الجـــــــــــــــــــمـهـوريـــــة الجـــــــــــــزائـريـة الديمـــــــقـراطيـة الـشـــــــــعبيـة

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التـــــــــــــــــــــــــــعـليـم العـــــــالي و البـــــــــحث العـــــلـمـــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جـــــــــــــــامعة أبي بكـــر بـلـقـايد – تـلمســـــــــــــان –

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Génie Industriel

Spécialité : Ingénierie des Systèmes

Par : LARBI Ghania
BELMENGAA Hamza

Sujet

Étude comparative des règles de priorité destinées à l'affectation des tâches dans un atelier flexible

Soutenu publiquement, le 25 / 06 / 2023 , devant le jury composé de :

Mme. Khedim Ouis Amaria	MCB	Université de Tlemcen	Président
Mme. Triqui Lamia	MCA	Université de Tlemcen	Examinateur 1
Mme. Abdellaoui Wassila	MCB	Université de Tlemcen	Examinateur 2
M. Hadri Abdelkader	MCB	Université de Tlemcen	Encadrant
Mme. Laribi Imane	MCB	Université de Tlemcen	Co- Encadrant

Année universitaire : 2022 /2023

Remerciement

Tout d’abord, nous remercions “ Dieu” le Tout Puissant de nous avoir donné le courage, la force et la patience pour faire aboutir ce travail.

Nous souhaitons exprimer nos sincères remerciements à nos respectueux encadrants, Mr. HADRI Abdelkaderet Mme. LARIBI Imane, pour leur précieuse contribution à notre projet. Leurs connaissances scientifiques, leurs orientations avisées, leurs conseils précieux et leur soutien intellectuel nous ont été d'une grande aide. Nous sommes également reconnaissants de leur présence constante et de leurs merveilleuses qualités humaines tout au long de ce travail.

Ces remerciements ne seraient pas complets sans une pensée pour nos parents, nos sœurs et nos frères. Merci de nous avoir encouragées et soutenues tout au long de ces années et de nous avoir permis de mener à bien nos études.

Enfin, nous tenons à exprimer nos sincères remerciements à l'Université Aboubakr Belkaïdet tous ceux qui nous ont aidés, directement ou indirectement, dans l'élaboration de ce travail.

Table des matières

Introduction générale 1
Chapitre I : L'ordonnancement de la production 3
I Introduction 3
II La production 3
II.1 Définitions 3
II.1.1 La production 3
II.1.2 Les systèmes de production 3
II.1.3 Les systèmes flexibles de production 5
II.2 La gestion de production 6
II.2.1 Définition 6
II.2.2 Le rôle de la gestion de production 6
II.2.3 Organisation hiérarchique de la gestion de production 7
L'ordonnancement de la production 7
III.1 Définition 7
III.2 Les éléments de problème d'ordonnancement 8
III.2.1 Les tâches 8
III.2.2 Les ressources 9
III.2.3 Les contraintes 9
III.2.4 Les objectifs 10
III.3 Typologie des problèmes d'ordonnancements 11
III.3.1 Modèles à une opération 11
III.3.1.1 Modèle à machine unique 11
III.3.1.2 Modèle à machines parallèles 11
III.3.2 Modèles à plusieurs opérations 12
III.3.2.1 Modèle flow-shop 12
III.3.2.2 Modèle job-shop 13
III.3.2.3 Modèle open-shop 13
III.4 Notation et classification des problèmes d'ordonnancement 14
III.5 Problème d'ordonnancement job shop 15
III.5.1 Définition 15
III.5.2 Représentations d'un problème d'ordonnancement job-shop 16
III.5.2.1 Le diagramme de Gantt 16
III.5.2.2 Le graphe Potentiel-Tâches 17

III.6. Caractéristiques générales des ordonnancements 18
III.6.1 Ordonnancement admissible 18
III.6.2 Ordonnancement semi-actif 19
III.6.3. Ordonnancement actif 19
III.6.4 Ordonnancements sans délais 20
Conclusion 20
Chapitre II : Les méthodes de résolutions 21
Introduction 21
Problème d'optimisation combinatoire 21
Complexité du système job shop 22
Classification des méthodes de résolutions 22
IV.1 Méthodes de résolution exactes 23
IV.1.1 Recherche exhaustive 23
IV.1.2 Programmation mathématique 23
IV.1.3 L'algorithme de retour arrière (Backtracking) 24
V.1.4 La méthode Branch and Bound (B&B) 25
IV.1.5 Méthodes basées sur les graphes de précédence 26
IV.2 Méthodes approchées 26
IV.2.1 Heuristiques basées sur des règles de priorité 26
IV.2.1.1 Shortest Processing Time (SPT) 27
IV.2.1.2 Shortest Remaining Processing Time (SRPT) 27
IV.2.1.3 Earliest Due Date (EDD) 27
IV.2.1.4 Longest Processing Time (LPT) 27
IV.2.1.5 First Come, First Solved (FCFS) 28
IV.2.1.6 Les avantages des heuristiques pour la résolution des problèmes d'ordonnancement 28
IV.2.2 Méta-heuristiques 29
IV.2.2.1 Les méta-heuristiques à base de solution unique 29
IV.2.2.2 Les méta-heuristiques à base de population de solutions 33
IV.2.3 Approches basées sur l'apprentissage et l'intelligence artificielle 38
IV.2.4 Les avantages et les inconvénients des méthodes approchées 39
Conclusion 40
Chapitre III : Étude comparative des règles de priorité destinées à l'affectation des tâches dans un atelier flexible de type job shop 41
III.1 Introduction 41
III.2 Etat de l'art : Les règles de priorité dans le contexte du job shop 41

III.3 Méthodologie : Étude comparative des règles de priorité appliquées sur le job shop 43
III.3.1 Description du problème et notations 43
III.3.1.1 Caractéristiques du système de job shop étudié 43
III.3.1.2 Contraintes spécifiques au problème 45
III.3.1.3 Fonction objectif 46
III.3.2 Présentation des règles de priorité 46
III.3.2.1 Shortest Total Processing Time (SPTsystème) 47
III.3.2.2 Longest Total Processing Time (LPTsystème) 47
III.3.2.3 Shortest Processing Time in machine (SPTmachine) 47
III.3.2.4 Longest Processing Time in machine (LPTmachine) 47
III.3.2.5 Short Accumulation Processing Time (SPTcumule) 47
III.3.2.6 Earliest Due Date (EDD) 48
III.3.2.7 Slack Time Remaining (STR) 48
III.3.2.8 Critical Ratio (CR) 48
III.3.2.9 La règle proposée Operating Range Sequence (ORS) 48
III.3.3 Méthodes expérimentales et mesures de performance 54
III.3.3.1 Résultats obtenus pour les petites instances 54
III.3.3.2 Résultats obtenus pour les moyennes instances 55
III.3.3.3 Résultats obtenus pour les grandes instances 55
III.3.3.4 Comparaison entre la règle ORS et la méthode exacte 56
IV. Conclusions de l'étude comparative 57
Conclusion générale 59

Liste des figures

Chapitre 01

Figure I.1 Décomposition d'un système de production. 4
Figure I.2 Système flexible de production. 5
Figure 1.3 Les activités de la gestion de production. 6
Figure I.4 Caractéristique d'une tâche. 9
Figure I.5 Modèle à machine unique. 11
Figure I.6 Modèle à machines parallèles. 12
Figure I.7 Modèle flow-shop. 12
Figure I.8 Modèle job-shop. 13
Figure I.9 Modèle open-shop. 14
Figure I.10 Diagramme de Gantt ressource pour un problème 4x3. 17
Figure I.11 Graphe Potentiel-Tâches d'un ordonnancement. 18
Figure I.12 Diagramme de Venn. 18
Figure I.13 Ordonnancement admissible. 19
Figure I.14 Ordonnancement admissible semi actif. 19
Figure I.15 Ordonnancement admissible actif. 20
Figure I.16 Ordonnancement sans délai. 20

Chapitre 02

Figure II.1 Des méthodes de résolutions de problèmes d'optimisation 23
Figure II.2 Arbre de recherche 24
Figure II.3 Déroulement de l'algorithme Branch and Bound 26
Figure II.4 Classification des méta-heuristiques 29
Figure II.5 Version la plus simple du recuit simulé 30
Figure II.6 Organigramme de l'algorithme tabou simple 32
Figure II.7 Organigramme d'un algorithme génétique 35
Figure II.8 Cycle de vie du système de colonies des fourmis 38

Chapitre 03

Figure III.1 La configuration 3D du système iCIM 3000. 44
Figure III.2 Sens de déplacement des palettes dans le système iCIM 3000. 45
Figure III.3 Diagramme de Gantt selon la règle SPT système. 48
Figure III.4 Diagramme de Gantt selon la règle LPT système. 50
Figure III.5 Diagramme de Gantt selon la règle SPT machine. 50

Figure III.6 Diagramme de Gantt selon la règle SPT cumule. 51
Figure III.7 Diagramme de Gantt selon la règle EDD. 51
Figure III.8 Diagramme de Gantt selon la règle STR. 52
Figure III.9 Diagramme de Gantt selon la règle CR. 53
Figure III.10 Diagramme de Gantt selon la règle ORS. 54
Figure III.11 Les résultats de Cmax obtenu par la règle ORS et le modèle CPLEX (3x4). 57
Figure III.12 Les résultats de Cmax obtenu par la règle ORS et le modèle CPLEX (4x4). 57

Liste des tableaux

Chapitre 01

Tableau I.1 Les paramètres de problème d'ordonnancement. 14
Tableau I.2 La représentation de la gamme opératoire des jobs dans un job-shop. 16

Chapitre 03

Tableau III.1 Temps opératoires sur chaque machine. 49
Tableau III.2 La gamme opératoire des jobs. 49
Tableau III.3 La séquence des jobs dans les machines selon SPT système. 49
Tableau III.4 La séquence des jobs dans les machines selon LPT système. 50
Tableau III.5 La séquence des jobs dans les machines selon SPT machine. 50
Tableau III.6 La séquence des jobs dans les machines selon LPT machine. 51
Tableau III.7 La séquence des jobs dans les machines selon SPT cumule. 51
Tableau III.8 La date d'échéance des jobs. 51
Tableau III.9 La séquence des jobs dans les machines selon EDD. 52
Tableau III.10 Marge de temps disponible des jobs. 52
Tableau III.11 La séquence des jobs dans les machines selon STR. 52
Tableau III.12 Ratio Critique des jobs. 53
Tableau III.13 La séquence des jobs dans les machines selon CR. 53
Tableau III.14 La séquence des jobs dans les machines selon ORS. 53
Tableau III.15 Les résultats de Cmax obtenu par MATLAB pour les petites instances. 55
Tableau III.16 Les résultats de Cmax obtenu par MATLAB pour les moyennes instances. 55
Tableau III.17 Les résultats de Cmax obtenu par MATLAB pour les grandes instances. 56

Introduction générale

Les entreprises industrielles sont généralement associées à la production en masse de biens standardisés. Cependant, ces dernières années, une tendance croissante vers la personnalisation des produits a émergé, ouvrant de nouvelles opportunités pour les entreprises industrielles. La personnalisation des produits permet aux entreprises de répondre aux besoins spécifiques et aux préférences uniques des clients, offrant ainsi des avantages concurrentiels significatifs.

Dans un contexte marqué par l'ouverture des marchés internationaux, l'évolution rapide des technologies et la mondialisation, les entreprises industrielles se tournent de plus en plus vers des systèmes de production flexibles. Cela remet en question les pratiques de production établies, en particulier la gestion des ateliers de production, qui joue un rôle crucial dans la productivité et la réduction des délais de production.

L'ordonnancement des opérations à un impact direct sur la productivité d'un atelier de production. En effet, une meilleure utilisation des ressources permet à l'atelier de réaliser une grande variété de produits tout en réduisant les coûts. L'ordonnancement joue donc un rôle essentiel dans l'optimisation des processus de production et la maximisation de l'efficacité.

Le domaine d'application de l'ordonnancement est vaste et diversifié. Il est utilisé dans la gestion de la charge des processus en informatique, la gestion de la production dans l'industrie manufacturière, la gestion de projets et bien d'autres domaines. L'objectif est toujours le même : optimiser l'allocation des ressources et planifier les opérations de manière à minimiser les temps d'attente, les retards et les coûts, tout en maximisant la productivité et la qualité des produits.

L'ordonnancement dans un environnement job shop est un défi complexe et crucial pour les entreprises industrielles. Le problème d'ordonnancement job shop se caractérise par la présence de plusieurs machines et plusieurs jobs, où chaque job doit passer par une séquence spécifique de machines dans un ordre donné.

L'ordonnancement job shop est un problème complexe pour lequel plusieurs méthodes de résolution ont été développées. Ces méthodes se divisent généralement en deux catégories : les méthodes exactes, qui visent à fournir une solution optimale, et les méthodes approchées, qui offrent des solutions proches de l'optimum. Parmi les méthodes approchées, les règles de priorité ont démontré leur efficacité dans de nombreux domaines.

Dans ce mémoire, nous nous intéressons particulièrement à l'application des règles de priorité dans le contexte spécifique du problème job shop. L'objectif est de déterminer quelle règle de priorité offre les meilleures solutions pour ce type de système d'ordonnancement.

La question qui se pose alors est la suivante : comment appliquer ces règles de priorité au problème job shop et quelle est la règle qui donne les résultats les plus performants ? Cette question revêt une importance cruciale, car la sélection d'une règle de priorité appropriée peut avoir un impact significatif sur l'efficacité et la qualité des solutions obtenues.

Ainsi, ce mémoire se propose de réaliser une étude comparative approfondie des règles de priorité appliquées à un système job shop. Nous examinerons plusieurs règles de priorité existantes, en analysant leurs performances respectives en termes de temps d'exécution des tâches, de délais et d'utilisation optimale des ressources.

L'objectif final de cette étude comparative est de fournir aux décideurs et aux gestionnaires des outils d'analyse et d'aide à la décision pour sélectionner la règle de priorité la plus appropriée en fonction des objectifs et des contraintes spécifiques. En identifiant la règle de priorité la plus performante, nous pourrons contribuer à l'amélioration de l'efficacité des opérations de production dans un contexte job shop.

Ce mémoire est organisé en trois chapitres :

Le premier chapitre débute en présentant brièvement les concepts de systèmes de production et de gestion de production. Ensuite, nous donnons des définitions générales de l'ordonnancement. Au fur et à mesure de l'avancement du chapitre, nous examinons de manière approfondie les divers types d'ateliers et étudions les particularités des ordonnancements qui leur sont associés.

Le deuxième chapitre se focalise sur les diverses approches de résolution décrites dans les travaux scientifiques. Nous examinons à la fois les techniques de résolution précises qui cherchent à obtenir une solution optimale, ainsi que les méthodes de résolution approximatives qui fournissent des solutions de qualité acceptable dans des délais raisonnables.

Le troisième et dernier chapitre constitue le cœur de notre étude comparative. Notre objectif principal est d'évaluer et de comparer les performances des règles de priorité couramment utilisées dans les systèmes job shop. Nous présentons les résultats obtenus après l'application de ces règles sur plusieurs exemples représentatifs. Ensuite, nous comparons les performances de la meilleure règle identifiée lors de notre étude avec les solutions obtenues par des méthodes de résolution exactes. Cette comparaison nous permet de mesurer l'écart entre les solutions obtenues par les règles de priorité et les solutions optimales.

Chapitre I : L’ordonnancement de la production

I Introduction

La gestion des systèmes de production, qu'ils soient axés sur la production de biens ou de services, peut être confrontée à divers défis en termes de gestion de la production, de commercialisation et de gestion des ressources humaines. Pour résoudre ces problèmes, il est souvent nécessaire de recourir à des techniques d'optimisation ou d'évaluation des performances provenant de différentes disciplines.

Dans ce chapitre, nous nous concentrerons sur un aspect particulier de la gestion de production, à savoir l'ordonnancement de la production. Plus précisément, nous étudierons les problèmes d'atelier. Nous avons présenté les notions clés de la gestion de production et les systèmes flexibles de production. Nous avons ensuite fourni des définitions générales de l'ordonnancement, nous avons également abordée les différents types de problèmes d'ordonnancement d'atelier.

II La production

II.1 Définitions

II.1.1 La production

La production peut être définie comme le processus de création d’un bien ou d’un service, apte à satisfaire une demande, à l’aide de facteurs de production acquis sur le marché [02].

Selon Giard [1], la production est vue comme le processus conduisant à la création de produits par l'utilisation et la transformation de ressources.

II.1.2 Les systèmes de production

Un système de production (ou de fabrication) est un ensemble d'éléments interconnectés qui travaillent ensemble pour transformer des ressources (matières premières, main-d'œuvre, équipements, etc.) en produits ou services finis. Il englobe les méthodes, les processus, les technologies, les équipements et les flux d'activités nécessaires pour réaliser la production.

Les systèmes de production peuvent être divisés en trois sous-systèmes, à savoir le système physique de production, le système de prise de décision et le système d'information, comme présenté dans la figure I.1. Cette décomposition est organisée en fonction des différents flux qui traversent chaque système.

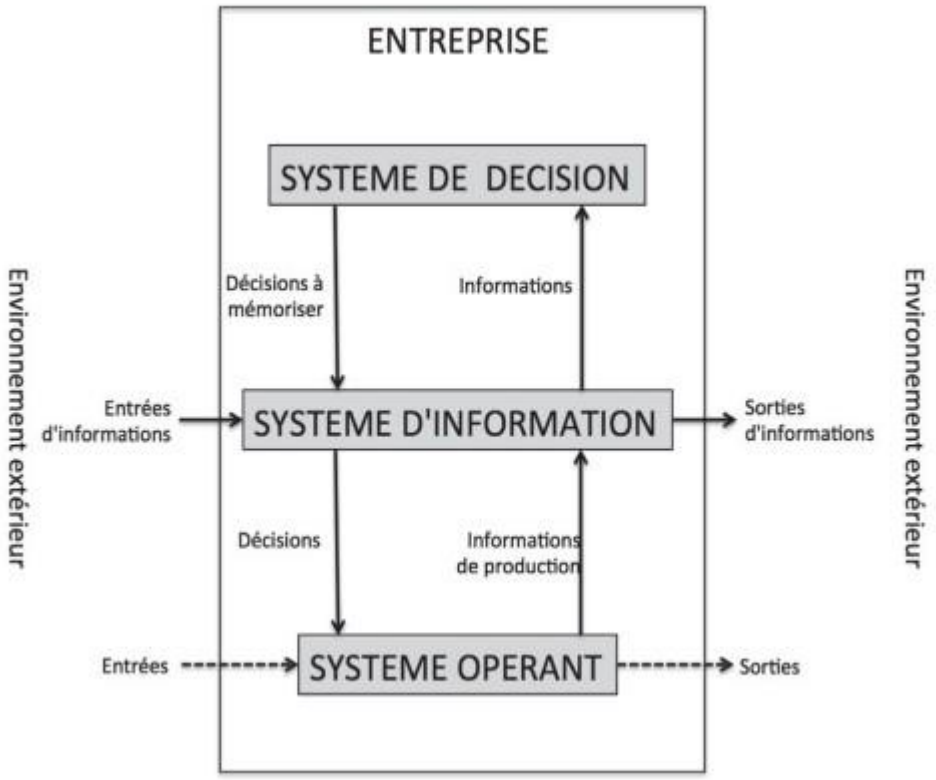


Figure I.1: Décomposition d'un système de production [07]

- **Le système physique de production (système opérant) :** il est également nommé système opératoire ou système de production, ce système réalise la production physique des biens et des services. Son activité est contrôlée par le système de décision. Il est relié à l'environnement par des flux physiques externes et aux autres sous-systèmes par des flux internes d'information.
- **Le système de décision :** également appelé système de pilotage ou système de management. Ce dernier, finalise l'entreprise en lui assignant ses objectifs. Il analyse l'environnement et le fonctionnement interne de l'entreprise. Il assure le contrôle des tâches et la régulation du système. Il est relié aux autres sous-systèmes par des flux internes d'information.
- **Le système d'information :** ce système joue un rôle central puisqu'il alimente l'entreprise en informations. Pour cela, il mémorise les informations, les traite et les communique aux deux autres sous-systèmes auxquels il est relié. Toutes les informations de l'entreprise, d'origine externe ou interne, passe donc par le Système d'Information [07].

II.1.3 Les systèmes flexibles de production

Les systèmes flexibles de production (SFP ou FMS en anglais), sont des systèmes de fabrication automatisés conçus pour produire une variété de produits différents avec un minimum de temps d'arrêt et de coûts de réglage. Les SFP sont souvent utilisées dans des environnements de production où il y a une demande variable pour les produits ou où il est nécessaire de produire des produits personnalisés.

Les SFP intègrent différents équipements automatisés tels que des robots, des machines-outils, des convoyeurs, des systèmes de stockage et de récupération automatisés, ainsi que des logiciels de planification et de contrôle centralisés pour coordonner le flux de matériaux et d'informations à travers le système.

Le SFP est un ensemble de machines interconnectées par un système de transport. Ce système sert à transporter les pièces (produits) vers les machines sur des palettes ou d'autres unités d'interface de telle sorte que la synchronisation des pièces-machines est précise, rapide et automatique. Dans cette catégorie, un ordinateur central contrôle à la fois les machines et le système de transport [08].

La figure ci-dessous présente un exemple d'un SFP avec six postes de travail. Ce système est destiné à la fabrication des bouteilles. Selon le type des opérations nécessaires à la réalisation des bouteilles, les six stations sont destinées chacune, à accomplir une fonction bien déterminée.

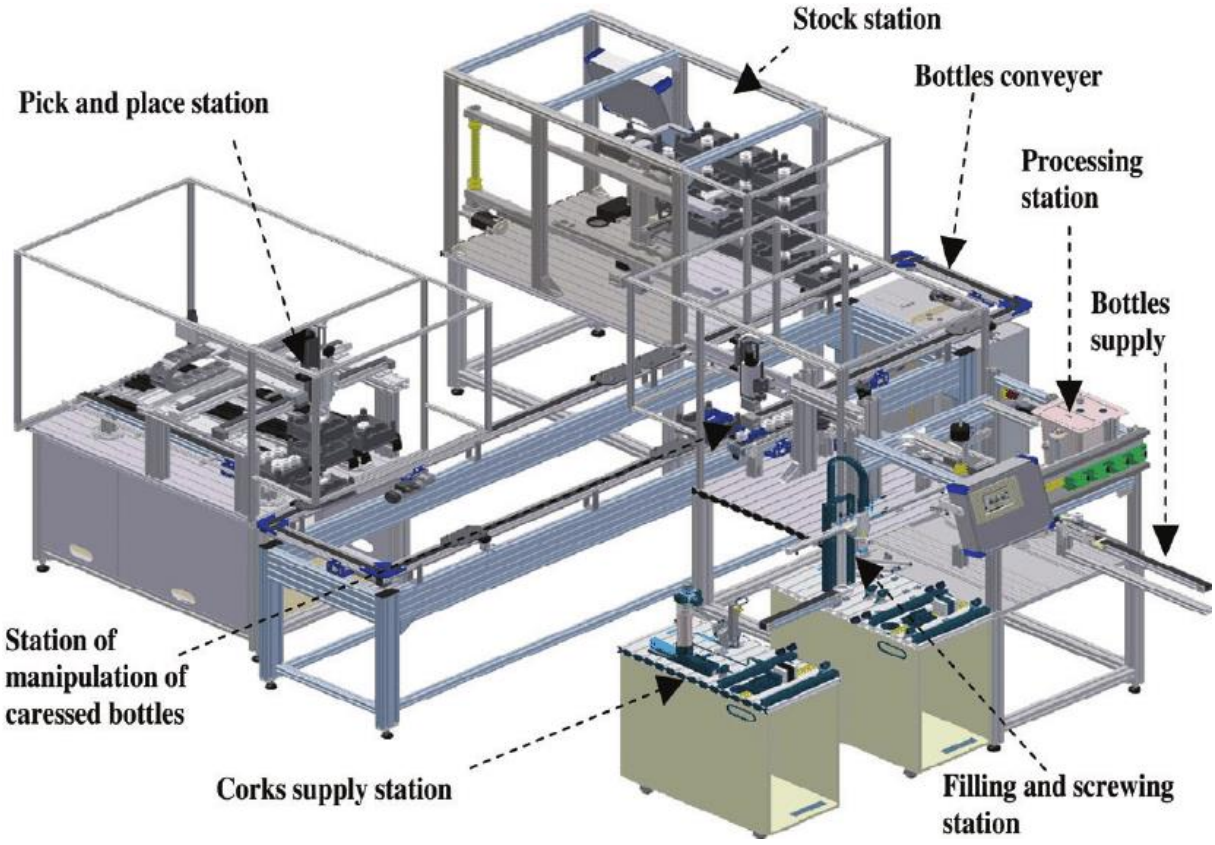


Figure I.2: Système flexible de production [14]

II.2 La gestion de production

II.2.1 Définition

La gestion de production est un ensemble de processus et de techniques qui visent à planifier, organiser, diriger et contrôler les activités de production d'une entreprise. Elle vise à optimiser l'utilisation des ressources, notamment les matières premières, les machines et la main-d'œuvre, pour atteindre les objectifs de production dans les délais impartis et avec un niveau de qualité optimal.

II.2.2 Le rôle de la gestion de production

La gestion de la production implique l'organisation du système de production afin de fabriquer les produits dans les quantités et délais souhaités, en tenant compte des ressources humaines et/ou technologiques disponibles.

La gestion de la production comprend diverses activités, comme le montre la figure I.3.

- **Gestion des données commerciales :**
  - Recevoir les commandes et établir les calendriers de livraison souhaités.
- **Gestion des données techniques :**
  - Décrire les produits et les familles de produits (nomenclatures).
  - Décrire les processus de fabrication (gammes).
- **Gestion des matières (gestion de stocks) :**
  - Assurer le stockage de produits fabriqués.
  - Assurer l'approvisionnement en matières premières ou en composants.
- **Gestion du travail :**
  - Planifier la réalisation des tâches dans le temps en allouant les ressources nécessaires. Cela prend en compte les données techniques, commerciales et celles du suivi de la production (quantités déjà fabriquées, état des ressources, etc.).

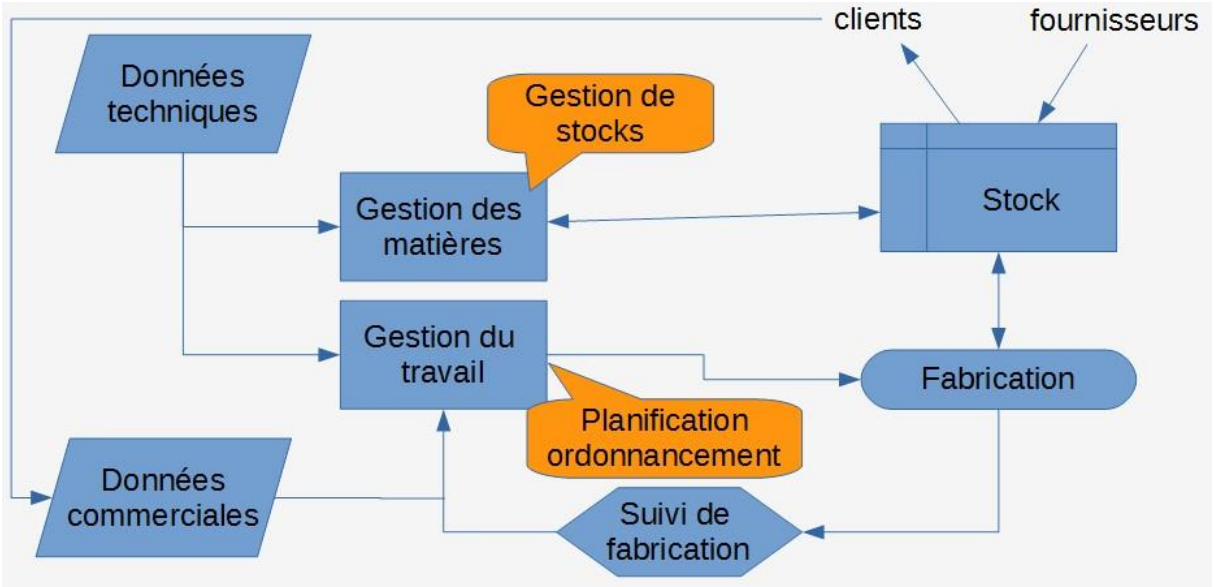


Figure I.3: Les activités de la gestion de production [01]

II.2.3 Organisation hiérarchique de la gestion de production

La gestion de la production est généralement organisée selon une structure hiérarchique comprenant trois échelles distinctes : l'échelle stratégique, l'échelle tactique et l'échelle opérationnelle.

- **L'échelle stratégique :**
  Au niveau stratégique, les décisions sont prises à long terme et concernent les objectifs globaux de l'entreprise en matière de production. Cela inclut la définition des orientations stratégiques, telles que le choix des marchés cibles, l'investissement dans de nouvelles technologies, l'optimisation des ressources à long terme, etc.

- **L'échelle tactique :**
  Le niveau tactique se situe entre le niveau stratégique et le niveau opérationnel. Il concerne la mise en œuvre des stratégies définies au niveau supérieur. Les décisions prises à ce niveau sont de portée moyenne et visent à optimiser l'utilisation des ressources disponibles. Cela comprend la planification de la production, l'affectation des ressources, la gestion des capacités, l'équilibrage des charges de travail, etc.

- **L'échelle opérationnelle :**
  Le niveau opérationnel est le niveau de gestion le plus bas, où les activités de production sont exécutées au jour le jour. Les décisions prises à ce niveau sont concrètes et immédiates, et visent à assurer le bon déroulement des opérations de production. Cela comprend la gestion des stocks, la coordination des flux de travail, l'ordonnancement des tâches, le contrôle de la qualité, etc.

  En résumé, la gestion de la production est organisée hiérarchiquement, avec le niveau stratégique se concentrant sur les objectifs à long terme, le niveau tactique sur la planification et l'optimisation des ressources, et le niveau opérationnel sur l'exécution quotidienne des activités de production.

L'ordonnancement de la production

III.1 Définition

Il existe de nombreuses définitions proposées pour le problème d'ordonnancement d'atelier. Nous avons sélectionné trois définitions dans la littérature, que nous présentons ci-dessous :

- **Definition 1 [10]**

L'ordonnancement est le processus d'organisation, de choix et de synchronisation de l'utilisation des ressources pour effectuer toutes les activités nécessaires à la production des résultats souhaités aux moments souhaités, tout en satisfaisant un grand nombre de contraintes temporelles et relationnelles entre les activités et les ressources.

- **Definition 2 [11]**

L'ordonnancement concerne l'allocation de ressources rares aux activités dans le but d'optimiser une ou plusieurs mesures de performance.

- **Definition 3 [12]**

L'ordonnancement est un processus de prise de décision qui est utilisé régulièrement dans de nombreuses industries manufacturières et de services. Il traite de l'allocation des ressources aux tâches sur des périodes données et son objectif est d'optimiser un ou plusieurs objectifs.

D'après ces définitions, nous pouvons identifier le point commun de l'allocation de ressources aux tâches. Ainsi, nous pouvons dire que l'ordonnancement d'ateliers consiste à planifier dans le temps l'exécution des tâches en fonction de la disponibilité des ressources, afin d'atteindre un ou plusieurs objectifs, tout en respectant les contraintes techniques de fabrication [13].

III.2 Les éléments de problème d'ordonnancement

Les éléments clés constituant un problème d'ordonnancement incluent :

III.2.1 Les tâches

Une tâche est une unité de travail élémentaire qui est définie dans le temps par une date de début t_i et une date de fin c_i, La tâche a une durée de traitement p_i, qui représente le temps nécessaire pour l'accomplir.

Habituellement, les symboles utilisés pour décrire les caractéristiques d'une tâche, notée i, sont les suivants :

- Une date de disponibilité r_i : l'exécution de la tâche *i* ne peut pas débuter avant cette date.
- Une date échue notée d_i : la tâche *i* doit être achevée avant cette date.
- Un poids w_i : représente le facteur de priorité qui dénote l'importance de la tâche *i* relativement aux autres.

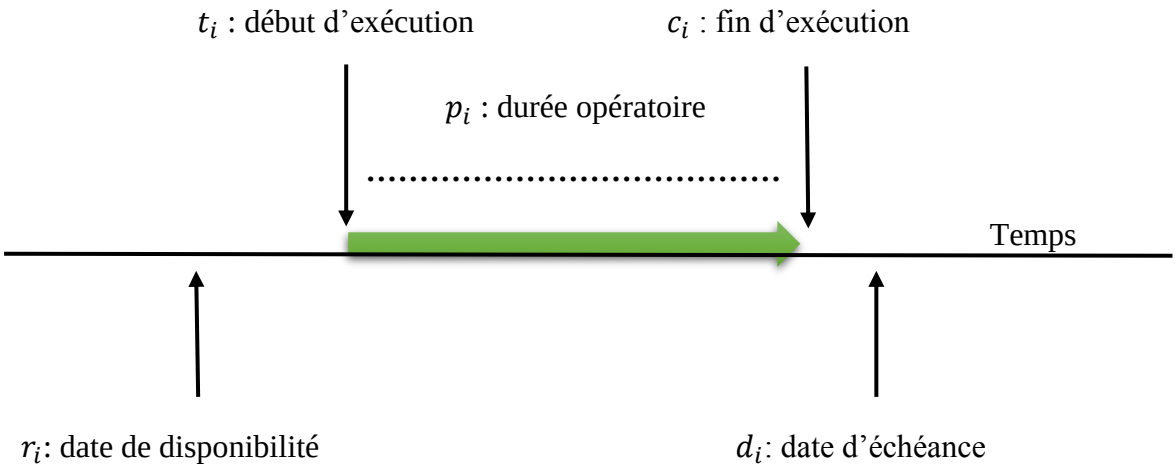


Figure I.4: Caractéristique d'une tâche

III.2.2 Les ressources

Une ressource est un moyen technique et/ou humain utilisé pour accomplir une tâche. Dans un système de production, plusieurs types de ressources sont identifiés [13] [15].

- **Ressources consommables :** Les ressources sont considérées comme consommables lorsque leur disponibilité diminue après avoir été allouées à une tâche. C'est le cas des matières premières, du budget, etc.
- **Ressources renouvelables :** Ce sont des ressources qui, une fois utilisées par une ou plusieurs tâches, sont à nouveau disponibles en quantités identiques (personnel, machines, espace, équipement, etc.).
- **Ressources à double contrainte :** Ces ressources ont à la fois une utilisation instantanée limitée et une consommation totale limitée (sources d'énergie, financement, etc.).
- **Ressources cumulatives (ou partageables) :** Ce sont des ressources qui peuvent être utilisées simultanément par plusieurs tâches (équipe d'ouvriers, poste de travail, etc.).
- **Ressources disjonctives (ou non-partageables) :** Ce sont des ressources qui ne peuvent exécuter qu'une seule tâche à la fois (machine-outil, robot manipulateur, etc.).

III.2.3 Les contraintes

Les contraintes représentent les limitations sur les valeurs que certaines variables peuvent prendre. Dans les problèmes d'ordonnancement, on distingue généralement deux types de contraintes : les contraintes de ressources et les contraintes temporelles.

- **Les contraintes de ressources :** Ces contraintes sont généralement liées à la capacité et/ou à la disponibilité des ressources. En termes de capacité des ressources, on distingue deux types de contraintes en fonction de la nature des ressources utilisées : les contraintes de ressources disjointes où les ressources ne peuvent être utilisées que par une seule tâche à la fois, et les contraintes de ressources cumulatives où les ressources peuvent être utilisées par plusieurs tâches simultanément [17] [18]. En ce qui concerne la disponibilité, les contraintes de ressources sont liées à la présence des ressources au moment de leur utilisation (par exemple, pour les ressources consommables) et/ou à leur disponibilité (c'est-à-dire l'absence de panne au moment de leur utilisation).

- **Les contraintes temporelles :** Ces contraintes incluent les restrictions sur le temps alloué aux tâches (comme les délais de livraison), la durée totale de l'ordonnancement, les contraintes de précédence ou de dépendance entre les tâches, les contraintes de calendrier liées aux plages horaires de travail, etc. [09].

III.2.4 Les objectifs

La résolution de tout problème d'ordonnancement implique généralement la recherche d'un critère à minimiser ou à maximiser, en améliorant au moins l'un des trois facteurs suivants : coût, qualité ou délais. En effet, plusieurs catégories d'objectifs liés à l'ordonnancement sont identifiées [17].

- **Objectifs liés aux ressources :** Les objectifs de cette catégorie comprennent la maximisation de la charge d'une ressource, la minimisation de la consommation des ressources consommables ou la réduction du nombre de ressources nécessaires pour effectuer un ensemble de tâches.
- **Objectifs liés au temps :** Ces objectifs incluent la minimisation du temps total d'exécution (makespan), la minimisation du temps moyen d'achèvement des tâches, la minimisation des retards par rapport aux dates de livraison, etc.
- **Les objectifs liés au coût :** Ces objectifs visent généralement à minimiser les coûts de lancement, de production, de stockage, de transport, de consommation d'énergie, etc.

Les objectifs (critères d'évaluation) peuvent être classés en critères réguliers et en critères irréguliers :

- **Critères réguliers :** Un critère est considéré comme régulier s'il est une fonction décroissante des dates de fin d'exécution des opérations. Par exemple,

la durée totale de fabrication d'un produit, la minimisation du retard maximum par rapport aux dates d'achèvement, la minimisation de la moyenne des retards par rapport aux dates d'achèvement [15].

- **Critères irréguliers :** Ces critères ne sont pas des fonctions monotones des dates de fin d'exécution des opérations. Parmi ces types de critères, on trouve la minimisation des en-cours, la minimisation du coût de stockage des matières premières, les délais de fabrication, les avances, les retards et l'équilibrage des charges des machines.

III.3 Typologie des problèmes d'ordonnancements

Les problèmes d'ordonnancement sont généralement classés en deux principaux modèles dépendamment du nombre d'opérations que requièrent les jobs : des modèles à une opération (machine unique et machines parallèles) et des modèles à plusieurs opérations (flow-shop, open-shop et job-shop).

III.3.1 Modèles à une opération

III.3.1.1 Modèle à machine unique

Dans un modèle à machine unique, toutes les tâches à réaliser sont exécutées par une seule machine. Ce type de configuration est souvent rencontré dans les systèmes de production où une machine spécifique, appelée goulot, exerce une influence significative sur l'ensemble du processus. Cette configuration est illustrée dans la Figure I.5. [20]

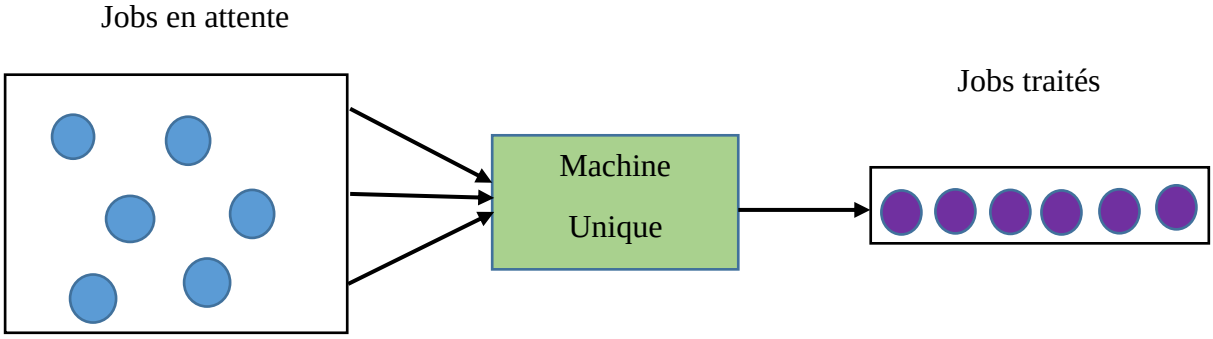


Figure I.5 : Modèle à machine unique

III.3.1.2 Modèle à machines parallèles

Le modèle de production à machines parallèles implique l'utilisation simultanée de plusieurs machines pour effectuer des tâches, ce qui permet d'augmenter la capacité de production et d'accélérer le processus global. Les tâches sont attribuées aux machines en fonction de leurs caractéristiques, telles que la durée de traitement et la disponibilité, dans le but d'optimiser l'utilisation des machines et de réduire les temps d'attente. Ce modèle est largement utilisé dans des industries telles que la fabrication, l'automobile et l'électronique, où une production rapide et efficace est essentielle. Des techniques d'ordonnancement avancées sont appliquées pour optimiser les performances globales du système de production en planifiant et en gérant l'allocation des tâches aux machines parallèles.

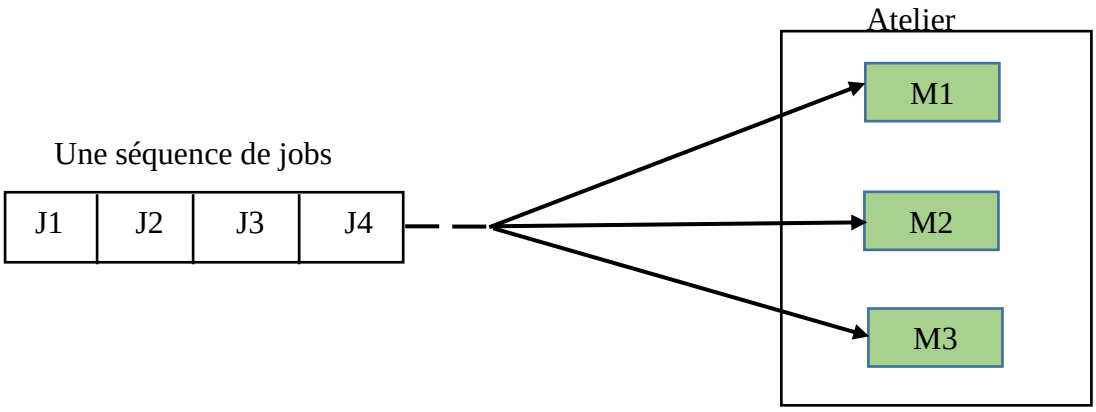


Figure I.6 : Modèle à machines parallèles

III.3.2 Modèles à plusieurs opérations

Le modèle à plusieurs opérations concerne les situations où une tâche doit passer par plusieurs machines avec des caractéristiques spécifiques pour être réalisée. On identifie trois modèles en fonction de l'ordre de passage des tâches sur les machines : le modèle de flow-shop, le modèle de job-shop et le modèle d'open-shop.

III.3.2.1 Modèle flow-shop

Le modèle de production flow-shop implique un ordre fixe des tâches lorsqu'elles passent par une séquence de machines. Chaque tâche est traitée successivement par chaque machine selon une séquence préétablie, où chaque machine effectue une opération spécifique sur la tâche avant de la transférer à la machine suivante. Ce modèle est largement utilisé dans des industries telles que l'automobile, l'électronique et la production en série, où une séquence bien définie est cruciale pour assurer la qualité et l'efficacité de la production. Des techniques d'ordonnancement sont appliquées pour déterminer l'ordre optimal des tâches sur les machines, réduisant ainsi les temps d'attente et optimisant les performances globales du système de production flow-shop.

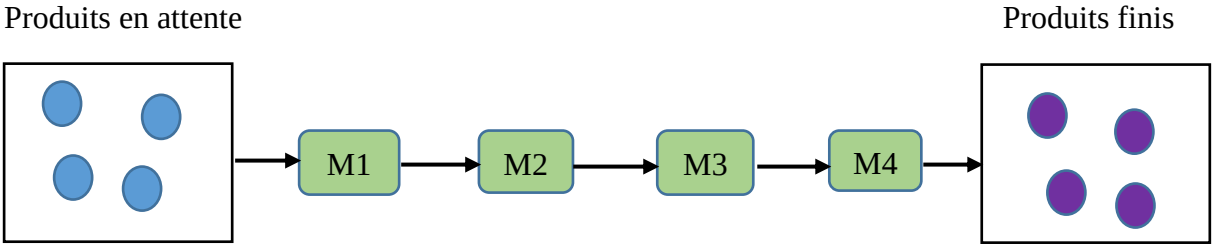


Figure I.7 : Modèle flow-shop

III.3.2.2 Modèle job-shop

L’atelier job shop (ou l’atelier à cheminements multiples) est un atelier dans lequel les jobs sont réalisés selon des gammes opératoires spécifiques. C’est-à-dire que le passage de chaque job à travers les machines n’est pas le même pour tous les produits comme le montre la figure I.8.

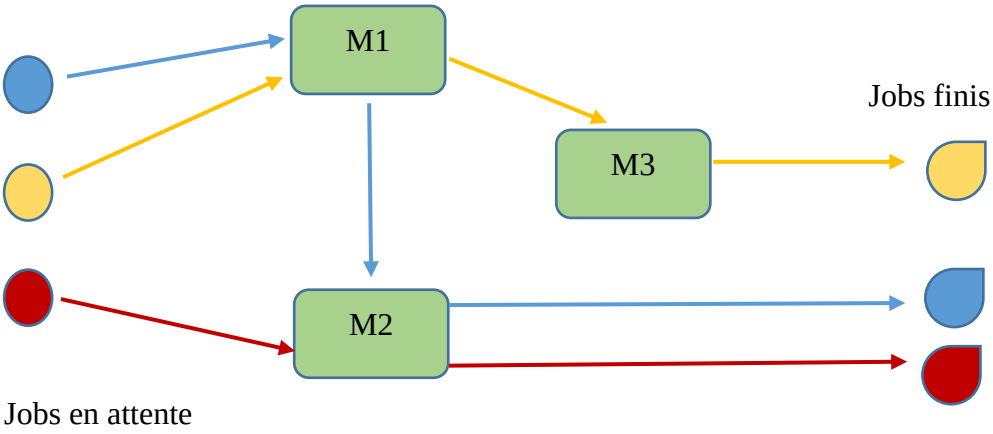


Figure I.8 : Modèle job-shop

III.3.2.3 Modèle open-shop

L'open shop est un modèle dans lequel les opérations peuvent être effectuées dans n'importe quel ordre, sans qu'il y ait de contraintes spécifiques concernant la séquence des tâches à effectuer. Dans un environnement open shop, les ressources (machines, travailleurs, équipements) sont généralement flexibles et peuvent être affectées à différentes tâches selon les besoins.

Contrairement aux modèles de production tels que le flow-shop ou le job-shop, où l'ordre des opérations est prédéfini, l'open shop offre une plus grande liberté en permettant la planification et l'exécution des tâches de manière plus flexible. Cela signifie qu'il n'y a pas de contraintes strictes sur la séquence d'exécution des différentes opérations.

Dans la figure I.9, nous pouvons observer un ensemble de quatre jobs et quatre machines. À droite de la figure, il est évident que chaque job a suivi un ordre de passage distinct sur les quatre machines.

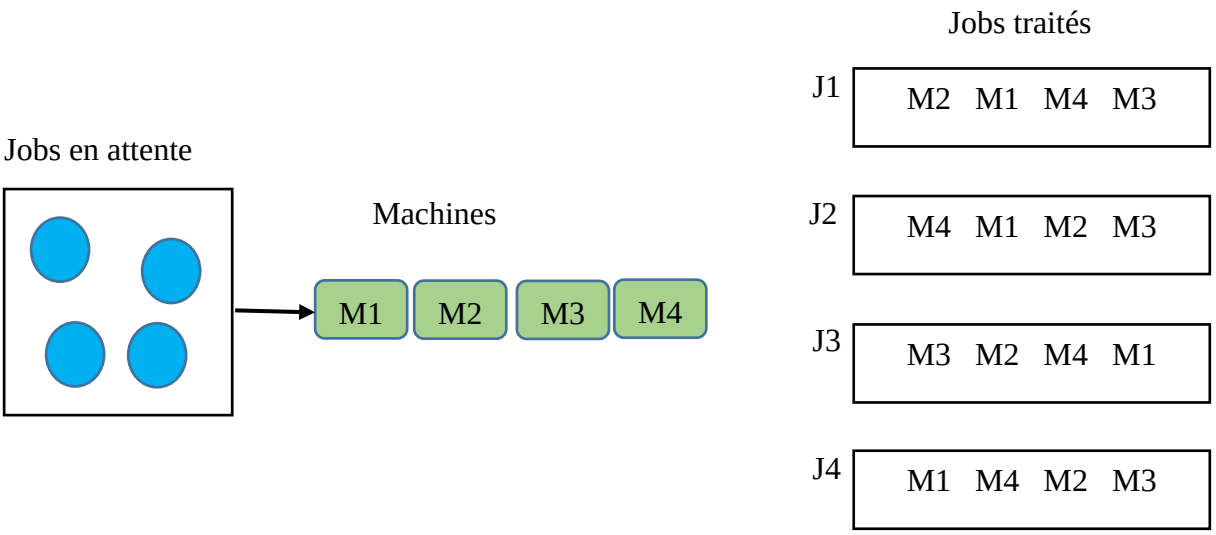


Figure I.9 : Modèle open-shop

III.4 Notation et classification des problèmes d'ordonnancement

La classification standard la plus populaire est la classification proposée par Graham et al. [21]. Dans cette classification, les problèmes d'ordonnancement sont distingués par la notation $\boldsymbol{\alpha|\beta|\gamma}$, où le premier paramètre ($\boldsymbol{\alpha}$) décrit l'organisation de l'atelier et le nombre de machines lequel il est constitué. Le deuxième paramètre ($\boldsymbol{\beta}$) définit les caractéristiques des travaux à réaliser et les contraintes associées. Le dernier paramètre ($\boldsymbol{\gamma}$) représente le critère d'optimisation sur lequel l'ordonnancent sera évalué. Cette notation permet de présenter de manière précise le problème d'ordonnancement à traiter. Le tableau suivant décrit les principales valeurs de ces trois paramètres [09].

Paramètres (Champ)	**Notation**	**Description**
$\boldsymbol{\alpha 1}$	1	Problème à une seule machine
	P	Problème à machines parallèles identiques
	Q	Problème à machines parallèles uniformes
	R	Problème à machines parallèles indépendantes
	F	Flow-Shop
	J	Job-Shop
	O	Open-Shop
	FH	Flow-Shop hybride
	JF	Job-Shop flexible
	OG	Open-Shop généralisé

$\alpha 2$	$\{m\}$	Problème à m machines
β	pmtn	La préemption des opérations est autorisée.
	prec	Existence des contraintes de précédence entre les opérations.
	res	L'opération nécessite l'emploi d'une ou plusieurs ressources supplémentaires.
	nwt	Les opérations de chaque job doivent se succéder sans attente.
	$p_i = p$	Les temps d'exécution des tâches sont identiques et égaux à *p*.
	r_i	Une date de début au plus tôt est associée à chaque job *i*.
	d_i	Une date d'échéance est associée à chaque job *i*.
γ	$C_{max} = max\ C_i$	La durée totale de l'ordonnancement
	$\Sigma\ w_i\ C_i$	La date de fin pondérée des travaux
	$T_{max} = max\ (C_i - d_i, 0)$	Le retard maximum des travaux
	$\Sigma\ w_i\ T_i$	Le retard pondéré des travaux
	$\Sigma\ T_i$	Le retard total
	$L_{max} = max\ (C_i - d_i)$	Le retard algébrique maximum des travaux
	$\Sigma\ w_i\ L_i$	Le retard algébrique pondéré des travaux
	$\Sigma\|L_i\|$	Le retard total absolu
	$\Sigma\ U_i = \|J_i\|$ si $C_i > d_i$	Le nombre de travaux en retard

Tableau I.1 : Les paramètres de problème d'ordonnancement

Exemples :

- $F_2||C_{max}$ dénote un problème d'ordonnancement d'un atelier de type Flow-Shop à deux machines sans contraintes avec minimisation de makespan.

- $P_m|pmtn|L_{max}$ désigne le problème de la minimisation du retard maximum L_{max} dans un environnement à *m* machines parallèles identiques où la préemption est autorisée [13].

III.5 Problème d'ordonnancement job shop

III.5.1 Définition

Un problème d'ordonnancement job-shop est un problème d'optimisation combinatoire qui consiste à ordonnancer un ensemble de tâches à effectuer sur un ensemble de machines. Chaque tâche doit être exécutée sur une machine particulière et chaque machine ne peut traiter qu'une tâche à la fois comme illustré dans le tableau I.2.

Ce problème est couramment rencontré dans l'industrie manufacturière, où des produits doivent être fabriqués en utilisant différentes machines. Par exemple, dans une usine automobile, chaque voiture doit être assemblée en utilisant différentes machines qui effectuent des opérations telles que la soudure, l'installation de composants électroniques, etc.

Le problème d'ordonnancement job-shop est un problème difficile, car il existe un grand nombre de combinaisons possibles pour l'ordonnancement des tâches sur les différentes machines, et le choix d'un ordonnancement optimal peut avoir un impact significatif sur le temps et les coûts de production. Il existe plusieurs algorithmes heuristiques et méthodes d'optimisation qui peuvent être utilisées pour obtenir des solutions proches de l'optimum en un temps raisonnable.

	1	2	3	4	1	2	3	4	1	2	3	4
J_1	M_1	M_2	M_3		10	8	4		O_{11}	O_{21}	O_{31}	
J_2	M_2	M_1	M_4	M_3	8	3	5	6	O_{22}	O_{12}	O_{42}	O_{32}
J_3	M_1	M_2	M_4		4	7	3		O_{13}	O_{23}	O_{43}	
	Machines				Durée opératoire (p_{ij})				Opérations			

Tableau I.2 : La représentation de la gamme opératoire des jobs dans un job shop [22]

III.5.2 Représentations d'un problème d'ordonnancement job-shop

On retrouve fréquemment dans la littérature deux techniques de représentation du problème d'ordonnancement. La première technique est le diagramme de Gantt, qui utilise une échelle de temps pour représenter l'ordre des opérations. La deuxième technique est le graphe Potentiel-Tâches, où les opérations sont représentées par des nœuds connectés les uns aux autres par des flèches.

III.5.2.1 Le diagramme de Gantt

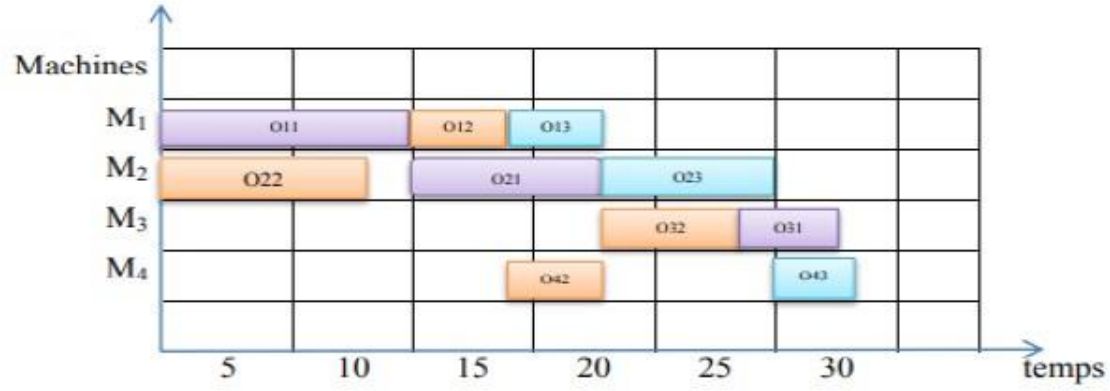
Le diagramme de Gantt, également connu sous le nom de graphique à barres, est un outil précieux pour la planification et le suivi des projets, notamment dans le domaine de l'ordonnancement des tâches. Il offre une représentation claire et concise des différentes tâches à effectuer, de leur durée respective et de leurs dépendances.

Dans le contexte de l'ordonnancement, le diagramme de Gantt est utilisé pour déterminer la séquence des tâches en tenant compte de leur durée et de leurs interdépendances. Il permet également d'identifier les tâches critiques, c'est-à-dire celles qui doivent être réalisées dans les délais impartis pour mener le projet à bien dans les temps.

L'objectif de ce diagramme est de présenter rapidement et de manière claire une solution à un problème d'ordonnancement. Deux types de diagrammes de Gantt sont couramment utilisés : le diagramme de Gantt des ressources et le diagramme de Gantt des tâches.

Le diagramme de Gantt des ressources est composé d'une ligne horizontale pour chaque ressource (machine). Les périodes d'exécution des différentes opérations sont représentées en séquence, ainsi que les périodes d'inactivité de la ressource.

Le diagramme de Gantt des tâches permet de visualiser les séquences d'opérations pour chaque tâche. Chaque tâche est représentée par une ligne où sont indiqués les périodes d'exécution des opérations et les périodes d'attente des ressources.



Figue I.10 : Diagramme de Gantt ressource pour un problème 4x3 [22]

III.5.2.2 Le graphe Potentiel-Tâches

Le graphe Potentiel-Tâches est une représentation schématique des différentes tâches d'un projet, où chaque tâche est représentée par un nœud ou un bloc, relié par des flèches qui indiquent la séquence dans laquelle les tâches doivent être réalisées. Les nœuds du graphe représentent les étapes distinctes du projet, tandis que les flèches représentent les dépendances entre les tâches.

Dans le graphe Potentiel-Tâches, comme illustré dans la figure I.11, les arcs peuvent être de deux types :

- Les arcs conjonctifs qui représentent les contraintes de précédence entre les tâches et indiquent également les durées respectives des tâches.
- Les arcs disjonctifs qui expriment les contraintes liées aux ressources nécessaires pour effectuer les tâches.

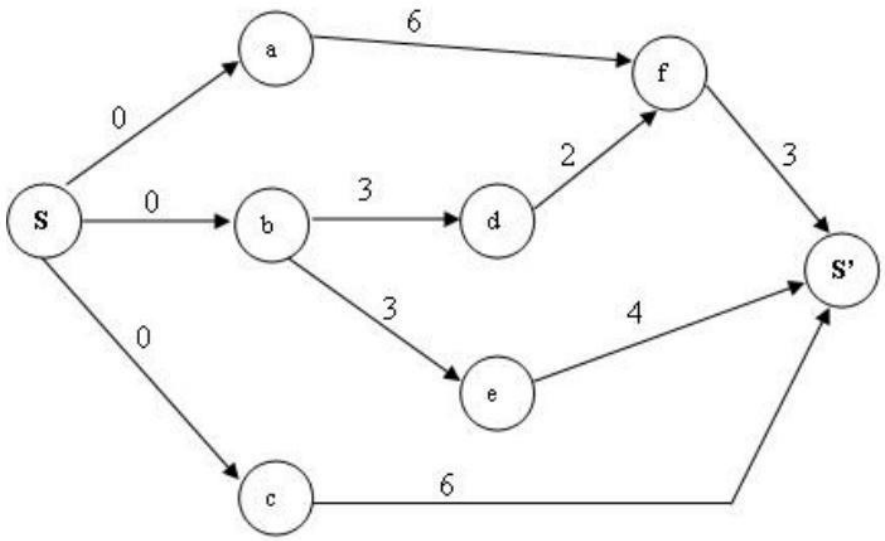


Figure I.11 : Graphe Potentiel-Tâches d'un ordonnancement

III.6. Caractéristiques générales des ordonnancements

L'un des facteurs clés pour générer une planification optimale est d'exploiter au maximum le temps disponible, car un simple ajustement des opérations peut éviter les périodes d'inactivité ou les temps morts. En effet, plus une planification est compacte, meilleure est sa qualité. Cette notion de compacité est à la base de différents types de planification : admissible, actif, semi-actif et sans délai [17].

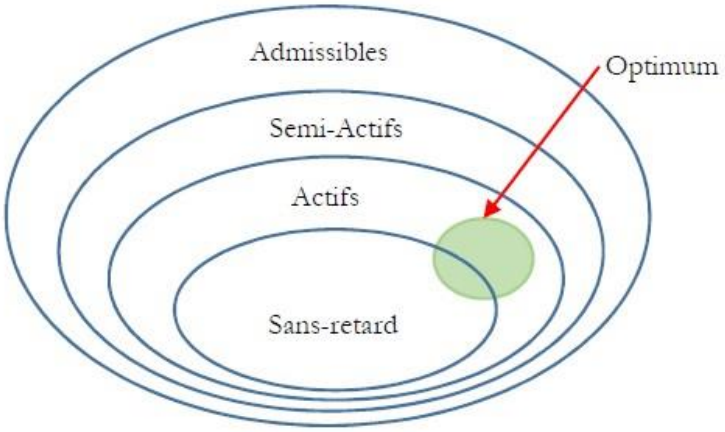


Figure I.12 : Diagramme de Venn [22]

Ainsi, les ordonnancements sans retard sont inclus dans le sous-ensemble des ordonnancements actifs, qui sont eux-mêmes inclus dans le sous-ensemble des ordonnancements semi-actifs.

III.6.1 Ordonnancement admissible

L'ordonnancement admissible (ou ordonnancement faisable) est une séquence de tâches à exécuter sur une machine qui respecte les contraintes de précédence, de ressources et de temps. Cela garantit que toutes les tâches seront exécutées dans un ordre optimal en respectant toutes les contraintes du problème. La figure I.13 présente un exemple d'ordonnancement admissible d'un problème à deux jobs et deux machines. Dans cet exemple une seule contrainte

est imposée, c'est la restriction sur la capacité des machines (chaque machine traite un seul job à la fois).

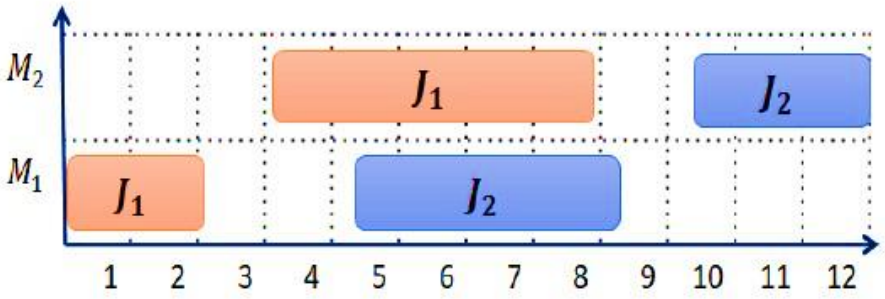


Figure I.13 : Ordonnancement admissible [09]

Comme démontré dans la figure I.13 ci-dessus, il est évident que la solution obtenue à partir d'un ordonnancement admissible n'est pas nécessairement la solution optimale. Dans de tels cas, il est souvent nécessaire d'apporter des améliorations afin d'atteindre une solution optimale, ou du moins une solution qui s'en rapproche [09].

III.6.2 Ordonnancement semi-actif

Un ordonnancement est considéré comme semi-actif lorsqu'il est valide et qu'il n'est pas possible de déplacer une opération vers la gauche pour obtenir un nouvel ordonnancement valide tout en préservant la séquence des opérations sur les différentes ressources. En d'autres termes, dans un ordonnancement semi-actif, chaque opération démarre dès que possible.

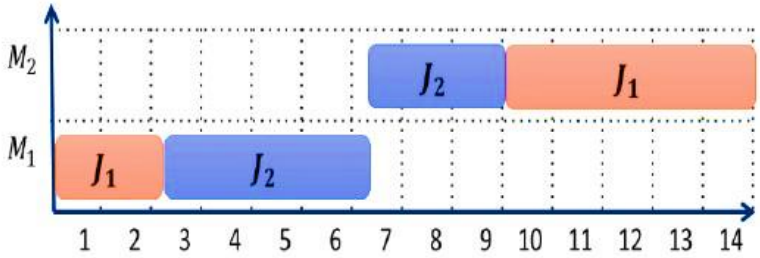


Figure I.14 : Ordonnancement admissible semi actif [09]

III.6.3. Ordonnancement actif

Un ordonnancement est dit actif s'il est impossible d'avancer le début d'exécution d'une opération sans retarder une autre tâche ou violer une contrainte. Il est dit, actif généralisée, lorsqu'il est impossible de translater un sous-ensemble d'opérations vers la gauche, pour obtenir un nouvel ordonnancement valide, sans en retarder un autre. [22]

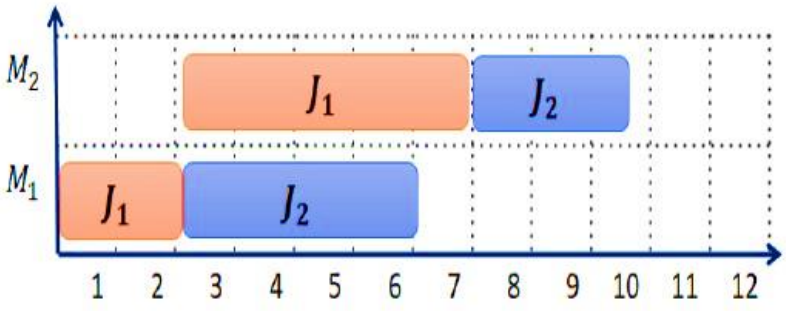
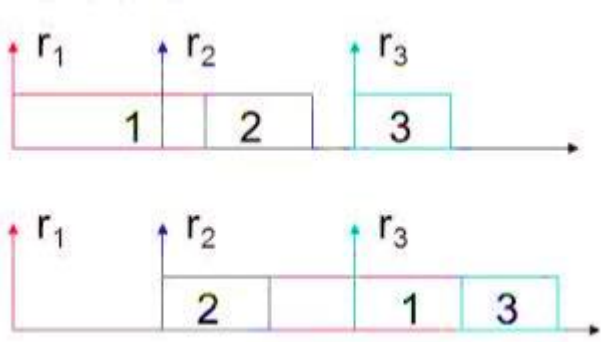


Figure I.15 : Ordonnancement admissible actif [09]

III.6.4 Ordonnancements sans délais

Un ordonnancement valide est sans délai ou sans retard (non-delay), si et seulement si aucune opération n'est mise en attente, alors, aucune ressource ne reste inoccupée.



Un ordonnancement sans retard

Un ordonnancement avec retard

Figure I.16 : Ordonnancement sans délai

Conclusion

Dans ce chapitre, nous avons présenté les notions clés de la gestion de production et les systèmes flexibles de production. Nous avons ensuite fourni des définitions générales de l'ordonnancement, en mettant en évidence l'importance de cette étape pour optimiser la production et la rentabilité de l'entreprise. Nous avons également abordé les différents types de problèmes d'ordonnancement d'atelier, qui peuvent se présenter dans des situations variées.

La planification efficace d'un ordonnancement est un processus crucial qui a un impact direct sur la productivité du système, quel que soit son type de production. En effet, un mauvais ordonnancement entraîne une utilisation inappropriée des ressources de production. C'est pourquoi le deuxième chapitre fournira une vue d'ensemble des méthodes de résolution des problèmes d'ordonnancement, visant ainsi à améliorer la qualité de la planification et à optimiser les performances du système.

Chapitre II : Les méthodes de résolutions

Introduction

Les problèmes d'optimisation sont importants dans de nombreux domaines et visent à trouver la meilleure solution en maximisant ou minimisant une fonction objectif tout en respectant des contraintes.

Dans ce chapitre, nous étudierons en détail les méthodes de résolution les plus utilisées pour ces problèmes. Nous présenterons les principes de chaque méthode, discuterons de leurs avantages et inconvénients, et illustrerons leur application avec des exemples concrets. En comprenant ces approches, nous pouvons choisir la méthode la plus appropriée pour résoudre efficacement nos problèmes d'optimisation en tenant compte de leurs caractéristiques et des contraintes imposées.

Problème d'optimisation combinatoire

Un Problème d'Optimisation Combinatoire (POC) est un type de problème dans le domaine de l'optimisation où l'objectif est de trouver la meilleure solution parmi un ensemble fini de solutions possibles, en respectant un ensemble de contraintes spécifiques.

Dans un POC, on cherche à optimiser une fonction objectif, c'est-à-dire une fonction qui évalue la qualité de chaque solution en attribuant une valeur numérique. L'objectif est de minimiser ou maximiser cette fonction objectif en trouvant la solution qui donne la valeur la plus basse ou la plus élevée, respectivement.

La particularité des problèmes d'optimisation combinatoire réside dans la nature discrète et combinatoire de l'espace de recherche. Les solutions possibles sont généralement représentées par des combinaisons ou des permutations d'éléments, et l'exploration de l'ensemble des solutions peut être très coûteuse en termes de temps de calcul, en raison de l'explosion combinatoire.

Les POC peuvent être rencontrés dans divers domaines, tels que la logistique, la planification, l'ordonnancement, la conception de réseaux, la gestion des ressources, la bio-informatique, etc. Ils posent des défis complexes en raison du grand nombre de possibilités à explorer et des contraintes qui doivent être satisfaites.

Pour résoudre un POC, différentes techniques peuvent être utilisées, telles que les méthodes exactes qui garantissent la recherche de la solution optimale, les heuristiques qui cherchent des solutions de qualité sans garantie d'optimalité, et les méta-heuristiques qui sont des méthodes d'optimisation générales , capables de s'adapter à différents problèmes.

Complexité du système job shop

La complexité du système job shop réside dans la résolution des problèmes d'ordonnancement, ce qui est rendu difficile en raison de divers facteurs. Ces facteurs incluent les combinaisons complexes de tâches et de ressources, ainsi que les contraintes temporelles et de précédence. En raison de ces défis, il est souvent impossible d'explorer de manière exhaustive toutes les possibilités, en particulier pour les problèmes de grande taille. Par conséquent, des techniques heuristiques et des méta-heuristiques sont utilisées pour trouver des solutions de haute qualité, bien que leur optimalité ne soit pas garantie. En conclusion, la complexité du système job shop pose des défis importants lorsqu'il s'agit de résoudre les problèmes d'ordonnancement.

Classification des méthodes de résolutions

Les méthodes de résolution des problèmes d'optimisation combinatoire peuvent être classées en différentes catégories. Les approches exactes, bien qu'elles puissent fournir des solutions optimales, sont souvent laborieuses et peu pratiques en raison de la complexité de ces problèmes. En pratique, il est souvent impossible de passer en revue toutes les instances et de comparer leurs qualités en raison du temps nécessaire. Par conséquent, la recherche de la solution optimale devient un défi majeur en raison de sa complexité temporelle, ce qui confère aux problèmes d'optimisation combinatoire leur réputation de difficulté.

Certains problèmes d'optimisation combinatoire peuvent être résolus de manière exacte en temps polynomial en utilisant des algorithmes spécifiques tels que les algorithmes gloutons, les algorithmes de programmation dynamique ou en les reformulant comme des problèmes d'optimisation linéaire. Cependant, il y a des situations où une solution exacte en temps polynomial n'est pas réalisable. Dans de tels cas, on peut recourir à des méthodes d'approximation telles que les heuristiques ou les méta-heuristiques pour obtenir des solutions approximatives de haute qualité dans un délai de résolution raisonnable. Bien que ces solutions ne soient pas parfaites, elles sont souvent considérées comme satisfaisantes pour résoudre de nombreux problèmes d'optimisation combinatoire.

Une méthode d'approximation (heuristique ou méta-heuristique) est un algorithme qui vise à trouver une solution réalisable sans garantie d'optimalité, par opposition à une méthode exacte qui garantit une solution exacte. Les algorithmes de solution exacte pour les problèmes difficiles sont d'une complexité exponentielle, il est donc parfois utile d'utiliser des méthodes d'approximation pour calculer une solution approximative au problème ou pour accélérer le processus de solution exacte. Les heuristiques sont généralement développées pour un problème spécifique, mais les méta-heuristiques peuvent inclure des principes plus généraux [23].

Les méthodes de résolution du système job shop en particulier et des problèmes d’optimisation en général peuvent être classées en plusieurs catégories en fonction de leurs caractéristiques et de leurs approches. La figure ci-dessous montre une classification courante des méthodes de résolution de problèmes d’optimisation.

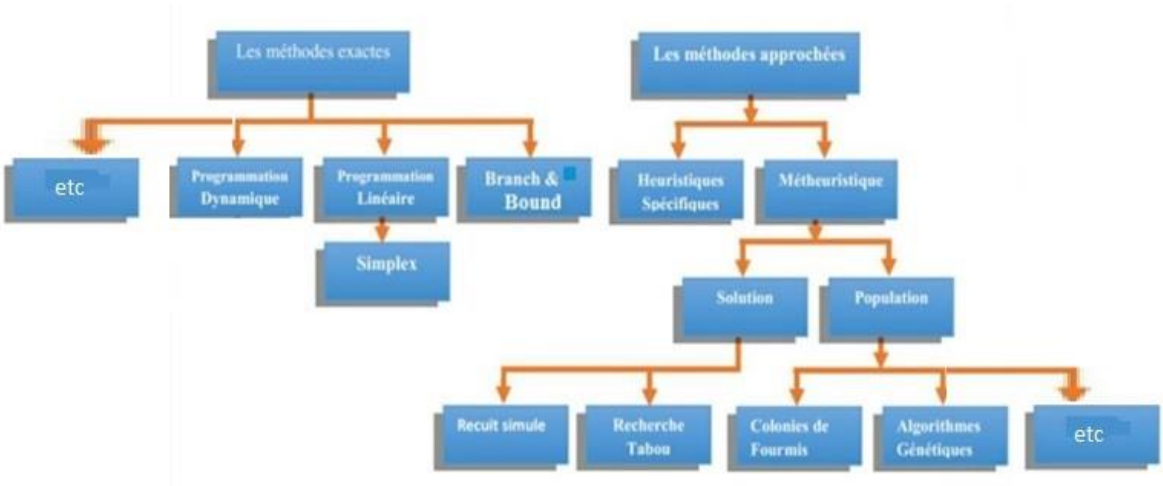


Figure II.1 : Classification des méthodes de résolutions de problèmes d'optimisation [24]

IV.1 Méthodes de résolution exactes

Les méthodes exactes visent à trouver la solution optimale en explorant exhaustivement toutes les combinaisons possibles de tâches et de ressources dans le système job shop. Elles utilisent des techniques telles que la programmation mathématique ou la recherche branch-and-bound pour résoudre le problème de manière précise. Toutefois, ces méthodes peuvent être chronophages et ne sont généralement applicables que pour des instances de petite taille. L'avantage des méthodes exactes réside dans leur capacité à garantir une solution optimale pour le problème traité. En effet, elles permettent d'explorer l'intégralité de l'espace de recherche de manière à inclure toutes les solutions potentiellement meilleures que la solution optimale trouvée lors de la recherche [25].

Nous présentons ci-dessous quelques méthodes de résolution exacte couramment utilisées.

IV.1.1 Recherche exhaustive

Cette méthode consiste à énumérer toutes les permutations possibles des tâches et à évaluer chaque permutation pour trouver la meilleure séquence d'exécution. Cela peut être réalisé en utilisant des algorithmes de recherche complète tels que la recherche exhaustive ou la recherche par force brute. Bien que cette méthode garantisse l'optimalité, elle est généralement inefficace pour les problèmes de grande taille en raison de l'explosion combinatoire.

IV.1.2 Programmation mathématique

Cette méthode consiste à formuler le problème job shop sous la forme d'un modèle mathématique, généralement un programme linéaire ou un programme d'optimisation en nombres entiers (PNE), et à résoudre ce modèle à l'aide de solveurs mathématiques. La programmation mathématique offre une garantie d'optimalité, mais sa complexité peut rendre son application difficile pour les problèmes de grande taille.

Voici quelques modèles mathématiques :

- **LP** (Programme linéaire) : toutes les expressions du modèle sont linéaires et il y a aucune restriction sur les valeurs des variables.

- **ILP** (Programme linéaire entier) : toutes les expressions sont linéaires, mais il y a néanmoins un sous-ensemble de variables qui doit être limité à des valeurs entières.
- **PILP** (Programme linéaire entier pur) : toutes les expressions sont linéaires et toutes les variables sont limitées à des valeurs entières.
- **NLP** (Programme non linéaire) : au moins, une des expressions dans le modèle est non linéaire par rapport aux variables.
- **MILP** (un modèle linéaire en nombres entiers mixtes) : un modèle mathématique linéaire, composé d'un ensemble de variables entières et un ensemble de variables binaires, est appelé (Programmation linéaire mixte en nombres entiers).

IV.1.3 L'algorithme de retour arrière (Backtracking)

Les algorithmes de backtracking permettent de tester systématiquement l'ensemble des affectations potentielles à un problème donné de manière beaucoup plus optimisé qu'un simple test itératif. Imaginez un arbre binaire complet de profondeur *n*, composé de nœuds rouges excepté un seul. Le but serait d'obtenir le plus rapidement un chemin qui mène au un nœud non-rouge sans connaître sa position à l'avance. Nous rappelons simplement les caractéristiques essentielles de cet algorithme. Dans l'état initial, aucune variable n'est affectée. À une étape donnée, l'ensemble des variables est divisé en deux groupes, les variables affectées et les variables non affectées. La solution donnée par les variables affectées est cohérente par rapport aux sous-ensembles de contraintes qui ne portent que sur ces variables. La prochaine étape va consister à affecter une nouvelle variable, le système choisit une variable et une valeur dans l'ensemble de celles qui sont possibles, et vérifie la cohérence. Si le système est cohérent, il passe à l'étape suivante, si le système est incohérent, il essaie une autre valeur pour la variable considérée. Si toutes les valeurs ont été considérées, il désaffecte une des variables affectées, et repart à l'étape précédente.

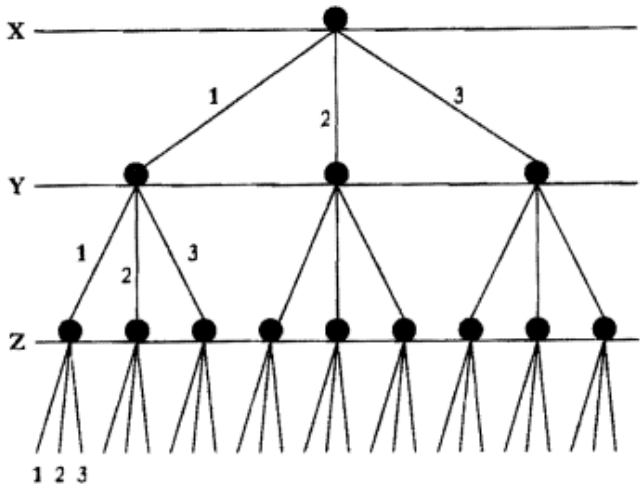


Figure II.2 : Arbre de recherche [26]

Dans l'exemple de la Figure II.2, on peut voir un arbre de recherche dans le cas où nous aurions des contraintes qui portent sur trois variables X, Y et Z (chacune étant définie dans l'intervalle [1...3]). Le système dans un premier temps évaluera X, en lui affectant par exemple

la valeur l, dans un deuxième temps il évaluera Y puis Z. Si cette évaluation ne satisfait pas le système de contraintes, il choisira une nouvelle valeur pour Z. Si aucune des valeurs de Z ne permet de satisfaire l'ensemble de contraintes, le système choisira une nouvelle valeur pour Y, et ainsi de suite. Il existe des heuristiques qui permettent d'améliorer et de personnaliser cet algorithme de manière à orienter la recherche de solutions (par exemple les heuristiques portant sur le choix de la variable à remettre en cause lors d'un retour arrière). L'avantage majeur de cette technique est sa complétude : on trouve toujours la solution si elle existe, et la possibilité pour l'utilisateur de contrôler la recherche de solution en cas de réponses multiples. Ses inconvénients sont :

- l'obligation de travailler sur des domaines finis, or nos systèmes de contraintes manipulent des domaines infinis.

- le coût généralement très grand de cette technique, qui procède par énumération, lorsque le domaine des variables est trop grand [26].

V.1.4 La méthode Branch and Bound (B&B)

La méthode Branch and Bound (B&B) découpe l'espace de recherche en sous-espaces plus petits, appelés branches, et utilise une stratégie de recherche intelligente pour éviter d'explorer les branches non-prometteuses. Elle utilise des bornes supérieures et inférieures pour évaluer les solutions partielles et guide la recherche vers les régions les plus prometteuses de l'espace de recherche. La méthode branch-and-bound peut être combinée avec des techniques telles que la recherche arborescente ou la programmation dynamique pour résoudre les problèmes du système job shop.

L'algorithme de Branch-and-Bound (B&B) consiste à construire progressivement l'arbre d'énumération de toutes les solutions possibles du modèle de programmation par contraintes de manière intelligente, en utilisant des bornes inférieures et supérieures pour éviter de générer tous les nœuds de l'arbre. Le processus commence par diviser récursivement le problème initial en sous-problèmes plus petits et plus faciles à résoudre, appelé phase de séparation (branching). Ensuite, lors de la phase d'évaluation (bound), les sous-ensembles qui peuvent contenir la solution optimale sont identifiés et ceux qui ne la contiennent pas sont éliminés. L'algorithme de B&B construit et parcourt l'arbre de recherche, où la racine représente l'espace réalisable du problème initial et les nœuds représentent les espaces réalisables des sous-problèmes. À chaque itération, l'algorithme de B&B résout la relaxation du sous-problème en cours, fournissant ainsi une borne inférieure sur la valeur optimale du sous-problème.

Le Branch-and-Bound est basé sur trois axes principaux :

- L’évaluation.
- La séparation.
- La stratégie de parcours.

La principale caractéristique de cette méthode est qu'elle n'explore pas les sous-branches inutiles. Grâce à la connectivité des nœuds, on peut déterminer à l'avance qu'il n'est pas possible d'améliorer la solution trouvée, ce qui permet de trouver une solution appropriée dans un temps de recherche raisonnable [25].

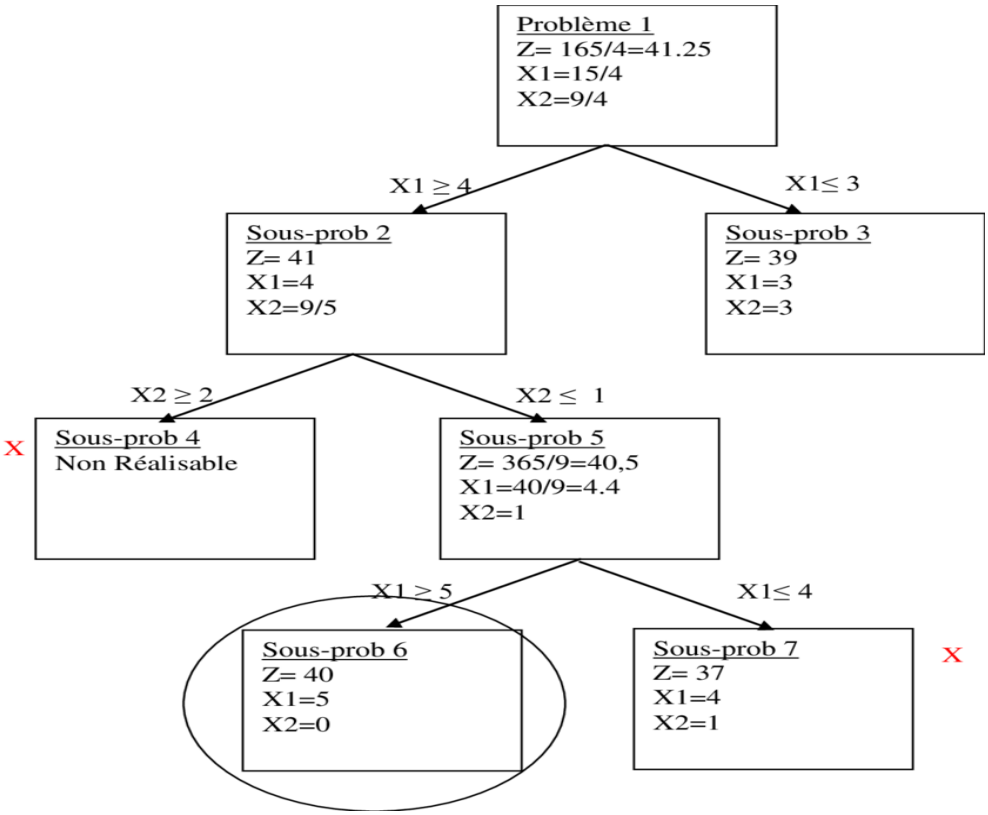


Figure II.3 : Déroulement de l'algorithme Branch and Bound [29]

IV.1.5 Méthodes basées sur les graphes de précédence

Ces méthodes modélisent le problème job shop sous forme d'un graphe de précédence, où les nœuds représentent les tâches et les arcs représentent les dépendances entre les tâches. En utilisant des algorithmes de recherche de plus court chemin ou de recherche topologique, il est possible de trouver la séquence optimale des tâches.

Ces méthodes de résolution exacte sont souvent utilisées pour résoudre des instances de petite ou moyenne taille du système job shop, où la garantie d'optimalité est cruciale. Cependant, en raison de l'explosion combinatoire, elles peuvent être coûteuses en termes de temps de calcul pour les problèmes de grande taille. Dans de tels cas, les méthodes approchées sont généralement préférées.

IV.2 Méthodes approchées

Les méthodes de résolution approchée du système job shop sont des approches qui visent à trouver des solutions de haute qualité, même si elles ne garantissent pas l'optimalité. Ces méthodes sont souvent utilisées pour résoudre des problèmes de grande taille, où les méthodes exactes deviennent trop coûteuses en termes de temps de calcul. Les méthodes approchées exploitent des heuristiques, des méta-heuristiques et d'autres techniques pour explorer efficacement l'espace de recherche et trouver des solutions prometteuses.

IV.2.1 Heuristiques basées sur des règles de priorité

Les méthodes d'approximation, également connues sous le nom d'heuristiques, permettent de trouver rapidement une solution réalisable à un problème donné. Cependant, cette solution n'est pas nécessairement la meilleure possible. L'objectif des heuristiques est donc de

trouver une solution plus rapide tout en se rapprochant autant que possible de la solution exacte. Les heuristiques basées sur des règles de priorité sont évaluées en comparant leur solution à la valeur optimale, et une faible différence indique une meilleure performance de l'heuristique. Ces heuristiques sont regroupées en différentes catégories selon leur mode de fonctionnement, et elles sont largement utilisées dans des domaines tels que l'optimisation combinatoire, la théorie des graphes, la complexité algorithmique et l'intelligence artificielle. Elles sont indispensables lorsque l'algorithme de solution exacte devient extrêmement complexe, ce qui est souvent le cas pour de nombreux problèmes difficiles. Les heuristiques peuvent être utilisées pour obtenir une solution approximative d'un problème ou pour accélérer la recherche d'une solution exacte. Elles sont généralement développées en tenant compte de la structure spécifique du problème, mais elles peuvent également intégrer des principes plus généraux. [30].

Les heuristiques basées sur des règles sont des approches couramment utilisées pour résoudre des problèmes job shop. Elles consistent à appliquer des règles spécifiques pour déterminer l'ordre d'exécution des tâches dans le système. Ces règles sont généralement simples et intuitives, et permettent de trouver rapidement des solutions réalisables, bien que leur qualité puisse varier.

Ci-dessous quelques exemples de règles couramment utilisées dans les heuristiques basées sur des règles pour le système job shop :

IV.2.1.1 Shortest Processing Time (SPT)

SPT est une heuristique qui consiste à ordonner les tâches dans l'ordre croissant des temps de traitement. Le principe est que lorsqu'une machine se libère, la tâche qui possède le moins de temps de traitement doit être traitée.

IV.2.1.2 Shortest Remaining Processing Time (SRPT)

Cette règle consiste à donner la priorité à la tâche qui a le temps de traitement restant le plus court parmi les tâches en attente d'exécution. Cela permet de prendre en compte le progrès réalisé dans l'exécution des tâches et d'optimiser l'utilisation des ressources disponibles.

IV.2.1.3 Earliest Due Date (EDD)

Cette règle attribue la priorité aux tâches ayant la date d'échéance la plus proche. Elle vise à minimiser les retards et à respecter les contraintes de délai pour les tâches critiques.

IV.2.1.4 Longest Processing Time (LPT)

Contrairement à la règle SPT, cette règle donne la priorité aux tâches ayant le temps de traitement le plus long. Cela peut être utile dans certains cas pour équilibrer la charge de travail entre les ressources.

Ces règles peuvent être utilisées individuellement ou en combinaison, selon les besoins spécifiques du système job shop. Les heuristiques basées sur des règles sont simples à mettre en œuvre et peuvent donner de bons résultats pour des problèmes de taille moyenne. Cependant, elles peuvent être sensibles aux caractéristiques spécifiques du problème et ne garantissent pas l'obtention de la solution optimale.

IV.2.1.5 First Come, First Solved (FCFS)

La règle FCFS donne la priorité aux tâches en fonction de leur ordre d'arrivée. Elle est simple à comprendre et à mettre en œuvre, mais peut entraîner des temps d'attente élevés si des tâches de longue durée arrivent en premier.

IV.2.1.6 Les avantages des heuristiques pour la résolution des problèmes d'ordonnancement

Les heuristiques jouent un rôle crucial dans la résolution des problèmes d'ordonnancement en fournissant des méthodes approximatives, mais efficaces, pour trouver des solutions acceptables dans un délai raisonnable. Voici quelques-uns des avantages et de l'importance des heuristiques dans ce contexte :

1. Temps de calcul réduit : Les problèmes d'ordonnancement sont souvent NP-difficiles, ce qui signifie qu'il est peu probable de trouver une solution optimale en un temps raisonnable. Les heuristiques offrent une alternative en fournissant des approches rapides pour trouver des solutions de bonne qualité, même si elles ne sont pas optimales.

2. Adaptabilité : Les heuristiques peuvent être conçues pour s'adapter à des problèmes d'ordonnancement spécifiques ou à des contraintes particulières. Elles peuvent prendre en compte des facteurs tels que les priorités des tâches, les dépendances, les ressources disponibles, les contraintes temporelles, etc. Cette adaptabilité permet aux heuristiques de fournir des solutions qui répondent aux besoins spécifiques de chaque problème.

3. Solutions proches de l'optimalité : Bien que les heuristiques ne garantissent pas des solutions optimales, elles peuvent souvent produire des solutions proches de l'optimalité. Ces solutions sont généralement suffisamment bonnes pour être utilisées dans la pratique et peuvent être améliorées davantage si nécessaire.

4. Facilité d'implémentation : Les heuristiques sont souvent relativement simples à implémenter et à comprendre. Elles peuvent être basées sur des principes intuitifs ou des stratégies de recherche locales qui ne nécessitent pas une connaissance approfondie des mathématiques avancées ou des techniques de résolution complexes.

5. Robustesse face aux variations du problème : Les heuristiques sont souvent capables de gérer efficacement les variations des problèmes d'ordonnancement, telles que l'ajout de nouvelles tâches, la modification des contraintes ou l'adaptation à des environnements dynamiques. Elles peuvent être facilement ajustées ou étendues pour tenir compte de ces changements.

6. Exploration de l'espace de recherche : Les heuristiques peuvent explorer rapidement un grand nombre de solutions potentielles et guider la recherche vers des régions prometteuses de l'espace de recherche. Cela permet de trouver rapidement des solutions de qualité sans examiner toutes les combinaisons possibles, ce qui serait généralement impossible pour les problèmes d'ordonnancement complexes.

Il convient de noter que les heuristiques peuvent présenter quelques inconvénients, notamment la possibilité de converger vers des solutions sous-optimales, la difficulté à évaluer la qualité des solutions trouvées et la sensibilité aux paramètres d'ajustement. Cependant,

malgré ces limitations, les heuristiques restent une approche très utile et largement utilisée pour résoudre les problèmes d'ordonnancement dans de nombreux domaines.

IV.2.2 Méta-heuristiques

Les méta-heuristiques sont des approches de résolution heuristique utilisées pour résoudre des problèmes d'optimisation complexes, y compris le système job shop. Contrairement aux méthodes de résolution exactes, les méta-heuristiques ne garantissent pas de trouver la solution optimale, mais elles visent à fournir des solutions de haute qualité dans un temps de calcule raisonnable. Nous présentons dans la suite quelques exemples de méta-heuristiques couramment utilisées pour résoudre le système job shop.

Les méta-heuristiques peuvent être classées en deux sous-catégories comme le montre la figure II.4 :

- Les méthodes à base de solution unique
- les méthodes à base de population de solutions

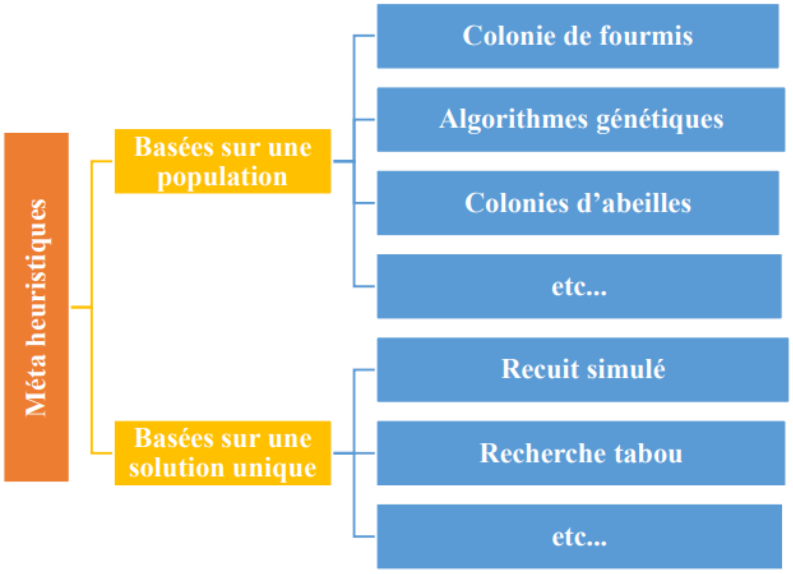


Figure II.4 : Classification des méta-heuristiques [31]

IV.2.2.1 Les méta-heuristiques à base de solution unique

Ces méthodes utilisent la recherche locale, également connue sous le nom de recherche par voisinage. Elles commencent avec une solution initiale de l'espace de recherche et effectuent une série de modifications dans le but de trouver des solutions plus performantes. Des exemples de ces méthodes sont le recherche taboues et le recuit simulé. Toutefois, ces algorithmes sont souvent susceptibles de converger vers des optima locaux, ce qui limite leur capacité à trouver la meilleure solution globale [31].

IV.2.2.1.1 Le recuit simulé

Le Recuit Simulé (RS) est une méthode de recherche locale inspirée du processus métallurgique qui consiste à refroidir progressivement un matériau pour réduire son énergie.

Cette méthode est basée sur l'algorithme de Métropolis-Hastings, qui décrit l'évolution d'un système thermodynamique. Dans cette analogie, la fonction à minimiser est l'énergie du système, représentée par une solution donnée, et la température fictive T est introduite. L'algorithme du recuit simulé ajuste itérativement la solution en refroidissant le système. Les solutions trouvées peuvent améliorer le critère d'optimisation recherché, correspondant à une réduction de l'énergie du système, mais elles peuvent également le détériorer. En acceptant des solutions qui améliorent le critère, on explore le voisinage de la solution initiale à la recherche de l'optimum. Contrairement à d'autres méthodes de recherche locale, le recuit simulé peut accepter des solutions de moindre qualité en fonction de la détérioration de la solution considérée, permettant ainsi une exploration plus complète de l'espace de recherche [32].

Le recuit simulé s'est révélé efficace pour résoudre des problèmes combinatoires classiques, notamment pour de grands échantillons. Par exemple, des expériences ont démontré son fonctionnement pour résoudre le problème du voyageur de commerce avec plus de 800 villes. Des tests réussis ont également été réalisés sur des problèmes de partitionnement de graphes, révélant son efficacité pour une catégorie spécifique de problèmes possédant des propriétés particulières [33].

Le principe de base du recuit simulé est donc de simuler le processus de refroidissement d'un matériau afin d'explorer l'espace de recherche et de rechercher des solutions de meilleure qualité.

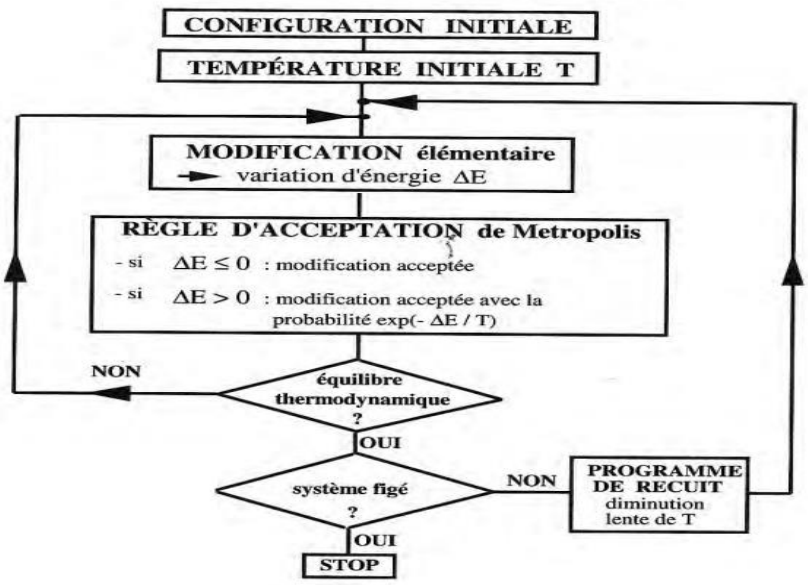
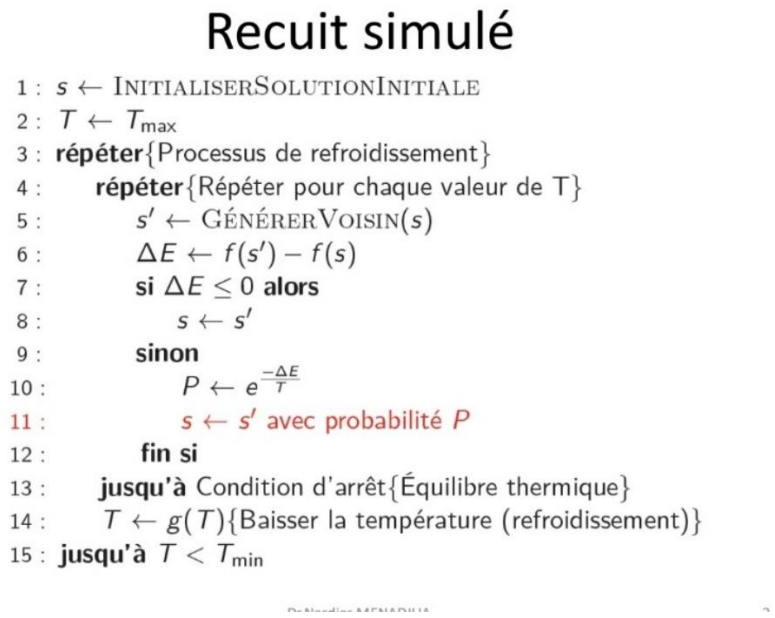


Figure II.5 : La version la plus simple du recuit simulé [33]

L'algorithme 1 Algorithme de recuit simulé



Exemples d'application d'algorithme recuit simulé

L'algorithme du recuit simulé est une méthode méta-heuristique utilisée pour résoudre des problèmes d'optimisation. Voici quelques exemples d'application de cette méthode :

- *Problèmes d'ordonnancement* : L'algorithme du recuit simulé peut être utilisé pour résoudre des problèmes d'ordonnancement, tels que l'optimisation des plannings de production, des emplois du temps ou des horaires de travail. Il permet de trouver des solutions qui minimisent les temps d'attente, maximisent l'utilisation des ressources et satisfont les contraintes spécifiques.
- *Problèmes de placement* : L'algorithme du recuit simulé peut être appliqué aux problèmes de placement, où l'objectif est de déterminer la disposition optimale des objets ou des éléments dans un espace donné. Par exemple, dans le domaine de l'agencement d'usines, il peut aider à optimiser l'emplacement des machines ou des postes de travail pour améliorer l'efficacité de la production.
- *Problèmes de conception de circuits* : L'algorithme du recuit simulé est utilisé dans la conception de circuits électroniques pour optimiser la disposition des composants sur une puce, minimisant ainsi la distance entre eux et améliorant les performances électriques. Il permet d'explorer différentes configurations et de trouver une solution de conception satisfaisante.

IV.2.2.1.2 La Recherche Tabou

Le principe de base de la recherche tabou est d'utiliser une mémoire appelée « liste tabou » pour guider l'exploration de l'espace de recherche et éviter de revisiter les solutions déjà explorées. La recherche taboue est une méta-heuristique largement utilisée pour résoudre des problèmes d'optimisation combinatoire, y compris le système job shop.

Voici l'organigramme simple de l'algorithme tabou.

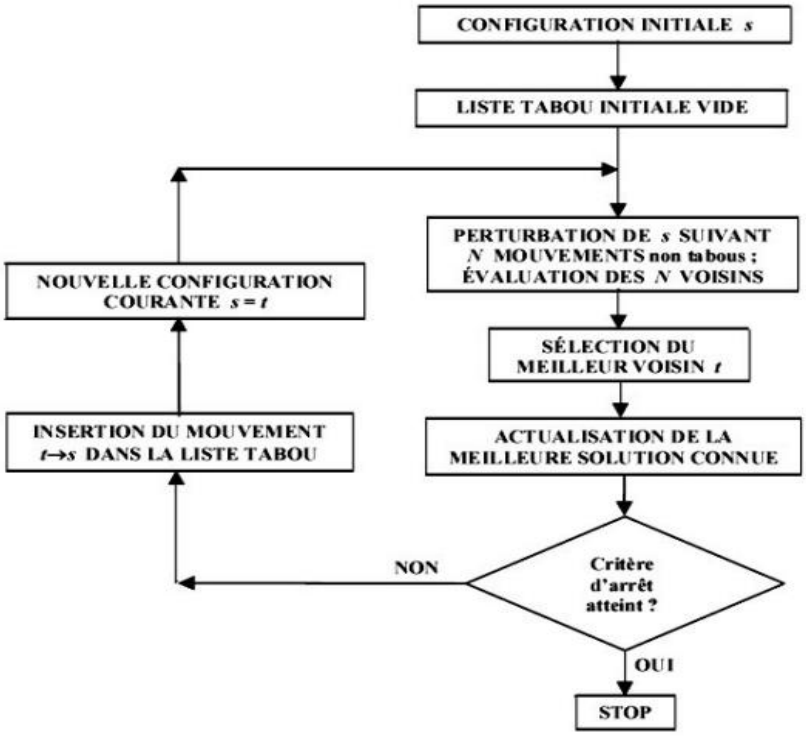


Figure II.6 : Organigramme de l'algorithme tabou simple [33]

L'algorithme 2 L'algorithme de recherche tabou

Recherche Tabou

Algorithme

```
1.  s ← solution initiale
2.  best ← s
3.  L ← ∅      // Tabou
4.  while (Conditions) loop
5.     s' ← Meilleur-Voisin(N(s), L)  // Voisin non tabou
6.     if f(s') < f(best) then
7.         best ← s'
8.     end if
9.     Actu_Tabu(s', L)
10. end while
11. Return best
```

Exemples d'application d'algorithme de recherche Tabou

L'algorithme de recherche tabou est une méthode heuristique utilisée pour résoudre des problèmes d'optimisation. Voici quelques exemples d'application de cette méthode :

- *Planification des horaires* : L'algorithme de recherche tabou peut être utilisé pour optimiser la planification des horaires dans différents domaines tels que les transports, les services de santé, les industries, etc. Il permet de trouver des horaires qui minimisent les conflits de ressources, maximisent l'utilisation des ressources disponibles et respectent les contraintes spécifiques.
- *Problèmes d'ordonnancement* : L'algorithme de recherche tabou peut être appliqué aux problèmes d'ordonnancement dans la production, la logistique ou tout autre domaine où il est nécessaire de déterminer l'ordre optimal d'exécution des tâches. Il peut aider à minimiser les temps d'attente, à optimiser l'utilisation des ressources et à réduire les coûts de production.
- *Problèmes de routage* : L'algorithme de recherche tabou peut être utilisé pour résoudre des problèmes de routage, tels que le problème du voyageur de commerce, où l'objectif est de trouver l'itinéraire optimal qui visite un ensemble de villes tout en minimisant la distance parcourue. L'algorithme de recherche tabou permet d'explorer efficacement l'espace des solutions et de trouver des solutions de qualité.
- *Optimisation de réseaux* : L'algorithme de recherche tabou peut être appliqué à l'optimisation de réseaux, tels que les réseaux de télécommunications, les réseaux de distribution d'énergie, etc. Il permet de trouver des configurations optimales pour les nœuds du réseau, de minimiser les coûts d'installation et d'optimiser la performance globale du réseau.
- *Problèmes de placement* : L'algorithme de recherche tabou peut être utilisé pour résoudre des problèmes de placement, tels que l'optimisation de l'agencement d'équipements dans une usine ou la disposition de meubles dans un espace restreint. Il peut aider à maximiser l'utilisation de l'espace, à optimiser l'efficacité des opérations et à minimiser les déplacements.

IV.2.2.2 Les méta-heuristiques à base de population de solutions

Ces approches mettent l'accent sur la recherche globale en utilisant un sous-ensemble de solutions appelé population, qui est ensuite modifiée pour construire et évaluer de nouvelles solutions. L'objectif est de guider efficacement la recherche vers des solutions de qualité dans l'espace de recherche. Parmi ces méthodes, on trouve les algorithmes évolutionnaires, les algorithmes basés sur les colonies de fourmis, les algorithmes inspirés des systèmes immunitaires [31].

IV.2.2.2.1 Algorithmes génétiques

Les algorithmes génétiques (AG) sont des approches de recherche inspirées par l'évolution biologique. Initialement introduits par J.H. Holland dans les années 1970, ils ont suscité un intérêt limité en raison de leur coût d'exécution élevé. Cependant, depuis les années 1990, leur développement a connu une croissance significative, en particulier avec l'avènement des architectures massivement parallèles qui exploitent leur parallélisme intrinsèque [35].

L'algorithme génétique est une méthode d'optimisation et de recherche inspirée du processus de sélection naturelle observé dans l'évolution biologique. Il est utilisé pour résoudre des problèmes complexes en générant et en évaluant des solutions potentielles de manière itérative.

Voici les étapes principales d'un algorithme génétique :

1. **Initialisation** : Une population initiale de solutions potentielles, appelées individus, est générée de manière aléatoire. Chaque individu est représenté sous forme d'un ensemble de gènes, qui peuvent être des valeurs numériques, des chaînes de caractères, des structures de données, etc.
2. **Évaluation** : Chaque individu de la population est évalué en fonction d'une fonction objectif prédéfinie. Cette fonction mesure la qualité de chaque individu en termes de performance par rapport au problème à résoudre. Plus la valeur de la fonction objectif est élevée, meilleure est la solution.
3. **Sélection** : Une sélection probabiliste est effectuée pour choisir les individus les plus performants de la population. Les individus de meilleure qualité ont plus de chances d'être sélectionnés pour la prochaine génération. Cela permet de favoriser les solutions prometteuses et d'éviter les solutions médiocres.
4. **Croisement** : Des opérations de croisement sont effectuées entre les individus sélectionnés pour créer de nouvelles solutions. L'idée est de combiner les caractéristiques des individus parents afin de générer une descendance qui peut potentiellement être meilleure que les parents.
5. **Mutation** : De manière aléatoire, certaines parties des nouvelles solutions peuvent être modifiées pour introduire de la diversité dans la population. La mutation permet d'explorer de nouvelles régions de l'espace de recherche et d'éviter de rester bloqué dans des optima locaux.
6. **Remplacement** : Les nouveaux individus issus du croisement et de la mutation remplacent les individus les moins performants de la population précédente.
7. **Répétition** : Les étapes 2 à 6 sont répétées pour un certain nombre de générations ou jusqu'à ce qu'une condition d'arrêt prédéfinie soit atteinte. La condition d'arrêt peut être un nombre fixe d'itérations, l'obtention d'une solution satisfaisante ou l'épuisement des ressources disponibles.

En résumé, l'algorithme génétique explore un espace de recherche en générant, évaluant, sélectionnant, croisant et mutant des solutions potentielles pour trouver une solution optimale ou proche de l'optimum à un problème donné. Il est couramment utilisé dans des domaines tels que l'optimisation de fonctions mathématiques, la conception de systèmes, l'ingénierie, la bio-informatique, etc.

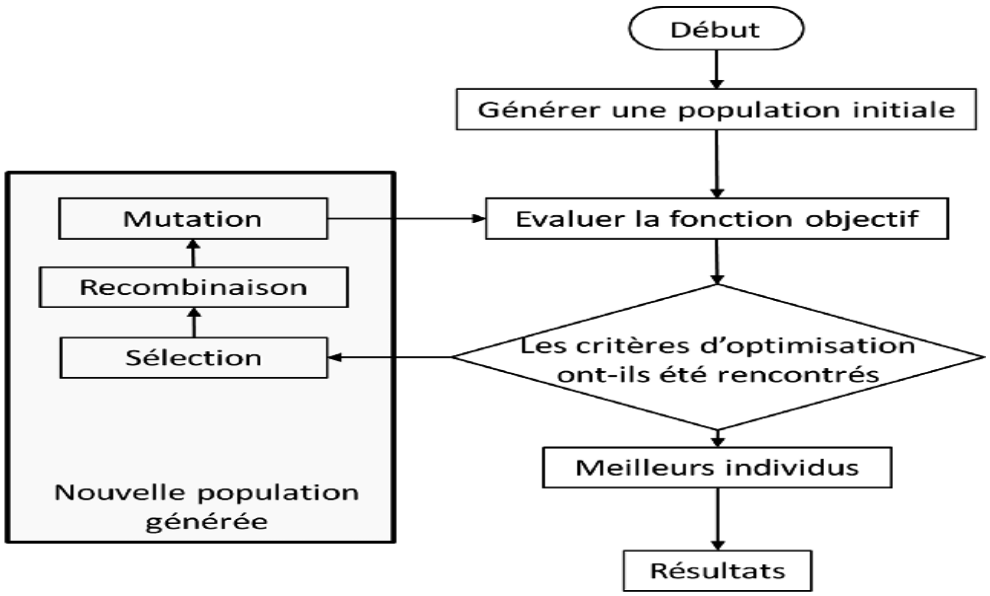


Figure II.7 : Organigramme d'un algorithme génétique [36]

L'algorithme 3 Algorithme génétique

L'algorithme général

initialiser la population (générer aléatoirement une population de *N* chromosomes *x*)
calculer le degré d'adaptation *f(x)* de chaque individu
Tant que non fini ou non convergence
  reproduction des parents
    sélectionner 2 individus à la fois
    appliquer les opérateurs génétiques
  calculer le degré d'adaptation *f(x)* de chaque enfant
  sélectionner les survivants parmi les parents et les enfants
fin Tant que
conclure

Exemples d'application d'algorithme génétique

L'algorithme génétique est une méthode méta-heuristique utilisée pour résoudre des problèmes d'optimisation. Voici quelques exemples d'application de cette méthode :

- *Optimisation des paramètres* : L'algorithme génétique peut être utilisé pour optimiser les paramètres dans des domaines tels que l'apprentissage automatique, la modélisation mathématique ou l'optimisation de systèmes. Il peut aider à trouver les valeurs optimales des paramètres pour maximiser la performance d'un modèle ou d'un système.
- *Problème du sac à dos* : L'algorithme génétique est souvent utilisé pour résoudre le problème du sac à dos, où l'objectif est de maximiser la valeur totale des objets pouvant

être placés dans un sac à dos avec une capacité limitée. Il permet d'explorer efficacement l'espace des combinaisons possibles et de trouver la meilleure combinaison d'objets.

- *Conception de circuits* : L'algorithme génétique peut être appliqué à la conception de circuits électroniques, où l'objectif est de trouver une disposition optimale des composants sur une puce en fonction de contraintes de performance, de consommation d'énergie, etc. Il permet d'explorer différentes configurations et de trouver une solution de conception efficace.
- *Problème de l'emploi du temps* : L'algorithme génétique peut être utilisé pour résoudre le problème de l'emploi du temps, où l'objectif est d'optimiser l'affectation des cours et des ressources dans un établissement scolaire ou universitaire. Il peut aider à minimiser les conflits d'emploi du temps, à maximiser l'utilisation des ressources et à satisfaire les contraintes spécifiques.
- *Problème de routage de véhicules* : L'algorithme génétique peut être appliqué au problème de routage de véhicules, où l'objectif est de déterminer les itinéraires optimaux pour un ensemble de véhicules afin de livrer des marchandises ou de fournir des services. Il permet d'optimiser l'utilisation des véhicules, de minimiser les coûts de transport et de respecter les contraintes de temps et de capacité.

IV.2.2.2.2 Algorithmes de colonies de fourmis

L'algorithme des colonies de fourmis est un algorithme d'optimisation inspiré du comportement des fourmis réelles. Il consiste à modéliser un problème d'optimisation sous la forme d'un graphe, où les nœuds représentent les solutions possibles. Les fourmis artificielles construisent itérativement des solutions en choisissant les nœuds voisins en fonction de la quantité de phéromone présente sur les arêtes du graphe. Les fourmis déposent également de la phéromone sur les arêtes en fonction de la qualité de leur solution. Pendant chaque itération, les fourmis sont guidées par la présence de phéromones pour explorer les parties prometteuses d'un graphe. L'algorithme des colonies de fourmis s'arrête lorsque certaines conditions prédéfinies sont satisfaites. Grâce à ce processus, il est possible de trouver une ou plusieurs solutions de haute qualité dans un espace de recherche complexe.

Les algorithmes de colonies de fourmis possèdent plusieurs caractéristiques intéressantes, mentionnons notamment :

- **Robustesse :** une colonie peut maintenir son activité même si certains individus présentent des défaillances.

- **Flexibilité :** une colonie de fourmis est capable de s'adapter aux modifications de l'environnement.

- **Décentralisation :** une colonie ne dépend pas d'une autorité centrale pour prendre des décisions.

- **Auto-organisation :** une colonie trouve elle-même une solution sans avoir connaissance de celle-ci à l'avance.

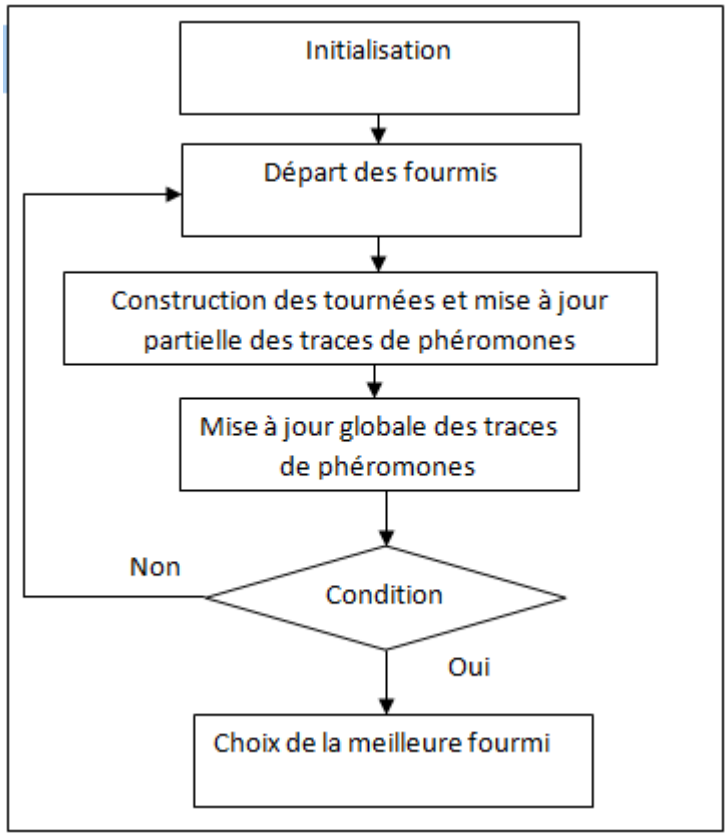


Figure II.8 : Organigramme du système de colonies des fourmis [38]

Exemples d'application d'algorithme de colonies fourmis

Les algorithmes de colonies de fourmis sont des techniques d'optimisation inspirées du comportement des fourmis dans la recherche de nourriture. Ils sont utilisés dans de nombreux domaines pour résoudre des problèmes d'optimisation et de recherche. Voici quelques exemples d'application des algorithmes de colonies de fourmis :

- *Problème du voyageur de commerce (TSP)* : L'algorithme de colonies de fourmis est largement utilisé pour résoudre le problème du voyageur de commerce. Dans ce problème, l'objectif est de trouver le chemin le plus court pour visiter un ensemble de villes une seule fois et revenir à la ville de départ. Les fourmis artificielles sont utilisées pour explorer différentes routes et déposer des phéromones sur les chemins les plus courts. Les phéromones attirent les autres fourmis à suivre les chemins les plus prometteurs, favorisant ainsi la découverte de la solution optimale.
- *Optimisation des réseaux de télécommunications* : Les algorithmes de colonies de fourmis sont appliqués pour résoudre des problèmes d'optimisation dans les réseaux de télécommunications, tels que la recherche du chemin le plus court dans un réseau de routage. Les fourmis artificielles peuvent être utilisées pour explorer différentes routes possibles, en tenant compte des contraintes de qualité de service telles que la latence ou la bande passante disponible.
- *Ordonnancement de production* : Les algorithmes de colonies de fourmis sont utilisés pour résoudre des problèmes d'ordonnancement de production dans diverses industries. Par exemple, dans le problème de l'ordonnancement de laproduction dans un atelier, les

fourmis artificielles peuvent être utilisées pour optimiser la séquence d'exécution des tâches afin de minimiser les temps de production, maximiser l'utilisation des ressources ou respecter les contraintes spécifiques du processus de production.

- *Routage de véhicules* : Les algorithmes de colonies de fourmis sont appliqués dans le domaine du routage de véhicules, où l'objectif est de trouver les itinéraires les plus efficaces pour un ensemble de véhicules qui doivent desservir différents emplacements. Les fourmis artificielles peuvent être utilisées pour explorer différentes combinaisons de routes et optimiser l'affectation des véhicules à des destinations spécifiques, en tenant compte des contraintes de capacité, des délais de livraison, etc.
- *Problèmes d'allocation de ressources* : Les algorithmes de colonies de fourmis peuvent également être utilisés pour résoudre des problèmes d'allocation de ressources, tels que l'affectation optimale de tâches à des ressources disponibles. Les fourmis artificielles peuvent explorer différentes combinaisons d'affectations et déposer des phéromones sur les solutions les plus efficaces, facilitant ainsi la recherche de la meilleure allocation possible.

Ces exemples illustrent la polyvalence des algorithmes de colonies de fourmis et leur capacité à résoudre une grande variété de problèmes d'optimisation dans différents domaines. Ils offrent une approche inspirée de la nature pour trouver des solutions efficaces et adaptatives.

IV.2.3 Approches basées sur l'apprentissage et l'intelligence artificielle

Les approches basées sur l'apprentissage et l'intelligence artificielle offrent une alternative intéressante pour la résolution des problèmes du système job shop. Elles tirent parti des techniques avancées d'apprentissage automatique et d'intelligence artificielle pour optimiser les opérations et les séquences de production.

Une approche courante consiste à utiliser des algorithmes de recherche locale basés sur des modèles. Ces algorithmes exploitent des modèles appris à partir de données historiques pour améliorer la planification et l'ordonnancement des tâches dans le système job shop. Les modèles peuvent être construits à l'aide de techniques telles que la régression, les arbres de décision, les machines à vecteurs de support, ou encore des méthodes plus avancées comme les réseaux de neurones. Ces modèles permettent de prédire les performances de différentes séquences d'opérations et d'identifier celles qui conduisent à une meilleure efficacité ou à des temps de production réduits.

Les réseaux de neurones, en particulier, sont largement utilisés dans les approches basées sur l'apprentissage et l'intelligence artificielle pour résoudre les problèmes du système job shop. Les réseaux de neurones peuvent être formés pour apprendre des règles complexes et des schémas dans les données historiques du système job shop. Ils peuvent être utilisés pour prédire les temps de traitement des différentes opérations, pour déterminer les priorités des tâches ou pour optimiser les séquences de production.

Une autre approche utilisée est l'apprentissage par renforcement. Cette méthode consiste à entraîner un agent intelligent à prendre des décisions dans un environnement complexe en récompensant les actions qui conduisent à de bonnes performances. Dans le contexte du système job shop, l'agent peut être entraîné à sélectionner les séquences d'opérations qui

minimisent les temps de production ou maximisent l'utilisation des ressources. L'apprentissage par renforcement nécessite souvent une quantité importante de données d'apprentissage et de temps de calcul, mais il peut aboutir à des stratégies d'ordonnancement optimales et adaptatives.

Ces approches basées sur l'apprentissage et l'intelligence artificielle ont un potentiel prometteur pour la résolution des problèmes du système job shop. Elles peuvent améliorer considérablement l'efficacité et les performances du système en exploitant des modèles et des techniques d'optimisation avancées. Cependant, il convient de noter que ces approches nécessitent souvent une quantité importante de données d'apprentissage de haute qualité et des ressources de calcul suffisantes pour effectuer des entraînements complexes.

IV.2.4 Les avantages et les inconvénients des méthodes approchées

Dans ce chapitre, nous avons exploré différentes méthodes de résolution pour les problèmes d'optimisation, en mettant l'accent sur les méthodes heuristiques et méta-heuristiques. Bien que ces méthodes offrent des avantages tels que leur capacité à trouver des solutions proches de l'optimum dans des délais raisonnables, elles présentent également quelques inconvénients.

L'un des principaux inconvénients des méthodes heuristiques et méta-heuristiques est qu'elles ne garantissent pas de trouver la solution optimale. Ces méthodes sont basées sur des règles empiriques et des stratégies de recherche approximatives, ce qui signifie qu'elles peuvent parfois donner des résultats sous-optimaux. Il est important de garder à l'esprit que les solutions trouvées par ces méthodes peuvent être satisfaisantes, mais pas nécessairement les meilleures.

Un autre inconvénient est que ces méthodes peuvent être sensibles aux paramètres choisis et aux conditions initiales. Des résultats différents peuvent être obtenus en modifiant légèrement les paramètres ou en utilisant différentes stratégies de recherche. Il est donc essentiel de comprendre ces paramètres et de les ajuster de manière appropriée pour obtenir des résultats de qualité.

En ce qui concerne les avantages des méthodes heuristiques et méta-heuristiques, nous pouvons prendre l'exemple de l'algorithme des essaims particulaires (PSO). Cette méthode s'inspire du comportement de recherche en groupe des oiseaux ou des poissons pour explorer l'espace des solutions. Les particules de l'algorithme PSO se déplacent dans l'espace de recherche en fonction de leur expérience personnelle et de l'expérience collective du groupe, afin de trouver une solution optimale.

Une autre méthode méta-heuristique couramment utilisée est l'algorithme génétique. Cette méthode utilise des concepts de la théorie de l'évolution, tels que la sélection, le croisement et la mutation, pour explorer l'espace des solutions. L'algorithme génétique est particulièrement efficace pour résoudre des problèmes d'optimisation combinatoire, tels que le problème du voyageur de commerce.

Ces méthodes heuristiques et méta-heuristiques offrent des approches flexibles et puissantes pour résoudre des problèmes d'optimisation complexes. Cependant, il est important de comprendre leurs limites et de les comparer aux méthodes exactes lorsque la recherche d'une solution optimale est primordiale. Il est également crucial d'ajuster les paramètres et de

sélectionner la méthode la mieux adaptée en fonction des caractéristiques spécifiques du problème et des contraintes de temps et de ressources disponibles.

Conclusion

En conclusion, après avoir examiné les problèmes d'ordonnancement et leurs principales caractéristiques dans le chapitre 1, ce chapitre présente différentes méthodes, à la fois exactes et approximatives, qui peuvent être utilisées pour résoudre ces problèmes. Parmi les méthodes exactes, nous avons abordé la recherche exhaustive, la programmation mathématique, les algorithmes de backtracking, la méthode Branch and Bound, ainsi que les méthodes basées sur les graphes de précédence.

Le recuit simulé, la Recherche Tabou, les algorithmes génétiques et les algorithmes de colonies de fourmis ainsi que les approches basées sur l'apprentissage et l'intelligence fait l'objet de la présentation pour les méthodes approchées ou méta- heuristiques.

Dans le chapitre suivant, nous appliquerons des règles de priorité pour faire une étude comparative afin d'extraire les meilleures solutions dans un système job shop à quatre machines.

Chapitre III : Étude comparative des règles de priorité destinées à l'affectation des tâches dans un atelier flexible de type job shop

III.1 Introduction

L'affectation des tâches dans un atelier flexible de type job shop est une tâche complexe qui implique la considération de multiples facteurs, tels que les contraintes de temps, les priorités des tâches et les capacités des ressources. Pour assurer une performance optimale de l'atelier, il est essentiel de mettre en place des règles de priorité appropriées afin de guider le processus d'affectation des tâches.

Cependant, la littérature propose de nombreuses règles de priorité, ce qui rend souvent difficile la détermination de celle qui convient le mieux à un contexte spécifique. Dans ce chapitre, nous présenterons une étude comparative entre différentes règles de priorité.

L'objectif de cette étude est de fournir aux décideurs et aux gestionnaires des outils d'analyse et d'aide à la décision pour sélectionner la règle de priorité la plus adaptée à leurs objectifs et contraintes spécifiques. Pour ce faire, nous avons examiné les différentes règles de priorité existantes, telles que la règle du plus court temps de traitement (SPT), la règle de la date d'échéance la plus proche (EDD), la règle du plus long temps de traitement (LPT), ainsi qu'une nouvelle règle que nous avons proposée. Nous avons utilisé le makespan comme indicateur de performance pour évaluer les résultats obtenus.

Cette étude comparative vise à approfondir notre compréhension des règles de priorité dans le contexte d'un atelier flexible de type job shop. Elle offre aux décideurs et aux gestionnaires des informations précieuses pour améliorer leurs processus d'affectation des tâches et prendre des décisions éclairées.

III.2 Etat de l'art : Les règles de priorité dans le contexte du job shop

Au fil des années, de nombreuses recherches se sont concentrées sur la résolution des problèmes d'ordonnancement dans le contexte du job shop. Ces études ont exploré différentes approches, qu'elles soient exactes ou approximatives, pour parvenir à des solutions optimales ou à des solutions proches de l'optimum. Les méthodes exactes, telles que la programmation linéaire ou la programmation par contraintes, visent à trouver la solution optimale en considérant toutes les contraintes et les combinaisons possibles. Cependant, ces approches peuvent être coûteuses en termes de temps de calcul, en particulier pour des problèmes de grande taille. Par conséquent, des méthodes approximatives ont également été développées, telles que les heuristiques de priorité que nous avons mentionnées précédemment. Ces heuristiques cherchent à fournir des solutions qui se rapprochent de l'optimalité en utilisant des règles de priorité basées sur des critères spécifiques. Ces approches sont souvent plus rapides à mettre en œuvre et peuvent être plus adaptées pour des problèmes de taille importante. L'objectif de ces travaux de recherche est d'améliorer l'efficacité des processus de production, en minimisant le temps d'exécution, en optimisant l'utilisation des ressources et en répondant aux contraintes spécifiques du job shop.

Dans le domaine des méthodes exactes, une contribution significative a été apportée par Pinson en 1988 [28] dans l'étude du problème du job shop en utilisant une modélisation basée sur un graphe disjonctif. L'auteur a développé un algorithme polynomial qui a été utilisé pour construire à la fois une heuristique et une méthode exacte. Cette approche a permis d'obtenir des solutions optimales pour le problème en tenant compte de toutes les contraintes et des différentes combinaisons possibles. Cela a constitué une avancée importante dans la résolution efficace du problème du job shop en utilisant des méthodes mathématiques rigoureuses.

En 2005, Zribi [34] a développé des méthodologies pour résoudre le problème du job shop flexible en prenant en compte les contraintes de disponibilité des machines. Deux méthodes ont été proposées dans cette étude. Tout d'abord, une méthode exacte de type Branch and Bound (B&B) a été utilisée pour trouver des solutions optimales en explorant de manière exhaustive l'espace de recherche. Ensuite, une méthode approchée basée sur une heuristique a été mise en œuvre pour obtenir une répartition efficace des charges de travail, suivie d'une recherche Taboue pour améliorer les solutions.

L'utilisation des règles de priorité dans le contexte du job shop a été largement étudiée et explorée dans la littérature scientifique. Ces règles jouent un rôle crucial dans la planification et l'ordonnancement des opérations dans les environnements de production complexes. L'état de l'art dans ce domaine se concentre sur le développement et l'évaluation de différentes heuristiques de priorité visant à optimiser des critères tels que le makespan, la minimisation des temps d'attente ou l'utilisation efficace des ressources. Les chercheurs ont proposé diverses règles de priorité, telles que SPT (Shortest Processing Time), LPT (Longest Processing Time), EDD (Earliest Due Date), ainsi que des approches plus sophistiquées basées sur des algorithmes génétiques, des méthodes de recherche locale ou des techniques d'apprentissage automatique. L'étude comparative de ces règles vise à identifier les approches les plus performantes pour résoudre les problèmes de job shop, offrant ainsi des pistes pour améliorer l'efficacité et l'optimisation des processus de fabrication.

En 1994, Penz [39] a présenté une approche novatrice pour résoudre les problèmes d'ordonnancement de type job shop. Cette approche a proposé une alternative aux algorithmes traditionnels basés sur des règles de priorité couramment utilisées. Le principe fondamental de cette approche consiste à construire une séquence d'ordonnancements partiels en agrégeant progressivement les jobs les uns après les autres.

En 1997, Ghedjati et Pomerol [40] ont publié un article proposant la résolution du problème d'ordonnancement d'ateliers de type "job-shop généralisé" en utilisant plusieurs méthodes heuristiques statiques et dynamiques originales. Leur approche tenait compte de la charge potentielle des machines au fur et à mesure de la construction de la solution, ce qui permettait une meilleure optimisation de l'utilisation des ressources.

En 1999, Hentous [41] a soutenu sa thèse qui traitait des problèmes d'ordonnancement de type job shop simple et job shop hybride. Ces problèmes appartiennent à la classe des problèmes NP-difficiles de la combinatoire. Pour aborder progressivement la complexité inhérente au job shop, il a proposé une première heuristique pour résoudre le cas à trois machines, puis a généralisé cette méthode pour résoudre des instances plus complexes.

Au cours des dernières années, plusieurs chercheurs ont également contribué à la résolution du problème du job shop. En 2016, Driss [42] a développé deux approches basées sur les algorithmes génétiques et la recherche "Coucou" pour minimiser la date de fin de toutes les opérations dans le job shop classique et/ou flexible. En 2019, Nadir [43] a proposé des méthodes pour résoudre ce problème d'ordonnancement dans un environnement dynamique, en visant à minimiser le makespan et à insérer de nouvelles commandes dans le plan prévisionnel.

Les règles de priorité se sont révélées efficaces dans de nombreuses situations de problèmes d'ordonnancement, notamment en ce qui concerne la minimisation des temps de calcul. Cependant, les résultats obtenus par ces méthodes ne sont généralement pas optimaux. C'est pourquoi des améliorations ont été développées pour se rapprocher davantage de la solution exacte.

En effet, des approches combinant différentes règles de priorité par agrégation ont été élaborées. Par exemple, la concaténation de l'algorithme de Johnson avec la règle FAM (First Available Machine) [3] a été utilisée pour résoudre le problème du flow-shop hybride à deux étages. Cette approche permet de tirer parti des avantages des différentes règles de priorité et d'obtenir de meilleures performances.

Plus récemment, d'autres extensions des règles de priorité ont été proposées pour résoudre des problèmes d'ordonnancement avec d'autres objectifs. Par exemple, le travail de Xiong et al [9], examine une analyse basée sur la simulation des règles de répartition pour l'ordonnancement job shop dynamique avec livraison par lots, tout en tenant compte de la contrainte de priorité technique étendue. Cette approche élargit le champ d'application des règles de priorité en prenant en considération des contraintes et des objectifs supplémentaires.

En 2022, Hadri [9] s'est intéressé à l'étude du problème d'ordonnancement en temps réel d'un atelier job shop en tenant compte des pannes du système de transport. Il a développé un programme linéaire en nombres entiers pour résoudre ce problème, ainsi qu'une méta-heuristique basée sur les algorithmes génétiques et plusieurs heuristiques basées sur les règles de priorité.

III.3 Méthodologie : Étude comparative des règles de priorité appliquées sur le job shop

III.3.1 Description du problème et notations

Le problème de Job Shop Scheduling (ordonnancement en atelier) est un problème d'optimisation combinatoire largement étudié en recherche opérationnelle. Il concerne l'ordonnancement de tâches (jobs) sur différentes machines dans un atelier de production. Chaque tâche est caractérisée par un ensemble d'opérations qui doivent être exécutées dans un ordre spécifique, et chaque machine peut exécuter une seule opération à la fois.

III.3.1.1 Caractéristiques du système de job shop étudié

Notre étude se concentre sur le système flexible de production iCIM 3000 situé dans le laboratoire de productique (MELT) à l'Université de Tlemcen. Le système flexible de

production iCIM 3000 est un système avancé de fabrication qui offre une solution intégrée pour la production, la planification et le contrôle de la qualité.

La configuration du système flexible de production iCIM 3000 est modulaire, ce qui permet de l'adapter facilement aux besoins de production spécifiques. Le système est constitué de différentes stations de travail et d'équipements tels que :

1) Le premier élément du système iCIM est un système de stockage/déstockage automatique (AS/RS), qui constitue le stock principal de toutes les matières utilisées pendant la production. Ce système englobe les matières premières, les produits semi-finis et les produits finis

2) Le deuxième composant du système iCIM est un système de convoyage utilisant des convoyeurs à palettes. Ce système permet le transfert des produits entre le stock principal et les différentes stations de travail, assurant ainsi un flux continu des matériaux dans le processus de production

3) Le troisième élément du système iCIM est la machine de fraisage 105, équipée d'un robot d'alimentation flexible. Cette machine est capable de réaliser des opérations de fraisage et de perçage sur les pièces.

4) Le quatrième élément du système iCIM est la machine de tours 105, équipée d'un robot d'alimentation flexible. Cette machine est dédiée à la production de pièces rotatives spécifiques

5) Le cinquième élément du système iCIM est la station de qualité et de manutention (QH 200). Cette station joue un rôle crucial dans le processus de production en assurant à la fois les tests de qualité des pièces et l'alimentation manuelle du système en palettes.

6) Le sixième élément du système iCIM est constitué des cellules d'assemblage robotisées flexibles (FAC). Ces cellules ont pour fonction principale d'effectuer l'assemblage des produits à partir des semi-produits créés par les machines CNC, ainsi que d'autres produits stockés dans la station de stockage automatique (AS/RS) [9].

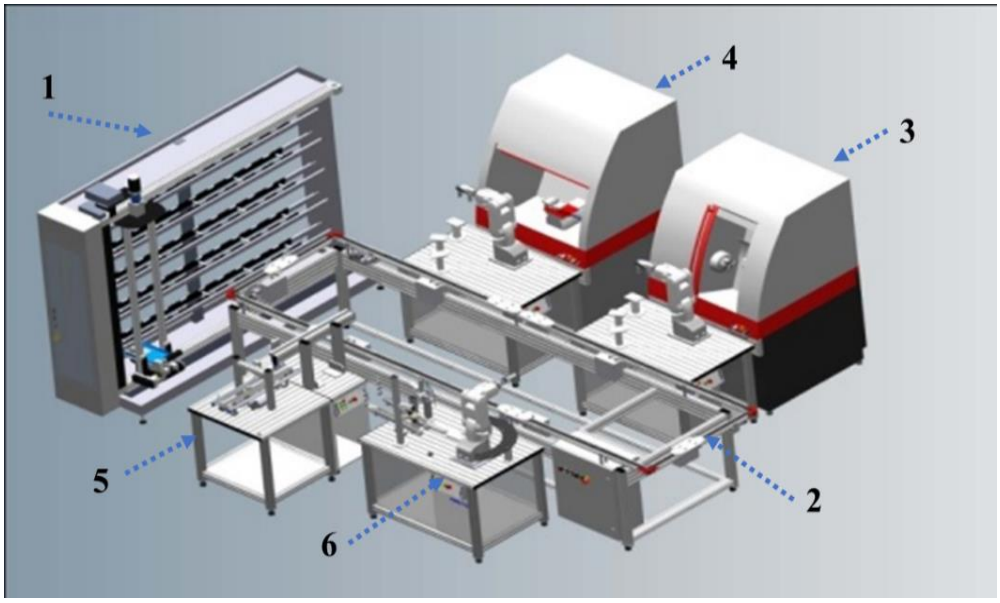


Figure III.1 : La configuration 3D du système iCIM 3000 [9]

Les stations de travail sont connectées entre elles par un convoyeur automatisé, qui transporte les pièces ou les produits entre les différentes étapes de production. Les pièces sont transportées entre les stations sur des palettes standard avec une capacité unitaire. Les palettes sont constamment disponibles et se déplacent (à vide) en permanence sur le tapis roulant.

Chaque station du système que nous étudions est capable de traiter une seule pièce à la fois, et dispose d'un stock de pièces considéré comme illimité. De plus, nous supposons que le temps de chargement, de déchargement et d'attente d'une pièce est inclus dans le temps de déplacement de cette pièce

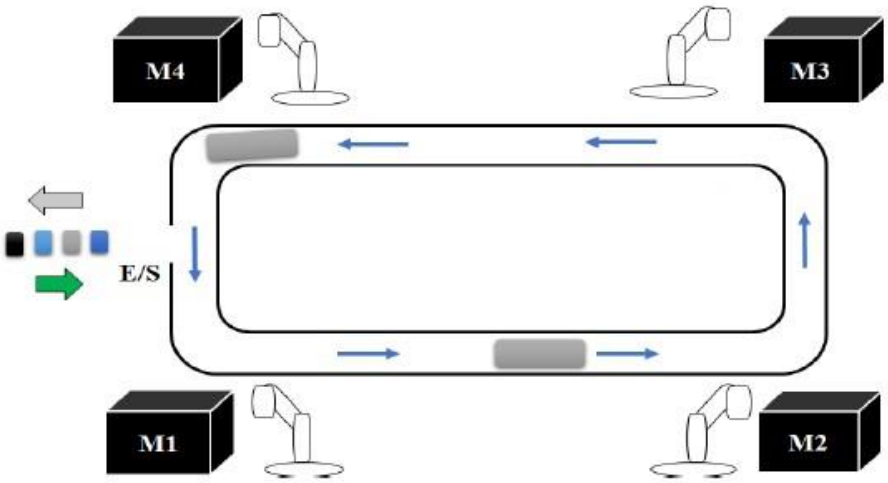


Figure III.2 : Sens de déplacement des palettes dans le système iCIM 3000 [9]

La configuration du système peut être adaptée pour répondre à différents besoins de production, tels que la production de petites séries, la production en flux tendu, la production à la demande, etc. Les différentes stations de travail peuvent être configurées pour traiter différents types de pièces ou de produits, ce qui permet de produire une grande variété de produits avec un même système.

Le système flexible de production iCIM 3000 est également évolutif, ce qui signifie qu'il peut être mis à jour ou amélioré avec des équipements plus récents ou plus performants. Il peut également être étendu pour ajouter de nouvelles stations de travail si nécessaire pour répondre à une demande de production croissante

III.3.1.2 Contraintes spécifiques au problème

Le problème d'ordonnancement traité dans le contexte de ce mémoire est un problème job shop où un ensemble de n jobs $J_1, J_2, \dots J_n$vont être exicutés sur m machines (stations) $M_1, M_2, \dots M_m$. Chaque job J_j est constitué de k tâches (ou opérations) à effectuer selon une séquence bien déterminée. Chaque tâche T_{ijk} du job j nécessite une durée de traitement D_{ijk} pur qu'elle soit traitée sur la machine i.

Les contraintes considérées sont :

- Une machine ne peut traiter qu'un seul produit à la fois
- Un produit ne peut s'exécuter que sur une seule machine à la fois
- Pas de préemption (le produit doit terminer son exécution avant de quitter la machine)
- Les tâches doivent être exécutées dans l'ordre spécifié pour chaque job.

Pour encadrer notre problématique nous avons proposé un ensemble d'hypothèses concernant le fonctionnement de notre système.

- Les produits sont toujours disponibles à l'instant t=0.
- Pas de panne de machines.
- Pas de recirculation (un produit ne peut exécuter sur une machine qu'une seule fois)
- En cas de conflit dans la séquence, le choix de produit à déplacer se fait selon les règles proposées.
- On ne prend pas en considération le temps de déplacement du ou vers le stock.

III.3.1.3 Fonction objectif

Dans le domaine de l'ordonnancement, il existe plusieurs fonctions objectifs couramment utilisées pour évaluer et optimiser les performances des systèmes de production (minimisation du temps d'attente, minimisation des coûts, minimisation de l'utilisation des ressources, minimisation de retarde etc).

Afin d'analyser l'efficacité des règles de priorité, dans notre étude, nous avons choisi la fonction objectif minimisation du makespan (C_{max}). Le makespan représente la durée totale nécessaire pour compléter l'ensemble des tâches dans le système.

En optimisant le makespan, nous visons à minimiser le temps global de production et à améliorer l'efficacité du processus. En réduisant le makespan, on améliore l'utilisation des ressources et la productivité globale. Cela peut entraîner des économies de coûts et une meilleure rentabilité.

Alors, la fonction objectif est décrit par la formule 01, dont le C_j^k détermine la date de fin d'exécution du job J_j dans la machine M_k

$$\text{Min } (C_{max}) = \min [max_j^k(C_j^k)] \qquad (01)$$

III.3.2 Présentation des règles de priorité

Dans cette partie, nous procéderons à une présentation détaillée des différentes règles de priorité couramment utilisées dans l'affectation des tâches dans un atelier flexible de type job shop. Ces règles de priorité jouent un rôle crucial dans l'optimisation des opérations de production et permettent de guider le processus d'affectation des tâches en fonction de différents critères.

Nous commencerons par examiner les règles les plus couramment utilisées (SPT, LPT, EDD…), Ensuite, nous présenterons la nouvelle règle que nous avons proposée dans le cadre

de cette étude comparative. Cette règle repose sur des critères spécifiques et offre une approche novatrice pour l'affectation des tâches dans un atelier flexible de type job shop.

Cette section vise à fournir une vue d'ensemble complète des règles de priorité utilisées dans l'affectation des tâches d'un atelier flexible de type job shop. Cette présentation approfondie des règles de priorité servira de base pour la comparaison ultérieure de leur performance dans la section suivante

III.3.2.1 Shortest Total Processing Time (SPTsystème)

La règle « SPTsystème » consiste à ordonner les jobs en fonction de la somme croissante de leurs temps de traitement. Cette somme est obtenue en additionnant les durées de toutes les opérations d'un même job. Prenons, par exemple, le job J_1 composé de deux opérations T_{111} et T_{212}, avec des temps de traitement respectifs de 5 et 7 unités. La somme totale est donc de 12 unités de temps. D'autre part, si un deuxième job J_2 a un temps de traitement estimé de 4 unités de temps sur la machine M_1 et de 6 unités de temps sur M_2, la somme totale est de 10 unités de temps. Selon la règle « SPTsystème » le job J_2 sera exécuté en premier sur le système.

III.3.2.2 Longest Total Processing Time (LPTsystème)

A l'inverse de l'heuristique précédente cette règle consiste à ordonnancer les jobs dans l'ordre décroissant de leurs sommes de temps de traitement. Dans ce cas, pour l'exemple donné précédemment, le job J_1 serait exécuté en premier. Cela signifie que la tâche avec la somme totale la plus élevée serait priorisée par rapport à celle avec la somme totale la plus faible.

III.3.2.3 Shortest Processing Time in machine (SPTmachine)

La règle "SPTmachine" est une heuristique qui consiste à ordonnancer les jobs sur chaque machine selon l'ordre croissant de leurs temps de traitement sur cette machine. Cela signifie que les tâches nécessitant moins de temps de traitement devront exécuter en premier, Par exemple, si deux jobs J_1 et J_2 vont être réalisés sur la machine M_1 avec un temps de traitement 5 et 7 respectivement, la priorité selon la règle "SPTmachine" est donnée au job J_1.

III.3.2.4 Longest Processing Time in machine (LPTmachine)

Cette heuristique consiste à ordonnancer les jobs dans l'ordre décroissant de leurs temps de traitement sur chaque machine. Cela signifie que les tâches qui nécessitent plus de temps de traitement sont exécutées en premier. Alors pour l'exemple des deux jobs J_1 et J_2 qui ont les temps 5 et 7 respectivement sur la machine M_1 le job J_2 est prioritaire selon cette règle.

III.3.2.5 Short Accumulation Processing Time (SPTcumule)

L'heuristique SPTcumule consiste à ordonnancer les jobs dans l'ordre croissant de leurs temps de traitement cumulé sur chaque machine. Pour cette heuristique, lorsqu'une machine est libérée, le travail choisi pour être traité en premier est celui qui a la valeur cumulée la plus

faible. Cette valeur cumulée est calculée par la somme des durées de toutes les opérations précédentes (précédentes de l'opération à effectuer sur cette machine). Cette technique vise à placer au début de l'ordonnancement les jobs qui consomment le moins de temps d'exécution pour quitter la machine.

III.3.2.6 Earliest Due Date (EDD)

La règle EDD (Earliest Due Date), également appelée "règle du plus proche délai", est une méthode de gestion de file d'attente utilisée pour ordonner les tâches en fonction de leur date d'échéance. Selon cette règle, la tâche avec la date d'échéance la plus proche est traitée en premier, avant les autres tâches en attente. « Pour notre problème la date due de tous les jobs est la somme des temps de traitement »

III.3.2.7 Slack Time Remaining (STR)

La Slack Time Remaining (STR), également appelée "marge de temps disponible", est une mesure de la quantité de temps qu'une tâche peut être retardée sans affecter la date d'échéance du projet (pour notre problème nous avons considéré la somme des temps de traitement de tous les jobs comme date d'échéance du projet). La marge « STR » est calculée en soustrayant le temps de traitement estimé de la tâche de la date d'échéance du projet.

Par exemple, si un job a une durée de traitement estimée de 5 jours et que la date d'échéance est dans 10 jours, la STR est de 5 jours. Cela signifie que le job peut être retardé jusqu'à 5 jours sans affecter la date d'échéance.

La mesure de la STR est utile pour identifier les tâches qui peuvent être retardées sans affecter le résultat final du projet.

III.3.2.8 Critical Ratio (CR)

Pour cette heuristique les jobs sont ordonnancés selon l'ordre croissant du Ratio Critique (CR). Ce ratio est calculé en divisant la date d'échéance sur le temps de traitement estimé de la tâche. C'est-à-dire CR = Due date / Procession time

III.3.2.9 La règle proposée Operating Range Sequence (ORS)

La règle ORS, est une règle de priorité qui vise à affecter les opérations disponibles aux machines disponibles en suivant une approche séquentielle. L'idée de base de la règle ORS est de commencer par la première opération de chaque job et de les attribuer à la machine disponible. Si plusieurs jobs partagent la même machine, la priorité est donnée au job ayant le numéro le moins élevé.

L'objectif principal de la règle ORS est de minimiser le temps d'attente. En donnant la priorité à la première opération disponible de chaque job, cette règle vise à réduire le temps nécessaire pour commencer le traitement de chaque job.

Exemple illustratif d'un problème 4 jobs

Afin de bien illustrer les règles de priorité présentées auparavant, nous allons présenter un exemple de quatre jobs qui vont être exécutés sur les quatre machines. Les temps de traitement ainsi que les gammes opératoires des jobs sont montrés dans les deux tableaux suivants.

Jobs / Machines	M1	M2	M3	M4	Somme
J1	6	7	11	0	24
J2	4	5	4	7	20
J3	9	10	10	9	38
J4	8	6	7	8	29

Tableau III.1: Temps opératoires sur chaque machine

Job / opération	Opération 1	Opération 2	Opération 3	Opération 4
J1	M1	M3	M2	/
J2	M3	M2	M1	M4
J3	M1	M3	M2	M4
J4	M3	M4	M2	M1

Tableau III.2 : La gamme opératoire des jobs

La séquence des jobs, la valeur de C_{max}. et le diagramme de GANTT obtenus par l'application de chaque règle sont les suivants :

a. Application de la règle SPTsystème

- La séquence :

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J2	J1	J4	J3
M2	J2	J1	J4	J3
M3	J2	J1	J4	J3
M4	J2	J4	J3	/

Tableau III.3: La séquence des jobs dans les machines selon SPT système

- La valeur de C_{max} = 97
- Le diagramme de Gannt :

Figure III.3: Diagramme de Gantt selon la règle SPT système

b. Application de la règle LPT système

- La séquence :

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J3	J4	J1	J2
M2	J3	J4	J1	J2
M3	J3	J4	J1	J2
M4	J3	J4	J2	/

Tableau III.4 : La séquence des jobs dans les machines selon LPT système

- La valeur de C_{max} = 100
- Le diagramme de Gannt :

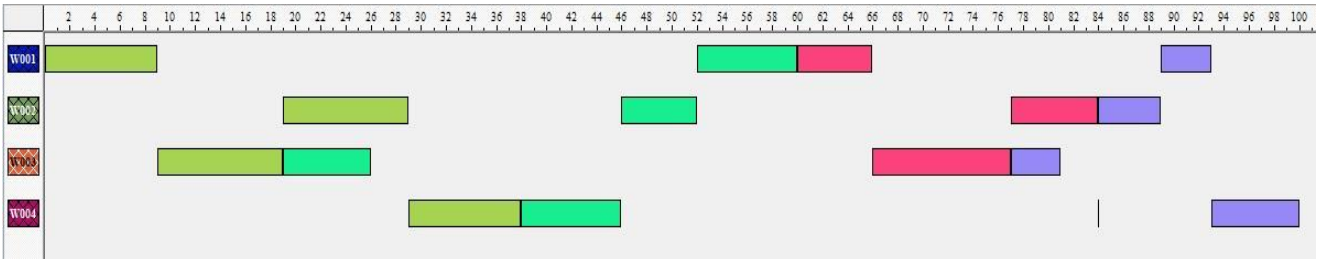


Figure III.4 : Diagramme de Gantt selon la règle LPT système

c. Application de la règle SPT machine

- La séquence :

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J2	J1	J4	J3
M2	J2	J4	J1	J3
M3	J2	J4	J3	J1
M4	J2	J4	J3	/

Tableau III.5 : La séquence des jobs dans les machines selon SPT machine

- La valeur de C_{max} = 98
- Le diagramme de Gannt :

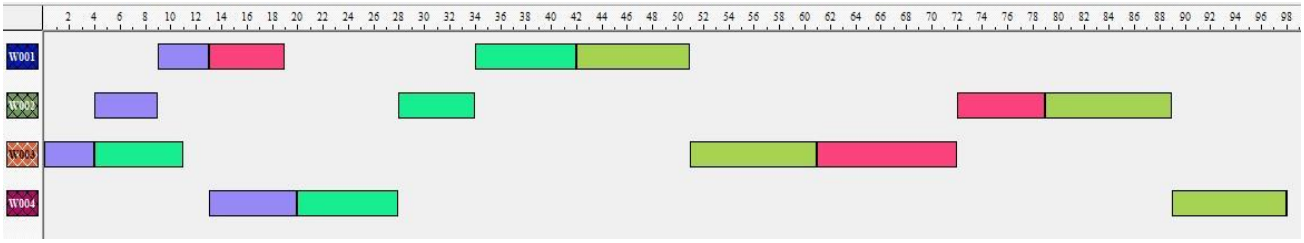


Figure III.5 : Diagramme de Gantt selon la règle SPT machine

d. Application de la règle LPT machine

- La séquence :

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J3	J3	J1	J2
M2	J3	J1	J4	J2
M3	J1	J3	J4	J2
M4	J3	J4	J2	/

Tableau III.6 : La séquence des jobs dans les machines selon LPT machine

Selon la règle LPT machine on à une solution inadmissible dans cet exemple

e. Application de la règle SPT cumule

- La séquence des jobs dans les machines selon SPT cumule :

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J2	J1	J4	J3
M2	J2	J1	J4	J3
M3	J2	J4	J1	J3
M4	J2	J4	J3	/

Tableau III.7 : La séquence des jobs dans les machines selon SPT cumule

- La valeur de C_{max} = 89
- Le diagramme de Gannt :

Figure III.6 : Diagramme de Gantt selon la règle SPT cumule

f. Application de la règle EDD

- La séquence :

Jobs	J1	J2	J3	J4
Due Date	45	35	40	50

Tableau III.8 : La date d'échéance des jobs

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J2	J3	J1	J4
M2	J2	J3	J1	J4
M3	J2	J3	J1	J4
M4	J2	J3	J4	/

Tableau III.9 : La séquence des jobs dans les machines selon EDD

- La valeur de C_{max} = 73
- Le diagramme de Gannt :

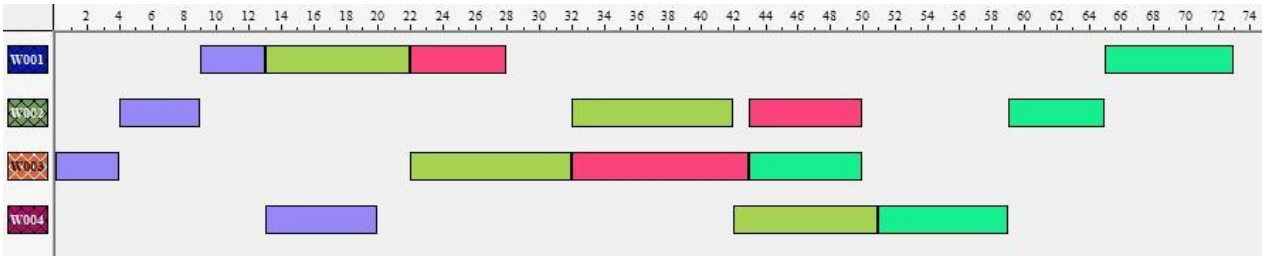


Figure III.7 : Diagramme de Gantt selon la règle EDD

g. Application de la règle STR

Jobs	J1	J2	J3	J4
Slack Time	21	15	2	21

Tableau III.10 : Marge de temps disponible des jobs

- La séquence :

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J3	J2	J1	J4
M2	J3	J2	J1	J4
M3	J3	J2	J1	J4
M4	J3	J2	J4	/

Tableau III.11 : La séquence des jobs dans les machines selon STR

- La valeur de C_{max} = 84
- Le diagramme de Gannt :

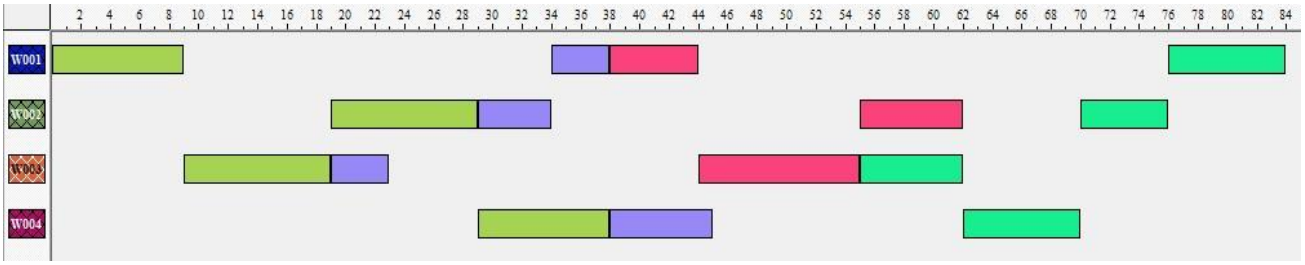


Figure III.8 : Diagramme de Gantt selon la règle STR

h. Application de la règle CR

Jobs	J1	J2	J3	J4
Critical Ratio	1.87	1.75	1.05	1.72

Tableau III.12 : Ratio Critique des jobs

- La séquence :

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J3	J4	J2	J1
M2	J3	J4	J2	J1
M3	J3	J4	J2	J1
M4	J3	J4	J2	/

Tableau III.13 : La séquence des jobs dans les machines selon CR

- La valeur de C_{max} = 88
- Le diagramme de Gannt :

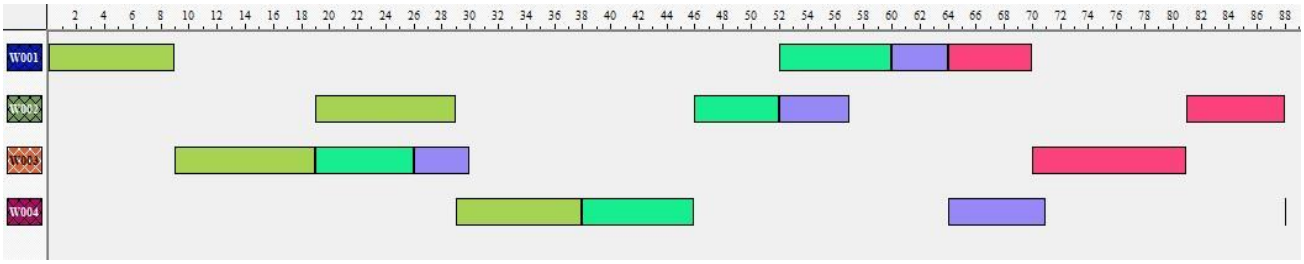


Figure III.9 : Diagramme de Gantt selon la règle CR

i. Application de la règle ORS

- La séquence :

	Ordre 01	Ordre 02	Ordre 03	Ordre 04
M1	J1	J3	J2	J4
M2	J2	J1	J3	J4
M3	J2	J4	J1	J3
M4	J4	J2	J3	/

Tableau III.14 : La séquence des jobs dans les machines selon ORS

- La valeur de C_{max} = 56
- Le diagramme de Gannt :

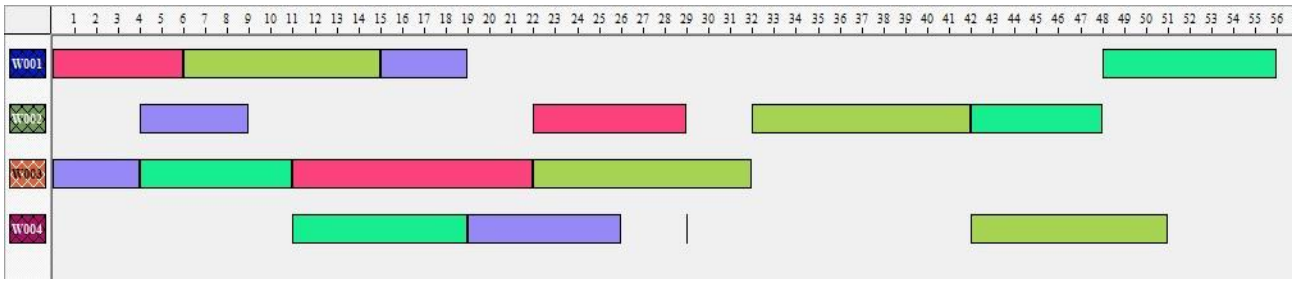


Figure III.10 : Diagramme de Gantt selon la règle ORS

III.3.3 Méthodes expérimentales et mesures de performance

Pour simuler les règles de priorité utilisées, nous avons développé un programme sur le logiciel Matlab qui permet de calculer la valeur de C_{max} (le makespan) pour chaque règle. Nous avons divisé notre application, selon la taille des instances choisies, en trois parties : les petites instances ; les moyennes instances et les grandes instances. Pour chaque instance, nous avons effectué 10 réplications afin d'obtenir la moyenne de C_{max} calculé.

En utilisant la fonction objectif du makespan, nous pouvons évaluer les performances des différentes règles de priorité en comparant les valeurs de makespan obtenues pour chaque règle. Cela nous permet de déterminer quelle règle de priorité est la plus efficace pour réduire les temps d'exécution et optimiser la séquence des tâches dans le job shop.

III.3.3.1 Résultats obtenus pour les petites instances

Dans cette première partie de test nous avons étudié trois problèmes de petites tailles (2x4, 3x4 et 4x4). Les temps de traitement des jobs sont générés aléatoirement entre 1 et 50 unités de temps. Le test des règles proposées effectué sur dix réplications donne les résultats mentionnés dans Le Tableau III.15.

À travers les résultats obtenus en appliquant ces méthodes sur les trois instances, on constate que pour l'instance (2x4), l'heuristique SPTmachine basée sur le choix de la petite valeur du temps de traitement, génère des solutions avec une moyenne de C_{max} minimale par rapport aux autres règles.

En ce qui concerne l'instance (3x4), on remarque que l'heuristique LPTmachine basée sur le choix de la plus grande valeur du temps de traitement, fournit des solutions avec une moyenne de C_{max} minimale par rapport aux autres règles. Pour l'instance (4x4), on constate que l'heuristique CR qui dépendent de ratio critique donne des valeurs de C_{max} minimale par rapport aux autres règles.

Nous pouvons remarquer également, à travers les résultats mentionnés dans le tableau 15, que la règle proposée (ORS) produire des meilleures solutions en termes de C_{max} par rapport aux règles. L'efficacité de cette heuristique est bien prouvée pour toutes les instances.

	SPT système	LPT système	EDD	STR	CR	SPT machine	LPT machine	SPT cumule	ORS
2 produits	178.2	173.1	169.7	170.5	173.1	162.3	164.4	186.4	140.8
3 produits	230.7	231.3	241.4	235	229.9	240	227	230.3	168.3
4 produits	202.3	198.9	216.1	202.5	197.1	226.1	254	226.4	145.4

Tableau III.15 : Les résultats de C_{max} obtenu par MATLAB pour les petites instances

III.3.3.2 Résultats obtenus pour les moyennes instances

Dans cette partie, l'expertise a été faite sur les problèmes de 8, 10 et 15 jobs qui doivent être réalisés sur les quatre machines. La variation du temps de traitement des opérations dans ce cas aussi suit une loi uniforme entre 1 et 50. Le test des règles proposées effectué sur dix exemples donne les résultats mentionnés dans Le Tableau III.16.

A travers ces résultats, nous pouvons constater que pour l'instance (8x4) l'heuristique SPTsystème basée sur le choix de la petite valeur de la somme du temps de traitement, fournit des solutions avec une moyenne minimale de C_{max}. D'autre part, pour les instances (10x4) et (15x4), l'heuristique SPTcum, basée sur le choix de la petite valeur du temps de traitement cumulé, génère des solutions meilleures par rapport aux autres approches utilisées. En comparant les valeurs de makespan obtenues pour chacune de ces deux règles avec la règle ORS nous constatons que cette dernière fournit des meilleures solutions de C_{max} pour les trois instances.

	SPT Système	LPT système	EDD	STR	CR	SPT machine	LPT machine	SPT cumule	ORS
8 produits	449.8	457.3	469.9	455.6	457.7	608.3	589.6	471.6	327.4
10 produits	632.5	602	607.4	603.6	595.3	715.6	747	586	383.2
15 produits	927.1	922.9	943.4	950.3	926	1331	1326	883.6	538.7

Tableau III.16 : Les résultats de C_{max} obtenu par MATLAB pour les moyennes instances

III.3.3.3 Résultats obtenus pour les grandes instances

Les problèmes sélectionnés dans cette partie sont les problèmes de taille 20, 30 et 50 jobs qui doivent être réalisés sur les quatre machines. Comme dans les parties précédentes, Les temps de traitement sont générés aléatoirement entre 1 et 50 unités de temps. Le test des règles proposées effectué sur dix réplications donne les résultats mentionnés dans Le Tableau III.17.

Là encore, on peut noter que l'heuristique SPTcum, qui se base sur la sélection de la plus petite valeur de temps de traitement cumulé, conduit à des valeurs faibles de C_{max} pour l'instance (20x4). De même, pour les instances (30x4) et (50x4), l'heuristique SPTsystème, qui priorise la plus petite somme des temps de traitement, génère des solutions avec un moyen de C_{max} minimisé.

Cependant, en comparant les valeurs de makespan obtenues pour chacune de ces deux règles avec la règle ORS, il est clair que cette dernière fournit les meilleures solutions de C_{max} pour les trois instances étudiées.

Le point commun entre les trois catégories des instances est que l'heuristique ORS donne toujours les meilleures solutions avec des valeurs de C_{max} plus minimales. Ces valeurs démontrent clairement l'efficacité de cette heuristique par rapport aux autres.

	SPT Système	LPT système	EDD	STR	CR	SPT machine	LPT machine	SPT cumule	ORS
20 produits	1097.9	1091.7	1100.5	1118	1115.6	1737.6	1732.6	1072	654.6
30 produits	1716.1	1957.5	2088	2016.4	1959.2	2614.5	2632.5	1772	904.5
50 produits	2562.5	3465.1	3250	3454.8	3251.1	4738	4721.3	3138.5	1443

Tableau III.17 : Les résultats de C_{max}obtenu par MATLAB pour les grandes instances

III.3.3.4 Comparaison entre la règle ORS et la méthode exacte

Après les bons résultats montrés par la règle ORS par rapport au reste des règles, Nous avons décidé de le comparer avec la solution exacte, Pour cela nous avons utilisé un modèle Mathématique que existe déjà dans [44] et nous avons simule le model sur CPLEX pour résoudre des problèmes de petit instance (3x4) et (4x4).

Les résultats présentés dans la figure III.11 et III.12 illustrent la différence entre la solution obtenus par la règle ORS et la solution obtenus par la méthode exacte, on voit clairement que la moyenne de C_{max}obtenus par le modèle CPLEX est faible par rapport à la moyenne de C_{max}obtenus par la règle ORS, Par conséquent, la règle ORS n'obtient pas la solution optimale, mais plutôt la solution approximative avec des performances très proche des solutions optimales obtenues par CPLEX.

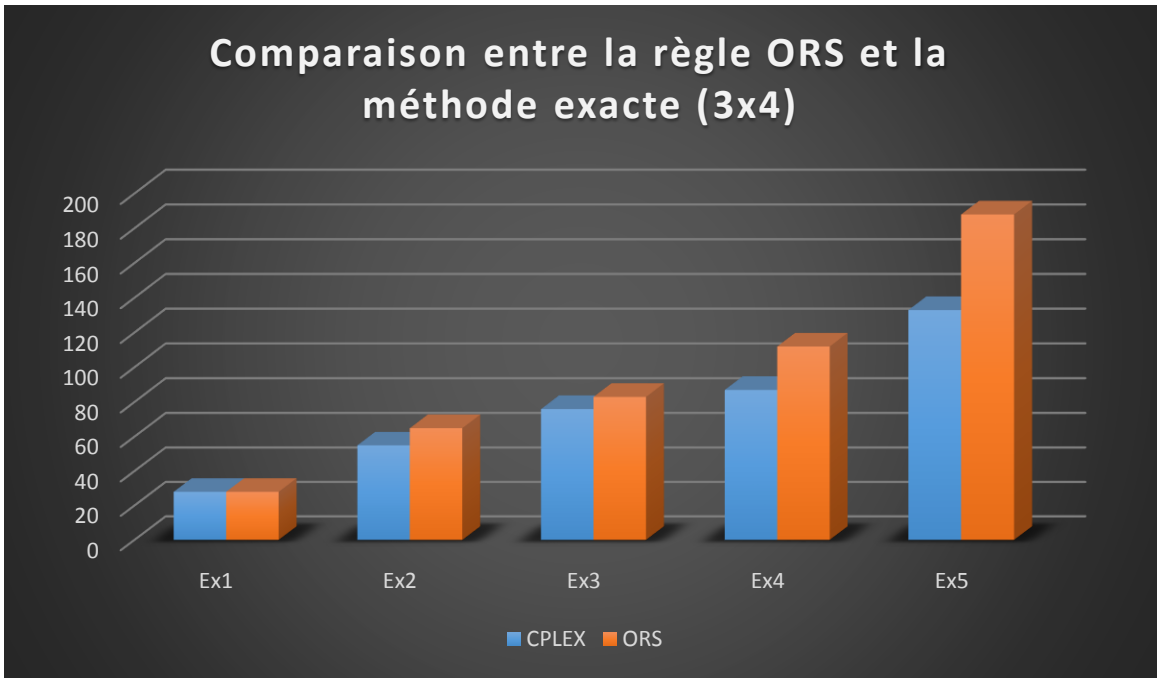


Figure III.11 : Les résultats de C_{max}obtenu par la règle ORS et le modèle CPLEX (3x4)

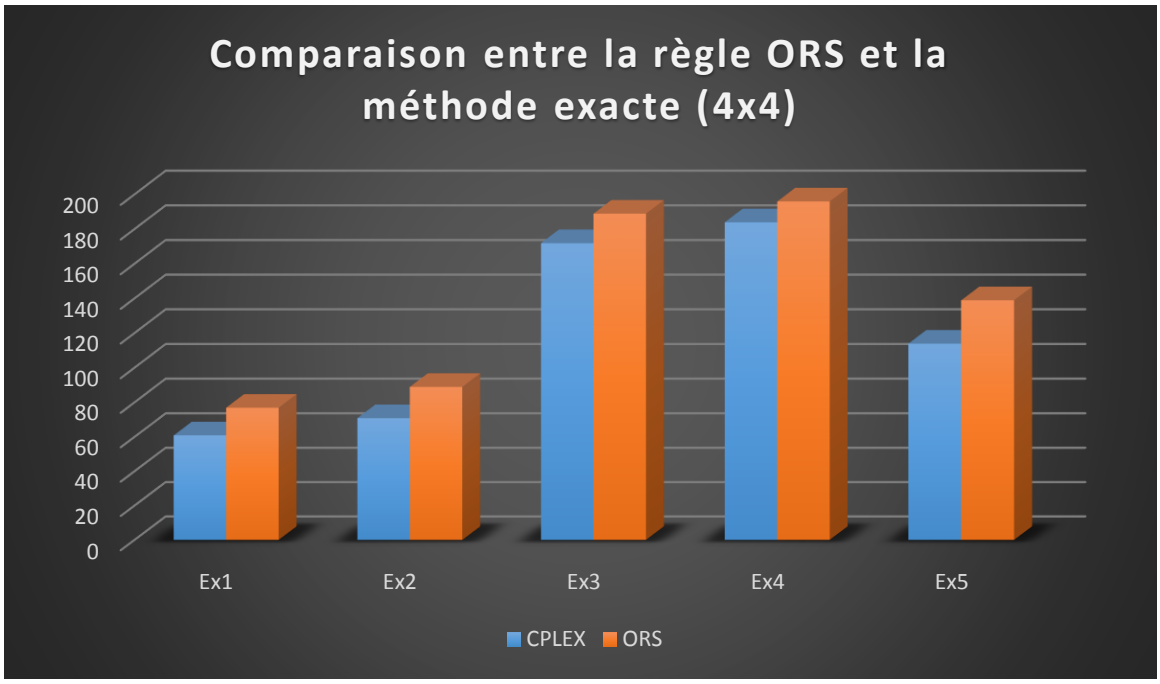


Figure III.12 : Les résultats de C_{max}obtenu par la règle ORS et le modèle CPLEX (4x4)

IV. Conclusions de l'étude comparative

La conclusion de l'étude comparative des règles de priorité appliquées sur le job shop met en évidence l'efficacité supérieure de la règle **ORS** par rapport aux autres règles examinées. L'objectif de cette étude était de déterminer la règle de priorité la plus adaptée pour optimiser les opérations dans un environnement de job shop, où différentes tâches doivent être planifiées et exécutées sur différentes machines.

Au cours de cette étude, plusieurs règles de priorité ont été évaluées, notamment la règle SPTmachine, la règle SPTsystème, la règle du plus court délai (EDD) et la règle de séquence la plus critique (CR).

Les résultats obtenus ont clairement démontré que la règle **ORS** surpassait les autres règles en termes de performances globales. La règle **ORS**, qui signifie " **Operating Range Sequence** ", se base sur le principe de donner la priorité aux tâches qui sont les plus disponibles. Cette approche permet, non seulement de minimiser le makespan, mais aussi de réduire les temps d'attente, d'équilibrer la charge de travail entre les différentes machines et d'optimiser l'utilisation des ressources disponibles.

En comparaison avec les autres règles, la règle **ORS** a montré une amélioration significative en termes de temps de traitement global. Elle a permis d'obtenir des performances plus élevées en minimisant les temps d'inactivité des machines et en favorisant une meilleure utilisation des ressources.

En complément de notre étude comparative des règles de priorité pour l'affectation des tâches dans un atelier flexible de type job shop, nous avons réalisé une analyse de performance spécifique à la règle Operating Range Sequence (ORS). Dans le cadre de cette analyse, nous avons comparé les résultats obtenus avec la règle ORS à des solutions optimales fournies par le logiciel **CPLEX**.

Les résultats de cette analyse démontrent que la règle **ORS** se rapproche significativement de l'optimum dans la plupart des cas étudiés. Bien que les solutions générées par la règle ORS ne soient pas strictement optimales, elles se situent dans une fourchette de performance très proche des solutions optimales obtenues par **CPLEX**.

Cette constatation confirme la pertinence et l'efficacité de la règle ORS dans la résolution du problème d'affectation des tâches dans un contexte de job shop. En offrant des résultats de qualité comparable à des solutions optimales, la règle ORS se positionne comme une méthode prometteuse pour les décideurs et les gestionnaires à la recherche de solutions réalisables et performantes dans des délais raisonnables.

Conclusion générale

Le problème d'ordonnancement des ateliers de type Job Shop est un sujet de recherche largement exploré et considéré comme extrêmement complexe. Il est classé parmi les problèmes combinatoires difficiles au sens fort, en raison de l'explosion combinatoire du nombre de solutions qui croît de manière exponentielle avec la taille du problème. Par conséquent, l'utilisation de méthodes exactes pour obtenir des solutions optimales apparaît souvent irréaliste.

Face à cette complexité, l'utilisation de méthodes approchées, telles que les heuristiques, est devenue incontournable. Bien que ces approches ne garantissent pas l'optimalité, elles permettent de trouver des solutions de qualité acceptable dans un délai raisonnable. Les heuristiques présentent des avantages significatifs dans la résolution du problème du job shop. Elles sont généralement plus rapides que les méthodes exactes et peuvent être adaptées pour tenir compte des caractéristiques spécifiques du système de production.

Dans ce contexte, notre mémoire s'est concentré sur les problèmes d'ordonnancement des ateliers flexibles de production, plus précisément les problèmes de job-shop. L'objectif est de faire une étude comparative sur les règles de priorité destinées à l'affectation des tâches dans ce type d'atelier. En examinant différentes règles de priorité, nous avons cherché à identifier celles qui offrent les meilleures performances en termes d'efficacité et de qualité des solutions obtenues.

Au terme de cette étude comparative, nous avons pu analyser et évaluer les performances de plusieurs règles classiques ainsi que d'une nouvelle règle que nous avons proposée, à savoir la règle Operating Range Sequence (ORS). Cette étude, basée sur plusieurs exemples des différentes instances, nous a permis de mieux comprendre l'impact de ces règles sur les performances globales de l'atelier et d'identifier celle qui donne les meilleurs résultats.

Nos résultats ont révélé que la règle « ORS » se distingue par sa capacité à améliorer significativement les performances de l'atelier flexible. En considérant non seulement les temps de traitement et les dates d'échéance, mais également les plages opératoires des ressources, la règle ORS permet de mieux équilibrer la charge de travail et de minimiser les temps d'attente. Cette approche innovante et adaptative s'est avérée particulièrement efficace pour maximiser l'utilisation des ressources et réduire les temps de production.

Cette étude comparative fournit aux décideurs et aux gestionnaires un ensemble d'outils d'analyse et d'aide à la décision pour sélectionner la règle de priorité la plus appropriée en fonction de leurs besoins. Les résultats obtenus permettent de prendre des décisions éclairées pour optimiser l'affectation des tâches dans un atelier flexible de type job shop.

À la lumière des résultats prometteurs obtenus avec la règle « Operating Range Sequence (ORS) » dans notre étude comparative, plusieurs perspectives intéressantes peuvent être explorées pour améliorer davantage l'affectation des tâches dans un atelier flexible de type job shop. Parmi ces perspectives, nous proposons notamment la combinaison de la règle ORS avec d'autres règles de priorité telles que la règle du plus court temps de traitement (SPT), la règle du plus court délai (LPT) ou encore la règle de la date d'échéance la plus proche (EDD).

La combinaison de règles de priorité peut offrir des avantages complémentaires en exploitant les points forts de chaque règle pour optimiser les performances globales de l'atelier. Par exemple, en combinant la règle ORS avec la règle SPT, on peut bénéficier à la fois d'un équilibrage de la charge de travail basé sur les plages opératoires des ressources et d'une réduction des temps de traitement grâce à la priorisation des tâches à court temps d'exécution.

Bibliographie

[01] Giard V. (1988). Gestion de la production, 2ème Edition, Economica, Paris.

[02] Lacroix J. & Bonenfant J. (2016). La production, chambre de commerce et d'industrie de paris.

[03] Johnson, S. M., (1954) Optimal two-and three-stage production schedules with setup times included.

[04] Groover, M. P. (2016). Automation, production systems, and computer-integrated manufacturing (éd. 4th). Pearson Higher Education.

[05] Letouzey A, (2001) .Ordonnancement interactif basé sur des indicateurs : Application à la gestion de commandes incertaine et à l'affectation des opérateurs, Thèse de doctorat, Institut National Polytechnique de Toulouse.

[06] Javel G, (2004). Organisation et Gestion de production, 3ème édition,DUNOD,Paris.

[07] Editions ellipses.fr (2018). Introduction aux Systèmes d'Information, 1-8.

[08] Shivanand, H. K., Benal, M., & Koti, V. (2006). Flexible manufacturing system. New Age International.

[09] Hadri Abdelkader , Contribution à la résolution des problèmes d'ordonnancement en temps réel dans un système de production de type Job Shop , Thèse de Doctorat .Université Batna 2

[10] Thomas Morton and David W Pentico. (1993) Heuristic scheduling systems : with applications to production systems and project management, volume 3. John Wiley & Sons.

[11] Joseph YT Leung. (2004).Handbook of scheduling : algorithms, models, and performance analysis. Chapman and Hall/CRC, Boca Raton, Florida.

[12] Michael Pinedo. (2016) Scheduling : Theory, Algorithms, and Systems. Springer.

[13] Laribi, I. (2018). Résolution de problèmes d'ordonnancement de type Flow-Shop de permutation en présence de contraintes de ressources non-renouvelables. Thèse de Doctorat. Université de Tlemcen.

[14] M. Sayed-Mouchaweh. (2012) Decentralized Fault Free Model Approach for Fault Detection and Isolation of Discrete Event Systems. European Journal of Control, 18 82-93.

[15] N. Mouhoub, (2011) Algorithmes de construction de graphes dans les problèmes d'ordonnancement de projet, Sétif, Thèse de Doctorat.

[16] M. kebabla, (2008) Utilisation des stratégies Métaheuristique pour l'ordonnancement des ateliers de type Job Shop, Batna, Mémoire de Magister.

[17] Esquirol, P., & Lopez, P. (1999). L'ordonnancement. Economica. Paris.

[18] Billaut, J. C., & Roubellat, F. (1996). A new method for workshop real time scheduling. International Journal of Production Research, 34(6), 1555-1579.

[19] H. Boukef Ben Othman, (2009) Sur l'ordonnancement d'ateliers job-shop flexibles et flowshop en industries pharmaceutiques optimisation par algorithmes génétiques et essaims particulaires, Tunis, Thèse de Doctorat.

[20] Chergui .A & Dahmani. A & Adda abbou .A. (2017). Ordonnancement d'un flow-shop par métaheuristique hybride . Université Abou Bekr Belkaid – Tlemcen.

[21] Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan., A.H.G. (1979), Optimisation and approximation in deterministic sequencing and scheduling: a survey, Annals of Discrete Mathematics; 5: 236-287.

[22] Y. Bahmani, (2017) Optimisation multicritère de l'ordonnancement des activités de la production et de la maintenance intégrées dans un atelier Job Shop, Batna, Thèse de Doctorat.

[23] Monmarché, N. (2000). Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation (Doctoral dissertation, Université François Rabelais-Tours).

[24] YASMINA, A. Une approche basée sur l'optimisation des colonies de fourmis pour la recherche binaire.

[25] Gherboudj, A. (2013). Méthodes de résolution de problèmes difficiles académiques. *Université de Constantine2*.

[26] MAREDJ, A., IDER, A., & HAMMOU, A. La Programmation par Contraintes pour le formatage spatial d'un document.

[27] Grainia, S. (2015). L'algorithme de Branch and Price and Cut pour le problème de conception de réseaux avec coûts fixes et sans capacité.

[28] Pinson, E. (1988). Le problème de job-shop (Doctoral dissertation, Paris 6).

[29] LOUCIF, R. (2014). Parallélisation d'algorithmes d'optimisation combinatoire (Doctoral dissertation, Université de Batna 2).

[30] HEMMAK, A. (2017). Support de cours d'optimisation combinatoire Focus sur les méthodes de résolution approchée.

[31] B. C. &. M. Abdelatif, (2015) «Résolution des problèmes d'optimisation par algorithmes évolutionnaires inspirées-,» Université L'arbi Ben M'hidi -Oum El Bouaghi.

[32] E.H.L. Aarts& J. Korst, (1989) Simulated annealing, boltzmann machines: a stochastic approach to combinatorial and neural computing. Wiley edition, Chichester.

[33] Préaux, J. P. (s.d). Méthode du recuit simulé. http://www.i2m.univ-amu.fr. Consulter le 2023-03-2, à l'adresse http://www.i2m.univ-amu.fr/~preaux

[34] Zribi, N. (2005). Ordonnancement de job-shops flexibles sous contraintes de disponibilité des machines (Doctoral dissertation, Ecole Centrale de Lille).

[35] CLERC et SIARRY, (2004) « Une nouvelle méta-heuristique pour l'optimisation difficile : la méthode des essaims particulaires » Vol. 3-7 France Télécom R&D; Université Paris 12.

[36] Carlet, C., Guilley, S., Nitaj, A., & Souidi, E. M. (2019). Codes, cryptology and information security.Lecture Notes in Computer Science.

[37] Fournier, J. R. L. (2002). Application de la méta-heuristique d'optimisation par colonie de fourmis et de la technique K-OPT à l'affectation de cellules aux commutateurs. École Polytechnique de Montréal.

[38] Ouaddi, K., Benadada, Y., & Mhada, F. Z. (2016, May). Multi period dynamic vehicles routing problem: Literature review, modelization and resolution. In 2016 3rd International Conference on Logistics Operations Management (GOL) (pp. 1-8). IEEE.

[39] Penz, B. (1994). Aggregation methods for static, dynamic and reactive job-shops. HAL, 1994.

[40] Ghedjati, F., &Pomerol, J. C. (1997). Résolution du problème d'ordonnancement de type Job-Shop généralisé par des heuristiques dynamiques (Doctoral dissertation, LIP6).

[41] Hentous, H. (1999). Contribution au pilotage des systèmes de production de type Job Shop (Doctoral dissertation, Lyon, INSA).

[42] DRISS, I. (2016). Analyse d'un système job shop aspect ordonnancement (Doctoral dissertation, Université de Batna 2).

[43] Nadir, R. (2019). Un problème d'ordonnancement de type Job Shop dans un environnement dynamique (Doctoral dissertation, UNIVERSITE MOHAMED

[44] Ku, W. Y., & Beck, J. C. (2016). Mixed integer programming models for job shop scheduling: A computational analysis. Computers & Operations Research, 73, 165-173.

ملخص

في هذه المذكرة قمنا بدراسة مقارنة بين قواعد الأولوية في واحدة من مشاكل الجدولة الأكثر تعقيدا ، وهي جدولة من نوع العمل جوب شوب . دراستنا تتمحور حول نظام إنتاجي يتكون من أربع آلات تعالج عدة أنواع من القطع . قمنا في هذه الدراسة بمقارنة تسع قواعد أولوية (SPT, LPT, EDD, etc.) بما في ذلك القاعدة التي اقترحناها (ORS)

(Makespan)سعياً لتحديد القاعدة التي تعطي الحل الأقرب للحل الأمثل. المعيار المستخدم هنا هو التقليل من المدة الإجمالية لتنفيذ المهام ، أي تحسين Cmax

كلمات البحث:. الجدولة، ورشة العمل جوب شوب، قواعد الأولوية.

Résumé

Dans ce mémoire, nous avons examiné une étude comparative des règles de priorité dans l'un des problèmes d'ordonnancement les plus complexes, qui est l'ordonnancement de type job shop. Notre étude s'articule autour d'un système de production composé de quatre machines qui traitent plusieurs types de pièces. Dans cette étude, nous avons comparé neuf règles de priorité dont (SPT, LPT, EDD, etc.) en plus de la règle que nous avons proposée (ORS) qui a donné de bons résultats en cherchant à identifier la règle qui donne la solution la plus proche à la solution optimale. Le critère retenu ici est la minimisation de la durée totale de l'exécution des tâches, soit l'optimisation du Cmax (Makespan).

Mots clés : Ordonnancement ; Job shop ; règles de priorités.

Abstract

In this work, we have examined a comparative study of priority rules in one of the most complex scheduling problems, which is job shop scheduling. Our study revolves around a production system composed of four machines that process several types of parts. In this study, we compared nine priority rules including (SPT, LPT, EDD, etc.) in addition to the rule that we proposed, which gave good results by seeking to identify the rule that gives the closest solution to the optimal solution. The criterion retained here is the minimization of the total duration of the execution of the tasks, i.e. the optimization of the Cmax (Makespan).

Keywords: Scheduling, Job-shop, Priority rule,