



UNIVERSITE MOHAMED BOUDIAF - M'SILA
FACULTE DES MATHEMATIQUES ET
DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE



MEMOIRE de fin d'étude
Présenté pour l'obtention du diplôme de MASTER
Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Informatique Décisionnel et Optimisation

Par : KACIMI EL HASSANI Aicha
ABASSI Hayat

SUJET

**L'optimisation à base de simulation pour les chaînes
logistiques**

Soutenu publiquement le : / /2020 devant le jury composé de :

Dr. LOUNASS Billal
Dr. BOUNIF Mohamed Elhadi
Dr. MOKHTARI Rabeh

Université de M'sila
Université de M'sila
Université de M'sila

Président
Rapporteur
Examineur

Promotion : 2019 /2020

Dédicaces

A nos très chers parents.

Remerciements

On remercie Dieu le tout puissant, qui nous a donné la force et la patience pour l'accomplissement de ce travail.

*Nous remercions en particulier Mr. **BOUNIF Mohamed Elhadí**, pour l'honneur qu'il nous a fait de bien vouloir nous encadrer, et pour les conseils donnés lors de la réalisation de ce travail.*

On adresse nos remerciements au membre de jury pour avoir accepté de nous prêté de leur attention et évaluer notre travail.

Table des Matières

INTRODUCTION GENERALE	1
 CHAPITRE 1 : LES CHAINES LOGISTIQUES	
Introduction	5
1 Les chaînes logistiques	5
1.1 Définition	5
1.1.1 Chaîne logistique d'un produit	5
1.1.2 Chaîne logistique d'une entreprise	5
1.2 Structure de la chaîne logistique	5
1.2.1 Dyadique	6
1.2.2 Série	6
1.2.3 Convergente	6
1.2.4 Divergente	7
1.2.5 Réseau	7
1.3 Les flux d'une chaîne logistique	8
1.3.1 Le flux d'information	9
1.3.2 Le flux physique	9
1.3.3 Le flux financier	9
1.4 Les fonctions de la chaîne logistique	9
1.4.1 L'approvisionnement	9
1.4.2 La production	10
1.4.3 Le stockage	10
1.4.4 Distribution	10
1.4.5 La vente	10
2. La gestion de la chaîne logistique (SCM)	10
2.1 Les niveaux de décisions dans une chaîne logistique	10
2.1.1 Le niveau stratégique	11
2.1.2 Le niveau tactique	11
2.1.3 Le niveau opérationnel	11

2.2 Modélisation de la chaîne logistique	11
2.2.1 Les modèles conceptuels	11
2.2.2 Les modèles mathématiques	11
2.2.3 Les modèles par simulation	12
2.3 Planification de la chaîne logistique	12
2.4 Les mesures de la performance de la chaîne logistique	12
3. problème d'ordonnancement	13
3.1 définitions et notions fondamentales	13
3.2 Les éléments du problème d'ordonnancement	13
3.2.1 Les ressource	13
3.2.2 Les tâches	14
3.2.3 Les objectifs	14
3.2.4 Les contraintes.....	15
3.3 Schémas de classification des problèmes d'ordonnancement	15
3.4 Les types des problèmes d'ordonnancement d'atelier	16
3.4.1 Une seule machine	16
3.4.2 Machines identiques en parallèle	16
3.4.3 Atelier mono-gamme (Flow shop)	17
3.4.4 Atelier mono-gamme flexible (Flow shop hybride)	17
3.4.5 Atelier multi-gammes (job shop)	18
3.4.6 Atelier ouvert (open shop)	18
Conclusion	18

CHAPITRE 2 : OPTIMISATION BASEE SUR LA SIMULATION

Introduction	20
1 Optimisation des chaînes logistiques	20
1.1 Principaux concepts en optimisation	20
1.3 Face à un problème d'optimisation	21
1.4 Les outils pour l'optimisation de la chaine logistique	22
1.4.1 Méthodes exactes	22
1.4.2 Méthodes approchées	23
1.4.3 Heuristique	23

1.4.4 Métaheuristiques	23
1.4.4.1 Les méthodes de recherche locale (Local Search Methods)	24
1.4.4.2 Les méthodes de recherche globale (Global Search Methods) ...	24
1.4.5 Algorithmes à performance garantie	24
1.4.6 Approches hybrides	24
2 La simulation des chaînes logistiques	25
3 Le couplage simulation et optimisation	25
Conclusion	26
 CHAPITRE 3 : ALGORITHMES GENETIQUES ET OPTIMISATION MULTICRITERES	
Introduction	28
1. Algorithmes génétiques multicritères	28
1.1 Principes de base des algorithmes génétiques	28
1.2 Fonctionnement des AGs	28
1.3 Représentation chromosomique (Codage)	30
1.3.1 Codage binaire	30
1.3.2 Codage réel	30
1.4 Evaluation	30
1.5 Sélection	31
1.5.1 Méthode de la loterie biaisée	31
1.5.2 Sélection par tournoi	31
1.6 Opérateurs génétiques	32
1.6.1 Croisement	32
1.6.2 Mutation	33
2. L'optimisation multi-objectif	33
3. Optimisation multi-objectif par algorithmes génétiques	35
3.1. Méthodes agrégées	36
3.1.1. La moyenne pondérée	36
3.1.2. Goal Programming	36

3.2. Méthodes non-agrégées non-Pareto	36
3.2.1. Vector evaluated genetic algorithm (VEGA)	36
3.3 Méthodes Pareto	37
3.3.1. Multiple Objectives Genetic Algorithm (MOGA)	37
3.3.2. Non-dominated Sorting Genetic Algorithm II (NSGA II)	38
Conclusion	41
 CHAPITRE 4 : LA METHODE DE RESOLUTION ET IMPLEMENTATION	
Introduction	43
1. La méthode de résolution	43
1.1 L'objectif de la méthode	43
1.2 L'architecture de la méthode	44
2.Outils et environnement de développement	45
3. Quelque exemple sur l'implémentation	45
3. 1 Illustration de l'application	51
Conclusion	55
CONCLUSION GENERALE	56
REFERENCES BIBLIOGRAPHIQUES	59

Liste des Figures

Figure 1.1 Chaîne dyadique	6
Figure 1.2 Chaîne série	6
Figure 1.3 Chaîne convergente	7
Figure 1.4 Chaîne divergente	7
Figure 1.5 Chaîne réseau	8
Figure 1.6 Les Flux de la chaîne logistique	8
Figure 1.7 Une seule machine	16
Figure 1.8 Machines identiques en parallèle	16
Figure 1.9 Flow shop	17
Figure 1.10 Flow shop hybride	17
Figure 1.11 job shop	18
Figure 2.1 Classification des méthodes d'optimisation	22
Figure 3.1 Fonctionnement de l'algorithme génétique	29
Figure 3.2 Codage binaire de trois variables	30
Figure 3.3 Codage réel	30
Figure 3.4 Méthode de sélection de la loterie biaisée	31
Figure 3.5 : Croissement en un seul point	32
Figure 3.6 Croissement en deux points	32
Figure 3.7 Mutation d'un chromosome	33
Figure 3.8 Problème d'optimisation multi objectif (2 variables de décision et 3 fonction objectifs)	34
Figure 3.9 Front de Pareto	35
Figure 3.10 Vector Evaluated Genetic Algorithm (VEGA)	37
Figure 3.11 le fonctionnement du NSGA II	40
Figure 3.12 Distance de crowding (les points noirs sont les solutions appartenant au même front)	41
Figure 4.1 L'architecture de la méthode utilisée	44
Figure 4.2 Résultats de meilleurs modèles obtenus par NSGA II de problème 4x7. ...	46
Figure 4.3 Exemple de Diagramme de Gant obtenu par NSGA II de problème 4x7. ...	48
Figure 4.4 Résultats de meilleurs modèles obtenus par NSGA II de problème 3x9. ...	49
Figure 4.5 Exemple de Diagramme de Gant obtenu par NSGA II de problème 3x9...	51
Figure 4.6 Interface principale de l'application	52
Figure 4.7 La représentation du courbe	53

Figure 4.8 l'ordonnancement optimale	54
Figure 4.9 Diagramme de Gantt	54
Figure 4.10 La comparaison entre l'état initiale et l'état optimale	55

Liste des Tables

Table 4.1 La table des tâches Exemple 1	45
Table 4.2 La table des temps du traitement de la solution initiale	46
Table 4.3 La table des temps du traitement de la solution optimale	47
Table 4.4 La table de la solution optimale avec NSGA II Exemple 1	47
Table 4.5 La table des tâches Exemple 3	48
Table 4.6 La table des temps du traitement de la solution initiale	49
Table 4.7 La table des temps du traitement de la solution optimale	50
Table 4.8 La table de la solution optimale avec NSGA II Exemple 3	50

Liste des Algorithmes

Algorithme 3.1 L'algorithme de NSGA II	39
---	----

INTRODUCTION GENERALE

Dans de nombreux secteurs industriels, en particulier celui de la gestion de production, les chaînes logistiques sont soumises à des aléas importants qui menacent leur développement et réduisent souvent leurs performances. Dans ces circonstances, le domaine de gestion des chaînes logistiques a suscité ces dernières années un grand intérêt.

La chaîne logistique englobe toutes les opérations réalisées pour la fabrication d'un produit ou d'un service, de l'extraction des matières premières à la livraison du client final, en passant par les différentes étapes de transformation, de stockage, d'approvisionnement et de Distribution.

Les chaînes logistiques sont un résultat de la vague de mondialisation et de la globalisation des marchés. En effet, dans un environnement de plus en plus concurrentiel, de ce fait la concurrence pour réduire en permanence les coûts, maintenir la qualité des produits finis et améliorer la qualité de service aux clients est devenu de rigueur plus que jamais.

À tout moment, la chaîne logistique peut devenir victime des retards de livraison, d'une mauvaise estimation, de marchandises perdues et de nombreux autres événements imprévisibles. Tous ces événements imprévus peuvent entraîner une pénurie de marchandises dans n'importe quelle partie de la chaîne logistique, ce qui peut entraîner d'énormes conséquences et même de grosses pertes. Ceci justifie le besoin de créer un outil qui permettrait de simuler des situations logistiques.

Notre travail s'articule autour de cette problématique dans le but est de proposer une méthodologie pour l'optimisation et la simulation des chaînes logistiques, ainsi que le développement d'un logiciel supportant cette méthodologie. Plus précisément, nous utilisons une approche d'optimisation multicritères à base de la simulation pour la conception optimale des chaînes logistiques. Cette approche comprend un module d'optimisation basé sur des algorithmes génétiques multicritères et un module de simulation permettant l'évaluation de la chaîne étudiée au cours de l'optimisation.

Le mémoire est organisé en quatre chapitres.

Le premier chapitre de ce mémoire propose, une vue d'ensemble sur les concepts d'une chaîne logistique et de sa gestion avec les différents niveaux décisionnels, ainsi que les principaux types ou architectures de chaînes logistiques, décrivons quelques outils et notions tel que : les différentes fonctions et les mesures de performance. Et dernière section, nous introduisons les concepts d'ordonnancement. Nous rappelons brièvement les notions d'ordonnancement d'atelier.

Le deuxième chapitre nous introduisons les concepts, les caractéristiques générales d'approche d'optimisation et les différentes méthodes de résolution d'un problème d'optimisation. Ainsi qu'une brève introduction sur la simulation des chaînes logistiques. Ensuite, nous présentons le couplage de ces deux approches.

Dans le troisième chapitre, nous présentons les aspects importants relatifs aux algorithmes génétiques et à l'optimisation multicritères, ainsi qu'une description plus détaillée de l'algorithme génétique multicritères implémenté.

Enfin dans Le quatrième chapitre nous présentons une vue sur l'architecture et l'objectif de la méthode de résolution. Puis nous allons présenter l'implémentation ainsi que les résultats obtenus par l'approche utilisée, ou on a pris l'algorithme génétique et particulièrement l'algorithme de NSGA II avec quelques exemples. Ce mémoire se termine par une conclusion.

CHAPITRE 1
LES CHAINES LOGISTIQUES

Introduction

La logistique occupe une place primordiale dans le fonctionnement de l'entreprise. Le cycle de cette fonction commence des fournisseurs des matières premières jusqu'à la livraison des produits final aux clients finaux.

Dans ce chapitre nous allons d'abord définir les différents concepts de base des chaînes logistiques : leurs définitions, leurs structures, leurs fluxes, leurs fonctions. Ensuite nous intéressons à la gestion de la chaîne logistique et à la prise de décision aux différents niveaux de la chaîne, ainsi que la définition des modèles et des outils de mesure de la performance logistique. Et en dernière section on abordera les problèmes d'ordonnancement en ateliers, d'une manière générale, nous essayons de donner les notions nécessaires pour comprendre notre problématique.

1 Les chaînes logistiques

1.1 Définition

Dans cette partie, nous définissons le concept de chaîne logistique ou Supply Chain. Plusieurs définitions existent dans la littérature et elles s'accordent toutes sur la finalité d'une chaîne logistique. Cependant, Thierry et Bel distinguent deux approches : celles qui définissent la chaîne logistique d'un produit et celles qui définissent la chaîne logistique d'une entreprise [1].

1.1.1 Chaîne logistique d'un produit

La chaîne logistique d'un produit fini se définit comme l'ensemble des entreprises qui interviennent dans les processus de fabrication, de distribution et de vente du produit, du premier des fournisseurs au client ultime [1].

1.1.2 Chaîne logistique d'une entreprise

La chaîne logistique peut être considérée comme le réseau d'entreprises qui participent, en amont et en aval, aux différents processus et activités qui créent de la valeur sous forme de produits et de services apportés au consommateur final. En d'autres termes, une chaîne logistique est composée de plusieurs entreprises, en amont (fourniture de matières et composants) et en aval (distribution), et du client final [1].

1.2 Structure de la chaîne logistique

La structure d'une chaîne logistique dépend évidemment de sa nature et des objectifs

souhaités lors de sa conception [2]. Croom et al identifient trois types de structures : dyadique, chaîne, réseau. Huang et al décomposent plus précisément ces structures en : série, divergente, dyadique, convergente et réseau. Dans cette section, nous présentons les différents types de structures d'une chaîne logistique [1].

1.2.1 Dyadique

Dans ce niveau, nous considérons deux entreprises, typiquement : un donneur d'ordre et un sous-traitant (Figure 1.1), ou un fournisseur et un détaillant. Cette liaison dyadique constitue la liaison de base dans la chaîne logistique [1].

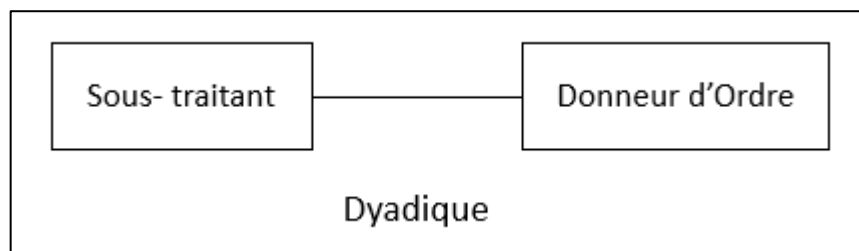


Figure 1.1 Chaîne dyadique [1]

1.2.2 Série

Dans une structure en série (Figure 1.2), nous considérons une succession de structures dyadiques constituant ensemble une chaîne logistique [1]. Ce type de structure chaîne consiste en l'existence d'un pôle ou entité alimentant une seule autre entité [2].

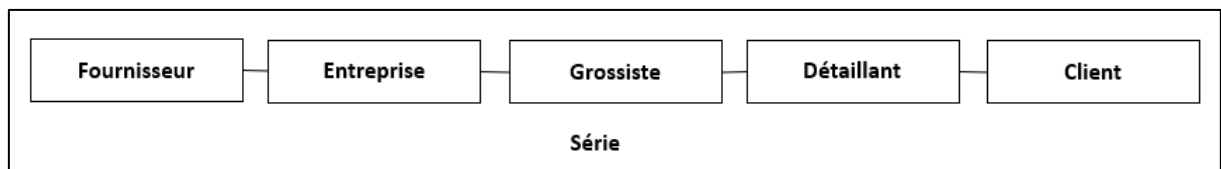


Figure 1.2 Chaîne série [1]

1.2.3 Convergente

Une structure convergente (Figure 1.3), consiste en l'existence de plusieurs fournisseurs alimentent un client ou une entreprise [2].

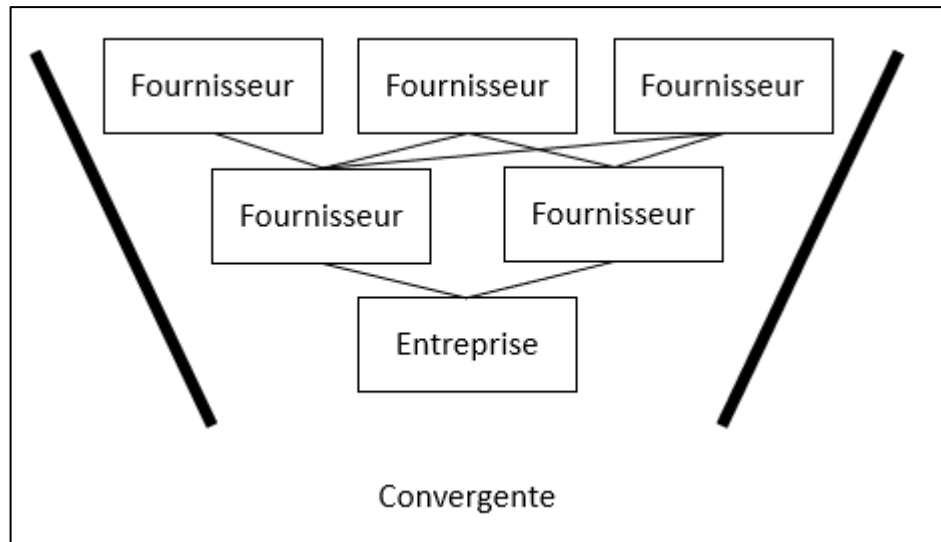


Figure 1.3 Chaîne convergente [1]

1.2.4 Divergente

Une structure divergente (Figure 1.4), est une chaîne qui alimente plusieurs clients, plusieurs fournisseurs ou réseaux [2].

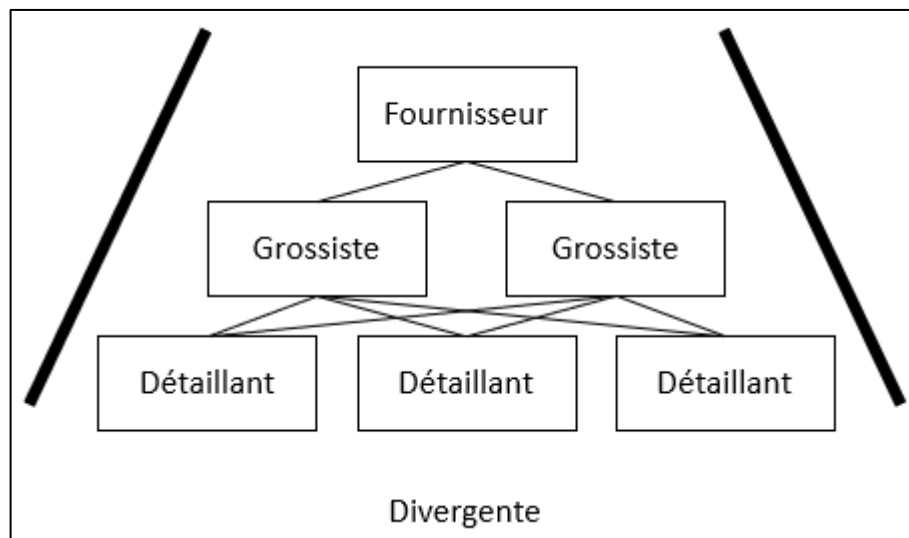


Figure 1.4 Chaîne divergente [1]

1.2.5 Réseau

Ce type de structure est la composition entre la structure convergente et divergente. (Figure 1.5) [2].

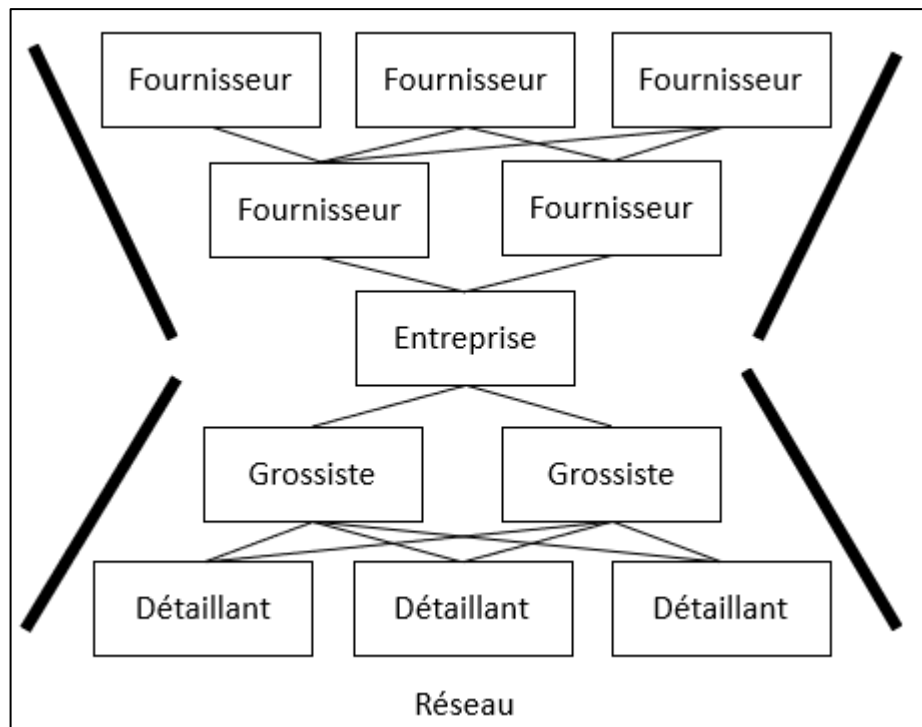


Figure 1.5 Chaîne réseau [1]

1.3 Les flux d'une chaîne logistique

Pour assurer le bon fonctionnement de l'entreprise il est essentiel de gérer les flux logistiques car ils sont directement liés à la performance organisationnelle d'une entreprise. On distingue trois types de flux : les flux d'information, les flux physiques et les flux financier. (Figure 1.6) [18][3].

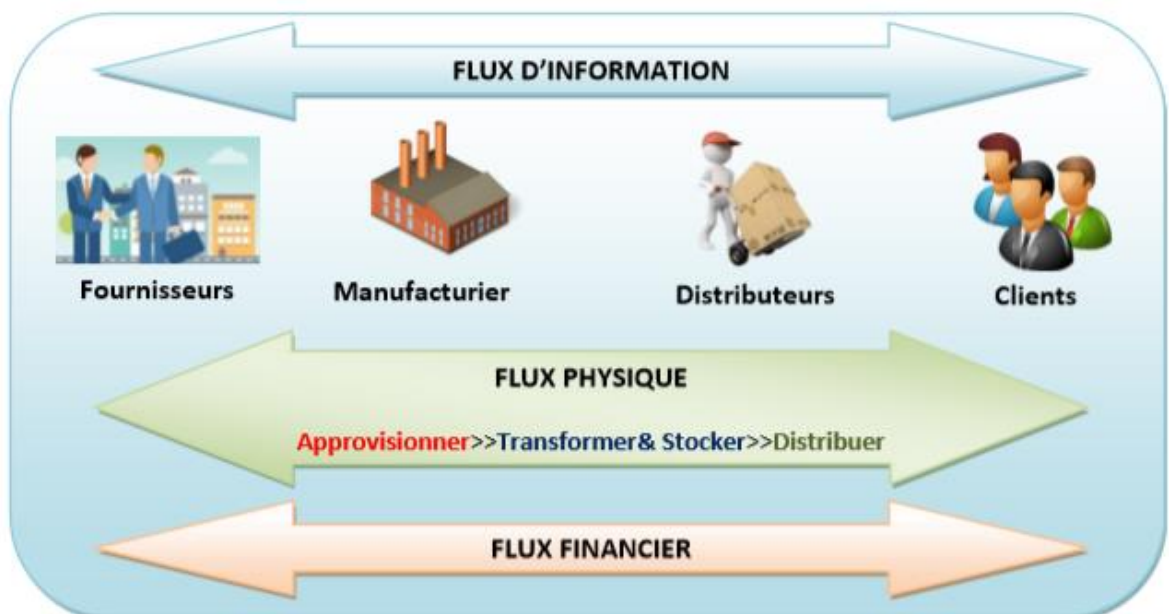


Figure 1.6 Les Flux de la chaîne logistique [8]

1.3.1 Le flux d'information

Le flux d'information représente les données qui circulent entre les différents acteurs de la chaîne logistique [2]. Par exemple Lorsque le client passe une commande, il déclenche un flux d'information. Ce flux est composé d'un flux de donnée et de décision qui sont essentiels pour un bon fonctionnement d'une chaîne logistique [4].

1.3.2 Le flux physique

Appelés également flux produit, le flux physique est constitué par le déplacement et le stockage des matières ou des marchandises qui circule au niveau de la chaîne logistique du fournisseur vers le client afin de fournir de la valeur ajoutée au client final. Ces matières peuvent être des composantes, des produits semi-finis, des produits finis ou des pièces de rechange [2][4].

Ces flux constituent le cœur d'une chaîne logistique, sans lesquels les autres flux n'existeraient pas. Ils peuvent être regroupés en trois étapes : produire, stocker et transporter [4].

1.3.3 Le flux financier

Le flux financier concerne toute la gestion budgétaire des entreprises : ventes des produits, achats de composants ou de matières premières, mais aussi des outils de production, de divers équipements, de la location d'entrepôts, du salaire des employés, etc. Le flux financier est généralement géré de façons centralisées dans l'entreprise dans le service financier ou comptabilité, en liaison toutefois avec la fonction production par les services achats et le service commercial [4].

1.4 Les fonctions de la chaîne logistique

Ganeshan et Hrisson définissent une chaîne logistique comme étant : « le réseau des moyens de production et de distribution qui assurent les tâches d'approvisionnement en matières premières, la transformation de ces matières premières en produits semi finis et en produits finis, et la distribution de ces produits finis aux clients ». De la définition ci-dessus on distingue, que les fonctions d'une chaîne logistique vont de l'approvisionnement, la production, le stockage et la vente [2].

1.4.1 L'approvisionnement

L'activité d'approvisionnement, permet à l'entreprise d'amener dans les unités les produits de base, composants et sous-ensembles nécessaires à la production [7].

1.4.2 La production

La fonction de production est au cœur de la chaîne logistique, Elle consiste à apporter aux unités de production les matériaux et composants nécessaires à la production. Le but est de tendre vers l'optimisation en améliorant les performances à chaque étape de production [7].

1.4.3 Le stockage

Le stockage inclut toutes les quantités stockées tout au long du processus en commençant par le stock des matières premières, le stock des composants, le stock des en-cours et finalement le stock des produits finis. Les stocks sont donc partagés entre les différents acteurs : les fournisseurs, les producteurs et les distributeurs. Ici aussi se pose la question de l'équilibre à trouver entre une meilleure réactivité et la réduction des coûts. La gestion des stocks est l'une des clés de la réussite et l'optimisation de toute une chaîne logistique [4].

1.4.4 Distribution

Processus qui intervient tout le long de la chaîne, en commençant par le transport des matières premières, a celui des composants entre les différentes usines, aux centres de stockage pour terminer dans des centres de distribution et finir par la livraison aux clients [2]. D'une manière générale, elle consiste à apporter au consommateur final les produits dont il a besoin [7].

1.4.5 La vente

Une vente est l'opération par laquelle un bien ou un droit détenu par un vendeur est cédé à un acheteur en échange d'une contrepartie, généralement la remise d'une somme d'argent [4].

2. La gestion de la chaîne logistique (SCM)

L'adoption de la démarche supply chain management ou la gestion de la chaîne logistique apparaît comme un outil de performance pour l'entreprise, puisque son ambition affichée est de répondre au triple objectif d'amélioration des niveaux de services, de réduction des coûts et de création de valeur, en gérant les relations avec les fournisseurs et les clients [8].

2.1 Les niveaux de décisions dans une chaîne logistique :

Les problématiques traitées dans la gestion des chaînes logistiques concernent généralement trois niveaux de décisions : le niveau stratégique, le niveau tactique et le niveau opérationnel [1].

Chacun de ces niveaux est lié à un aspect temporel différent. En effet, le niveau stratégique

concerne les décisions qui sont prises sur le long terme. Le niveau tactique concerne les décisions qui sont prises pour le moyen terme. Le niveau opérationnel concerne les décisions qui sont prises sur le court terme [1].

2.1.1 Le niveau stratégique

Le niveau stratégique appelé aussi Strategic Management regroupe toutes les décisions qui portent sur le long terme pouvant aller de six mois à plusieurs années. Dans ce niveau, il s'agit de déterminer les orientations stratégiques que va adopter la chaîne logistique sur plusieurs années [1].

2.1.2 Le niveau tactique

Le niveau tactique regroupe toutes les décisions pouvant aller d'une ou deux semaines à quelques mois [1]. Dans ce niveau, il s'agit de prendre les décisions à moyen terme qui devront être mis en application pour déployer la stratégie décidée par l'entreprise [2].

2.1.3 Le niveau opérationnel

Le niveau opérationnel concerne les décisions à court terme de l'entreprise. Ce sont des décisions prises dans le quotidien afin d'assurer la bonne mise en place des décisions tactiques. Ce type de décisions est généralement pris par les chefs d'équipes dans le but de gérer par exemple : les opérations quotidiennes d'une usine ou d'un centre de distribution [1].

2.2 Modélisation de la chaîne logistique

La modélisation de la chaîne logistique est une construction abstraite qui permet de comprendre le fonctionnement de la chaîne et une meilleure gestion de ce système. Il existe différents modèles de chaînes logistiques. Le choix du modèle dépend directement du type de problème et de structure que l'on veut étudier. [2]

2.2.1 Les modèles conceptuels

Les modèles conceptuels sont de loin les plus simples. Il s'agit en fait d'une description basique d'un système économique comme la chaîne logistique qui peut s'exprimer sous formes de diagrammes ou d'explications verbales. Le format utilisé dépend en grande partie de l'expérience du modélisateur [8].

2.2.2 Les modèles mathématiques

Les modèles mathématiques sont très utilisés pour la conception des chaînes logistiques et pour l'optimisation des coûts. Ils consistent à modéliser un système réel par un ensemble

d'équations exprimant les contraintes et les objectifs. Contrairement aux modèles conceptuels qui aident seulement à la compréhension du système, les modèles mathématiques résolvent les problèmes d'optimisation. Une autre différence avec les modèles conceptuels est que l'utilisation des modèles mathématiques requiert des compétences spéciales dans les mathématiques [8].

2.2.3 Les modèles par simulation

Les modèles par simulations sont très pratiques dans le cas de systèmes où il est difficile de représenter toutes les hypothèses par des équations, et de ce fait, on ne peut pas utiliser les modèles mathématiques. Ces modèles essaient d'imiter le comportement des composants d'un modèle et donc de pouvoir faire des prévisions et des évaluations de performances [8].

2.3 Planification de la chaîne logistique

La planification des chaînes logistiques (SCP supply chain planning) concerne la coordination et l'intégration des activités métier réalisées par une entreprise, depuis l'approvisionnement en matières premières jusqu'à la livraison du produit final aux clients. Les buts finaux de la planification de la chaîne logistique visent :

- L'utilisation optimale des ressources.
- La minimisation des stocks à tous les niveaux.
- La satisfaction de la demande des clients. [2]

2.4 Les mesures de la performance de la chaîne logistique

Il y a un adage qui dit : « You can't manage what you don't measure » dont la traduction en français pourrait être « vous ne pouvez pas gérer ce que vous ne mesurez pas ». La question qui est posée est donc : comment mesurer l'efficacité d'une chaîne logistique ?

La gestion de la chaîne logistique cherche à améliorer le système global de production. Pour atteindre cet objectif, il faut connaître sa performance effective et déterminer une cible ou un objectif à atteindre. La première étape de suivi les performances consiste donc à « mesurer la performance ». Ensuite, il faut prendre des décisions de réingénierie et agir sur le système à travers des variables de décision afin de tendre vers la cible choisie [9].

Nous devons donc définir un ensemble d'indicateurs pour mesurer cette performance. Un indicateur de performance est défini comme « une donnée quantifiée qui mesure l'efficacité de tout ou partie d'un processus ou d'un système, par rapport à une norme, un plan ou un objectif qui aura été déterminé et accepté, dans le cadre d'une stratégie d'ensemble »

Chopra et Meindil[26] identifient six indicateurs de performance :

- **Les infrastructures** : ce sont les sites où l'on produit, assemble et stocke les composants. La performance de la chaîne logistique est influée par leur rôle, localisation, flexibilité et capacité.
- **Les stocks** : la réactivité d'une chaîne logistique peut être influée par tout changement de politique de stockage.
- **Les transports** : les transports, représentant une lourde charge pour la chaîne logistique, influent sur la performance, la réactivité et l'efficacité de celle-ci.
- **L'information** : est l'indicateur le plus important car elle influe directement sur l'ensemble des autres indicateurs et sur l'efficacité de la chaîne.
- **Le Sourcing** : définit le travail que chacun doit effectuer à travers la chaîne. C'est aussi la répartition des activités de l'entreprise mère.
- **Les prix** : donner une valeur aux biens et services (produits) en fonction de plusieurs paramètres car leurs répercussions influencent le comportement (décisions) des clients et les performances de l'entreprise [2].

3. problème d'ordonnancement

3.1 définitions et notions fondamentales

Un problème d'ordonnancement peut être considéré comme un sous problème de planification dans lequel il s'agit de décider de l'exécution opérationnelle des tâches (jobs) planifiées, et ainsi d'établir leur planning d'exécution et leur allouer des ressources visant à satisfaire un ou plusieurs objectifs sous une ou plusieurs contraintes. En se basant sur les concepts de tâche, ressource, contrainte et objectif [32].

3.2 Les éléments du problème d'ordonnancement

3.2.1 Les ressource

Une ressource est un moyen technique ou humain dont la disponibilité limitée ou non est connu à priori [32]. Plusieurs classifications des ressources peuvent être distinguées :

- Ressources renouvelables vs Ressources consommables :

Une ressource est dite renouvelable, si après avoir été utilisée par une ou plusieurs tâches, elle est à nouveau disponible en même quantité comme : les hommes, les machines...ect

Par contre, une ressource est consommable, lorsqu'elle devient non disponible après avoir été utilisée ; comme la matières premières ...ect

- Ressources disjonctives vs Ressources cumulatives

Une ressource est dite disjonctive (ou non partageable), si elle ne peut être affecter qu'à une seule tâche à la fois. Ce type concerne surtout les ressources renouvelables.

Par contre, une ressource cumulative (partageable) peut être utilisée par plusieurs tâches simultanément.

Dans notre problème, les ressources qui sont des machines, sont renouvelables et disjonctives.[33]

3.2.2 Les tâches

Une tâche est un travail élémentaire dont la réalisation nécessite un certain nombre d'unités de temps (sa durée) et d'unités de chaque ressource, on distingue deux types de tâches :

- Les tâches morcelables (préemptives) qui peuvent être exécutées en plusieurs fois, facilitant ainsi la résolution de certains problèmes
- Les tâches non morcelables (indivisibles) qui doivent être exécutées en une seule fois et ne sont interrompues qu'une fois terminées [32]

3.2.3 Les objectifs

Les objectifs des entreprises sont diversifiés et le processus d'ordonnancement est devenu de plus en plus multicritère. Les critères que doivent satisfaire un ordonnancement sont variés.

D'une manière générale, on distingue plusieurs classes d'objectifs concernant un ordonnancement :

- Les objectifs liés au temps : on trouve par exemple la minimisation du temps total d'exécution (C_{max}), du temps moyen d'achèvement, des durées totales de réglage ou des retards par rapports aux dates de livraisons (L_{max} ou T_{max})
- Les objectifs liés aux ressources : maximiser la charge d'une ressource ou minimiser le nombre de ressources nécessaire pour réaliser un ensemble des tâches.
- Les objectifs liés au coût : minimiser les coûts de lancement, de production, de stockage, de transport, etc[32].

3.2.4 Les contraintes

Une contrainte exprime des restrictions sur les valeurs que peuvent prendre simultanément les variables représentant les relations reliant les tâches et les ressources. On distingue deux types de contraintes, les contraintes temporelles et les contraintes de ressources.

- Les contraintes temporelles : Les contraintes temporelles concernent les délais de fabrication imposés. Ces contraintes peuvent être :
 - Des contraintes de dates butoirs, certaines tâches doivent être achevées avant une date préalablement fixée.
 - Des contraintes de précédence, une tâche i doit précéder la tâche j .
 - Des contraintes de dates au plus tôt, liées à l'indisponibilité de certains facteurs nécessaires pour commencer l'exécution des tâches.
- Les contraintes de ressources : Ces contraintes concernent la limitation de la quantité des ressources de chaque type. Dans ce cadre, deux types de contraintes de ressources sont distingués :
 - Les contraintes disjonctives : induisant une contrainte de réalisation des tâches sur des intervalles temporels disjoints pour une même ressource.
 - Les contraintes cumulatives : impliquant la limitation du nombre de tâches à réaliser en parallèle [32].

3.3 Schémas de classification des problèmes d'ordonnancement

Nous suivons les schémas de classification proposée par (Graham et al, 1979). Classification à trois champs α , β , γ :

- α : environnement machine.
 - α : Désigne le nombre de machine. Si α est entier positive, le nombre de machine supposé constant .si α est absent alors ce nombre est supposé arbitraire.
- β : les caractéristiques des tâches.
 - $\beta_1 = \text{pmtn}$ si la préemption des tâches est autorisée, sinon β_1 est absent.
 - S'il y a des contraintes de précédence entre les tâches $\beta_2 = \text{prec}$, sinon β_2 est vide.
 - $\beta_3 = r_j$ si les dates de début au plus tôt r_j (ou dates de disponibilité) des tâches ne sont pas forcément identiques, sinon ($j, r_j = 0$) β_3 est absent.
- γ : le (ou les) critère(s) à optimiser.[32]

3.4 Les types des problèmes d'ordonnancement d'atelier

3.4.1 Une seule machine

Toute tâche $J_j, j \in \{1, \dots, n\}$ de durée P_j , (processing time) s'exécute sur une machine qui ne peut traiter plus qu'une tâche à la fois. Le champ $\alpha = 1$. [32]

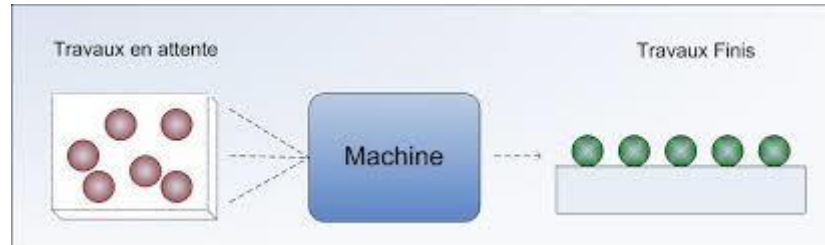


Figure 1.7 Une seule machine [32]

3.4.2 Machines identiques en parallèle

Toute tâche $J_j, j \in \{1, \dots, n\}$ peut être exécutée indifféremment sur une des m machines mises en parallèle

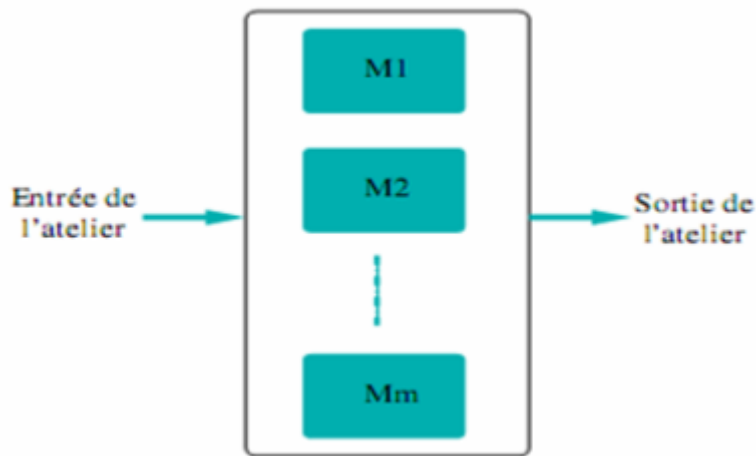


Figure 1.8 Machines identiques en parallèle [32].

- Si toutes les machines ont la même vitesse d'exécution des tâches, les machines sont appelées « identiques » et le problème noté (P).
- Si les machines sont différentes par leur vitesse d'exécution et la vitesse de chaque machine est constante et ne dépend pas de l'ensemble des tâches, elles sont dites « uniformes ». (Q).
- Si les vitesses d'exécution des machines dépendent des tâches et sont différentes alors elles sont dites « quelconques » ou différentes (R) [32].

3.4.3 Atelier mono-gamme (Flow shop)

Chaque ordre de fabrication doit être traité par chacune des m machines en série et ce, dans le même ordre. Tous les produits ont donc le même routage. En temps normal, toutes les files d'attente des machines (le cas échéant) opèrent selon la règle : FIFO First In First Out.[2]

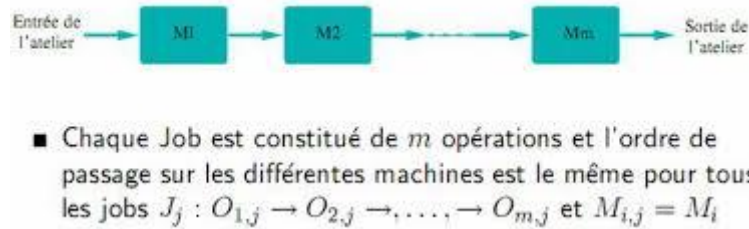


Figure 1.9 Flow shop[32]

3.4.4 Atelier mono-gamme flexible (Flow shop hybride)

C'est une généralisation de l'atelier mono-gamme et des machines parallèles en ce sens qu'au lieu de m machines en série, il y a s étages ou stations en série contenant des machines en parallèle. Chaque ordre de fabrication traverse les étages dans le même ordre, en utilisant une seule machine par étage. Les files d'attente fonctionnent généralement selon la règle FIFO.[2]

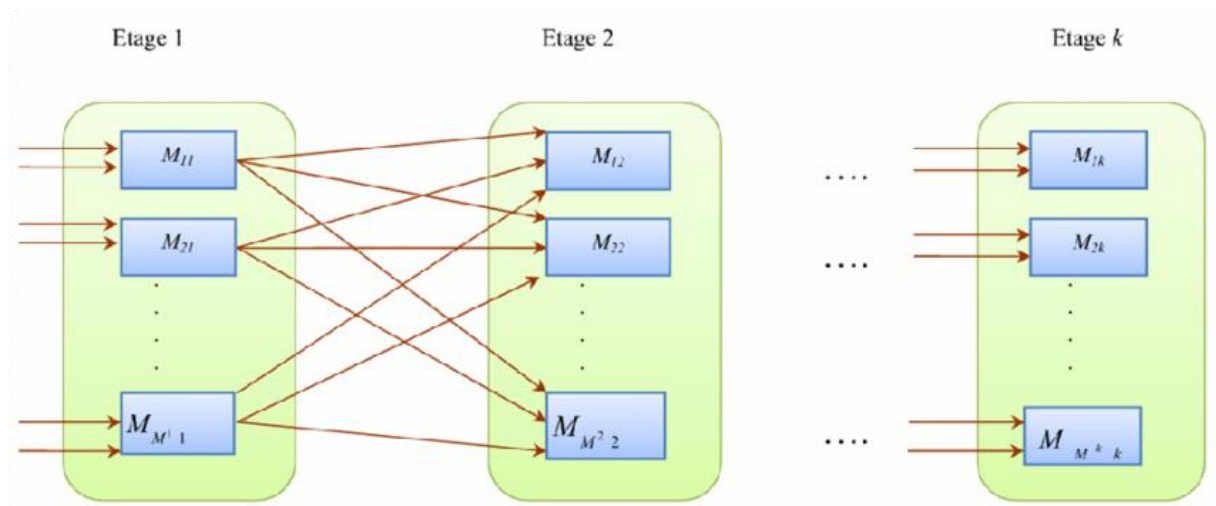


Figure 1.10 Flow shop hybride[4]

3.4.5 Atelier multi-gammes (job shop)

Chaque produit possède ici son propre routage, bien que certains puissent avoir à visiter une même machine plus d'une fois. C'est le phénomène de recirculation[2].

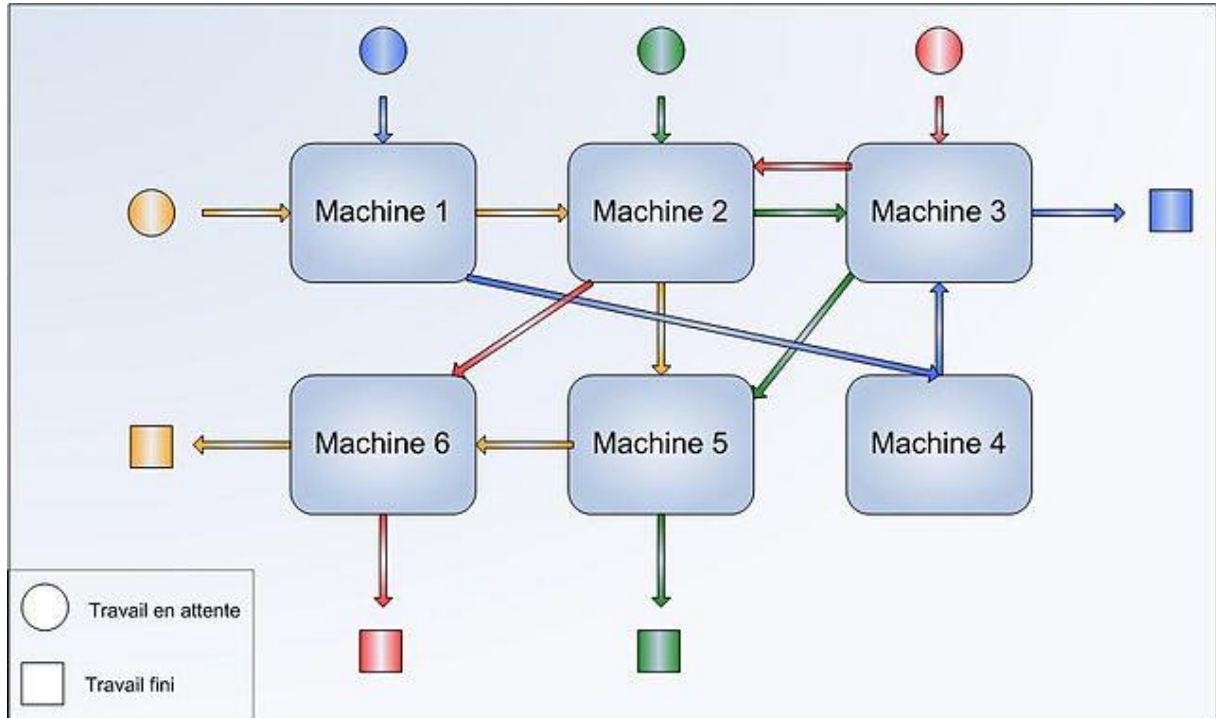


Figure 1.11 job shop [32]

3.4.6 Atelier ouvert (open shop)

Chaque ordre de fabrication doit être traité par les m machines. Cependant, il n'y a aucune restriction quant au routage emprunté par les produits, même que certains peuvent avoir des temps de traitement nuls à des machines en particulier.[2]

Conclusion

Dans ce chapitre, nous avons abordé une brève introduction sur les différents concepts de base des chaînes logistiques et de sa gestion. En faisant un tour sur les différents composants qui permettent par leur combinaison d'aboutir des meilleures mesures de performances, et une meilleure gestion de chaîne logistique et faire valoir l'importance du problème d'ordonnancement.

Dans le chapitre suivant, nous allons présenter l'approche d'optimisation basée sur la simulation.

CHAPITRE 2

OPTIMISATION BASEE SUR

LA SIMULATION

Introduction

Le présent chapitre présente une description de l'approche d'optimisation à base de la simulation, pour traiter les problèmes de conception des chaînes logistiques. Nous commençons ce chapitre par une brève introduction de l'optimisation des chaînes logistiques et ces principaux concepts, avec ces outils. Par la suite nous présenterons une vue sur la simulation des chaînes logistiques, Nous terminons ce chapitre par le couplage des deux approches la simulation et l'optimisation.

1 Optimisation des chaînes logistiques

L'optimisation de la chaîne logistique consiste à mettre en place les solutions qui s'imposent pour améliorer l'organisation de la gestion de la chaîne logistique d'une part, et de réduire les coûts relatifs au processus logistique [22]. L'optimisation de la chaîne logistique est donc un objectif pour toute société souhaitant atteindre l'excellence opérationnelle. Les entreprises doivent utiliser de manière optimale les moyens industriels & logistiques, améliorer les flux au sein de la chaîne logistique pour réduire les stocks tout en améliorant leur réactivité et réduire les délais de livraison [23].

1.1 Principaux concepts en optimisation

Tout d'abord nous définissons les notions communes à n'importe quelle méthode d'optimisation multicritère :

- **Fonction objectif**

Une fonction objectif est une fonction qui modélise le but à atteindre Dans le problème d'optimisation sur l'ensemble des critères. Il s'agit de la fonction qui doit être optimisée. Elle est notée $F(x)$ de manière générale $F(x)$ est un vecteur :

$$F(x) = [f_1(x), f_2(x), \dots, f_n(x)].$$

Elle est aussi appelée : critère d'optimisation, fonction coût, fonction d'adaptation, ou encore performance [3].

- **Variables de décision**

Les variables de décision sont des quantités numériques pour les quelles des valeurs sont à choisir. Cet ensemble de n variables est appelé vecteur de décision, il est noté : $\vec{x} = [x_1, x_2, \dots, x_n]$ avec : n le nombre de variables. Les différentes valeurs possibles prises par les variables de décision x_i constituent l'ensemble des solutions potentielles [4].

- **Contraintes**

Une contrainte du problème est une condition que doivent respecter les vecteurs de décision du problème. Une contrainte est notée : $g_i(\vec{x})$ avec : $i=1, \dots, q$
 q : le nombre des contraintes [4].

- **Espace de recherche**

Représentant l'ensemble des valeurs pouvant être prises par les variables [16].

- **Espace réalisable**

Représentant l'ensemble des valeurs des variables satisfaisant les contraintes [16].

- **Minimum local**

Un point x est un minimum local du problème s'il existe un intervalle $J \subset I$:

$$\forall x' \in J, f(x) \leq f(x') \quad I: \text{espace de recherche.}$$

- **Maximum global**

Un point x est un minimum global du problème si et seulement si :

$$\forall x' \in I, f(x) \leq f(x') \quad I: \text{espace de recherche.}$$

1.2 Les caractéristiques d'un problème d'optimisation

- Le domaine des variables de décision : soit Continu et on parle alors de problème continu, soit discret et on parle donc de problème combinatoire.
- La nature de la fonction objective à optimiser : soit linéaire et on parle alors de problème linéaire, soit non linéaire et on parle donc de problème non linéaire.
- Le nombre de fonctions objectifs à optimiser : soit une fonction scalaire et on parle alors de problème mono-objectif, soit une fonction vectorielle et on parle donc de problème multi objectif.
- La présence ou non des contraintes : on parle de problème sans contrainte ou avec contrainte.
- Sa taille : problème de petite ou de grande taille [17].

1.3 Face à un problème d'optimisation

- Elaborer un modèle (mathématiques) : l'expression de l'objectif à optimiser et les contraintes à respecter.
- Développer un algorithme de résolution.
- Evaluer la qualité des solutions produites [17].

1.4 Les outils pour l'optimisation de la chaîne logistique

Un problème d'optimisation consiste à trouver une solution d'un problème donné en minimisant ou bien en maximisant une fonction objectif donnée. En outre, une technique d'optimisation est la méthode qu'on l'utilise pour trouver une solution pour un problème de minimisation ou de maximisation. Au fil des années, de nombreuses techniques (méthodes) de résolution de problèmes d'optimisation ont été proposées [27, 28, 29, 30, 31]. En effet, nous pouvons regrouper ces méthodes en deux classes : la classe de méthodes approchées et la classe de méthodes exactes. Dans cette partie nous donnons une vue générale sur ces deux classes de méthodes. La figure 2.1 résume les différentes méthodes des deux classes :

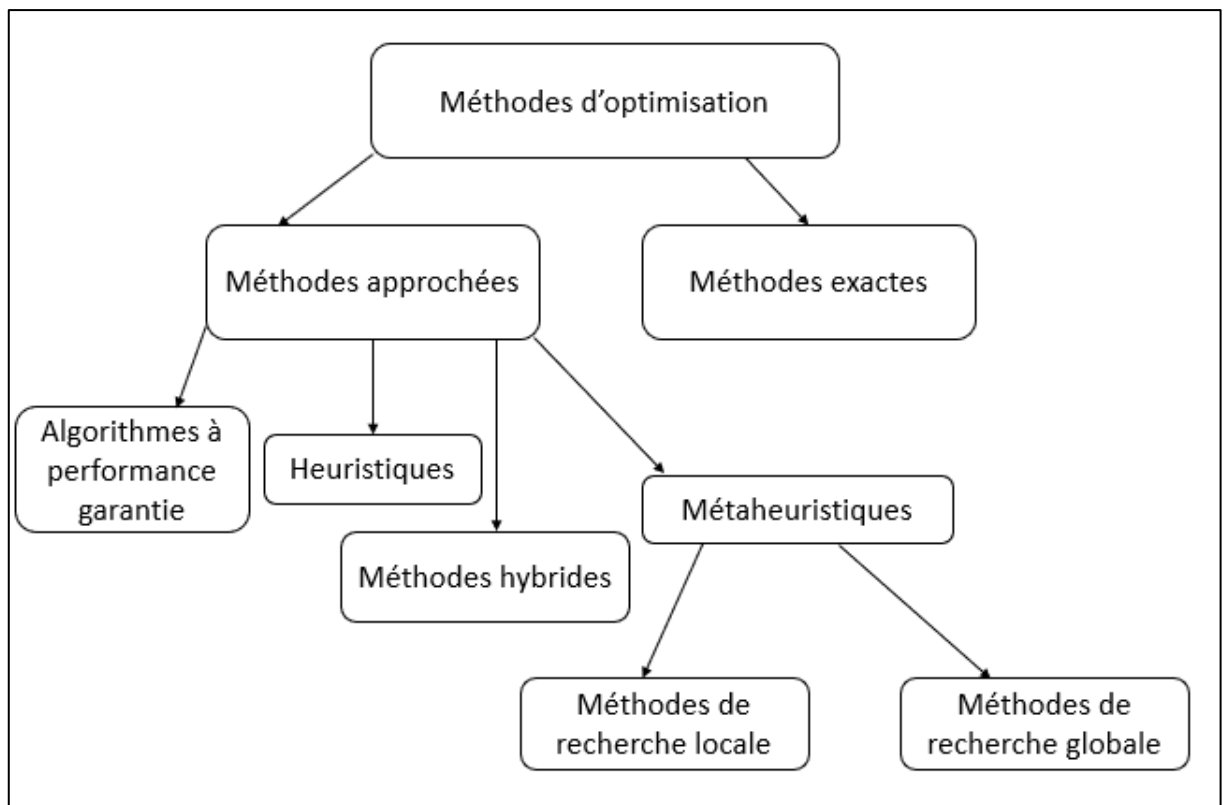


Figure 2.1 Classification des méthodes d'optimisation [25]

1.4.1 Méthodes exactes

Les méthodes exactes sont des méthodes qui cherchent la solution optimale d'un problème en explorant exhaustivement toutes les solutions possibles dans l'espace de recherche. Cependant, l'inconvénient majeur de ces méthodes est le temps d'exécution, car toutes les solutions sont évaluées une à une et le temps d'exécution augmente exponentiellement avec la taille du problème considéré. Donc, ces techniques restent inappropriées aux problèmes combinatoires de grandes tailles [13]. Comme exemple de ces méthodes, nous pouvons citer : l'algorithme A* ? les algorithmes de type : Branch and Bound [25].

1.4.2 Méthodes approchées

Les problèmes d'optimisation du monde industriel sont généralement de grandes tailles et disposent de nombreuses contraintes, donc les méthodes exactes ne sont pas utilisables. Dans ce cas, on doit chercher une bonne solution en un temps raisonnable au lieu d'attendre l'obtention d'une solution optimale après des années de calcul. Contrairement aux méthodes exactes, les méthodes approchées ne garantissent pas l'optimalité de la solution mais, elles permettent de trouver des solutions de très bonne qualité en un temps d'exécution raisonnable, c-à-dire, elles cherchent un bon compromis entre la qualité de la solution et le temps de calcul. Dans la littérature, de nombreuses méthodes approchées ont été proposées. Nous pouvons diviser ces méthodes en 4 catégories : des heuristiques, des méta-heuristiques, des algorithmes à performance garantie et des algorithmes hybrides. Par la suite nous présentons chaque catégorie [25].

1.4.3 Heuristique

Dans la littérature, il existe plusieurs définitions d'une heuristique, ici nous présentons celle de Feigenbaum et Feldman [14] : « une heuristique (règle heuristique, méthode heuristique) est une règle d'estimation, une stratégie, une astuce, une simplification, ou tout autre type de dispositif qui limite considérablement la recherche de solutions dans des espaces problématiques importants. Les heuristiques ne garantissent pas des solutions optimales. En fait, elles ne garantissent pas une solution du tout. Tout ce qui peut être dit d'une heuristique utile, c'est qu'elle propose des solutions qui sont assez bonnes la plupart du temps ».

En outre, dans le domaine de l'optimisation combinatoire on peut dire qu'une heuristique est une méthode approchée développée pour résoudre un problème particulier et elle nécessite des connaissances approfondies sur le problème traité. Le but d'une heuristique est de trouver de bonnes solutions (pas forcément optimales) pour un problème donné dans un temps d'exécution très réduit. Les méthodes constructives représentent un bon exemple des heuristiques [25].

1.4.4 Métaheuristiques

Selon Osman et Laporte : « une métaheuristique est un processus itératif qui subordonne et guide une heuristique, en combinant intelligemment plusieurs concepts pour explorer et exploiter tout l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions optimales, ou presque-optimales ».

Généralement, une heuristique est un algorithme développé pour résoudre un problème spécifique. Cependant, une métaheuristique est une stratégie générale pour qu'on peut appliquer pour résoudre un grand nombre de problèmes d'optimisation. Dans la littérature, de nombreuses métaheuristicues ont été proposées qui peuvent être subdivisées en deux grandes familles : les méthodes de recherche locale (à base de solution unique) et les méthodes de recherche globale (à base d'une population de solutions). Ces deux familles sont décrites ci-dessous [25][20].

1.4.4.1 Les méthodes de recherche locale (Local Search Methods)

Les méthodes de recherche locale sont des algorithmes à base de solution unique qui se basent généralement sur la notion de voisinage. Elles commencent à partir d'une solution initiale, ensuite à chaque itération elles essaient d'améliorer la solution courante en exploitant son voisinage. De nombreuses méthodes de recherche locale ont été proposées dans la littérature, ici nous citons les plus connus : la méthode de descente, la méthode du recuit simulé, etc.

1.4.4.2 Les méthodes de recherche globale (Global Search Methods)

Les méthodes de recherche globale, appelées aussi les algorithmes évolutionnaires [21], sont des méthodes à base de population de solutions et qui sont en général inspirées de la nature. Ces méthodes utilisent un ensemble de solutions appelé population. Elles commencent avec une population initiale et, à chaque itération, elles essaient de construire une nouvelle population meilleure basée sur la précédente afin de converger vers de bonne(s) solution(s). Comme exemples de ces méthodes nous pouvons citer : les algorithmes génétiques, l'algorithme de colonies de fourmis, etc.

1.4.5 Algorithmes à performance garantie

Les algorithmes à performance garantie sont des méthodes approchées basées plutôt sur des principes mathématiques. En se basant généralement sur des théorèmes prouvés mathématiquement, ces algorithmes nous permettent d'assurer que, dans le pire des cas, l'écart entre la solution trouvée et la solution optimale sera au plus d'une fraction donnée p [25].

1.4.6 Approches hybrides

Les méthodes approchées sont des méthodes puissantes, mais la combinaison entre ces méthodes augmente encore leur puissance. En outre, les approches hybrides sont des méthodes qui visent à combiner deux ou plusieurs algorithmes afin d'exploiter leurs points

forts et de compenser leurs faiblesses. Dans la littérature nous trouvons généralement deux types d'hybridations:

- L'hybridation entre les méthodes approchées, par exemple : l'hybridation entre les métaheuristiques basées population et les méthodes de recherche locale.
- L'hybridation entre les méthodes approchées et les méthodes exactes, soit en utilisant une méthode approchée pour calculer une borne inférieure et/ou supérieure pour un algorithme exact ou bien en utilisant un algorithme exact pour résoudre un sous-problème ou sein d'une méthode approchée (Mat-heuristiques) [25].

2 La simulation des chaînes logistiques

Par comparaison avec les modèles analytiques, la simulation offre la possibilité de modéliser toutes les complexités et la dynamique réelles de la chaîne logistique, sans faire recours à des hypothèses simplificatrices. Elle consiste à étudier le comportement et la dynamique d'un système afin de mesurer sa performance (output) à différents intrants (inputs) en fonction du temps. Les modèles de simulation sont de natures différentes et se distinguent par leur évolution face à la composante temporelle. Les modèles de simulation à événements discrets est le plus utilisés pour la simulation de chaînes logistiques. Ainsi, pour pouvoir assurer une solution optimale, plusieurs approches d'optimisation des modèles de simulation émergent dans la littérature [15].

3 Le couplage simulation et optimisation

Carson et Maria ont défini l'optimisation de la simulation comme « le processus de trouver les meilleures valeurs des variables d'entrée parmi toutes les possibilités sans évaluer explicitement chaque possibilité » Une approche d'optimisation basée sur la simulation comprend généralement un module d'optimisation qui a pour rôle de guider la direction de recherche des solutions et un module de simulation utilisé pour l'évaluation des performances des solutions suggérées par le module d'optimisation. En tant que problème dynamique complexe, la conception de réseau de la chaîne logistique est caractérisée par un nombre élevé de variables de décision et d'alternatives. Par conséquent, en simulant toutes les possibilités combinatoires possibles est impossible. L'optimisation est donc un outil essentiel pour trouver l'ensemble de valeurs des variables de décision qui conduit à des performances optimales. Par ailleurs, malgré que la simulation soit un outil puissant qui permette de modéliser et d'analyser le comportement opérationnel des chaînes logistiques complexes, il n'est pas

capable de garantir la solution optimale. Ainsi, l'intégration des algorithmes d'optimisation aux modèles de simulation constitue un outil hybride puissant en optimisation. Une analyse des avantages et des désavantages de l'utilisation de la simulation par rapport à l'optimisation mathématique pour étudier les chaînes logistiques est présentée dans Ingalls. L'auteur conclut que la simulation est un excellent outil d'analyse lorsque la configuration optimale de la chaîne logistique est déjà déterminée par un outil d'optimisation mathématique et il recommande d'utiliser des techniques hybrides d'optimisation-simulation, il y a deux types de relations entre la simulation et l'optimisation : (1) la simulation évaluant les résultats de l'optimisation, (2) le modèle de simulation créant et transmettant des données servant à l'optimisation [15].

Conclusion

Dans ce chapitre, nous avons présenté une introduction sur les deux approches l'optimisation des chaînes logistiques incluent ces concepts de base et ces outils et l'approche de simulation, ensuite on a vu le couplage de ces deux approches.

Dans le chapitre suivant nous allons apprendre les algorithmes génétiques et l'optimisation multi-objectif.

CHAPITRE 3

ALGORITHMES GENETIQUES ET OPTIMISATION MULTICRITERES

Introduction

Le présent chapitre est dédié à la description détaillée de l'algorithme d'optimisation. En particulier, nous allons présenter Les principes de base de l'optimisation multi-objectif, ainsi que les principaux algorithmes génétiques multi-objectifs (AGMOs) et une description plus détaillée sur quelque Algorithmes génétiques utilise dans l'optimisation multi-objectif.

1. Algorithmes génétiques multicritères

Cette section rappelle les principes de base des algorithmes génétiques multicritères (MOGA) qui constituent le moteur d'optimisation de l'approche développée dans cette thèse [2].

1.1 Principes de base des algorithmes génétiques

Les AGs font partie des algorithmes méta-heuristiques, ils travaillent sur un espace de recherche sous forme de population en y opérant d'une manière stochastique. Les caractéristiques qui font l'unicité de cette méthode sont :

- Les AGs ne travaillent pas directement sur les paramètres, ces derniers sont codés avant l'opération.
- Ils opèrent sur une population de solutions au lieu d'une seule solution.
- Les AGs se basent sur des données aléatoires, donc ils sont probabilistes.

Les AGs utilisent les éléments suivant pour se définir :

- Individu/chromosome/séquence : une potentielle solution du problème abordé.
- Population : un ensemble de chromosomes de l'espace de l'exploration
- Environnement : l'espace de l'exploration et de la recherche.
- Fonction d'évaluation (fitness) : la fonction objective qui sert à évaluer la précision de la solution du problème [17].

1.2 Fonctionnement des AGs

Les algorithmes génétiques comportent des phases à travers lesquelles une solution potentielle est estimée :

- **Initialisation** : Un ensemble de N chromosomes est aléatoirement créé formant une population de solutions potentielles.
- **Evaluation** : Décodage puis évaluation de chaque individus/chromosomes.

- **Sélection** : Après l'évaluation, une nouvelle population de N individus est créée à l'aide d'une méthode de sélection adéquate.
- **Opérations génétiques** : Croisement et mutation de certains individus au sein de la nouvelle population.
- **Retour** : Tant que la condition d'arrêt n'est pas satisfaite, l'algorithme recommence à partir de la phase d'évaluation.

Un AG génère une population en utilisant une représentation chromosomique des individus puis à l'aide de la fonction objectif (fitness), il évalue chaque individu et sélectionne les mieux adaptés pour former une nouvelle population de solutions potentielles à travers les opérateurs génétiques (croisement, mutation) là où se passe la reproduction afin de créer de nouveaux individus au sein de la nouvelle population [17]. Ce processus est illustré dans l'organigramme suivant :

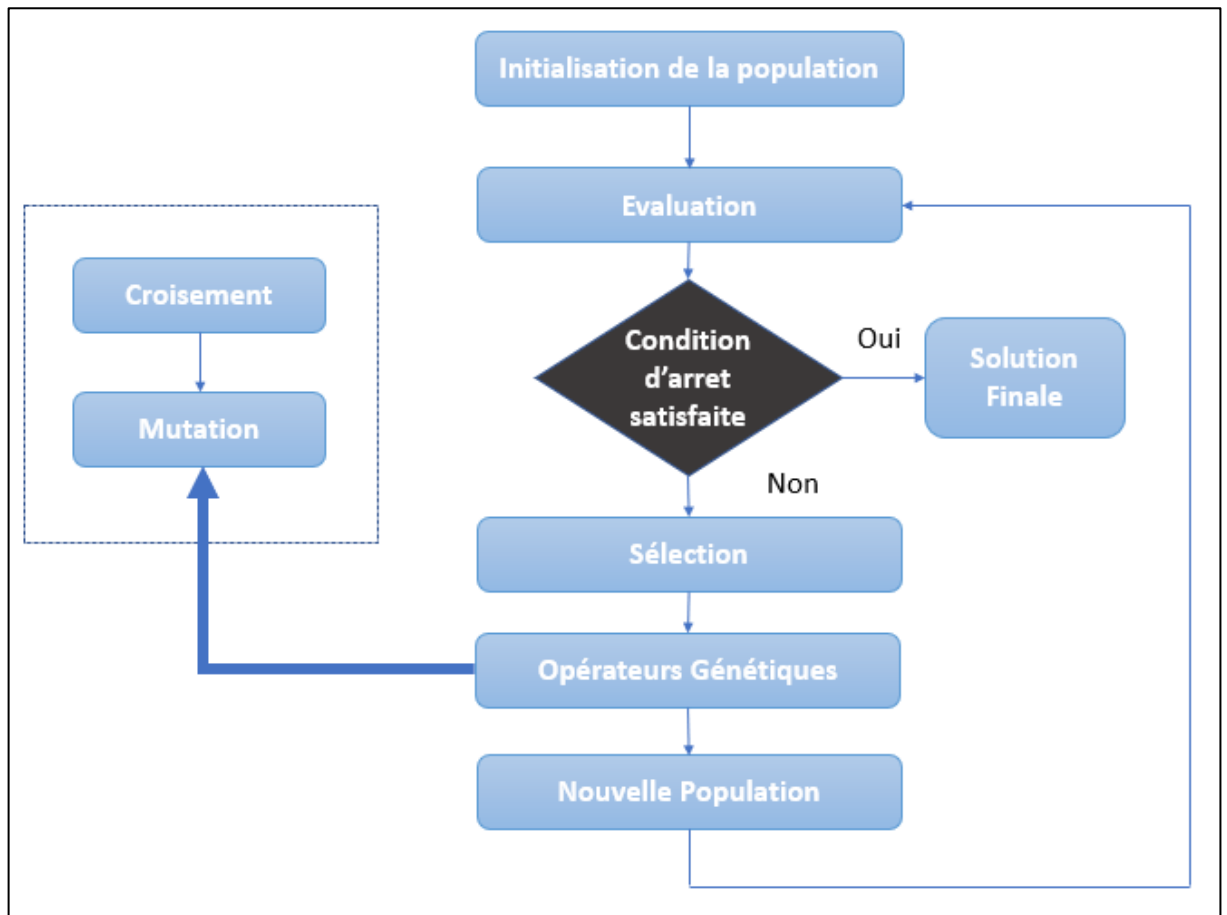


Figure 3.1 Fonctionnement de l'algorithme génétique [17].

1.3 Représentation chromosomique (Codage)

Au début, chaque individu (chromosomique) de la population est codé, représenté par un vecteur dont la longueur dépend de la précision requise, le codage donne à l'algorithme génétique une représentation génotype-phénotype. Le codage des paramètres d'un AG dépend du problème à optimiser, le codage binaire et réel sont les plus utilisés dans le domaine.

1.3.1 Codage binaire

Ce codage est caractérisé par sa simplicité car il ne comporte que deux caractères (0 et 1), chaque individu est représenté par un chromosome et chaque caractéristique de l'individu est un gène. Le chromosome est codé en une suite de bits appelée chaîne binaire, ceci est illustré à travers l'exemple suivant : Trois variables (203, 13, 45) codées sur 8 bits formant une chaîne binaire de taille 24 [17].

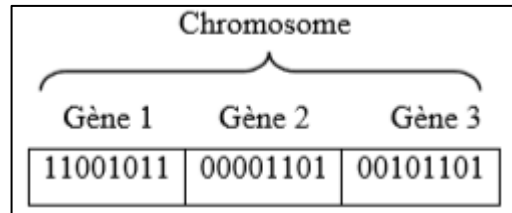


Figure 3.2 Codage binaire de trois variables [17]

1.3.2 Codage réel

L'ensemble des variables est représenté par un vecteur $\vec{x} = [x_1, x_2, \dots, x_n]$ où chaque x_i est un nombre réel.

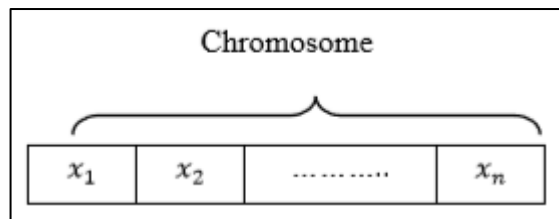


Figure 3.3 Codage réel [17]

1.4 Evaluation

La fitness (fonction d'évaluation) évaluera l'efficacité des individus et permettra la sélection des candidats aptes à la reproduction. Trois opérateurs primaires feront évoluer la population :

- La sélection.
- Le croisement.

- La mutation.

Employés dans cet ordre, ils créeront une nouvelle population d'individus [19]

1.5 Sélection

C'est dans cette étape où se décide quels sont les individus de la population qui vont être dupliqués dans la nouvelle population où ils deviendront des parents. Les individus ayant le meilleur fitness auront donc plus de chance d'être sélectionnés [17].

Il existe différentes méthodes de sélection, Parmi eux :

- La méthode de la "loterie biaisée" (roulette Wheel) [16].
- La sélection par tournoi [18].

1.5.1 Méthode de la loterie biaisée

La sélection par roulette biaisée (roulette Wheel sélection) de Goldberg est la plus utilisée des méthodes de reproduction connues. A partir de cette méthode, chaque chromosome sera copié dans une nouvelle population proportionnellement à sa fitness. La probabilité avec laquelle un individu chi de fitness $f(chi)$, sera sélectionné dans une population de taille N est déterminée généralement de la façon suivante [19] :

$$p(chi) = \frac{f(chi)}{\sum_{j=1}^N f(chi)}$$

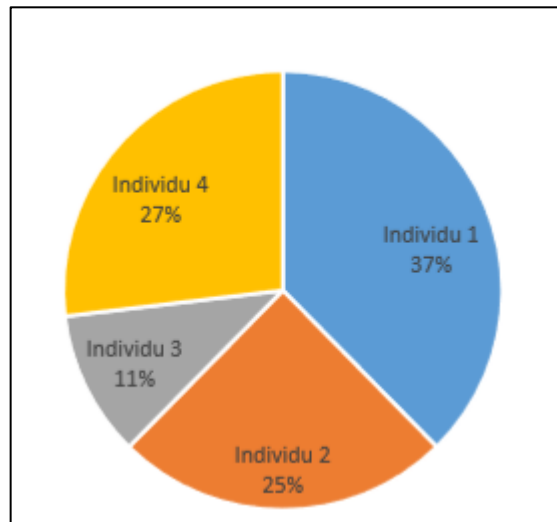


Figure 3.4 Méthode de sélection de la loterie biaisée [18]

1.5.2 Sélection par tournoi

Le tournoi (tournament selection) est l'une des méthodes les plus connues. Il choisit entre deux individus sélectionnés au hasard l'individu le plus important et le reproduit dans une nouvelle population. La même opération sera répétée jusqu'à avoir une population complète [19].

1.6 Opérateurs génétiques

1.6.1 Croisement

Pendant cette opération, la population est divisée en deux sous-populations, ensuite, l'opération de reproduction est appliquée sur chaque individu de la première sous population avec un autre de la deuxième sous-population, chaque couple de parents forme deux nouveaux chromosomes (enfants) dont les caractéristiques sont issues des deux parents. Le taux de croisement est défini en général entre 0.5 et 0.9, un taux excessivement élevé peut engendrer la perte de bonnes structures, mais aussi un taux faible provoque la stagnation puisque l'exploration est diminuée.

Deux méthodes classiques sont à mentionner, le croisement en un point et le croisement en deux points :

Chaque chromosome de longueur "l" est divisé en segments : deux segments pour le croisement en un seul point (Goldberg, 1989) et trois segments pour le croisement en deux points (Man et al. 1996). La position de chaque chromosome est tirée aléatoirement ensuite les gènes des individus parents sont échangés selon une probabilité de croisement P_c pour former deux nouveaux chromosomes [17].

Les schémas ci-dessous illustrent les deux méthodes :

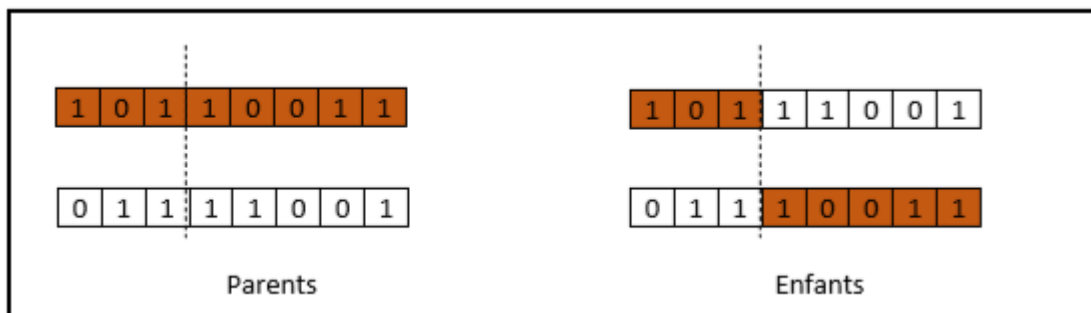


Figure 3.5 : Croisement en un seul point [17]

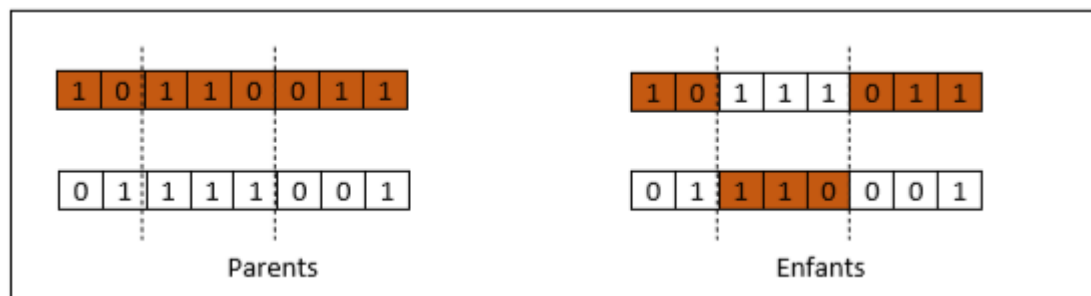


Figure 3.6 Croisement en deux points [17]

1.6.2 Mutation

Cet opérateur tire au hasard un ou plusieurs gènes d'un individu, puis les modifie aléatoirement à condition que cet individu reste une solution possible au problème.

Chaque bit d'un individu a une probabilité P_m de subir une mutation, un réel $r \in \{0,1\}$ est généré, si $r < P_m$ le bit sera donc remplacé par complémentaire [17].



Figure 3.7 Mutation d'un chromosome [17].

L'opérateur de mutation rapporte de la diversité à la population, en évitant la convergence et donc l'AG peut atteindre tous les points de l'espace de recherche et donc l'optimum global [17].

2. L'optimisation multi-objectif

L'optimisation multicritères est destinée aux problèmes de prise de décision que l'on rencontre à tous les échelles de gestion, qui ont des critères multiples et qui peuvent être en conflit. Par exemple, un nouveau processus de production peut être adopté pour l'amélioration des délais de livraison, la qualité des produits et la réduction des coûts.

Un problème d'optimisation avec objectifs multiples peut être représenté par le programme suivant :

Optimiser $y = (f_1(x), \dots, f_n(x))$

t.q. $x = (x_1, \dots, x_m) \in X$ et $y = (y_1, \dots, y_n) \in Y$

Où x est appelé vecteur de décision constitué de m variables de décision, dans un AGs, il représente un individu (solution potentielle). Et y vecteur objectif de n objectifs. X représente l'espace de décision, et Y l'espace des objectifs [17].

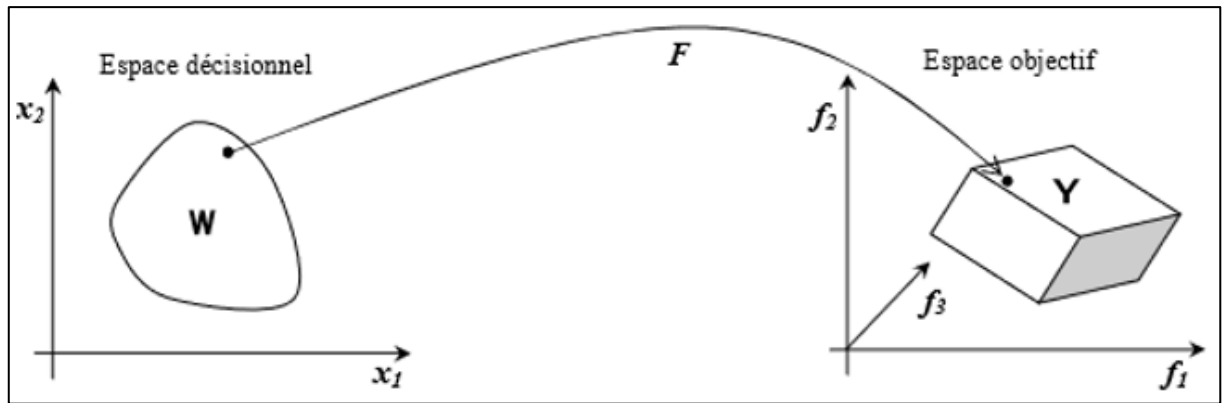


Figure 3.8 Problème d'optimisation multi objectif (2 variables de décision et 3 fonction objectifs)[17].

On doit suivre plusieurs phases pour arriver à résoudre un problème multicritère qui peuvent être considérés, comme suit :

- L'identification des critères d'optimisation.
- L'élaboration d'une solution en essayant d'exploiter les données disponibles.
- Enfin, une ou plusieurs solutions optimales seront disponibles selon la méthode adoptée.

Plusieurs fonctions d'objectif ($n \geq 2$) sont posées dans un problème d'optimisation multi objectif. Chaque fonction peut avoir un optimum différent. On doit arriver à de « bons compromis » plutôt qu'à une solution unique. Les notions de dominance et d'optimalité de Pareto remplace la notion d'optimum [2].

- **La dominance** : une solution A domine une solution B pour un problème de minimisation si et seulement si : $\forall i \in \{1, 2, \dots, n\} : f_i(A) \leq f_i(B)$
 $\exists i \in \{1, 2, \dots, n\} : f_i(A) < f_i(B)$

On dit que B est dominée par A ou entre les deux solutions, A est la solution non dominée[2].

- **Pareto optimum**

Le mathématicien italien Vilfredo Pareto au XIX^{ème} siècle, a formulé un autre concept du problème multi objectif appelé plus tard optimum de Pareto. Selon lui, il y a un équilibre entre les critères du problème multi objectif qui empêche d'améliorer d'un d'eux sans détériorer au moins un des autres critères. Un point x est appelé Pareto-optimal lorsque aucun autre point de l'espace de recherche X ne le domine, ces points sont également appelés solutions non inférieures ou non dominées. L'optimisation multi objectifs, est le résultat d'un ensemble de solutions optimales et non d'une solution unique. Toute solution de cet ensemble elle-même

optimale, à tel point qu'il est impossible d'améliorer les performances d'un critère sans dégrader les performances d'au moins un autre critère.

On appelle front de Pareto la représentation des solutions non dominées dans l'espace d'objectif. La (figure 3.9) montre un exemple de front de Pareto pour un problème de minimisation à deux objectifs. L'ensemble de points blancs représentent le front de Pareto[2][5].

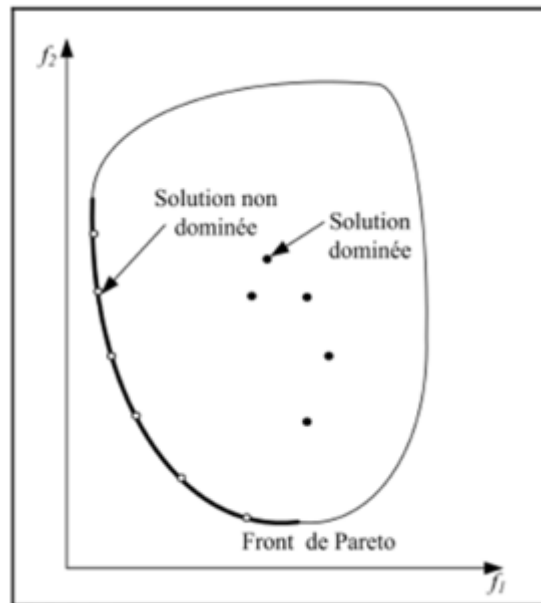


Figure 3.9 Front de Pareto [2]

3. Optimisation multi-objectif par algorithmes génétiques

Les algorithmes génétiques sont une méthode efficace dans la résolution des problèmes d'optimisation multi-objectif, ils sont répartis en trois classes [6] :

- Ceux qui transforment le problème multi-objectif en un problème mono-objectif en utilisant une fonction objectif équivalente Les AGs de cette classe sont appelés : **méthodes agrégées**.
- **Les méthodes non-agrégées non-Pareto** utilisent des sous-populations pour optimiser chaque objectif indépendamment.
- **Les méthodes Pareto** consistent à déterminer un ensemble de solutions optimales, elles favorisent les solutions non-dominées dans leurs processus de sélection en utilisant la notion de dominance au sens de Pareto.

3.1. Méthodes agrégées

Elles consistent à transformer le vecteur objectif en un scalaire, il en existe plusieurs types:

3.1.1. La moyenne pondérée

Cette méthode additionne tous les objectifs après avoir multiplié chacun d'eux par un coefficient de poids, ce dernier dépend de l'importance de l'objectif, donnant un problème mono-objectif :

$$\min \sum_{i=1}^n w_i f_i(x) \text{ avec } w_i \geq 0$$

$$\text{t.q. } \sum_{i=1}^n w_i = 1$$

3.1.2. Goal Programming

Chaque objectif $f(x)$ a un but L_i à atteindre, les L_i sont des contraintes introduites dans le calcul des solutions :

$$\min \sum_{i=1}^n |f_i(x) - L_i|$$

Un vecteur de buts L est imposé sur les objectifs $f(x)$. Le principe de cette méthode est de minimiser les écarts entre les objectifs $f(x)$ et leurs valeurs L_i à atteindre. La différence entre cette méthode et la méthode précédente est que dans cette méthode, les valeurs désirées de chaque objectif sont connues, tandis que dans la précédente, c'est l'importance relative des objectifs qui est requise.

3.2. Méthodes non-agrégées non-Pareto

Dans ce type d'approche, la population est utilisée pour diversifier la recherche, mais le principe de Pareto n'est pas directement incorporé dans le processus de sélection. Le modèle le plus représentatif de cette méthode est le Vector Evaluated Genetic Algorithm (VEGA)[21].

3.2.1. Vector evaluated genetic algorithm (VEGA)

Le VEGA (Vector evaluated genetic algorithm) proposé par Schaffer, a été la première méthode non agrégative utilisant les algorithmes génétiques pour résoudre un problème d'optimisation multi objectif. Cet algorithme considère une population de N individus. A chaque génération, la population est aléatoirement divisée en autant de sous-populations égal au nombre d'objectifs. Chaque sous-populations vont être créées en considérant un objectif particulier. Ensuite, ces sous populations sont mélangées afin d'obtenir une nouvelle

population de N individus et les opérateurs de croisement et de mutation sont appliqués. L'inconvénient majeur est qu'il a tendance à générer des solutions qui excellent dans un seul objectif, sans tenir compte des autres objectifs [2].

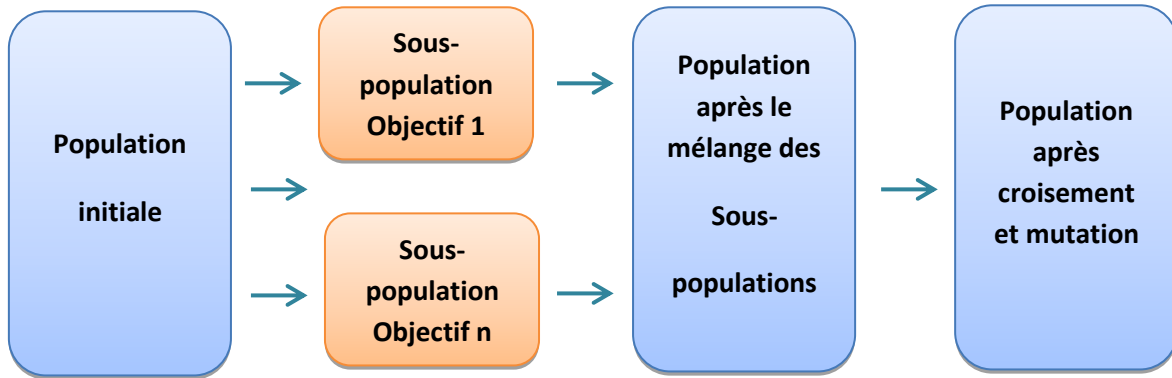


Figure 3.10 Vector Evaluated Genetic Algorithm (VEGA) [17].

3.3 Méthodes Pareto

Cette catégorie de méthodes introduit et prend en charge le concept d'optimalité de Pareto dans leurs mécanismes de sélection, on va citer quelques-unes :

3.3.1. Multiple Objectives Genetic Algorithm (MOGA)

Le MOGA algorithme, proposé par Fonseca et Fleming[11], est une méthode qui utilise la notion de dominance pour ranger les individus de la population, dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le domine. Pour démarrer l'algorithme, les relations de domination sont d'abord calculées pour chaque solution. Puis, pour un individu i , un rang égal à 1 plus le nombre d'individus n_i qui dominant l'individu i est attribué.

Un fitness est ensuite attribué à chaque solution en fonction de son rang, les individus avec les rangs les plus faibles ayant les meilleurs fitness [2]. Et pour maintenir la diversité entre les solutions non dominées, les auteurs utilisent le sharing. Cette méthode permet d'obtenir des solutions de bonne qualité et s'implante facilement [24].

- **La fonction de partage (sharing)**

Cette technique permet la formation de sous-populations (niches). Les individus d'une s/population sont des voisins proches. La technique utilise également le concept de la fonction de partage qui permet de dégrader la fonction coût de chaque individu par une quantité liée aux nombres d'individus qui lui sont similaires dans la population, c'est-à-dire de très proches

voisins. Ainsi, la fonction dégradée f' d'un individu ayant une fonction coût f est exprimée par :

$$f'(x) = \frac{f(x)}{m(x)}$$

$$\text{Où : } m(x) = \sum_{y \in pop} sh(d(x, y))$$

$m(x)$ est appelé le compteur de niche qui mesure le nombre approximatif d'individus dans le voisinage de l'individu x ; le paramètre sh représente la fonction de partage mesurant le niveau de similarité entre 2 individus x et y , et d est la distance entre ces individus. La fonction de partage est donnée par l'équation suivante :

$$\text{Si } d < \sigma_{share} \text{ Alors } sh(d) = 1 - \left(\frac{d}{\sigma_{share}}\right)^\alpha$$

$$\text{Sinon } sh(d) = 0$$

Où : α est un paramètre de réglage de la forme de la fonction de partage.

σ_{share} Est la mesure de similarité (rayon de niche) [24].

3.3.2. Non-dominated Sorting Genetic Algorithm II (NSGA II)

Comme son nom l'indique, le Non-dominated Sorting Genetic Algorithm II trie (rapidement) la population à base de non-dominance [19], voici l'algorithme de cette méthode :

<i>NSGA (N,g)</i>	
<i>P0=initialisation-de-la-population(N)</i>	
<i>Evaluation des objectifs</i>	
<i>Tri-non-dominance(P0)</i>	
<i>Générer une population d'enfants Q0</i>	
<i>Sélection par tournoi binaire</i>	
<i>mutation et recombinaison</i>	
<i>Pour t=1 jusqu'à g</i>	
<i>Rt=Pt ∪ Qt</i>	Combiner les 2 populations
<i>{ F1,F2,..Fk }=tri-non-dominance(Rt)</i>	Trier Rt
<i>Pt+1= ∅</i>	Initialiser Pt+1
<i>i=1</i>	Initialiser le compteur
<i>Tant que N- Pt + 1 ≥ Fi </i>	Tant que Pt+1 est incomplète
<i>Pt+1=Pt+1 ∪ Fi</i>	inclure les fronts Fi dans Pt+1
<i>i=i+1</i>	incrémenter i de 1
<i>Fin tant que</i>	
<i>Si N- Pt + 1 < Fi & N - Pt + 1 ≠ 0</i>	Si dernier front Fi est trop large
<i>assignement-distance-crowding(Fi)</i>	attribuer les distances
<i>tirer(Fi,<n)</i>	opérateur de crowding
<i>Pt+1=Pt+1 ∪ Fi [1:N- Pt + 1]</i>	inclure une partie de Fi ∪ Pt+1
<i>Sinon</i>	si Pt+1 est complète
<i>Qt+1=créer-nouv-pop(Pt+1)</i>	créer une nouvelle population
<i>t+1</i>	Prochaine génération

Algorithme 3.1 L'algorithme de NSGA II [19]

Explication : D'abord, une population de parents (P0) de taille (N) est générée, elle est tirée à base de non dominance, chaque solution se fait attribuer une fitness factice égale à son niveau de non-dominance (veut dire non-dominée donc meilleure fitness). Après l'application des opérations génétiques sur (P0), une population enfant Q0 de taille N est créée. Après, cette première procédure, une boucle commence. Tant que le critère d'arrêt n'est pas vérifié, les populations, parents et enfants sont mélangées pour produire une population ($R_t = P_t \cup Q_t$), puis tirées à base de non dominance pour identifier plusieurs fronts F1, F2, etc. [10] Les meilleurs individus vont se retrouver dans le ou les premiers fronts. Une nouvelle population parente (P_{t+1}) est formée en ajoutant les fronts au complet (premier front F1, second front F2, etc.) tant que ceux-ci ne dépassent pas N. Si le nombre d'individus présents dans (P_{t+1}) est inférieur à (N), une procédure de crowding est appliquée sur le premier front suivant, (F_i), non inclus dans (P_{t+1}) pour insérer les ($N - |P_{t+1}|$) meilleurs individus de (F_i) qui manquent dans la population (P_{t+1}). Les individus de ce front sont utilisés pour calculer la distance de crowding entre deux solutions voisines. Une fois que les individus de la population (P_{t+1})

sont identifiés, une nouvelle population enfant (Q_{t+1}) est créée par sélection, croisement et mutation [19][12].

L'organigramme ci-dessous illustre le fonctionnement du NSGA II

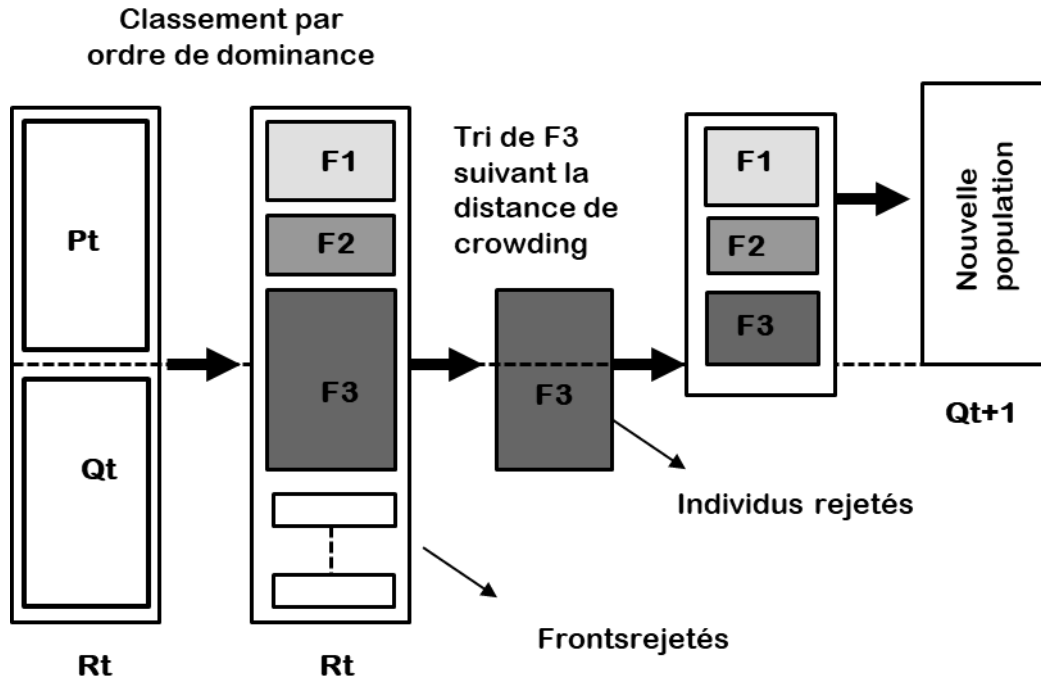


Figure 3.11 le fonctionnement du NSGA II [19].

Calcul de la distance de crowding (distance de surpeuplement)

Une procédure de crowding, basée sur un calcul de distance (distance de crowding) qui ne nécessite aucun paramétrage. La distance de crowding d'une solution particulière i se calcule en fonction du périmètre de l'hypercube ayant comme sommets les points les plus proches de i sur chaque objectif. Sur la figure 3.8, est représenté l'hypercube en deux dimensions associées au point i . Le calcul de la distance de crowding nécessite, avant tout, le tri des solutions selon chaque objectif, dans un ordre ascendant. Ensuite, pour chaque objectif, les individus possédant des valeurs limites se voient associés une distance infinie. Pour les autres solutions intermédiaires, on calcule une distance de crowding égale à la différence normalisée des valeurs de fonctions objectifs de deux solutions adjacentes. Ce calcul est réalisé pour chaque objectif. La distance de crowding d'une solution est obtenue en sommant les distances correspondantes à chaque objectif, le meilleur individu est l'individu qui possède la plus grande valeur de distance de crowding [19].

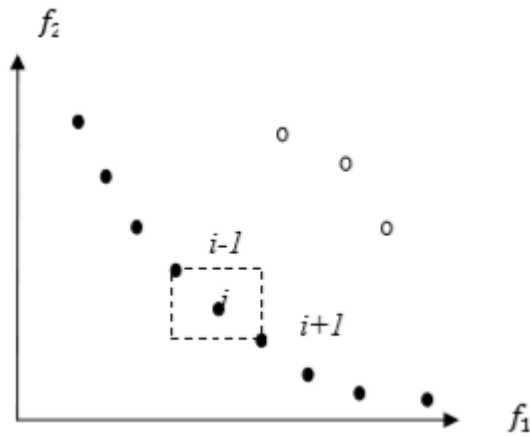


Figure 3.12 Distance de crowding (les points noirs sont les solutions appartenant au même front)
[19]

Conclusion

Ce troisième chapitre a été consacré à l'optimisation multi-objectif et aux algorithmes génétiques multi-objectifs. Nous avons présenté une introduction sur les principes de base des algorithmes génétiques et leur fonctionnement ensuite on a vu quelques AGs utilisés dans la résolution des problèmes d'optimisation multi-critères parmi eux le NSGAII considéré comme l'un des meilleurs MOGAs, a été adopté et intégré dans l'approche.

Le chapitre suivant sera consacré à l'implémentation et la mise en œuvre de l'algorithme génétique, ainsi l'étude et les résultats obtenus.

CHAPITRE 4

**LA METHODE DE RESOLUTION ET
IMPLIMENTATION**

Introduction

Dans ce chapitre, nous allons présenter une vue sur l'objectif et l'architecture de la méthode utilisée, ainsi qu'une brève introduction sur le langage et l'environnement de développement que nous avons utilisé. Ensuite, on a procédé quelques exemples pour tester. Après Nous avons présenté l'application.

1. La méthode de résolution

1.1 L'objectif de la méthode

L'objectif de notre travail est de donner un environnement d'optimisation basé sur une combinaison des méthodes mathématiques (pour l'optimisation) et la simulation (pour évaluation), pour optimiser globalement la conception des chaînes logistiques. Un modèle est un ensemble d'hypothèses sur le comportement d'un système. Ces hypothèses prennent la forme de relations mathématique ou logique.

Si les relations qui composent le modèle sont assez simples, il peut être possible d'utiliser des méthodes mathématiques pour obtenir des informations précises sur les questions d'intérêt c'est ce qu'on appelle une solution analytique. Cependant, la plupart des systèmes du monde réel sont trop complexes pour être évaluées analytiquement, ces modèles doivent être étudiés au moyen de la simulation.

Les modèles de simulation par ordinateur sont largement utilisés comme des modèles de systèmes complexes réels pour évaluer leurs réponses à certains stimulants. L'un des inconvénients de simulation est que ce n'est pas une technique d'optimisation. Dans la plupart des études, plusieurs algorithmes de recherche ont été liés aux simulations, les algorithmes génétiques ont montré la capacité de résoudre les grands problèmes de façon robuste, c'est pour cela nous utilisons l'approche de simulation à base d'optimisation.

1.2 L'architecture de la méthode

La solution suggérée est basée sur l'architecture représentée sur la figure 4.1

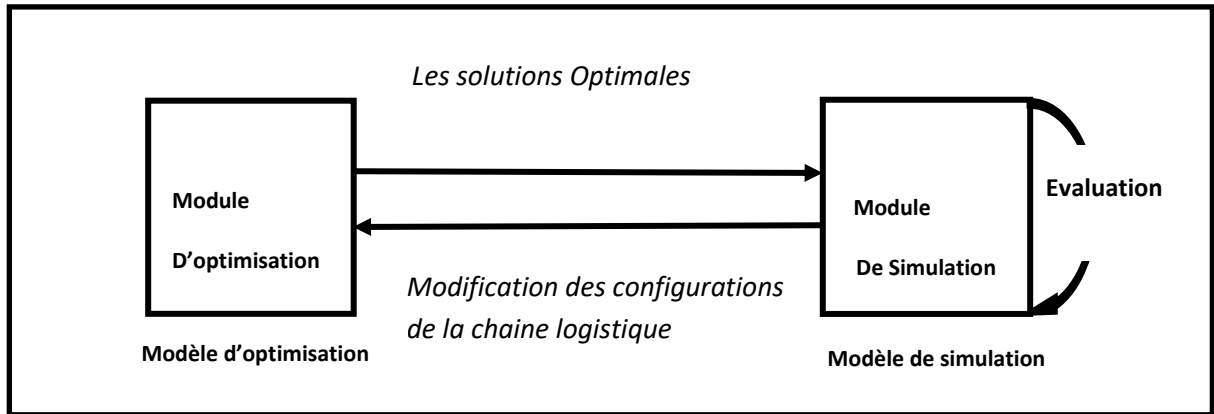


Figure4.1 L'architecture de la méthode utilisée

L'architecture globale est divisée en deux modèles (le modèle d'optimisation et le modèle de simulation), qui sont des entrées aux deux modules. Le premier module est le module d'optimisation et le second est le module de simulation. La raison pour laquelle on a utilisé la combinaison de deux modules (optimisation et simulation) est due au comportement complexe de la chaîne logistique, ainsi le processus de la séparation du comportement du modèle de la chaîne logistique à deux sous-modèles est nécessaire, pour gagner les avantages des méthodes analytiques et des méthodes de simulation.

Et afin de produire un modèle optimal d'une chaîne logistique notre système d'optimisation à base de simulation utilise une configuration initiale de la chaîne logistique, qui contient la table du temps de traitement, puis le module d'optimisation va générer une solution d'ordonnancement optimal par l'application de l'algorithme génétique plus précisément le NSGAI, qui explore les solutions candidates intéressantes dont les performances sont évaluées par le simulateur. Dès réception d'une solution candidate, le simulateur génère automatiquement le modèle de simulation correspondant et évalue la valeur d'une fonction objective, qui serait en fait la somme des retards, si on a du retard le module de simulation va modifier la configuration de la chaîne logistique, ainsi il va abaisser tous les temps de traitement des tâches et cela mène à l'augmentation du coût et de la pénalité). Dans la deuxième éventualité, lorsqu'on n'a pas de retard il ferait l'inverse, il va augmenter tous les temps de traitement des tâches et réduire les coûts sans la moindre pénalité.

Ce processus continuera jusqu'à atteindre un nombre maximum fixe d'itérations. Enfin on sera appelé à choisir parmi ces modèles le meilleur qui minimise la somme de C_{max} et

du cout sans le moindre retard, ainsi on aurai atteint une solution optimale déduit d'une méthode d'optimisation basée sur la simulation.

2. Outils et environnement de développement

Avant de commencer l'implémentation de notre travail, nous allons tout d'abord spécifier les outils utilisés qui nous ont semblé être un bon choix vu les avantages qu'ils offrent.

Langage JAVA :

Notre choix du langage java a été guidé par les avantages qu'offre la programmation orientée objet. Java utilise des processus qui augmentent les performances des entrées/sorties, facilitent l'internationalisation. Il examine le programme au fil de l'exécution et libère automatiquement la mémoire. Cette fonctionnalité diminue les risques de panne du programme et évite la possibilité de mal utiliser la mémoire [19].

Netbeans :

Notre travail est écrit en Netbeans version 8.2 sous Windows 10. Le choix de la programmation sous Windows a été pris à cause de l'interface graphique qu'offre cet environnement. Le choix de Netbeans est fondamental car c'est un logiciel permettant principalement le développement en java. Il fournit un environnement standard de développement pour créer des interfaces très puissantes [19].

3. Quelque exemple sur l'implémentation

Exemple 01 : Dans un problème de type M-machines identiques en parallèles compose de 4 machines et 7 jobs tels que les données sont :

Tâches	1	2	3	4	5	6	7
P (temps de traitement)	4	3	4	5	2	6	3
r (temps de disponibilité)	0	2	2	4	6	8	10
d (temps de livraison)	8	8	12	12	14	18	20

Table 4.1 La table des tâches Exemple 1.

En appliquant notre algorithme génétique (NSGAI) 10 fois, nous avons obtenu la courbe suivante :

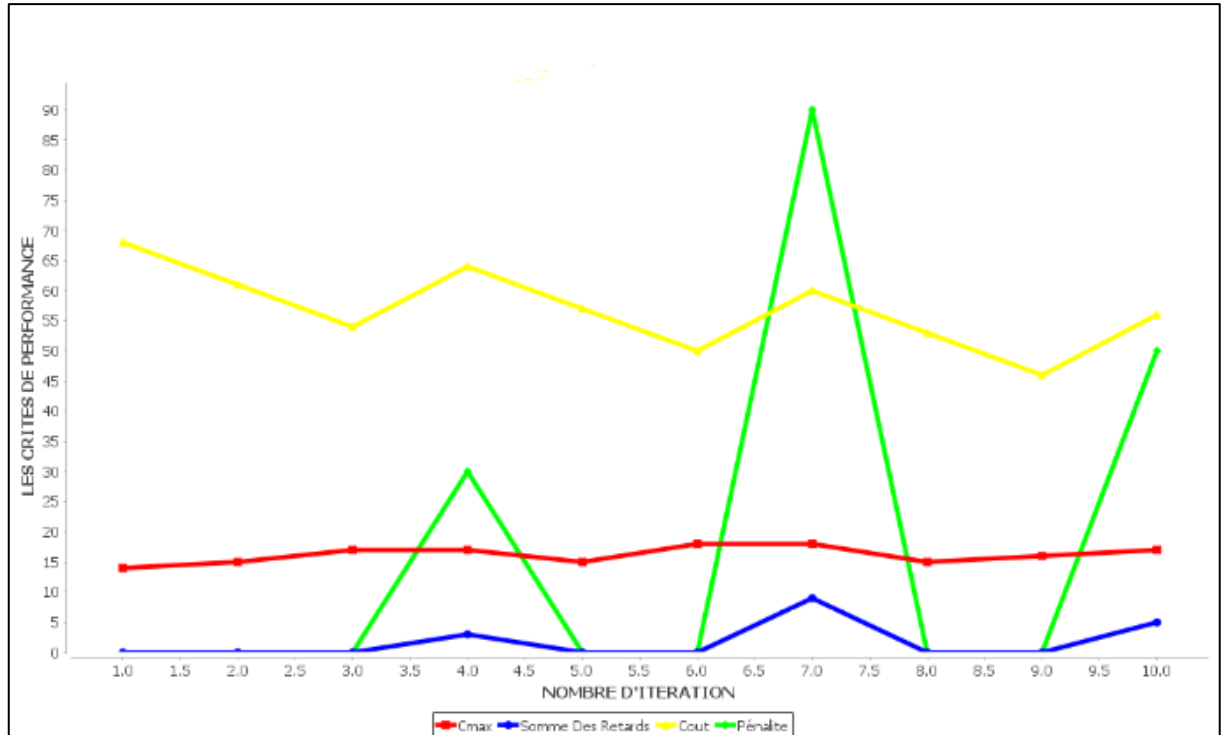


Figure 4.2 Résultats de meilleurs modèles obtenus par NSGA II de problème 4x7.

La (figure 4.2) donne les résultats de simulation pour les meilleurs modèles (axe des X). Les quatre courbes colorées représentent le coût total, le makespan Cmax, la pénalité de retard de la solution d'ordonnancement et la somme des retards ($\sum T$).

Lorsque nous faisons la comparaison entre les meilleurs modèles, nous constatons que le modèle numéro 9 est le meilleur, qui à zéro temps de retard et donc zéro pénalité et la valeur de la somme du coût total et de le makespan Cmax (cout + Cmax) la plus basse.

- **La solution initiale**

Avec notre implémentation nous avons la solution initiale suivante :

Tâches	1	2	3	4	5	6	7
P (temps de traitement)	4	3	4	5	2	6	3

Table 4.2 La table des temps du traitement de la solution initiale.

On a obtenu $C_{max}=14$, $\sum T=0$, Cout =68, Pénalité= 0.

- **La solution optimale**

Dans chaque modèle de ces 10 derniers, le module de simulation fait une modification des chaines logistique au niveau des temps de traitement (pi) dans la meilleure solution on obtient les temps de traitement (processing time pi) suivant :

Tâches	1	2	3	4	5	6	7
P (temps de traitement)	6	5	6	7	4	8	5

Table 4.3 La table des temps du traitement de la solution optimale.

▪ **L'ordre de l'ordonnancement optimale obtenu après l'application de NSGA II :**

Tâches	1	2	3	4	5	6	7
n (indice de machine)	2	1	3	4	2	1	3
K (indice de position sur la machine)	1	1	1	1	2	2	2

Table 4.4 La table de la solution optimale avec NSGA II Exemple 1.

Les matrices Pr, R, D, C, T sont :

$$Pr = \begin{pmatrix} 5 & 8 & 0 & 0 & 0 & 0 & 0 \\ 6 & 4 & 0 & 0 & 0 & 0 & 0 \\ 6 & 5 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad R = \begin{pmatrix} 2 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 \\ 2 & 10 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 8 & 18 & 0 & 0 & 0 & 0 & 0 \\ 8 & 14 & 0 & 0 & 0 & 0 & 0 \\ 12 & 20 & 0 & 0 & 0 & 0 & 0 \\ 12 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 7 & 16 & 0 & 0 & 0 & 0 & 0 \\ 6 & 10 & 0 & 0 & 0 & 0 & 0 \\ 8 & 15 & 0 & 0 & 0 & 0 & 0 \\ 11 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Alors que Pr c'est la matrice de temps opératoire de la tâche j , R c'est la matrice de date de début, D c'est la matrice date de fin la tâche j , C c'est la matrice de makespan, T c'est la matrice retard réel de la tâche j .

On a obtenu $Cmax=16$, $\sum T=0$, Cout =46, Pénalité= 0.

Diagramme de Gantt

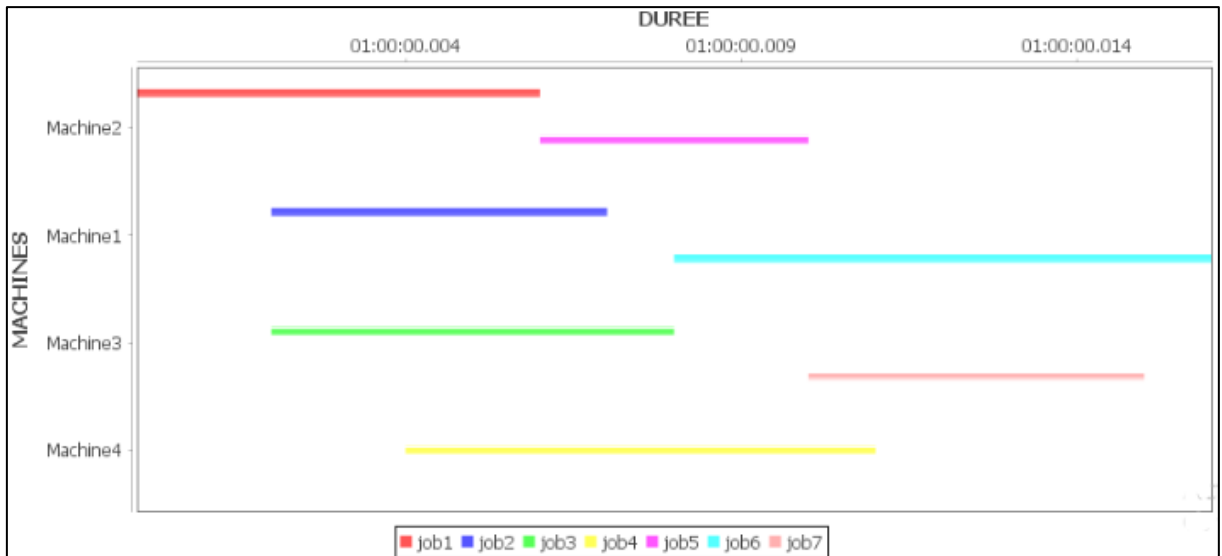


Figure 4.3 Exemple de Diagramme de Gant obtenu par NSGA II de problème 4x7.

Exemple 03 : Dans un problème de type M-machines identiques en parallèles compose de 3 machines et 9 jobs tels que les données sont :

Tâches	1	2	3	4	5	6	7	8	9
P (temps de traitement)	4	3	4	5	2	6	3	6	4
r (temps de disponibilité)	0	2	2	4	6	8	10	8	9
d (temps de livraison)	8	8	12	12	14	18	20	18	18

Table 4.5 La table des tâches Exemple 3.

En appliquant notre algorithme génétique (NSGAII) 10 fois, nous avons obtenu la courbe suivante :

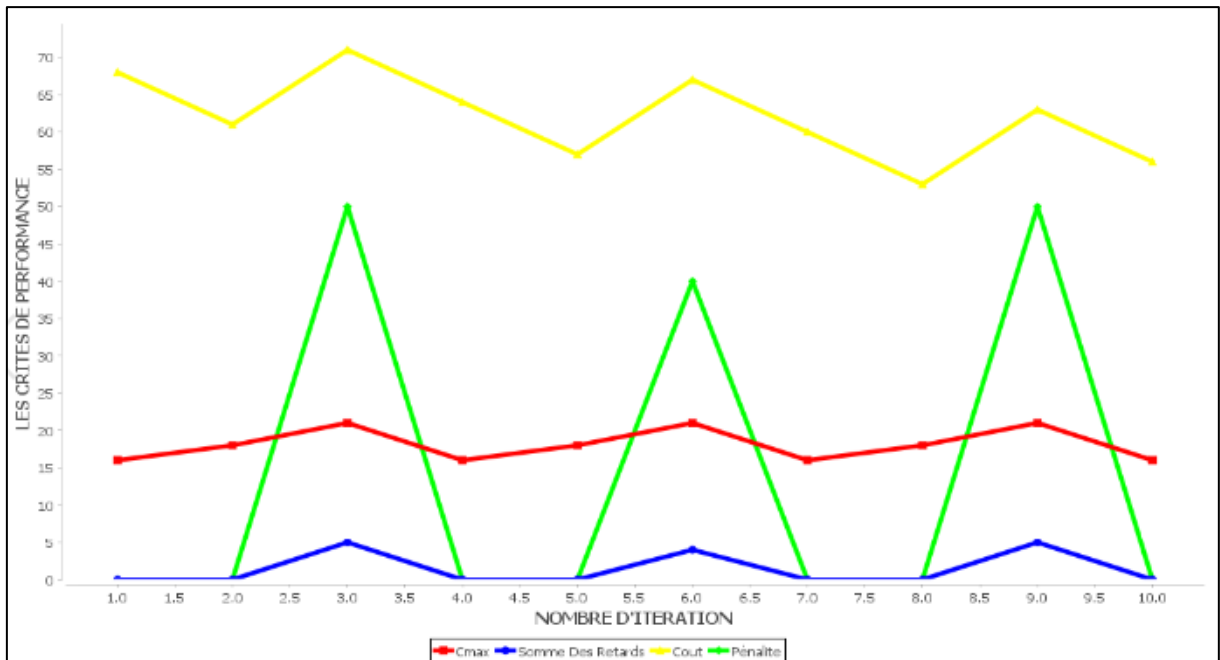


Figure 4.4 Résultats de meilleurs modèles obtenus par NSGA II de problème 3x9.

La (figure 4.4) donne les résultats de simulation pour les meilleurs modèles (axe des X). Les quatre courbes colorées représentent le coût total, le makespan Cmax, la pénalité de retard de la solution d'ordonnancement et la somme des retards ($\sum T$).

Lorsque nous faisons la comparaison entre les meilleurs modèles, nous constatons que le modèle numéro 8 est le meilleur, qui à zéro temps de retard et donc zéro pénalité et la valeur de la somme du coût total et de le makespan Cmax (cout + Cmax) la plus basse.

- **La solution initiale**

Avec notre implémentation nous avons la solution initiale suivante :

Tâches	1	2	3	4	5	6	7	8	9
P (temps de traitement)	4	3	4	5	2	6	3	6	4

Table 4.6 La table des temps du traitement de la solution initiale.

On a obtenu $C_{max}=16$, $\sum T=0$, Cout =68, Pénalité= 0.

- **La solution optimale**

Dans chaque modèle de ces 10 derniers, le module de simulation fait une modification des chaines logistique au niveau des temps de traitement (pi) dans la meilleure solution on obtient les temps de traitement (processing time pi) suivant :

Tâches	1	2	3	4	5	6	7	8	9
P (temps de traitement)	5	4	5	6	3	7	4	7	5

Table 4.7 La table des temps du traitement de la solution optimale.

▪ **L'ordre de l'ordonnancement obtenu après l'application de NSGA II :**

Tâches	1	2	3	4	5	6	7	8	9
n (indice de machine)	1	2	3	1	2	1	3	2	3
K (indice de position sur la machine)	1	1	1	2	2	3	3	3	2

Table 4.8 La table de la solution optimale avec NSGA II Exemple 3.

Les matrices Pr, R, D, C, T sont :

$$Pr = \begin{pmatrix} 5 & 6 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad R = \begin{pmatrix} 0 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 6 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 9 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 8 & 12 & 18 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 14 & 18 & 0 & 0 & 0 & 0 & 0 & 0 \\ 12 & 18 & 20 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 5 & 11 & 18 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 9 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 14 & 18 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Alors que Pr c'est la matrice de temps opératoire de la tâche j , R c'est la matrice de date de début, D c'est la matrice date de fin la tâche j , C c'est la matrice de makespan, T c'est la matrice retard réel de la tâche j .

On a obtenu $C_{max}=18$, $\sum T=0$, Cout =53, Pénalité= 0.

Diagramme de Gantt

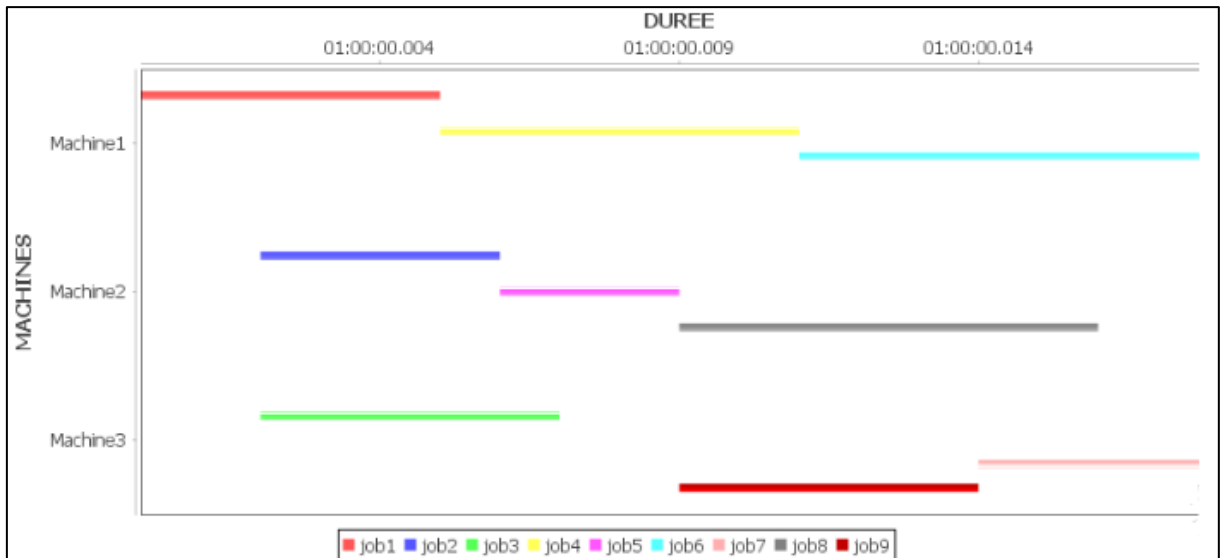


Figure 4.5 Exemple de Diagramme de Gant obtenu par NSGA II de problème 3x9.

- **La comparaison entre la solution initiale et la solution optimale**

On a appliqué dans la première itération uniquement l'algorithme NSGAII, ainsi on n'a inclut le module d'optimisation, par contre dans les autres itérations nous avons appliques le couplage entre les deux modules optimisation-simulation. Ceci permettrait d'améliorer la qualité de la solution.

3. Illustration de l'application

Notre application se compose de cinq fenêtres d'affichage, la premier représente l'interface principale qui permet de choisir des exemples existants qui contient deux exemple 4 machines et 7 jobs et 3 machines 9 jobs

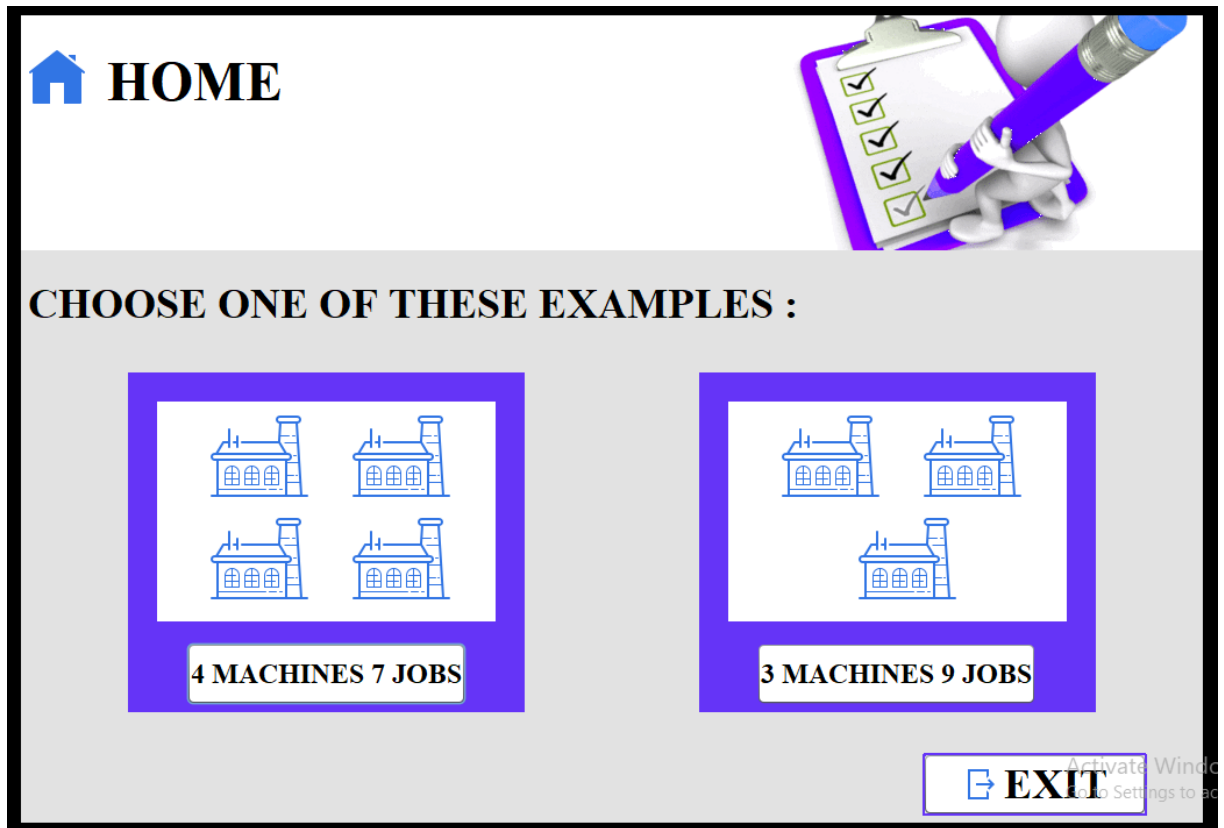


Figure 4.6 Interface principale de l'application

La deuxième interface qui serait l'image de la représentation du courbe des nombres d'itérations en terme des fonctions objectifs (C_{max} , $\sum T$) et les critères de performances (le cout, la pénalité)

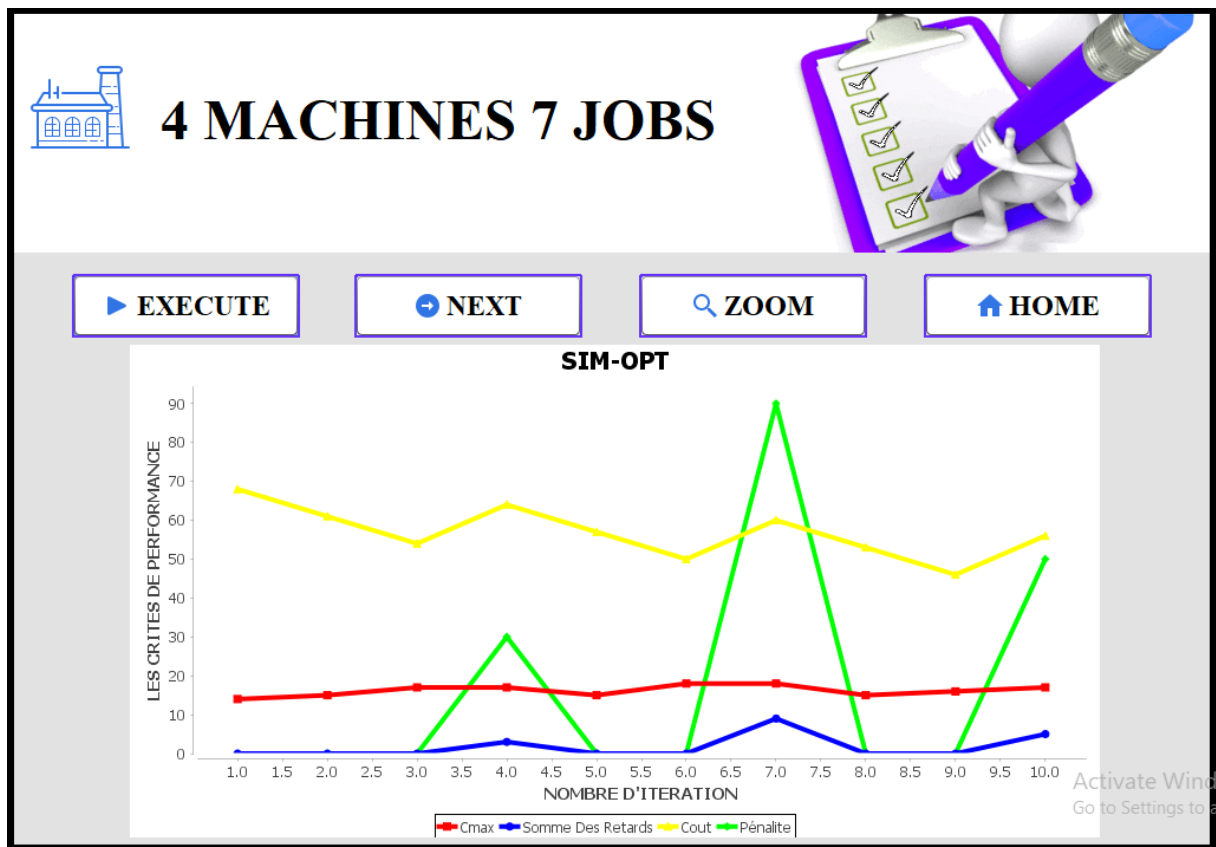


Figure 4.7 La représentation du courbe

La troisième interface inclut la représentation de la solution optimale (l'ordonnancement optimale) et les valeurs des deux fonctions objectives et les deux critères de performance

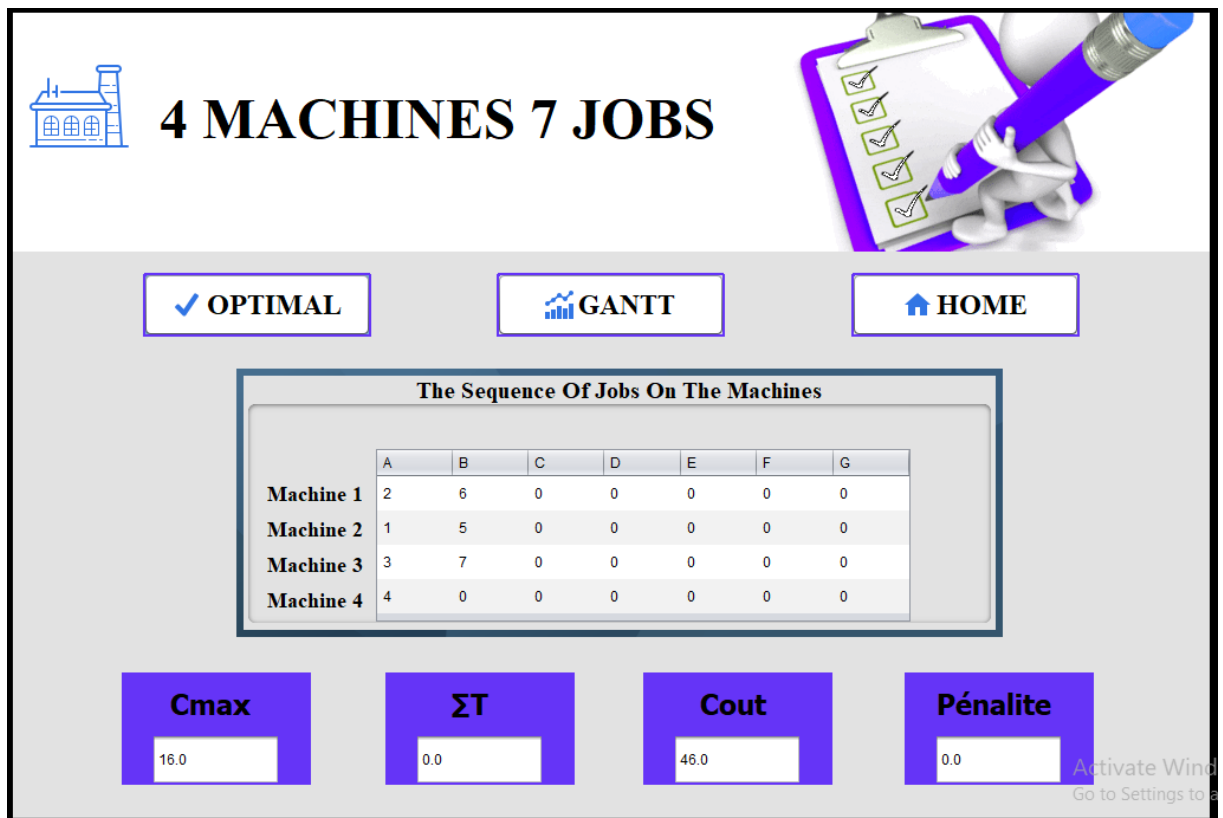


Figure 4.8 l'ordonnancement optimale

La quatrième interface représente le diagramme de Gantt

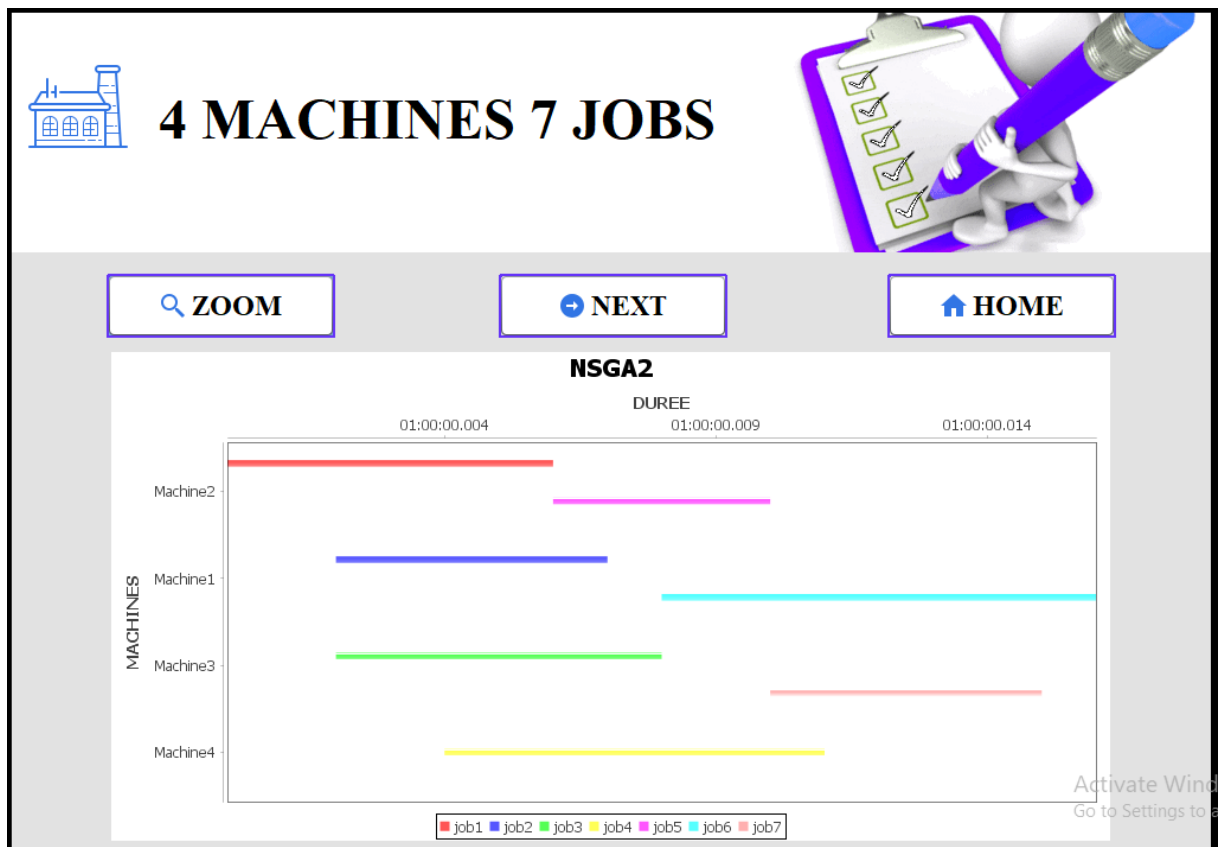


Figure 4.9 Diagramme de Gantt

La dernière interface contient la comparaison entre les temps de traitements (π_i) de l'état initiale et l'état optimale ainsi que la comparaison des fonctions objectifs et les critères de performance

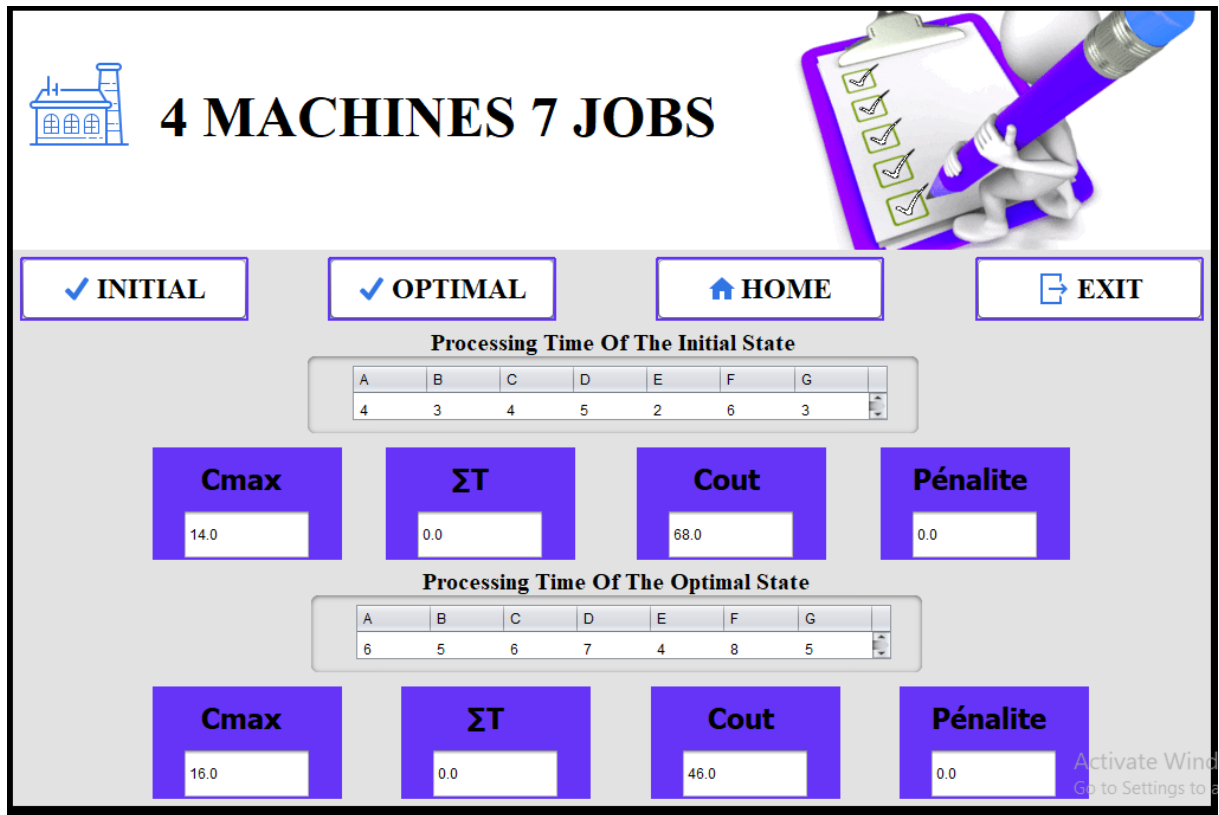


Figure 4.10 La comparaison entre l'état initiale et l'état optimale

Conclusion

Nous avons présenté dans ce chapitre l'objectif ainsi que l'architecture de la méthode de résolution, et les Outils et l'environnement de développement que nous avons utilisé pour résoudre le problème que nous avons abordé dans cette mémoire. Ensuite nous avons présenté le logiciel développé et nous avons pris des exemples ainsi que les résultats obtenus.

CONCLUSION GENERALE

Dans ce travail, nous avons présenté une méthode basée sur les techniques de simulation/optimisation pour optimiser la planification de la chaîne logistique en déterminant un plan faisable, qui serait à la hauteur des besoins et les limitations de chaîne en optimisant les plans par rapport aux buts de l'entreprise tels que la réduction des coûts et des temps de retard ...etc.

Ces études peuvent répondre aux besoins réels du monde des affaires, que ce soit dans l'environnement industriel de fabrication d'objets physiques ou dans l'environnement de service. Beaucoup de ces entreprises sont confrontées quotidiennement à des problèmes pour satisfaire leurs clients, qui sont de plus en plus exigeants en termes de performances. La livraison à temps, réduire les coûts de stockage, répondre rapidement à la demande du marché, tels sont les avantages de la mise en place d'un bon fonctionnement du système.

La simulation/optimisation peut être bénéfique pour les entreprises de plusieurs manières, particulièrement l'utilisation de la simulation a permis une évaluation précise des performances de la chaîne avec la prise en compte des incertitudes dans la chaîne. De plus l'algorithme génétique multicritères implémenté montre son efficacité pour trouver des solutions de bonne qualité avec un meilleur compromis entre les différents objectifs de l'entreprise.

Dans le premier chapitre, nous avons présenté tout d'abord une présentation générale sur la chaîne logistique. Ainsi, nous avons présenté les fonctions, les flux de base de la chaîne logistique, Ensuite nous avons défini la gestion de la chaîne logistique et les niveaux de décisions ainsi que la modélisation des chaînes logistiques et leur critères de performances. Nous avons clôturé ce chapitre par une généralité sur les problèmes d'ordonnancement avec les différents types d'atelier.

Dans le deuxième chapitre nous avons donné une description de l'approche d'optimisation et un certain nombre de méthodes de résolution, ainsi qu'une introduction de l'approche de simulation des chaînes logistiques. Nous avons clôturé ce chapitre par le couplage entre ces deux dernières approches.

Dans le troisième chapitre, nous avons également présenté les algorithmes génétiques et l'optimisation multicritères et une description plus détaillée de l'algorithme génétique multicritères implémenté est présenté.

Dans le quatrième et le dernier chapitre, nous avons présenté les objectifs et l'architecture de la méthode de résolution, ainsi que le langage et l'environnement de développement qui ont été utilisés. Ensuite, l'implémentation de l'algorithme NSGA II (Nondominated Sorting Genetic Algorithm II) qui a été appliqué avec succès pour la résolution de notre problème.

Les résultats obtenus montrent que même si l'optimisation est efficace en termes d'optimalité mais elle reste inefficace dans la modélisation des problèmes complexes, pareil pour la simulation, elle est un outil puissant qui permet de modéliser et d'analyser le comportement opérationnel des chaînes logistiques complexes, mais elle ne nous permet pas une garantie d'une solution optimale, donc on est arrivé à la conclusion ; que la simulation est un excellent outil d'analyse lorsque la configuration optimale de la chaîne logistique est déjà déterminée par un outil d'optimisation.

Comme perspective, nous allons examiner d'autres algorithmes génétiques multicritères et les comparer avec le MOGA actuellement utilisé. Ainsi que les tester dans un problème d'optimisation à trois objectifs.

REFERENCES BIBLIOGRAPHIQUES

- [1] O.KALLEL, L'impact des relations contractuelles entre donneurs d'ordres et sous-traitants sur la performance d'une chaîne logistique équitable, Doctorat, Toulouse, Tunis, 2012.
- [2] M.E.BOUNIF , Optimisation à base de simulation pour le développement des systèmes décisionnels, Doctorat, Msila, 2015.
- [4] S.MEGUIRECHE, Ordonnancement des systèmes de production flexible, Master, Msila, 2019.
- [5] Blum C., Andrea R., Meta-heuristics In Combinatorial Optimization: Overview And Conceptual Comparison, ACM Computing Survey, Vol. 35, n 3, (2003).
- [9] L.Pichot, Stratégie de déploiement d'outils de pilotage de chaînes logistiques : Apport de la classification, Thèse de doctorat, L'institut national des sciences appliquées de Lyon, 2006.
- [10] A. Benhama et Y. Benkhalouf ,Etude comparative des algorithmes génétiques multi-objectifs, Bejaia,2014.
- [11] Fonseca, C.M., P.J. Fleming, Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms (S. Forrest, Ed.). Morgan Kaufmann. San Mateo, California. pp. 416–423, (1993).
- [12] Deb, K., Pratap A., Agarwal S., Meyarivan T, A fast and elitist multi-objective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation 6(2), 182– 197, (2002).
- [13] A.Gherboudj. Méthodes de résolution de problèmes difficiles académiques. Thèse de doctorat, Université de Constantine 2, Algérie, 2013.
- [14] E. A Feigenbaum and j. Feldman. Computers and thought. McGraw-Hill, Inc. New York, NY, USA, 1963.
- [15] M.EL BOUNJIM, CONTRIBUTION À LA CONCEPTION DE LA CHAÎNE LOGISTIQUE VERTE EN BOUCLE FERMÉE, Doctorat, 2016.
- [19] M.KACIMI, Etude comparative des algorithmes génétiques pour un problème d'ordonnancement d'atelier M-Machines identiques en parallèles, Master, 2019.
- [20] I. H Osman and G. Laporte. Metaheuristics: A bibliography. Annals of Operations Research, 63(5):511-623, 1996.
- [21] R. Kammarti. Approches évolutionnistes pour la résolution du 1-PDPTW statique et dynamique. Thèse de doctorat, L'Ecole Centrale de Lille, France, 2006.
- [25] O.GUEMRI, Proposition de solution pour l'optimisation des chaînes logistiques, Doctorat,2017 .
- [26] Chopra S., Meindil P., Supply chain management: Strategy, planning, and Operations, third edition, Pearson Education, Inc. New Jersey (2007).
- [27] A. Bekrar. Résolution du problème de placement en deux dimensions : heuristiques, bornes inférieures et méthodes exactes. Thèse de doctorat, Université de Technologie de Troyes, France, 2007.
- [28] E. Grellier. Optimisation des tournées de véhicules dans le cadre de la logistique inverse : modélisation et résolution par des méthodes hybrides. Thèse de doctorat, Université de Nantes, France, 2008.

- [29] J. K Hao, P. Galinier, and M. Habib. Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. *Revue d'Intelligence Artificielle*, 13(2) :283-324, 1999.
- [30] S.Kande. Etude et résolution de problèmes de planification dans des réseaux logistiques multi-échelons. Thèse de doctorat, l'Université de Technologie de Troyes, France, 2015.
- [31] C. Maneel. Modélisation et résolution de problèmes d'optimisation combinatoire issus d'applications spatiales. Thèse de doctorat, INSA de Toulouse, France, 2004.
- [32] M.BOURZIK, Problèmes d'Ordonnancement d'Atelier, Diplôme de Licence Sciences et Techniques, 2016.
- [33] A.HANNACHE, A.LEMMOUI, Résolution d'un problème d'ordonnancement de type job shop avec contrainte de transport, Thèse de magister, TLEMCEN.
- [3] simcore, <https://www.simcore.fr/gestion-de-flux/flux-logistiques/> ,consulté le 5 février 2020
- [6] isatech, <https://www.isatech.fr/optimisation-logistique/>,consulté le 1 mars 2020
- [7] univ-bejaia,<http://www.univ-bejaia.dz/dspace/bitstream/handle/123456789/600/La%20logistique%20de%20distribution%20%2C%20Optimisation%20des%20coûts%20de%20transport.pdf?sequence=1&isAllowed=y> ,consulté le 6 Mars 2020
- [8] univ-bejaia,<http://www.univ-bejaia.dz/dspace/bitstream/handle/123456789/11795/LA%20CHAINE%20LOGISTIQUE%20ET%20LA%20GESTION%20DES%20STOCKS%20D'UNE%20ENTREPRISE.pdf?sequence=1&isAllowed=y> ,consulté le 6 Mars 2020
- [16] univ-biskra, <http://thesis.univ-biskra.dz/2124/4/chapitre02.pdf> ,consulté le 1 Mars 2020
- [17] univ-bejaia,<http://www.univ-bejaia.dz/dspace/bitstream/handle/123456789/8269/Etude%20comparative%20des%20algorithmes%20génétiques%20multi-objectifs.pdf?sequence=1&isAllowed=y> ,consulté le 6 Mars 2020
- [22] eurobios, <http://www.eurobios.com/optimisation-de-la-chaîne-logistique> ,consulté le 5 février 2020
- [23] eurodecision,<https://www.eurodecision.com/offres/optimisation-logistique-et-supply-chain>,consulté le 5 février 2020
- [24] dl.ummt, <https://dl.ummt.dz/bitstream/handle/ummt/3367/Belhassani%2C%20Saliha.pdf?sequence=1&isAllowed=y> ,consulté le 6 Marche 2020

ملخص

في المجالات الصناعية كما هو الحال في الإنتاج، التنافسية الشديدة للمحيط التجاري تلزم الشركات لتنويع عروضها للإجابة على نحو أفضل لطلبات عملائها وإدارة هذا التنوع هو المشكل المركزي بالإضافة إلى أن التحسين في السلسلة اللوجستية أصبح واحد من القضايا الرئيسية للشركات في السنوات الأخيرة

في هذا العمل هدفنا هو تطوير أداة لتحليل هيكل سلسلة التوريد والتي تعتمد منهجيته على الإقتران بين نهجي المحاكاة والتحسين الذي يحتوي على وحدة التحسين ووحدة محاكاة. يتضمن هذا النهج وحدة تحسين تعتمد على الخوارزميات الجينية ووحدة محاكاة تسمح بتقييم السلسلة التي تمت دراستها أثناء التحسين

الكلمات الرئيسية: السلسلة اللوجستية، التحسين، المحاكاة، الخوارزميات الجينية

Résumé

Dans le secteur industriel comme celui de la production, l'environnement commercial très concurrentiel oblige les entreprises à diversifier leur offre de produits pour mieux répondre aux besoins de leurs clients. La gestion de cette diversité est alors une problématique centrale, ainsi que l'optimisation de la chaîne logistique est devenue un des enjeux majeurs des entreprises ces dernières années.

Notre objectif était de développer un outil d'analyse de la structure de la chaîne logistique et de son efficacité. L'outil est basée sur le couplage des deux approches la simulation et l'optimisation qui contient un module d'optimisation basée sur les algorithmes génétiques multicritères et un module de simulation permettant l'évaluation de la chaîne étudiée au cours de l'optimisation

Mots clés : La chaîne logistique, simulation, optimisation, les algorithmes génétiques.

Abstract

In the industrial environments as in the production, the very competing commercial context obliges the companies to diversify their offers for better answering the needs of their customers. Managing this diversity has become a central issue, as well as optimizing the supply chain became one of the major issues of the companies in recent years.

Our objective was to develop a tool to analyze the structure of the supply chain and its efficiency. This tool is based on the combination of the two approaches optimization and simulation it contains an optimization module based on genetic algorithms and a simulation module allowing the evaluation of the studied chain during the optimization.

Key words: Supply chain, simulation, optimization, Genetic algorithms.