_{2j}), \ldots, best_j(v_{nj}))$ (1.3)

$V^- = (best_j(v_{1j}), best_j(v_{2j}), \ldots, best_j(v_{nj}))$ (1.4)

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

$worst_j(v_{1j}) = \begin{cases} \min(v_{ij}) \; si \; c_i \; est \; bénéficial \\ \max(v_{ij}) \; si \; c_i \; n'est \; pas \; bénéficial \end{cases}$

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 :

$$d_j^+ = \sqrt{\sum_{i=1}^{n}(v_{2j} - v_2^+)^2} \quad (1.5)$$

$$d_j^- = \sqrt{\sum_{i=1}^{n}(v_{2j} - v_2^-)^2} \quad (1.6)$$

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

$$D_j^+ = \frac{d_j^+}{d_j^- + d_j^+} \quad (1.7)$$

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

## 6. Conclusion

Nous avons défini, dans ce chapitre, le problème d'optimisation multi-objectif et présenté les différentes approches qui ont été proposées pour résoudre ces problèmes. Les deux métaheuristiques multicritère NSGA-II et SPEA-II, sont présentés en détail dans ce chapitre, ils ont été proposés comme des alter natives très performant. La méthode TOPSIS a été détaillée avec son principe.

# 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.

# 2. Évolution des systèmes manufacturiers

A la fin du XXe 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[20].

## 2.1 Les systèmes de production dédié (DMS : Dedicated Manufaturing Systems)

L'invention des lignes de production dédiées, nommé aussi lignes de transfert, a beaucoup facilité le paradigme de production de masse. Chaque ligne est conçue afin de produire une seule pièce à un taux de production qui est élevé en utilisant différents outils simultanément. Les lignes de transfert peuvent être économiquement fiables si elles sont utilisées le plus longtemps possible avec une capacité maximale. Tant que les volumes de production sont élevés, chaque unité de produit a un coup qui est relativement minimisés. Cela dit, dans un contexte caractérisé par une compétitivité globale et une saturation mondiale, les DMS ne sont pas suffisamment exploités, ce qui engendre pas mal de pertes [21].

## 2.2 Les systèmes de productions flexibles (FMS : Flexible Manufacturing Systems)

Dans les années 80 on constate une évolution du besoin des consommateurs. Ce qui fait qu'on passe d'un besoin de production de masse à un autre besoin de personnalisation de masse. Avoir des systèmes de production réactifs et flexibles est devenu nécessaire. Le concept de systèmes de production flexibles est nais en réponse au besoin de personnalisation de masse et afin de fournir une grande réactivité dans les changements de produits. Les objectifs stratégiques des entreprises durant la période 1980 - 1990 étaient : productivité, qualité et flexibilité. Les FMS se basent sur des machines à commandes numériques ainsi que d'autres machines robotisés qui permettent de produire, en moyenne série, une variété de produits avec le même système. Il y a un glissement de la ligne de production vers les ilots flexibles de production. L'acquisition de ces machines présente un coût important. A cause des investissements initiaux élevés, il a fallu presque 20 ans pour que ces systèmes pénètrent l'industrie automobile qui représentait en 2010 le plus gros marché des FMS. Une revue de la littérature permet de classifier 10 types de flexibilité du système de production[21].

## 2.3 LES SYSTEMES DE PRODUCTION RECONFIGURABLES (RMS : Reconfigurable Manufactoring Systems)

Un RMS est conçu dès le départ pour des changements rapides de sa structure (physique et logique) afin d'ajuster rapidement la capacité et la fonctionnalité de la production autour d'une famille de produits en réponse à des changements soudain dans le marché[22].

La reconfigurabilité est un concept qui a été proposé en 1995 par l'université du Michigan. Le système de production reconfigurable a été conçu dès le départ conçu avec une possibilité pour la modification de sa structure afin d'ajuster plus rapidement sa capacité et de se faite s'adapter à des fluctuations de volume et une rapidité d'ajustement de sa capabilité pour s'adapter à des produits nouveaux. Initialement le système RMS est installé avec les fonctionnalités et la capacité juste nécessaires et peut, si besoin est, l'ajustement dans le futur sa structure pour s'adapter aux évolutions des exigences du marché. Ceci permet de minimiser le besoin en investissement initial. A la différence d'un FMS, le focus de flexibilité d'un RMS est limité et porte sur une famille de produits. La reconfigurabilité concerne aussi bien la structure physique du système de production que sa structure logique. Pour être reconfigurable, un système de production doit posséder certaines caractéristiques clés[20].

# 3. Principes et caractéristiques de base des RMSs

## 3.1 Architecture d'un RMS

L'architecture d'un RMS est composée de deux entités dont le processus de fabrication et machine- outil reconfigurable. Le Figure (2.1) suivante représente L'architecture d'un RMS. [23]

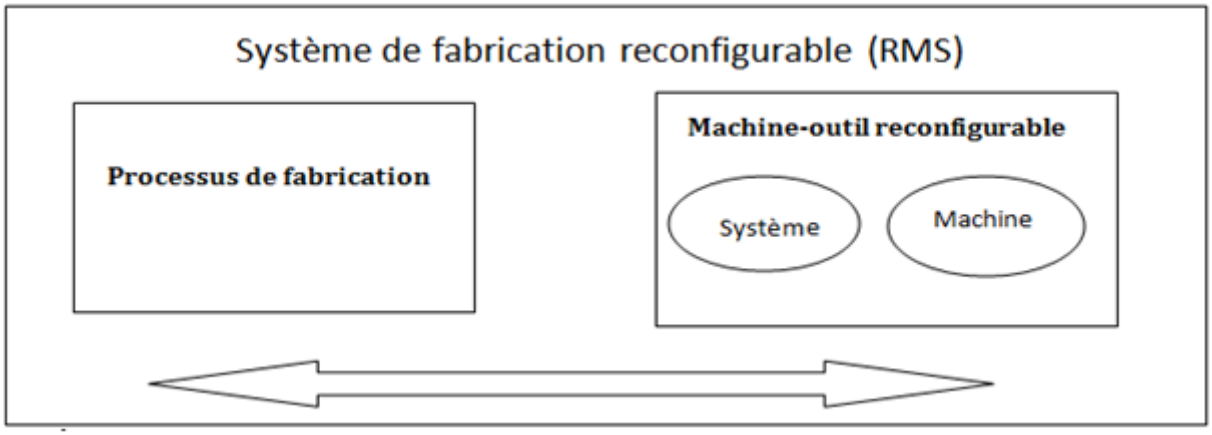


Figure2.1 : L'architecture d'un RMS [23]

- Machine-outil reconfigurable : 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.

  La Figure 2.2 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. [24]

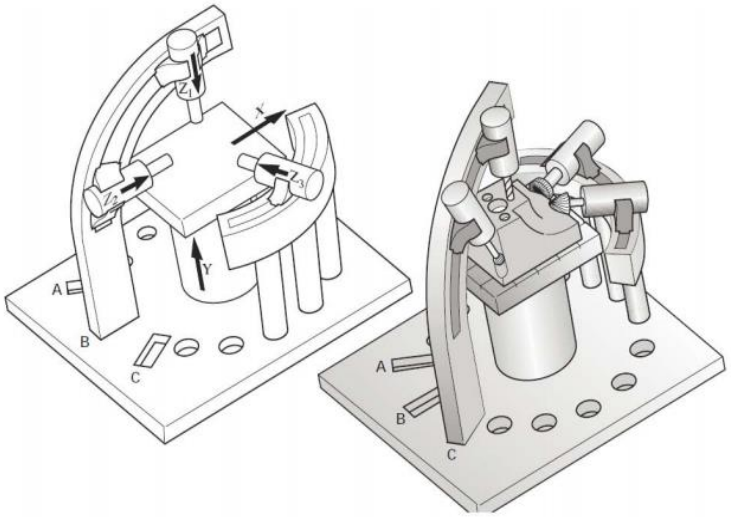


**Figure 2.2**. Deux configurations d'une même RMT [24]

- Le processus de fabrication : Défini comme étant la fonction pour laquelle le procédé de fabrication et machines doivent être utilisés pour effectuer les différentes opérations nécessaires à la production d'un composant.

## 3.2. Technologies clés des RMSs

Ces dernières années ou plus précisément les deux dernière l'émergence de deux nouvelles technologies a vus le jour, ils constituent 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 2.1).

| | Hardware fixé | Hardware reconfigurable |
|---|---|---|
| Pas de software | Machine manuelle, DMS | ------ |
| Software fixé | FMS, machine CNC, Robot | Machines reconfigurable |
| Software reconfigurable | AGV1, open architecture | RMS |

**Tableau 2.1.** Classes de machines et de systèmes manufacturiers [23].

Autrement dit, il est nécessaire d'avoir du matériel et des logiciels reconfigurable pour un RMS, mais cela ne garantit en aucun cas sa rentabilité. En effet, la conception des RMSs doit absolument prévoir les fonctionnalités opérationnelles possibles pendant 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 [23].

## 3.3. Caractéristiques de la reconfiguration

Un RMS doit comporter des modules hardware et software dès sa conception, qui par la suite pourront être intégrés plus facilement et d'une manière fiable. Dans le cas contraire, le processus de reconfiguration sera plus lent et non pratique [25]. Et donc on constate, qu'un RMS possède six caractéristiques primordiales :

**1) Personnalisation** ***:*** La personnalisation de la flexibilité importe que le RMS doit s'adapter à la plupart des caractéristiques qui dominent la famille de produit à manufacturer. La 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** : La convertibilité du système est la capacité à changer rapidement et facilement les fonctionnalités du système et ses machines installées afin de s'adapter aux nouveaux besoins de la production. Ainsi, la convertibilité du système inclut la convertibilité

des machines qui peut être caractérisée par le changement des outils, une partie des programmes, ou encore un réglage des degrés de liberté [22].

**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. [20]

**4) Modularité** dans un RMS tous les composants principaux doivent être modulaires (les ressources, les outils, le contrôle,…). Lorsque nécessaire, les composants modulaires peuvent être remplacés ou modernisés afin de mieux répondre aux nouvelles exigences et/ou applications. La sélection des modules de base et la façon dont ils sont connectés doivent permettre la création de systèmes qui peuvent être facilement intégrés, diagnostiqués et convertis [22].

**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[20].

**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. [24]

## 4. Approches et travaux connexes :

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 :

Chaube et al.[23] 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. [25] 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 [26] 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.

Bensmaine et al. [27] discuté du problème de la sélection des machines de configuration optimale pour effectuer toutes les opérations requises à faible coût et dans un temps de réalisation réduit. Ils ont proposé un NSGA-II adapté pour résoudre le problème multi-objectif de la sélection du processus de fabrication optimal. Ils ont modélisé chaque objectif (en minimisant le temps total et le coût total) avec une fonction mathématique, qui permet de formuler la fonction de fitness. La représentation individuelle est l'élément central d'un

algorithme génétique. Dans cet article, les chromosomes sont codés par des valeurs réelles, puis ils sont décodés pour obtenir les processus de fabrication réalisables.

Haddou Benderbal, H., Dahane, M. & Benyoucef [28] le problème de l'indisponibilité de l'une des machines sélectionnées en définissant l'indice de flexibilité, qui représente le nombre de machines alternatives pouvant éviter les perturbations causées par l'indisponibilité des machines sélectionnées. Les auteurs ont appliqué l'algorithme NSGA-II pour sélectionner les machines optimales, ils ont considéré deux objectifs (maximiser l'index de flexibilité et minimiser le temps d'achèvement). Après avoir appliqué l'algorithme NSGA-II et obtenu le Pareto-front, basé sur la méthode TOPSIS (Technique de Performance par similarité à Solution Idéale), les auteurs aident le décideur à choisir la meilleure solution parmi les solutions de Pareto-front en classant des solutions du front de Pareto selon les préférences et les capacités de l'entreprise. Ils ont suivi dans leur approche la formulation du problème et la technique de décodage proposé par (Bensmaine, Dahane, Benyoucef).

## 5. 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).La dernière section de ce chapitre a présenté quelques travaux liés au problème de la sélection des machines reconfigurables optimales.

# 1. Introduction

Après avoir appris les points théoriques nécessaires mentionnés dans le chapitre précédent. Ce chapitre donne une vue détaillée de l'approche adoptée pour la génération des processus de fabrication. 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 du produit.

# 2. L'Analyse

De nos jours, les chercheurs en fabrication sont prêts à lancer un processus de fabrication permettant de produire un produit à moindre coût et en moins de temps. Cette situation est un problème d'optimisation multi-objectif, car elle vise simultanément à minimiser les coûts et le temps. Les deux objectifs sont en conflit entre eux, ainsi le processus de résolution nécessite de nombreux compromis pour obtenir la solution optimale. Les méthodes classiques de recherche telles que la méthode de Benson et la somme pondérée qui sont basées sur le contexte mono-objectif ne permettent pas de résoudre ce genre de problèmes.

Notre objectif de projet est d'implémentés de nouvelles solutions basées sur des algorithmes évolutifs pour résoudre le Problème de la sélection du processus de fabrication optimal. Qui permet de produire un produit à moindre coût et en moins de temps.

## 2.1 Description du problème

Le Problème de la sélection du processus de fabrication optimal est décrit dans la figure 1.3

Les paragraphes ci-dessous résument, comment sélectionner l'ensemble optimal de machines qui permettant de produire un produit à moindre coût et en moins de temps. Sur la base des caractéristiques du produit, et parmi un ensemble de machines reconfigurables disponibles (m), ces machines candidates sont sélectionnées en fonction de leurs fonctionnalités (outils et mouvements autorisés). Après cela, nous décidons quelles sélections sont optimales. L'ensemble de ces machines représente le RMS, où si elles sont ensemble, elles peuvent réaliser le produit souhaité à moindre coût et en moins de temps.

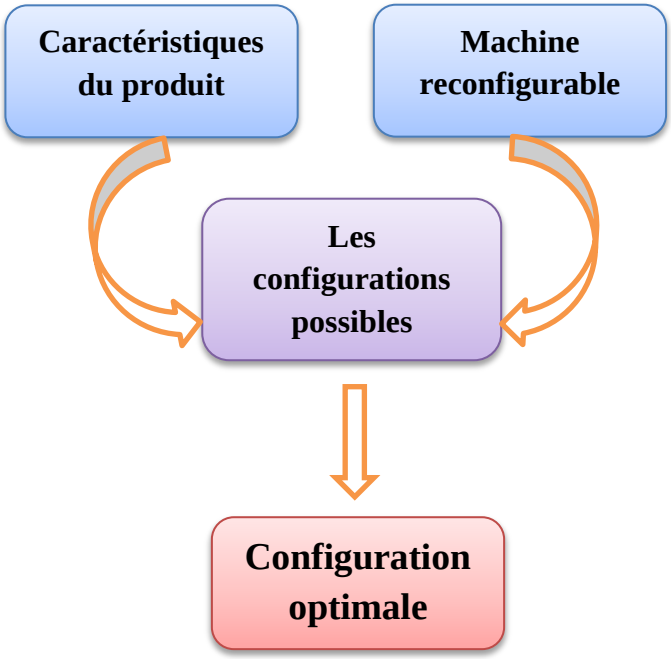


**Figure 3.1** : Problème de la sélection du de processus de fabrication optimal

Ce problème de sélection vise à atteindre les deux objectifs de minimisation du temps et du coût total de production. Les deux objectifs sont en conflit entre eux, d'où cette situation appartient à la famille des problèmes d'optimisation multi-objectifs (MOP).Comme nous l'avons mentionné dans la partie précédente (chapitre 1), ce type de problèmes est résolu à l'aide d'algorithmes évolutifs (EA). Dans notre cas, nous utiliserons les algorithmes "NSGA-II et SPEA-II" comme métaheuristique pour obtenir l'ensemble de machines pouvant réaliser le produit dans un temps et un coût optimal.

## 2.2 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 qui est en représentation textuelle, à un processus de fabrication sous forme matricielle [20]. Ainsi, nous représentons un processus de fabrication en tant qu'une matrice M×N, sous la forme illustrée dans le Tableau 3.1.

| 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 3.1. Un processus de fabrication sous forme de tableau [20]

Le processus de fabrication représenter dans le Tableau 3.1 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.

## 3. Conception

Après avoir fait l'analyse et la description du problème de la sélection du processus de fabrication optimal, Nous allons déterminer la l'architecture globale de la conception, et détaillée l'approche adoptée pour la génération des processus de fabrication et la formalisation mathématique du problème.

### 3.1 Conception globale

Figure 3.2 illustre l'architecture globale de conception.

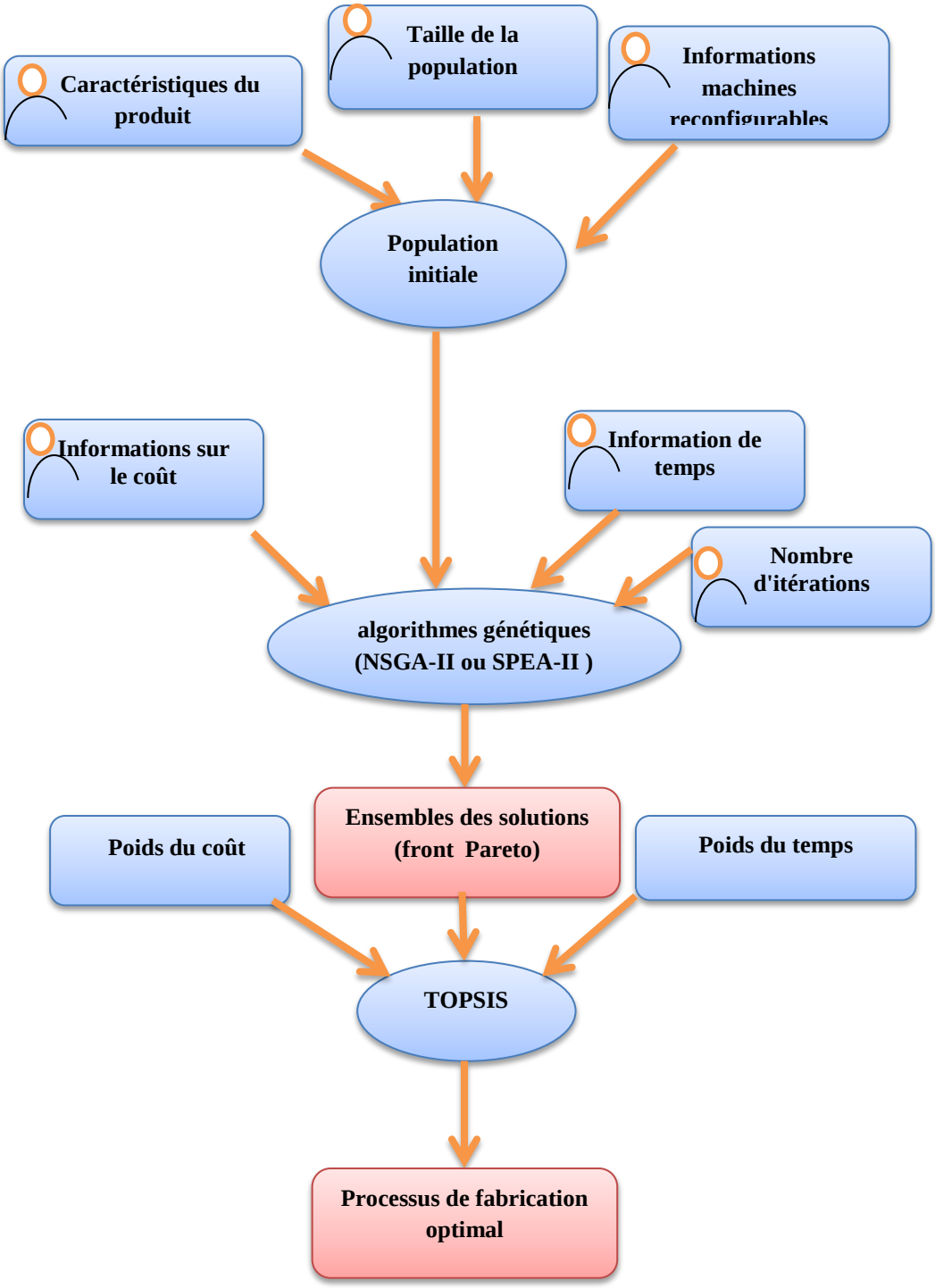


**Figure 3.2** : l'architecture globale de conception

L'utilisateur doit identifier la taille de la population qui est le nombre des individus (les processus de fabrication), qui sont les paramètres de "population initial", qui crée de manière aléatoire un ensemble de processus de fabrication. Basé sur le produit et caractéristiques et informations sur les machines reconfigurables, les individus créés sont décodés en appliquant des "méthodes de décodage".

Après cela, l'utilisateur peut exécuter les deux ‘algorithmes (NSGA-II et SPEA-II), qui vise à obtenir des ensembles de solution optimale. Dans notre cas ces algorithmes nécessitent plusieurs entrées: une population codée, et une population décodée, données de coût et de temps, et nombre maximal d'itérations. Les informations de temps et de coût sont utilisées pour l'évaluation de la qualité des solutions (fonction de fitness).

En fonction des objectifs de l'entreprise (minimisant le temps et le coût) et sa capacité à réaliser le produit, il détermine le poids de coût et le poids de temps, puis basé sur le décideur 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.

## 3.2. Modélisation des données

Dans cette sous-section, nous présentons les données d'entrée nécessaires et leur structure. Ces données d'entrée définissent quatre informations (produit, machines, coût et temps)[20].

Remarque: les notations suivantes sont utilisées pour définir à la fois les données d'entrée.

- $M_j(u)$: la machine $M_j$ effectuant l’opération d’index u
- $CM[M_j(u)]$ : le coût d'utilisation d'une machine $M_j$ pour effectuer l'opération u
- $C_j^l(u)$: la configuration de la machine l utilisée pour effectuer l’opération d’index u.
- $T_j^q(u)$ : l'outil q qui peut effectuer l'opération u sur la machine j

### 3.2.1 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, H et W représente les dimensions du produit.

#### b) Opérations

- TNOP : nombre total d'opérations requises pour produire un produit.
- PS = [1..NPS]: liste représentant le graphe de précédence, NPS : est le nombre de séquences d'opérations possibles.

#### c) Directions d'approche de l'outil (TAD : Tool approach directions) 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)

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

  Avec : OPTAD(x,d) : {1 si $OP_x$ a besoin du TADs d ,0 sinon}

x : index de l'opération

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*] {1 si l'outil t est capable d'effectuer l'opération x ,0 sinon}

X = 1.. OP index de l'opération

T = 1..NT index de l'outil

## 3.2.2 Structure de données des machines

La structure utilisée dans notre travail a été inspiré de la référence suivante [20].

### 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 ] {1 si l'outil t est disponible sur la machine m ,0 sinon}

m = 1..NM index de la machine

t =1..NT index de l'outil

- CTAD[1..NM][1..MAX(NC)][6] matrice des TAD disponibles par configuration de machine, avec : *CTAD*[*m*][*c*][*d*] {1 si la configuration c de la machine m offre le TAD d ,0 sinon}

m = 1..NM index de la machine

t =1..NT index de l'outil

d = 1..6 index de TAD

## 3.2.3 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)][1..n][1..MAX(NOP)][1..MAX(NC)][1..n][1..MAX(NOP)] : 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

TC[1..NM] c'est le coût d'utilisation de l'outil t = 1..NM

### e) Coût de changement des outils

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

## 3.2.4 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)][1..n][1..MAX(NOP)]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..NM] [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)][1..NT] : est le temps nécessaire pour effectuer une opération particulière sur une machine avec l'une de ses configurations en utilisant un outil approprié.

## 3.3 Méthode de résolution

### 3.3.1 Codages du processus de fabrication

Le Codage Des Solutions est une étape clé dans l'implémentation des métaheuristiques. Appelons que le codage consiste à passer de la représentation réelle des solutions vers une représentation codée[20]. 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 3.1.

C'est-à-dire une matrice M×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 1 et 0 , le étant non inclus (Tableau 3.2).

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 3.2. Un processus de fabrication codé en nombre réel [20]

L'algorithme suivant génère un processus de fabrication codé.

- M est égal à cinq lignes, chaque ligne est occupée respectivement pour les caractéristiques, opérations, machines, configurations et outils
- N correspond au nombre total d'opérations

**Algorithme :** générer un processus de fabrication codé

```
1 :  fonction
2 :  // Créer une matrice d'éléments réels M × N
3 :  // Affecter à chaque cellule de la matrice créée une valeur ∈] 0, 1 [
4 :  for i : 1..M do
5 :       for j : 1..N do
6 :            individu [i][j] ← random (0.01, 0.99)
7 : return individu
```

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.

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

### 3.3.2 Décodage du processus de fabrication

#### 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

4) Multiplier 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 p ($0 \leq p < n$).
5) Arrondir p au nombre entier immédiatement supérieur. Ainsi p sera entre 1 et n ($1 \leq p < n$).
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.

### 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.
3) Multipliez n par le nombre réel situé dans une cellule en cours de décodage. Cette étape renvoie un nombre réel p ($0 \leq p < n$).
4) Arrondir au nombre entier immédiatement supérieur. Ainsi sera entre et n ($1 \leq p < n$)
5) Dans la liste précédemment établie, l'opération se trouvant à la position 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-M2-M3-M4 , avec n = 3.

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 p entre 0 et n.
4) Arrondir p au nombre entier immédiatement supérieur. Ainsi p est compris entre 1 et n ($1 \leq p < n$)
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 (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 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.

3) Multiplier par le nombre réel se trouvant dans la solution codée, dans la cellule correspondante à la cellule à décoder. 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 p compris entre 0 et n .

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

5) Dans la liste précédemment établie, la configuration se trouvant dans la position 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 nombre d'outils capables de réaliser l'opération concernée. Par hypothèse, nous commençons par classer dans la liste les outils ayant un index inférieur (avant 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 p compris entre 0 et n .
4) Arrondir au nombre entier immédiatement supérieur. Ainsi, le nombre p sera compris entre 1 et n ($1 \leq p < n$).
5) Dans la liste précédemment établie, l'outil se trouvant dans la position p sera sélectionné pour être placé dans la cellule correspondante.

## 3.4 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 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

### 3.4.1 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 :

$$MUC = \sum_{u=1} C\,M[Mj(u)] \times PrTime[M_j(u)][C_l^j(u)][T_j(u)] \quad (3.1)$$

Avec :

- $M_j(u)$: la machine $M_j$ effectuant l'opération d'index u.
- $CM[M_j(u)]$ : le coût d'utilisation d'une machine $M_j$ pour effectuer l'opération u.
- $C_j^l(u)$: la configuration de la machine l utilisée pour effectuer l'opération d'index u.
- $PrTime[M_j(u)][C_l^j(u)][T_j(u)]$ 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 :

$$MCC = \sum_{u=1}^{T\,NOP} MCCost[Mj(u)][Mj'(u')] \qquad (3.2)$$

### 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 :

$$CCC = \sum_{u=1}^{T\,NOP} CCCost\,[Clj(u)][C_j^{l}{}'(u')] \qquad (3.3)$$

- $C_j^l(u)$: la configuration de la machine l utilisée pour effectuer l'opération d'index u.

### 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 :

$$TUC = \sum_{u=1}^{T\,NOP} CT[T(u)] \times PrTime[Mj(u)][Clj(u)][Tj(u)] \qquad (3.4)$$

Avec :

- Tj(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 de la machine l utilisée pour effectuer l'opération d'index u.
- $PrTime[M_j(u)][C_l^j(u)][T_j(u)]$ temps requis pour achever l'opération.

### 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 :

$$TCC = \sum_{u=1}^{T\,NOP} \text{TCCost}[T^{q}_{j}(u)][T^{q}_{j}{}'(u')] \qquad (3.5)$$

- Tj(u) l'outil utilisé pour effectuer l'opération d'index u.

Le premier objectif sera la minimisation de la somme des équations ( 3.1 ), ( 3.2 ), ( 3.3 ), (3.4 ) et ( 3.5 ). Ainsi, le premier objectif relatif au coût est exprimé sous la Forme Suivante :

$$\textbf{totalCost} = \text{MUC} + \text{MCC} + \text{CCC} + \text{T UC} + \text{T CC} \qquad (3.6)$$

### 3.4.2 Temps total

Le temps total requis pour effectuer toutes les opérations nécessaires d'un produit est calculé sur la base des quatre temps suivants :

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

C'est le temps nécessaire pour changer les configurations dans les machines. CCT formulé l'expression suivante :

$$\text{CCT} = \sum_{u=1}^{T\,NOP} \text{CCT ime}[\text{Clj}(u)][C^{l}_{j}{}'(u')] \qquad (3.7)$$

- $C_j^l(u)$: la configuration de la machine l utilisée pour effectuer l'opération d'index 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 :

$$TCT = \sum_{u=1}^{T\,NOP} T\,CT\,ime[T^{q}_{j}(u)][T^{q}_{j}{}'(u')] \qquad (3.8)$$

- Tj(u) l'outil utilisé pour effectuer l'opération d'index u.

### c. Temps de changement de machine

Comme mentionné dans l'équation, c'est le temps requis pour changer une machine j par une autre machine j ', pour effectuer deux successives différentes Opérations (u et u ').

$$MCT = \sum_{u=1}^{T\,NOP} M\,CT\,ime\,[Mj(u)][Mj'(u')] \qquad (3.9)$$

- $M_j(u)$: la machine $M_j$ effectuant l'opération d'index u.

### d. Temps opératoire

C'est le temps nécessaire pour effectuer une opération particulière sur une machine. L'expression du temps opératoire est donnée sous la forme suivante :

$$PT = \sum_{u=1}^{T\,NOP} PrTime[Mj(u)][C^{l}_{j}(u)][Tj(u)] \qquad (3.10)$$

- Mj(u) : la machine effectuant l'opération d'index u.
- Tj(u) : l'outil utilisé pour effectuer l'opération d'index u .
- $C^{l}_{j}(u)$ : la configuration d'index de la machine d'index effectuant l'opération d'index u .
- $PrTime[Mj(u)][C^{l}_{j}(u)][Tj(u)]$ temps requis pour achever l'opération .

le deuxième objectif sera la minimisation de la somme des équations ( 3.7 ), ( 3.8 ), ( 3.9 ), (3.10 ). Ainsi, la formule du temps total d'un processus de fabrication particulier est équivalente :

$$\textbf{totalTime} = CCT + T\,CT + MCT + PT \qquad (3.11)$$

## 4. Conclusion

Dans ce chapitre, nous avons présenté le problème de génération des processus de fabrication dans le cadre des systèmes reconfigurables. Un modèle mathématique du quel on s'est inspiré, 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'un produit. Une technique de codage en nombre réel nous inspiré pour éviter les solutions irréalisables, La prochaine étape du projet est l'implémentation de la conception.

# 1. Introduction

Après les étapes d'analyse et de conception mentionnées dans le chapitre précédent (chapitre 3), nous devons passer aux étapes suivantes du projet, qui sont le codage et le test. Ces phases visent à implémenter, tester et discuter les algorithmes que nous allons utiliser (NSGA-II et SPEA-II).

Ce chapitre comprend trois sections. La première section présente brièvement les outils de développement et les langages que nous avons appris et les exploite dans la réalisation de notre projet. La deuxième section présente les' interfaces graphiques de l'implémentation de notre application finale. Dans la dernière section, nous présentons et discutons quelques résultats expérimentaux.

# 2. Outils de développement et langage

Dans cette section, nous présentons différents outils et langages, qui nous aident lors de la réalisation de notre projet.

## 2.1 Langage de programmation Python

Python est un langage de programmation intelligent que nous avons utilisé dans l'implémentation de notre application. Il est facile à apprendre, parce qu'il est flexible, et sa syntaxe n'est pas difficile à apprendre. Un programme Python est court que les autres langage de programmation, en raison de la disponibilité de nombreuses fonctions mises en œuvre. Python est un langage de programmation open source et non typé. C'est disponible pour tous ces systèmes d'exploitation (Windows, LINUX, Mac OS).


**Figure 4.1** : Logo de Python

## 2.2 Éditeur de programmation PyCharm

PyCharm est un environnement de développement intégré (IDE :Integrated Development Environment), utilisé pour la programmation python. Il s'agit d'un assistant de copiage puissant, capable de mettre en évidence les erreurs et d'apporter des solutions rapides basées sur un débogueur Python intégré. C'est un éditeur approprié pour écrire et tester de nombreuses lignes de code et de classes, car il offre une vue de projet structurelle et une navigation rapide dans les fichiers.


**Figure 4.2** : Logo de PyCharm

## 2.3 Tkinter (Tool kit interface)

Est la bibliothèque graphique libre d'origine pour le langage Python, permettant la création d'interfaces graphiques. Nous avons préféré la boîte à outils Tkinter pour développer des interfaces graphiques de notre application, car il est simple à apprendre, et c'est une boîte à outils puissante. C'est disponible pour tous ces systèmes d'exploitation (Windows, LINUX, Mac OS).


**Figure 4.3** : Logo de Tkinter

## 3. L'implémentation

Les Algorithme évolutionnaire élitistes pour l'optimisation multiobjectif (NSGA-II et SPEA-II) sont implémentés en utilisant le paradigme de programmation orientée objet (OOP), et en utilisant un ensemble de logiciels et de matériels qui sont résumés dans le tableau suivant **(**Table 4.1).

| Logiciel /matériel | Version |
|---|---|
| SE | Microsoft Windows 8 Professional, 64bits, version 8.1 |
| CPU | Intel(R) Core(TM) i3-2310M CPU @2.50GHz |
| RAM | 4.00Go |
| Python | 3.4.0 |
| PyCharm | 2016.3.1 |
| Tkinter | 8.6 |

**Table 4.1** : Logiciel /matériel/ Version

### Interface principale :

Le Figure (4.4) suivante représente l'interface principale


**Figure 4.4** : l'interface principale

Nous avons créé l'interface principale pour faciliter l'utilisation de l'application. L'application contient une barre de menus avec trois menus (File, Optimisation, et Help) les figures (figure 4.5,figure 4.6 et figure 4.7) Ils représentent chaque menu et ses commandes.


**Figure 4.5**: menu « File »


**Figure 4.6** : menu « Optimisation »


**Figure 4.7** : menu « Help »

## Menu (File):

' File ' menu offre de nombreuses fonctionnalités pour l'utilisateur: créer un nouveau problème (machines et Information produit). Puis enregistrez les différentes données saisies dans VSV (valeurs séparées par des virgules). Formats qui sont des fichiers lisibles.

- Avec la commande '**NEW**', l'utilisateur peut créer de nouvelles machines et produits reconfigurables en tapant leurs informations selon la formulation au chapitre 3. Les données d'entrée sont entrées à travers la fenêtre représentée sur la figure 4.8.

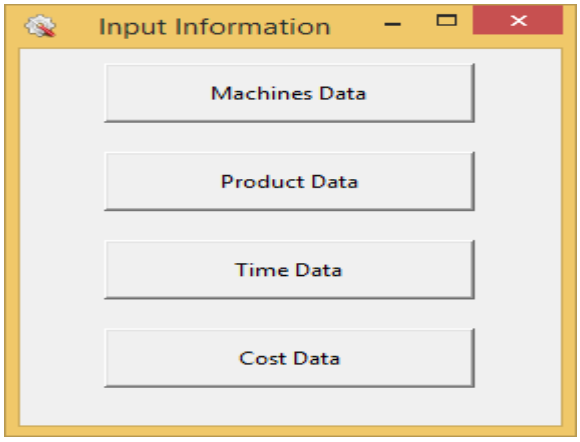


**Figure 4.8** : Les données d'entrée

Tout d'abord, l'utilisateur doit entrer les données des machines «machines data», puis les données produits «product data», après celle-ci peut entrer les données des de temps «time data» ou et de cout «cost data». Par exemple, si l'utilisateur veut taper «time data» ou «cost data» avant de taper «machines data» et «product data», un message d'avertissement s'affiche la figure 4.9.

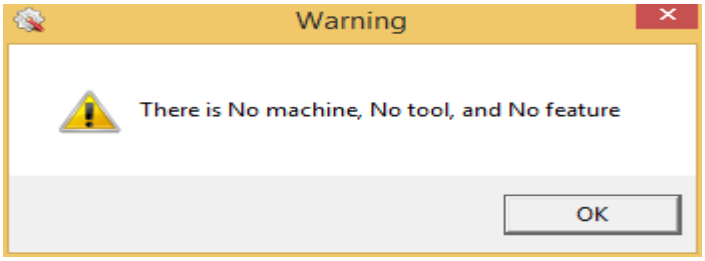


**Figure 4.9** : avertissement (entrer les données)

- En cliquant sur le bouton '**Machines Data**', une nouvelle fenêtre (Figure 4.10) s'affiche pour entrer les données des machines.

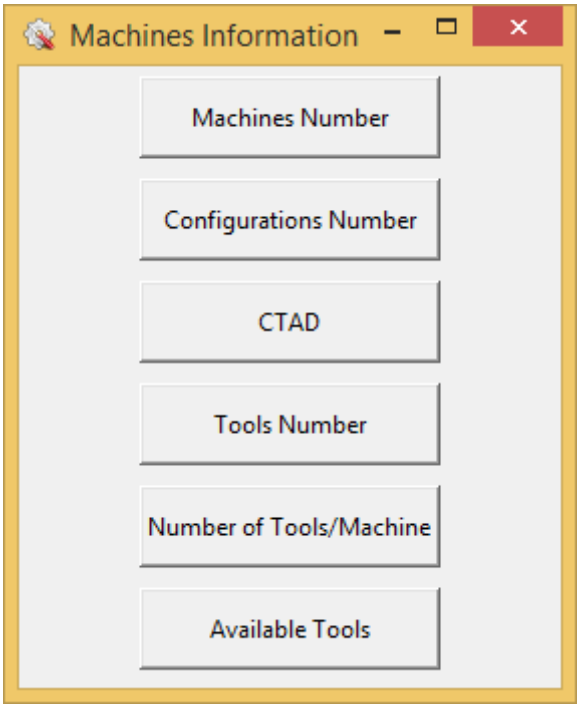


**Figure 4.10** : Les données des machines

Pour 'Configuration TAD', l'utilisateur doit identifier les valeurs des six axes de mouvement pour chaque configuration d'une machine comme indiqué dans les figures (4.11 et 4.12).

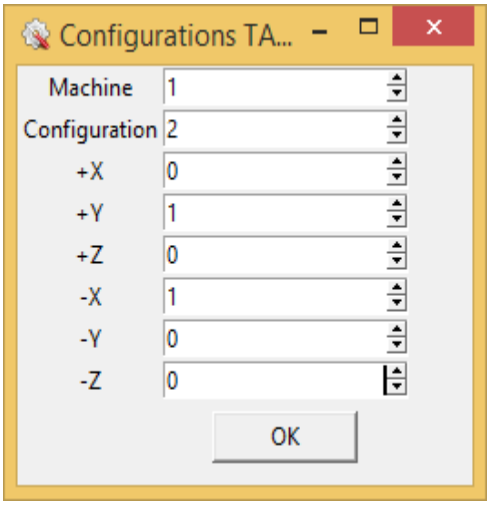


**Figure 4.11** : ConfigurationsTAD

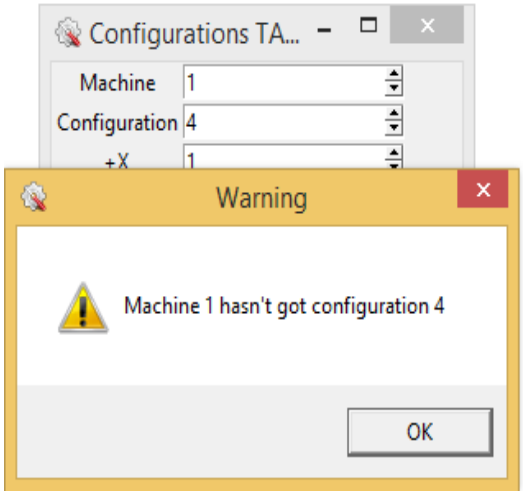


**Figure 4.12** : avertissement (ConfigurationsTAD)

La figure 4.11 montre que le Configurations TAD de la deuxième configuration de la machine 1 est un vecteur (0, 1, 0, 1, 0, 0).

L'ensemble des outils disponibles de chaque machine est rempli à l'aide de la case suivante, par exemple La figure 4.13 montre que la machine 1 peut utiliser l'outil 4.

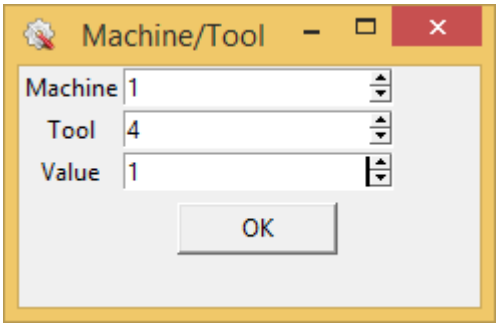


**Figure 4.13** : outils disponibles de chaque machine

- La deuxième étape des données d'entrée consiste à entrer des données de produit ' **product data**' (figure 4.8). Chaque information sur le produit entré en cliquant sur un bouton de la fenêtre '**Product Information**' (Figure 4.14).

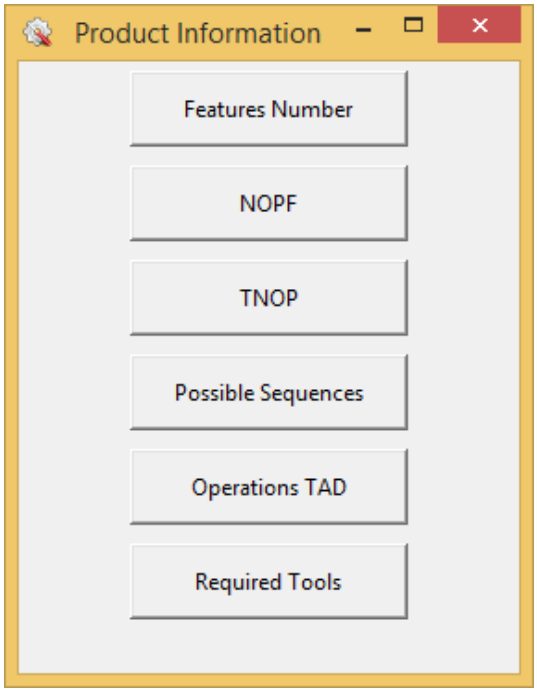


**Figure 4.14** : données de produit

Les séquences d'opérations possibles sont entrées en utilisant l'interface graphique suivante.

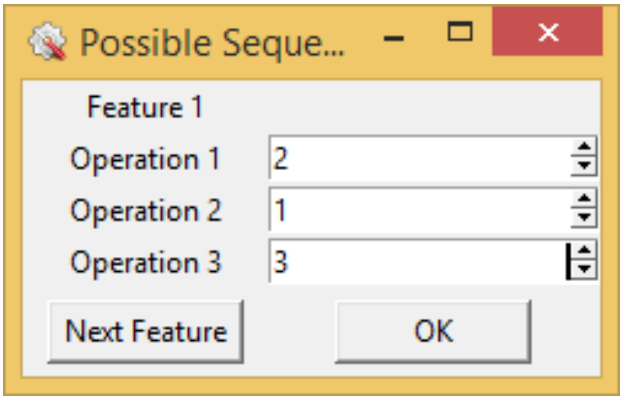


**Figure 4.15** : Les séquences d'opérations possibles

(Figure 4.15) illustre l'une des séquences d'opérations possibles de la caractéristique 1 qui est 2 $\longrightarrow$ 1 $\longrightarrow$ 3.

Les figures (4.16 et 4.17) montrent comment entrer le 'Operations TAD'. exemple : le figure 4.16 illustre TAD de l'opération 1 de la caractéristique 1 est un vecteur de six entrées (0, 1, 0, 1,

0, 0). le figure 4.18 montrent que Il n'y a pas de machine qui a un approprié configuration pour cette opération.

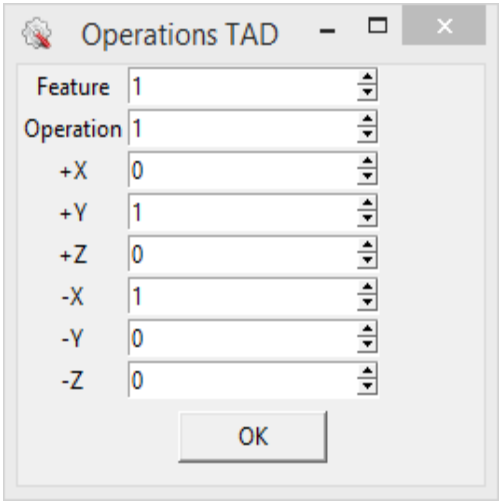


**Figure 4.16** : Operations TAD

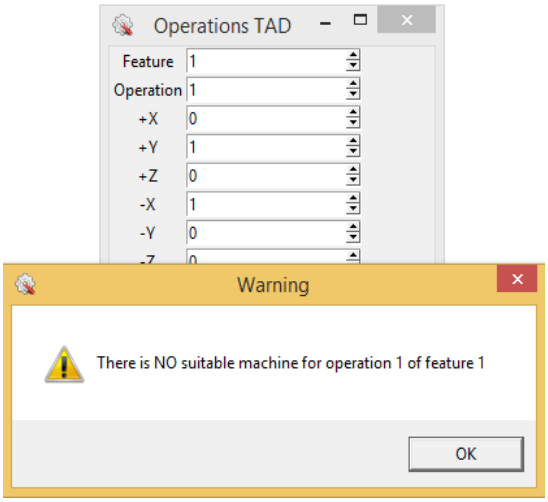


**Figure 4.17** : avertissement (Operations TAD)

Une fois que l'utilisateur a saisi toutes les données des machines et les informations sur le produit, il peut saisir les données de coût et de temps. Les figures 4.18 et 4.19 illustrent toutes les informations de temps et de coût nécessaires que l'utilisateur peut saisir.


**Figure 4.18** : données de coût

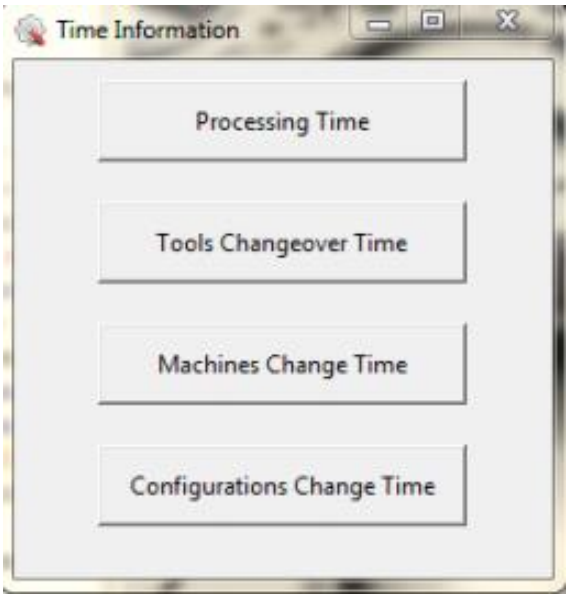


**Figure 4.19** : données de temps

Le temps de traitement d'une opération dépend du type d'opération, de son outil requis, sur la machine, et sa configuration. L'utilisateur peut faire face à de nombreux cas lors de la saisie du traitement le temps d'une opération, certains d'entre eux sont illustrés à la figure 4.20 et à la figure 4.21

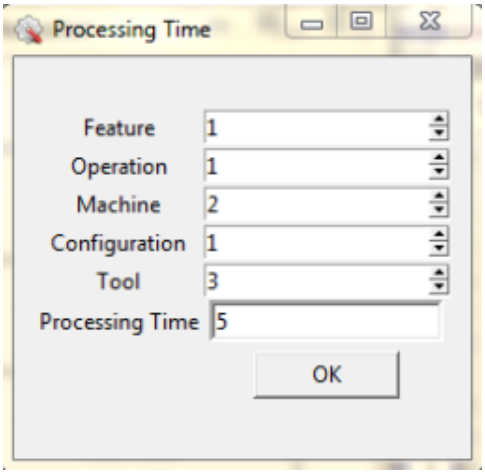


**Figure 4.20** : Le temps de traitement d'une opération

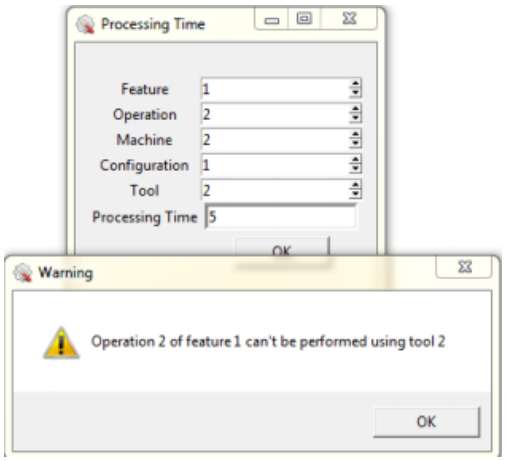


**Figure 4.21** : avertissement (erreur de saisie)

(Figure 4.20) illustre le temps requis pour effectuer la première opération de la caractéristique 1 sur la machine 2 avec la configuration 1 en utilisant l'outil 3, est égal à 5.

A propos des configurations, le coût de modification de la configuration dépend du type d'opération.

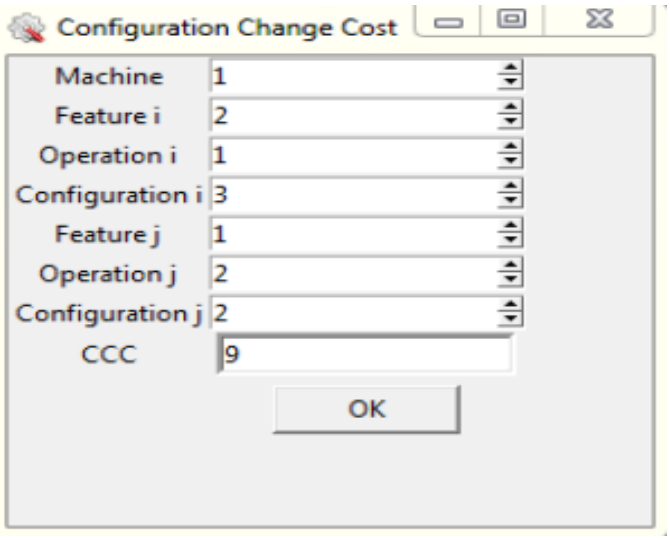


**Figure 4.22** : Coût de changement de configuration

Dans la figure 4.22, le coût de la modification de la configuration 3 à la configuration 2 de la machine 1, à effectuer deux opérations consécutives (opération 1 de la caractéristique 2 et opération 2 de la caractéristique 1) sur la même machine (machine 1) égale à 9.

Dans notre application, nous avons étudié le plus de cas particuliers de données d'entrée. Par exemple, la machine ne peut pas effectuer l'opération si elle ne peut pas utiliser l'un des outils nécessaires de l'opération, comme montré dans la figure 4.23.

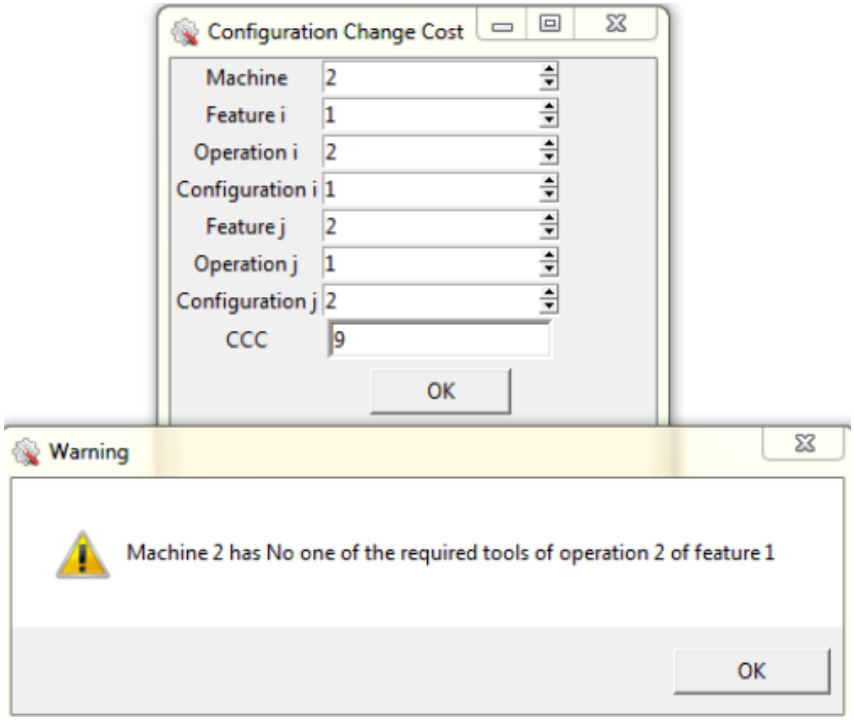


**Figure 4.23** : avertissement (erreur de saisie des outils)

## Menu(Optimisation) :

Permet à l'utilisateur d'exécuter l'algorithme NSGA-II ou SPEA-II chaque commande (NSGA-II ou SPEA-II) permette à l'utilisateur de saisir les entrées nécessaire et d'exécuter une des commandes. Exécutez ensuite la technique TOPSIS pour obtenir la solution optimale (voir Figure 4.24 et 4.25).

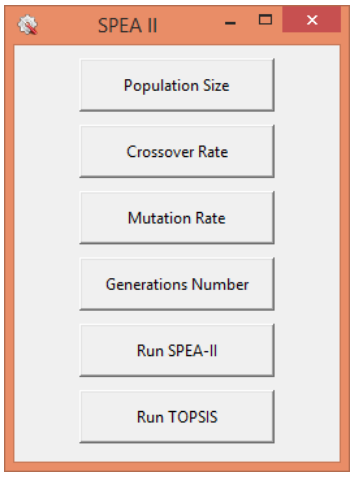


**Figure 4.24** l'algorithme SPEA-II

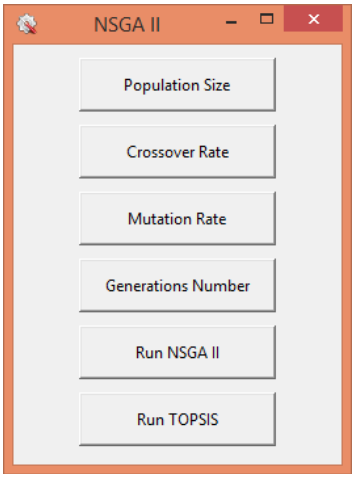


**Figure 4.25** : l'algorithme NSGA-II

L'utilisateur doit identifier les entrées nécessaires qui sont la taille de la population, le taux de croisement, le taux de mutation et le nombre d'itérations. Toutes ces entrées sont entrées à travers les cases à cocher des figures 4.26, 4.27, 4.28 et 4.29. La population initiale est créée lorsque l'utilisateur saisit la taille de la population.

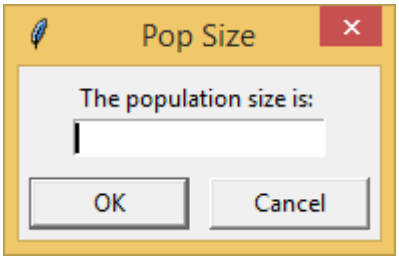


**Figure 4.26** la taille de la population

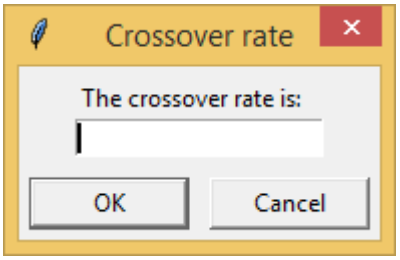


**Figure 4.27** le taux de croisement

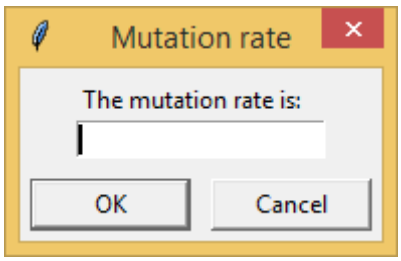


**Figure 4.28** le taux de mutation

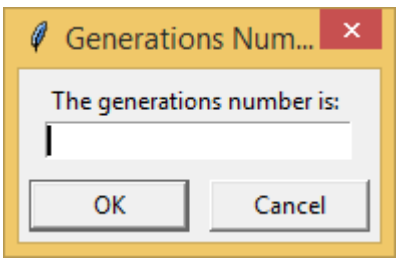


**Figure 4.29** le nombre d'itérations

L'utilisateur exécute NSGA-II ou SPEA-II pour obtenir le front de Pareto (l'ensemble des solutions optimales) la dernière génération, il peut lancer TOPSIS pour obtenir la meilleure solution, si l'utilisateur n'exécute pas NSGA-II ou SPEA-II en premier, une boîte de message d'avertissement apparaît (Figure 4.30).

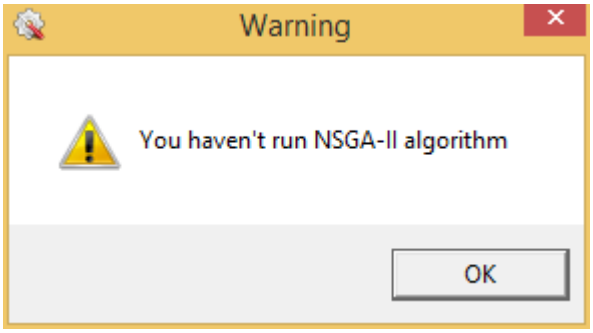


**Figure 4.30** avertissement TOPSIS

Pour exécuter l'approche TOPSIS, l'utilisateur doit identifier les entrées nécessaires de TOPSIS sont le poids du temps et le poids du coût, alors il peut cliquer sur le bouton «Exécuter TOPSIS» pour obtenir la solution optimale (Figure 4.31 et 4.32).

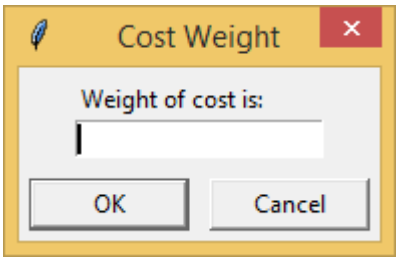


**Figure 4.31** le poids du coût

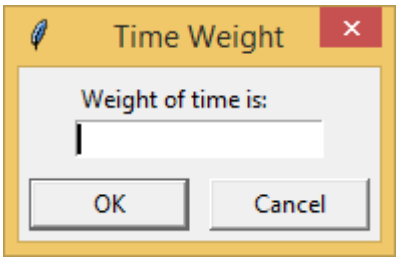


**Figure 4.32** le poids du temps

## 4. 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 des deux Algorithme évolutionnaire élitistes pour l'optimisation multiobjectif (NSGA-II et SPEA-II).

## 4.1 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.

Les données des machines d'entrée se composent de trois machines (M1, M2 et M3) qui partagent un ensemble d'outils (T1, T2, T3, T4, T5, T6 et T7). Le tableau 4.2 illustre les données d'entrée des machines inclure les configurations TAD et l'ensemble des outils disponibles pour chaque machine.

| **MACHINE** | **CONFIGURATION** | **TAD**<br>**+X –X +Y –Y +Z -Z** | **outils** |
|---|---|---|---|
| **MACHIN1** | CONFIGURATION1 | 1 1 1 0 0 0 | 1,2,4 |
| | CONFIGURATION2 | 1 0 1 0 1 0 | |
| | CONFIGURATION3 | 1 1 0 1 0 1 | |
| **MACHIN2** | CONFIGURATION1 | 1 0 0 0 1 1 | 1,2,3,6 |
| | CONFIGURATION2 | 0 1 1 1 0 0 | |
| | CONFIGURATION3 | 0 1 0 1 0 1 | |
| | CONFIGURATION4 | 0 0 1 1 1 1 | |
| **MACHIN3** | CONFIGURATION1 | 1 1 1 0 0 0 | 1,5,7 |
| | CONFIGURATION2 | 1 0 1 0 1 0 | |

**Table 4.2** Opérations TAD et outils requis

Dans cet exemple, le produit est composé de trois entités, la première caractéristique ayant trois opérations à accomplir, la deuxième caractéristique ayant cinq opérations à effectuer et la troisième caractéristique ayant quatre opérations à accomplir. Ainsi, le nombre total d'opérations est égal à six.

Selon le graphe de précédence (voir Figure 4.33), la première caractéristique n'a qu'une seule séquence d'opérations:

1 ⟶ 2 ⟶ 3

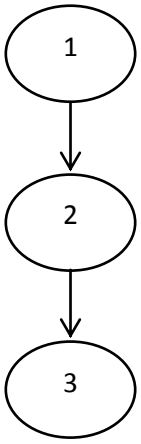


**Figure 4.33** Graphe de précédence pour la première caractéristique

En fonction du graphe de précédence de la deuxième caractéristique qui est illustrée à la Figure 4.34, il comporte six séquences d'opérations possibles:

1 ⟶ 2 ⟶ 3 ⟶4 ⟶ 5

1 ⟶ 2 ⟶ 3 ⟶5 ⟶ 4

1 ⟶ 3 ⟶ 2 ⟶4 ⟶ 5

1 ⟶ 3 ⟶ 2 ⟶5 ⟶ 4

1 ⟶ 3 ⟶ 4 ⟶5 ⟶ 2

1 ⟶ 3 ⟶ 5 ⟶4 ⟶ 2

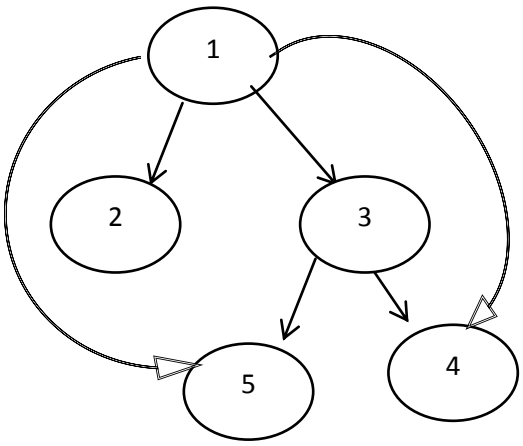


**Figure 4.34** Graphe de précédence pour la deuxième caractéristique

Selon le graphe de précédence de la figure 4.35, les séquences d'opérations possibles sont:

1 ⟶ 2 ⟶ 3 ⟶4

1 ⟶ 3 ⟶ 2 ⟶4

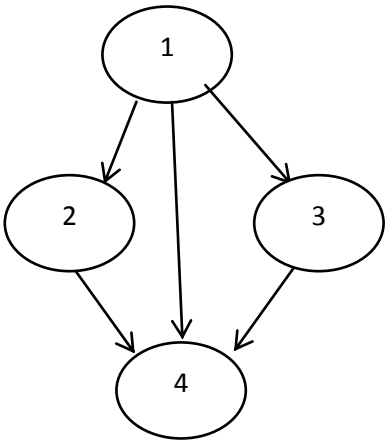


**Figure 4.35** Graphe de précédence pour la troisième caractéristique

Le Tableau 4.3 montre les données d'entrée du produit qui sont des caractéristiques et ses TAD d'opérations ainsi que l'outil requis pour chaque opération.

| Caractéristique | Opérations | TAD<br>+X –X +Y –Y +Z -Z | outils |
|---|---|---|---|
| Caractéristique1 | Opérations 1 | 1 1 0 0 0 0 | 1 |
| | Opérations 2 | 0 0 0 0 1 1 | 2 |
| | Opérations 3 | 1 0 0 0 0 0 | 1 |
| Caractéristique2 | Opérations 1 | 0 1 0 0 0 0 | 3 |
| | Opérations 2 | 0 0 0 0 1 1 | 5 |
| | Opérations 3 | 0 1 0 0 0 1 | 4 |
| | Opérations 4 | 0 0 0 0 1 0 | 1 |
| | Opérations 5 | 0 0 1 1 0 0 | 1 |
| Caractéristique3 | Opérations 1 | 0 0 1 0 0 0 | 3 |
| | Opérations 2 | 0 1 0 0 0 0 | 4 |
| | Opérations 3 | 0 0 0 0 1 0 | 4 |
| | Opérations 4 | 1 0 1 0 0 0 | 2 |

**Table 4.3** Opérations TAD et outils requis(produits)

Après avoir obtenu les informations sur les produits et les données des machines, nous pouvons identifier les données de temps et les données de coût.

## **Exécution les deux algorithmes** (NSGA-II et SPEA-II) :

À ce stade, toutes les données des machines, informations sur le produit, temps et coût sont identifiés. Pour exécuter L'algorithme (NSGA-II ou SPEA-II), il est founi pour

déterminer la taille de la population, le taux de mutation, le taux de croisement et le nombre de générations. Par conséquent, nous prendrons comme paramètres :

- Taille de la population = 30. Taux de croisement = 0.8
- Taux de mutation = 0.2. Nombre de générations = 10

## 4.2 Résultats obtenus

Après l'exécution  Les deux algorithmes ont donné deux fronts de Pareto différents Chaque point du front de Pareto représente un processus de fabrication, c'est-à-dire une matrice M×N (Tableau 3.1), 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 4.4.

La représentation graphique de ces résultats sous forme de fronts Pareto montre une capacité de convergences plus importante avec NSGA-II par rapport au SPEA-II (Figure 4.37et 4.38). En effet, NSGA-II a réussi à trouver un front plus optimal que celui de SPEA-II. La simplicité d' NSGA-II permet d'effectuer plusieurs itérations de recherche avec un faible effort de calcul.

| SPEA-II | | NSGA-II | |
|---|---|---|---|
| Coût total | Temps total (ms) | Coût total | Temps total (ms) |
| 1048 | 148 | 1043 | 149 |
| 1046 | 152 | 1032 | 150 |
| 1045 | 153 | 1025 | 150 |
| 1012 | 153 | 1022 | 150 |
| 1003 | 156 | 1019 | 151 |
| 1001 | 159 | 1019 | 151 |
| | | 1016 | 152 |
| | | 1007 | 156 |

**Table 4.4** Résultats obtenus par SPEA-II et NSGA-II

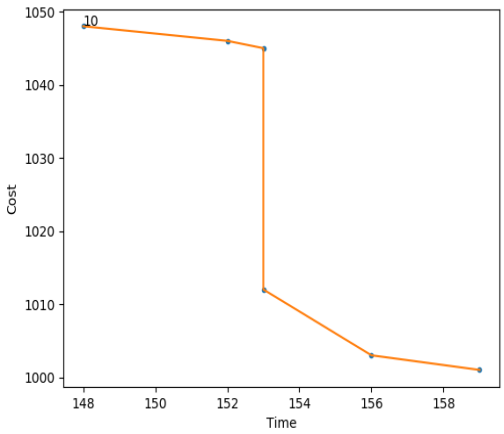


**Figure 4.36** le front de Pareto obtenus par SPEA-II

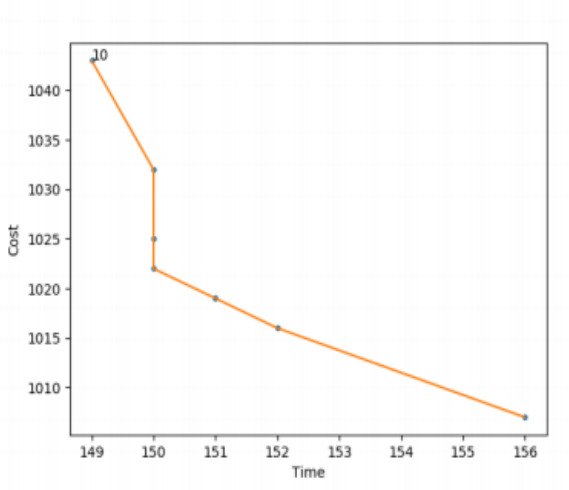


**Figure 4.37** le front de Pareto obtenus par NSGA-II

Le front de Pareto de la dernière génération de l'algorithme NSGA-II est composé de huit solutions, et selon l'approche TOPSIS (Technique for Order Performance by Similarity to Ideal Solution) comme support de décision, les solutions du front de Pareto sont classées dans l'ordre croissant, où la meilleure solution du front de Pareto est celle qui a le plus bas rang TOPSIS. Le classement des solutions du front de Pareto est présenté dans le tableau 4.33.

| NSGA-II | | **TOPSIS** |
|---|---|---|
| Coût total | Temps total | |
| 1043 | 149 | 6 |
| 1032 | 150 | 3 |
| 1025 | 150 | 2 |
| 1022 | 150 | **1** |
| 1019 | 151 | 4 |
| 1019 | 151 | 5 |
| 1016 | 152 | 7 |
| 1007 | 156 | 8 |

**Table 4.6** Résultats obtenus, triés avec TOPSIS

## Les paramètres TOPSIS sont:

- Poids de Coût = 5
- Poids de Temps = 8

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F3 | F1 | F1 | F2 | F3 | F3 | F2 | F3 | F2 | F1 | F2 | F2 |
| O1 | O1 | O3 | O1 | O2 | O3 | O3 | O4 | O4 | O2 | O5 | O2 |
| M2 | M3 | M3 | M2 | M2 | M1 | M1 | M1 | M1 | M2 | M2 | M3 |
| C2 | C2 | C2 | C2 | C1 | C3 | C3 | C1 | C2 | C1 | C4 | C1 |
| T3 | T1 | T1 | T3 | T4 | T4 | T4 | T2 | T1 | T2 | T1 | T5 |

**Table 4.6** La meilleure solution optimal

Le tableau 4.6 montre la meilleure solution du front de Pareto. Son temp total est de 150 et le cout total 1022.

## 5. Conclusion

Dans ce chapitre, nous avons présenté les outils que nous avons utilisés pour lancer notre application, puis les résultats de notre implémentation sous la forme d'un ensemble 'interfaces utilisateur graphiques (GUI). La dernière section a discuté des résultats de l'exécution de l'exemple proposé, les résultats ont démontré de meilleures performances pour NSGA-II par rapport au SPEA-II.

# *Conclusion générale*

Au cours des dernières années, il y a eu un intérêt croissant pour les systèmes de production reconfigurables (RMS). Notre projet traitait d'un problème d'optimisation multi-objectif qui sélectionnait les machines reconfigurables optimales pouvant libérer le produit souhaité dans un temps et un coût minimisés.

Sur la base de l'approche présentée dans [20], nous avons implémenté Les Algorithme évolutionnaire élitistes pour l'optimisation multiobjectif (NSGA-II et SPEA-II) sur la même formalisation des problèmes pour obtenir l'ensemble optimal de processus de fabrication pouvant réaliser le produit optimal. Après cela, nous avons mis en œuvre la technique TOPSIS pour aider l'entreprise a décider quelle est la meilleure solution selon ses préférences.

Nous avons implémenté ces algorithmes en utilisant le langage de programmation Python.

D'autres questions restent à être résolues dans des recherches futures, dont certaines sont :

- Implémentation d'autres algorithmes d'optimisation multi-objectifs récemment proposés tels que NSGA-III
- L' amélioration visée est le passage d'un problème d'optimisation multi-objectifs à deux objectifs (tel qu'exposé dans le projet) vers une optimisation atteignant plusieurs objectifs (plus de trois objectifs).

# Bibliographie

[1] : Mahdi Samir. Optimisation Multi-objectif Par Un Nouveau Schéma De Coopération Méta/Exacte. (2010). Université Mentouri de Constantine. Thèse de magister.

[2] : Vincent Barichard, Jin-Kao Hao. Une approche hybride pour l'optimisation multi objectif sous contraintes. (2003). RSTI/hors-série. JFPLC p43-45. Article scientifique.

[3] : Jessie Birman (2013). Quantification et propagation d'incertitude dans les phases amont de projets de conception d'avions : de l'optimisation déterministe à l'optimisation sous contraintes probabilistes. 2013. Université de Toulouse. Thèse de doctorat.

[4] : Ayadi Azzabi (2010). Optimisation Multicritere de La Fiabilite : Application du Modele de Goal Programming Avec Les Fonctions de satisfactions dans l'industrie de traitement de Gas. Thèse de doctorat.

[5] : Clarisse Dhaenens-Flipo. Optimisation Combinatoire Multi-Objectif : Apport des Méthodes Coopératives et Contribution à l'Extraction de Connaissances. (2005). Université des Sciences et Technologies de Lille. Thèse d'habilitation pour la direction des recherches de l'U.S.T.L.

[6] : Yann Collette, Patrick Siarry. Optimisation multi-objectif. (2002). Éditions eyrolles 61, Bld Saint-Germain 75240 Paris Cedex 05. Livre.

[7] : Claude Perdigou. Caractérisation de comportement dynamique en robotique mobile & application de la robotique évolutionniste. (2011). Université Pierre et Marie Curie, ISIR, École Normale Supérieure.

[8] : Faiza Sadi Algorithmes exacts et approchés pour les problèmes d'ordonnancement multi-agent à machines parallèles. (2015). Université François Rabelais de Tours. Thèse de doctorat.

[9]: Kalynmoy Deb. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. (2002) IEEE transactions on evolutionary computation, VOL. 6, NO. 2, pp 182-196 Article scientifique.

[10] : Leila Dridi. Développement et la validation d'une approche de remplacement des conduites pour les réseaux d'eau portable. (2005). Université du Québec INRS-ETE. Thèse de doctorat.

[11] : Ahmed Atahran (2012). Etude et résolution d'un problème de transport à la demande multicritère. Université François Rabelais de Tours. Thèse de doctorat.

[12] : Colas durand(2004),'Algorithmes génétiques et autres outils d'optimisation appliqués à la gestion du trafic aerien, Thèse d'habilitation pour la direction des recherches de l'U.N.P.T.

[13] : Souquet Amédée, Radet François-Gérard. Algorithme génétique. (2004). TE de fin d'année.

[14] : Les algorithmes génétiques. (2005). Site web : http://khayyam.developpez.com/articles/algo/genetic/, derrière consultation le : 17/02/2018.

[15] : Les Algorithmes évolutionnaires. Site web : http://www.pinville.com/algorithmes-evolutionnistes.php, dernière consultation le : 17/02/2018.

[16] : Félix-Antoine Fortin, Marc Parizeau. Revisiting the NSGA-II Crowding-Distance Computation. Laboratoire de vision et systèmes numériques. (2013). GECCO '13 Proceedings of the 15th annual conference on genetic and evolutionary computation, pp 623-630. Article scientifique.

[17] : Michael L.Pinedo. Scheduling: Theory, Algorithms, and Systems third edition. (2012). Springer New York Dordrecht Heidelberg London. Livre.

18] : Huixue Zhao. Analyse de Faisabilités pour l'ordonnancement de taches en graphe dans les systèmes temps réel. (2007). Université Paris XII. Thèse de doctorat.

[19] : Zitzler et Thiele, 1999. Comparison of Multiobjective Evolutionary Algorithms. Article scientifique.

[20] : Bensmaine, A. (2013), 'Algorithmes évolutionnaires et méthodes approchées multicritères pour la génération des processus de fabrication dans un environnement reconfigurable'. Thèse de doctorat.

[21] : Youssef Benama (2016), Formalisation de la démarche de conception d'un système de production mobile : intégration des concepts de mobilité et de reconfigurabilité. Thèse de doctorat.

[22] : Imad CHALFOUN (2014), 'Conception et déploiement des Systèmes de Production Reconfigurables et Agiles (SPRA) '. Thèse de doctorat.

[23]: 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).

[24] : Koren, Y. (2006), General rms characteristics. comparison with dedicated and flexible systems, in 'Reconfigurable manufacturing systems and transformable factories', Springer, pp. 27–45.

[25]: Goyal, K. K., Jain, P. K. & Jain, M. (2011), 'Multiple objective optimization of reconfigurable manufacturing system', Proceedings of the International Conference on SocProSpp.

[26] : 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.

[27] : Bensmaine, A., Benyoucef, L. & Dahane, M. (2011), 'Process plan generation in reconfigurable manufacturing systems using adapted NSGA-II and AMOSA', IEEE International Conference on Industrial Informatics (INDIN).

[28]: Haddou Benderbal, H., Dahane, M. & Benyoucef, L. (2017), 'Flexibility-based multiobjective approach for machines selection in reconfigurable manufacturing system (RMS) design under unavailability constraints', International Journal of Production Research.