

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



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



DEPARTEMENT D'INFORMATIQUE

**MEMOIRE de fin d'étude**

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

**Domaine : Mathématiques et Informatique**

**Filière : Informatique**

**Spécialité : Informatique Décisionnelle et Optimisation**

**Par : FATHI Zahira**

**Etude d'un problème d'ordonnancement  
Job-Shop avec des contraintes de transport**

**Soutenu publiquement devant le jury composé de :**

<b>Pr. Gasmi Abdelkader</b>	<b>Université de M'sila</b>	<b>Président</b>
<b>Dr. MOUHOUB Nasser Eddine</b>	<b>Université de M'sila</b>	<b>Rapporteur</b>
<b>Mr. Bouda Abdelghani</b>	<b>Université de M'sila</b>	<b>Examineur</b>

**Promotion : 2020/2021**

بسم الله الرحمن الرحيم والصلاة على  
رسوله الكريم ﷺ

## **Remerciements**

*Je remercie et loue Dieu Tout-Puissant pour sa grâce, son aide et son acceptation de mes supplications lorsque j'étais impuissant, et je le remercie d'avoir mobilisé les gens pour m'aider sans aucune pénurie. Sans cela, mon travail ne serait pas complet.*

*J'exprime mes remerciements et ma gratitude à l'honorable professeur, l'homme responsable, **Dr. MOUHOU B NASSER EDDINE** pour ses bons et précis conseils tout au long de la préparation du mémoire, Et je le remercie en tant que professeur pour son bon enseignement du matériel « l'ordonnancement dans la production » lié à ce mémo.*

*Enfin, Je tient à remercier ma mère avec amour pour sa encouragement, sa soin et sa présence à mes côté.*

## *Dédicaces*

*A ma belle et généreuse mère.*

*A mon cher père, que Dieu le guérisse de sa maladie : Je suis sûr, papa, que tu m'as donné ce que tu pouvais et m'a appris à construire mon avenir*

*A toute ma famille et ma petite belle « Ranou ».*

*A mes proches surtout « Amar Bassim ».*

*A mes petits chers « ACMMRT ».*

*A : Assil, C : Chihab, M : Mayar, M : Mohamed, R:Ranim, T : Tasnim*

*A mon encadreur **Mr.** Mouhoub Nasser Eddine.*

# Table des matières

Introduction générale.....	1
<b>Chapitre 01 : La gestion de production</b>	
1 Introduction.....	4
2 La gestion de production.....	4
2.1 Définitions .....	4
2.1.1 Système de Production (SP).....	4
2.1.2 Gestion de Production(GP) .....	4
2.2 Le rôle de la gestion de production .....	4
2.3 Les activités de la GP .....	5
2.3.1 Gestion des données techniques.....	6
2.3.2 Gestion des données commerciales.....	6
2.3.3 Gestion des matières.....	6
2.3.4 Gestion du travail .....	6
2.4 Les systèmes de production.....	6
2.4.1 Paramètre d'un système de production .....	6
2.4.2 Les sous systèmes de SP .....	7
2.4.3 Typologie des SP.....	8
2.5 Les types de décision.....	10
3 Les problèmes d'optimisation combinatoire.....	11
3.1 La complexité .....	12
3.2 Les Classes P et NP .....	12
3.3 La classe NP-Complet et NP-Difficile .....	12
4 Conclusion .....	13
<b>Chapitre 02 : Les problèmes d'ordonnancement</b>	
1 Introduction.....	15
2 Définitions générales des problèmes d'ordonnements.....	15
3 Les éléments d'un problème d'ordonnancement .....	16
4 Les typologies des problèmes d'ordonnement .....	19
5 Les méthodes de résolution.....	20
5.1 Les méthodes exactes .....	20

5.2	Les méthodes approchées .....	21
5.2.1	Les méthodes heuristiques .....	21
5.2.2	Les méta-heuristiques.....	22
6	Les algorithmes génétiques.....	24
6.1	Les opérateurs génétique .....	25
6.1.1	Le croisement .....	25
6.1.2	Mutation .....	25
6.1.3	Sélection .....	26
7	Représentation des solutions.....	26
7.1	Le diagramme de Gantt .....	26
8	Conclusion .....	27
<b>Chapitre 03 : Le problème Job-Shop</b>		
1	Introduction.....	29
2	$\alpha/\beta/\gamma$ .....	29
3	L'atelier Job shop.....	29
4	Complexité.....	30
5	Types de job shop .....	31
5.1	Job shop classique .....	31
5.1.1	Les contraintes liées aux les machines et aux tâches .....	31
5.2	Job shop flexible.....	32
6	Le problème d'ordonnancement job shop avec contrainte de blocage .....	33
6.1	Modélisation par un graphe alternatif.....	34
6.2	Arcs disjonctifs de blocage (arcs alternatifs).....	34
6.3	Exemple d'application : .....	35
7	Ordonnancement d'ateliers de type Job shop avec deux opérateurs.....	36
7.1	Description du problème .....	36
7.2	La Formulation de problème .....	37
8	Job shop à deux machines et algorithme de Johnson.....	37
8.1	Exemple d'application.....	38
9	Conclusion .....	39
<b>Chapitre 04 : Etude pratique</b>		
1	Introduction.....	41
2	L'environnement de développement.....	41

2.1	Matlab « <i>Mat (matrix) et laboratory (lab)</i> ».....	41
3	Formulation du problème.....	42
3.1	Description du System à étudier.....	42
3.2	Fonction objectif.....	42
3.3	Les contraintes.....	42
4	Démarche de résolution.....	42
4.1	Codage.....	42
4.1.1	Le codage basé sur les jobs.....	42
4.2	Création de Population initiale.....	43
4.3	Le croisement.....	43
4.3.1	Le croisement de l'ordre des jobs.....	43
4.4	Mutation.....	44
4.4.1	Mutation par position.....	44
4.5	Selection.....	45
4.5.1	La sélection par la roue de fortune.....	45
4.6	Evaluation des individus.....	46
5	Condition d'arrêt.....	46
6	Les données utiisés.....	46
7	Le programme :.....	47
7.1	Comment ça marche.....	47
7.2	Le schéma de résolution du problème.....	48
7.3	Illustration de logiciel.....	49
7.3.1	Illustration des barre de menus et Tools.....	56
8	Conclusion.....	58
	Conclusion générale.....	59
	Références bibliographiques.....	60
	Résumé.....	63

## Listes des figures

Figure 1: L'objectif de gestion de production[4] .....	5
Figure 2: Les activités de la gestion de production [5] .....	5
Figure 3:Les caractéristiques d'une tâche .....	17
Figure 4:Typologie des problèmes d'ordonnancement .....	19
Figure 5:Typologie des problèmes d'ordonnancement par les ressources .....	20
Figure 6:Le diagramme de Gantt ressource et Le Gantt tâche.....	26
Figure 7: Job shop simple et hybride .....	30
Figure 8 : Job shop classique.....	32
Figure 9 : Job shop flexible.....	33
Figure 10: arcs croisés dans un job shop avec contrainte de blocage.....	35
Figure 11:Graphe alternatif correspondant à l'exemple du tableau 1 .....	35
Figure 12:Graphe représentant une solution.....	36
Figure 13: Les propriétés de notre PC.....	41
Figure 14: Codage de chromosome .....	43
Figure 15:Le croisement par des jobs.....	44
Figure 16:Mutation entre 1 et 4 .....	44
Figure 17: Les ensembles des M-file.....	48
Figure 18:Le schéma de résolution du problème.....	49
Figure 19:la page de Connexion au programme .....	50
Figure 20:La page d'entrée de logiciel .....	51
Figure 21:Le bouton "Nouveau calcul" .....	51
Figure 22:la page qui permet de choisir le nombre des machines.....	52
Figure 23:La page qui permet de choisir le nombre des jobs.....	53
Figure 24 :La page qui permet de saisir la taille de population.....	53
Figure 25: après saisi la taille de population .....	54
Figure 26: Le contenu de la note .....	54
Figure 27: L'étape suivant qui permet de confirmer l'opération de calcul .....	55
Figure 28 : Les résultats obtenus .....	55
Figure 30: Le contenu de "Edit" .....	56
Figure 29: Cmax obtenu par le Diagramme de Gantt .....	56
Figure 31 : Le contenu de "Help" .....	57
Figure 32: Le contenu de L'icone imprimante .....	57
Figure 33: Le Contenu de l'icone "Ouvrir fichier" .....	58

## Liste des Tableaux

<i>Tableau 1:Récapitulatif des principales complexités d'un job shop .....</i>	<i>31</i>
<i>Tableau 2 : Exemple d'un job shop avec blocage à trois jobs et trois machines.....</i>	<i>35</i>
<i>Tableau 3:Les temps d'exécution des tâches .....</i>	<i>38</i>
<i>Tableau 4 : Les temps des opérations des jobs sur les machines.....</i>	<i>46</i>
<i>Tableau 5: La routage des jobs .....</i>	<i>47</i>
<i>Tableau 6 : Les temps de transport entre les machines .....</i>	<i>47</i>

## Liste des algorithmes

<i>Algorithme 2:Algorithme générale de l'algorithme génétique .....</i>	<i>24</i>
<i>Algorithme 1: Colonies de fourmis pour le problème du voyageur de commerce (TSP) .....</i>	<i>24</i>
<i>Algorithme 3: L'algorithme de Jhonson .....</i>	<i>38</i>

# Introduction générale

La compétition mondiale s'est considérablement renforcée au début des années 80. Les causes sont connues : disparition de nombreuses frontières, amélioration considérable des moyens de transport et de diffusion de l'information, dissémination des technologies et des connaissances, pour ne citer que les plus importantes. La conséquence de cette évolution sur les systèmes de production, dont les systèmes de fabrication sont une composante, a été un changement radical dans la conception de ces systèmes. À l'organisation classique par départements autour de métiers (réapprovisionnement, gestion des stocks, comptabilité, fabrication, contrôle de la qualité, expédition, maintenance, etc.) s'est substituée une organisation par projets dont la figure emblématique est le concept de chaînes d'approvisionnement [19].

Pourquoi cette évolution ?

Notons d'abord qu'elle n'a pas été immédiate. L'industrie a d'abord réagi par l'automatisation, gage de productivité et de régularité de la qualité, sans modifier l'organisation par départements. Malheureusement, le niveau des investissements à consentir et la rigidité des systèmes de fabrication automatisés ont rapidement montré leurs limites. Si l'automatisation des tâches s'est révélée très efficace pour la fabrication de masse, les industriels se sont rapidement aperçus que la variabilité croissante de la demande des consommateurs et l'évolution rapide des technologies nécessitaient de faire un pas supplémentaire en repensant l'organisation des systèmes de production : c'est l'émergence de l'organisation par projets que nous pouvons dater du début des années 90. Cette modification de l'organisation n'a été rendue possible que grâce aux progrès de l'informatique et de la communication [19].

La productivité peut être affectée directement par la qualité de l'ordonnancement des opérations sur les machines. car un atelier de production peut réaliser une grande variété de produit avec des coûts réduits, grâce à une meilleure utilisation des ressources.

On trouve une grande variété de problèmes d'ordonnancement qui sont liés à plusieurs paramètres : Les tâches ou opérations (préemptives et non préemptives, indépendantes ou non), les paramètres relatifs aux ressources (renouvelables consommables), types de contraintes sur les tâches (précédence, disjonctions), critères d'optimalité (minimisation de la

durée totale d'achèvement « makespan », minimisation du retard total des tâches, minimisation de la charge des machines...) etc.

Dans ce mémoire nous allons essayer de résoudre un problème d'ordonnement d'atelier de type job shop avec contrainte de transport où les produits sont transportés dans un sens unidirectionnel l'objectif est de trouver le meilleur ordonnancement de jobs lié au critère de minimisation de la durée de production (Makespan).

Des méthodes approchées ont été proposées pour résoudre ce problème en temps raisonnable. Parmi ces méthodes, apparaissent celles dites « Métaheuristiques ». Grâce à ces Métaheuristiques, on peut proposer des solutions approchées pour des problèmes difficiles et de plus grande taille qu'il était impossible de traiter auparavant.

Pour la résolution de notre problème d'ordonnement nous avons utilisé les Algorithmes Génétiques dont nous avons pu développer un logiciel qui nous a permis de calculer le makespan et nous donne les meilleures solutions.

Dans le chapitre 1, nous introduisons les différentes notions et quelques détails liées à la gestion de production, Puis on a parlé un peu du système de production et ses paramètres, nous avons mentionné ses sous systèmes de production.

Dans le deuxième chapitre, nous avons introduit des définitions sur les problèmes d'ordonnement (PO) afin d'aborder ses éléments et la typologie des (PO), à savoir les méthodes exactes et les méthodes approchées. Et nous avons parlé plus en détail sur les algorithmes génétiques et comment allons-nous représenter nos solutions, c'est-à-dire à travers le diagramme de Gantt.

Le troisième chapitre, nous présentons le problème d'ordonnement d'atelier de type Job-Shop.

Enfin, Dans le troisième chapitre, on abordera la description de la résolution de problème à étudier. Par une approche heuristique, l'algorithme génétique suivi d'une implémentation de notre logiciel ainsi que les résultats obtenus.

Ce mémoire se termine par une conclusion dans laquelle nous évaluons les résultats et présentons les perspectives de recherche.

# **Chapitre 01**

## **La gestion de production**

---

## **1 Introduction**

La production est le processus conduisant à la création de produits par l'utilisation et la transformation de ressources [5]. Le processus de production est alors, constitué d'un ensemble d'opérations qui sont les activités conduisant à la création de biens et de services [4], Le système de production, est l'ensemble de ressources réalisant une activité de production. C'est un ensemble de moyens divers : humains, matériels, informationnels et d'autres, constituant un tout, dont l'objectif est la réalisation de biens ou de services. Les systèmes de production industrielle se sont considérablement diversifiés et compliqués. En effet, ils peuvent se décomposer en plusieurs sous-systèmes, qui s'intègrent en vue d'assurer la pérennité et la compétitivité de l'entreprise.

## **2 La gestion de production**

### **2.1 Définitions**

#### **2.1.1 Système de Production (SP)**

Transformer un ensemble de matières premières ou de composants semi-finis en produits finis.

#### **2.1.2 Gestion de Production(GP)**

Assurer l'organisation du système de production afin de fabriquer les produits en quantités et temps voulus compte tenu des moyens (humains ou technologiques) disponibles

- La gestion de production est « un ensemble de processus qui permet de mener à bien la fabrication de produits à partir d'un ensemble de données et de prévisions »[1].
- F. Blondel [2], définit la gestion de production comme étant « la fonction qui permet de réaliser les opérations de production en respectant les conditions de qualité, délai, coûts qui résultent des objectifs de l'entreprise.

### **2.2 Le rôle de la gestion de production**

La gestion de production s'occupe d'un ensemble de problèmes liés à la production tels que la gestion des données, la planification, le contrôle de la production, la gestion des stocks,

la prévision, l'ordonnancement etc.[3] La figure suivante représente un schéma globale montrant le rôle de gestion de la production.

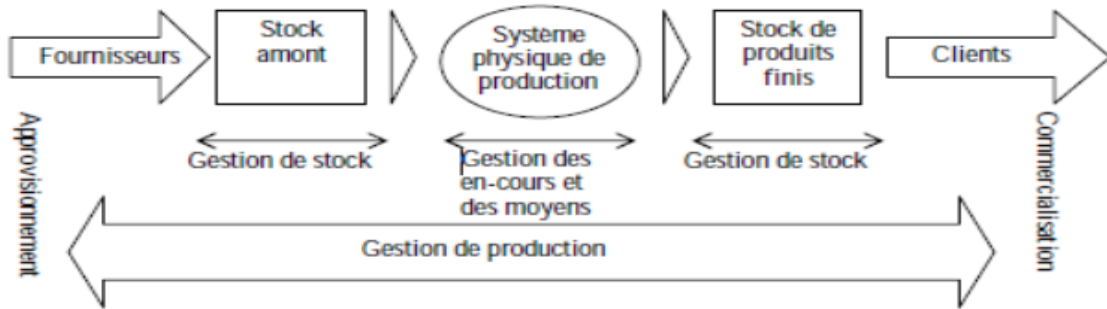


Figure 1: L'objectif de gestion de production[4]

### 2.3 Les activités de la GP

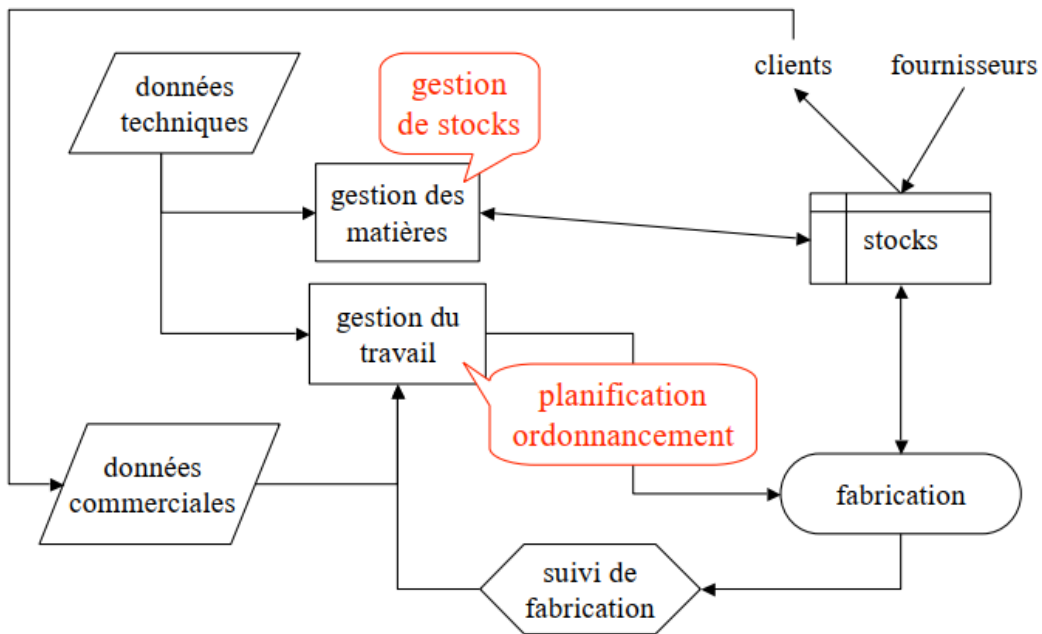


Figure 2: Les activités de la gestion de production [5]

### 2.3.1 Gestion des données techniques

- ❖ Description des produits et des familles de produits (nomenclatures)
- ❖ Description des processus de réalisation (gammes)

### 2.3.2 Gestion des données commerciales

- ❖ Reçoit les commandes et établit les calendriers de livraison souhaités

### 2.3.3 Gestion des matières

- ❖ Assurer l'approvisionnement en matières premières ou composants
- ❖ Assurer le stockage de produits fabriqués

### 2.3.4 Gestion du travail

- ❖ Organiser dans le temps la réalisation des tâches en leur attribuant les ressources nécessaires. Prend en compte les données techniques et commerciales et celles du suivi de fabrication (quantités déjà fabriquées, état des ressources...[28]).

## 2.4 Les systèmes de production

La charge d'un system de production est constitué de jobs .un job suit la réalisation d'un produit dans un toutes les étapes de production (de la matière première jusqu'au produit fini).chaque job a une gamme qui décrit la suite des opérations qu'il doit réaliser .la gamme décrit les opérations à réaliser en donnant : [8]

- Leur durée,
- La ou les machines qui doivent réaliser cette opération,
- Les ressources consommées (type de ressources et quantité utilisée),
- L'ordre entre les opérations à réaliser (facultatif).

### 2.4.1 Paramètre d'un système de production

selon [8] les paramètres d'un système de production sont :

– **Taux de production :**

Le taux de production est le nombre de produit fabriques par unité de temps. Si le système de système de production fabrique plusieurs types de produits. Le taux de production est donné sous forme de vecteur.

---

- **Capacité**

C'est le débit de sortie maximal pouvant être raisonnablement atteint compte-tenu des différentes contraintes. C'est le taux de production maximum pour le cas d'un seul type de produit [29].

- **Traitement en cours ou stock d'en cours**

C'est le produit dans ses différents stades d'élaboration dans l'usine, parmi les matières premières jusqu'au produit complètement terminé.

- **Temps de cycle**

C'est le temps moyen qu'un produit passe dans système de production.

- **Délai**

Le délai est le temps mis entre la connaissance du besoin du command et la fin de réalisation du produit corresponde.

- **Temps de chargement**

C'est le temps nécessaire pour passe de la fabrication d'un type de produit à un autre sur une unité de production donnée.

- **Temps d'exécution**

C'est le temps pendant lequel un produit subit une transformation lui conférant une valeur ajoutée.

- **Temps de transfert**

C'est le temps dépensé pour le transport des unités de produits entre les stations d'opération.

- **Temps d'attente**

C'est le temps, pendant lequel un produit ou un lot de produit reste près d'un poste de travail avant d'autre transféré au poste suivant.

- **Flexibilité**

C'est la faculté d'un système à s'adapter réellement aux changements de l'environnement.

## 2.4.2 Les sous systèmes de SP

Il ya trois sous systèmes de système de production [5] [30]

### Systeme physique

- éléments directement impliqués dans le processus de fabrication
- soumis aux perturbations émanant de l'environnement

### **Systeme décisionnel**

- Gestion des activités du système physique
- Prise en compte des contraintes environnementales
- besoin de réactivité

### **Systeme informationnel**

- Assurer les liens système physique-système décisionnel et système décisionnel environnement Besoin de réactivité.

## **2.4.3 Typologie des SP**

On mentionne deux, le premier en fonction de l'organisation de ressources, le deuxième est en fonction du pilotage en production [5][40] :

### **2.4.3.1 En fonction de l'organisation des ressources**

- Production continue : produit ne peut attendre qu'une ressource se libère (ex : industries de process–sidérurgie, pétrochimie, chimie lourde, pharmacie, agroalimentaire...)
- Production de masse : produits quasi-identiques, en grande quantité, moyens de production spécialisés et organisés afin d'optimiser le cycle de production (ex : industrie automobile...)
- Production en petite et moyenne séries : diversité forte des produits, type de demande (personnalisé) prohibe équipement spécialisé ⇒ ressources polyvalentes, lancement par lot afin de minimiser les coûts liés a une campagne de fabrication (ex : industries de sous-traitance : équipementiers, service...)
- Production unitaire (gestion de projet) : production en très faible quantité, cycle de production élevé, nombreuses ressources (ex : industrie aéronautique, spatial, BTP...)

### **Organisation en ligne de production**

- Ordre de passage sur les postes de travail toujours identique (*flow shop*).
- Importance de la fiabilité du matériel, la panne d'une machine provoquant l'arrêt de la chaîne ⇒ prépondérance de la maintenance

- Équilibrage de la chaîne afin que le produit passe un temps aussi constant que possible sur chaque poste. But : flux rythmé des produits sans accumulation ne d'en-cours ni perte de temps.

### **Organisation en ateliers**

- main d'œuvre qualifiée et équipements polyvalents
- Importance de la gestion des approvisionnements – coûts de manutention importants
- Diminution des coûts
- Multiplicité des routes entre postes de travail ⇒ prépondérance de la problématique d'ordonnement.

### **Organisation de type série unitaire**

- Mobilisation de toutes les ressources de l'entreprise pour la réalisation d'un projet, sur une assez longue période
- "survie" des entreprises réalisée par la gestion en parallèle de plusieurs projets, à des stades différents d'avancement
- Personnel qualifié exécute des tâches à caractère non répétitif équipements polyvalents
- Problème des stocks secondaire (produit fini non stockable et approvisionnements spécifiques au projet)
- Problèmes majeurs : respect des délais et maîtrise des coûts
- Rôle essentiel de l'ordonnement

### **2.4.3.2 En fonction du pilotage de la production**

#### **Production à la commande**

- Déclenchée par la commande ferme d'un client
- Grande variété de produits, demande aléatoire

#### **Production pour stock**

- Déclenchée par anticipation d'une demande solvable
- éventail des produits restreint, demande importante et prévisible.

#### **Production mixte**

- fabrication sur stocks et à la commande

#### **Production à flux tendus**

## 2.5 Les types de décision

La gestion de production est une fonction complexe. Cette complexité conduit généralement à la hiérarchiser afin de la simplifier et de rendre possible la gestion du système. Les types de décision couramment retenus sont en nombre de trois catégories.: stratégique, tactique et opérationnel [22].

### a. Décision stratégique

Les décisions stratégiques se traduisent par la formulation de la politique à long terme de l'entreprise (vision à plus de deux ans, en général), ce qui suppose une définition volontariste et cohérente du portefeuille d'activités qu'elle entend avoir à terme et des ressources stables qu'elle prévoit mettre en œuvre pour parvenir à ses fins. Les ressources stables visées sont :

- les machines : opérations d'investissement, de radiation, de cession, de modification de la disponibilité par une modification de maintenance, etc.
- les hommes : embauche, licenciement, préretraite, modification de qualification par des plans de formation, etc.
- les informations (procédures de gestion formalisées et bases de données techniques, c'est-à-dire les gammes, nomenclatures et schémas) détenues dans des systèmes d'information caractérisés par leur degré d'intégration et de latence C'est-à-dire le retard existant entre l'état du monde réel et sa transcription dans une base de données
- les encours de matières, de composants et de produits semi-finis, dont le niveau joue sur la rapidité de la propagation des problèmes d'un centre de production à un autre, en fonction d'une polyvalence plus ou moins grande donnée des équipements et des hommes.

### b. Décision tactique

Il s'agit de décisions à moyen terme. Elles assurent la liaison entre le niveau stratégique et le niveau opérationnel. L'objectif est de produire au moindre coût pour satisfaire la demande prévisible, en s'inscrivant dans le cadre fixé par le plan stratégique de l'entreprise. Parmi les décisions tactiques concernant la gestion de la production, on trouve : la planification de la production, qui est une programmation prévisionnelle de la production, agrégée par famille de produits, pour un ensemble de périodes dont l'amplitude varie entre la semaine et le mois (selon les entreprises) et qui est établie sur un horizon variant généralement entre 6 et 18 mois; la préparation du plan de transport, qui correspond à un ensemble de tournées types de distribution ou d'approvisionnement qui seront utilisées dans

les entreprises où ces problèmes sont relativement stables et ont une certaine incidence économique. Ces décisions s'inscrivent dans un cadre contraint par les décisions stratégiques, mais l'horizon de planification est normalement trop court pour permettre une modification importante des capacités de production [32].

### c. Décision opérationnel

Il s'agit des décisions à court et à très court terme. Les décisions opérationnelles assurent la flexibilité quotidienne nécessaire pour faire face aux fluctuations prévues de la demande et des disponibilités de ressources (mode prévisionnel) et réagir aux aléas (mode correctif), dans le respect des décisions tactiques. Les décisions opérationnelles concernant la gestion de la production comprennent entre autres la gestion des stocks, qui assure la mise à disposition des matières premières et des composants, **et l'ordonnancement**, qui consiste en une programmation prévisionnelle détaillée des ressources mobilisées (opérateurs, équipements et outillages) dans l'exécution des opérations nécessaires à la production élémentaire de biens ou de prestations de service (pour un client final ou pour des besoins internes, s'il s'agit d'opérations de maintenance ou de manutention), sur un horizon ne dépassant pas quelques dizaines d'heures, dans le cadre d'un découpage temporel généralement de l'ordre de la minute. S'y ajoute, dans un certain nombre de systèmes productifs, mais avec une granularité temporelle et spatiale plus fine, tout ce qui concerne le pilotage informatique en temps réel d'ensembles productifs jouissant d'une certaine autonomie de conduite par rapport au reste du système productif (machines à commande numérique, ateliers flexibles, magasins automatiques, automates bancaires, billetterie automatique ferroviaire, etc.)

On trouve l'ordonnancement dans le niveau opérationnelle, Les problèmes d'ordonnancement font partie des problèmes d'optimisation combinatoire.

## 3 Les problèmes d'optimisation combinatoire

Un problème d'optimisation combinatoire est un problème qui consiste à chercher une meilleure solution parmi un ensemble de solutions réalisables. Aussi, le nombre de solutions réalisables des problèmes combinatoires augmente exponentiellement en fonction de la taille du problème, et c'est ce qui exclut des méthodes de résolution basées sur l'énumération de toutes les solutions réalisables [20].

Le plus souvent, un problème d'optimisation se présente comme un objectif à poursuivre (maximisation d'un profit, minimisation d'une distance, ou d'un coût, etc.) [21].

Un facteur majeur qui permet la classification des problèmes d'ordonnancement c'est la complexité [33].

### 3.1 La complexité

*La complexité d'un algorithme A est une fonction C (AN), donnant le nombre d'instructions caractéristiques exécutées par A dans le pire des cas, pour une donnée de taille [21]. Elle est notée d'un algorithme par O[34].*

Dans le monde des problèmes d'optimisations, on peut classer ces problèmes en deux classes, des problèmes faciles et des problèmes difficiles à résoudre. On dit un problème facile si l'en résoudre par un algorithme efficace. Un algorithme est dit de complexité  $O(g(n))$  s'il se termine après un nombre d'opérations élémentaires ne dépassant pas au pire cas  $c.g(n)$ , avec  $c$  constant et  $n$  la taille d'une instance. On dira d'un algorithme de complexité  $O(g(n))$  qu'il est efficace ou polynomial s'il existe un polynôme en  $n$  majorant  $g(n)$  (i.e. il existe deux constantes  $c, k \geq 0$  et  $n_0 \in \mathbb{N}$ , telles que pour tout  $n \geq n_0$ ,  $g(n) \leq c.n^k$ ). [20]

### 3.2 Les Classes P et NP

La classe NP est celle des problèmes d'existence dont une proposition de solution est Oui et qui est vérifiable polynomialement. Parmi les problèmes décidables, les plus simples à résoudre sont regroupés dans la classe NP. La classe NP est également décomposée en trois catégories qui permettent d'identifier les problèmes les plus simples et les problèmes les plus compliqués de la classe. Un problème est dit polynomial s'il existe un algorithme de complexité polynomiale permettant de répondre à la question posée dans ce problème, quelle que soit la donnée de celui-ci. La classe P est l'ensemble de tous les problèmes de reconnaissance polynomiaux. Pour le reste de la classe NP, on n'est pas sûr qu'il n'existe pas un algorithme polynomial pour résoudre chacun de ses problèmes. Ainsi, on sait que P est incluse dans NP mais on n'a pas pu prouver que P n'est pas NP. [21]

### 3.3 La classe NP-Complet et NP-Difficile

La classe NP-Complet regroupe les problèmes les plus difficiles de la classe NP. Elle contient les problèmes de la classe NP tels que n'importe quel problème de la classe NP leur

est polynomialement réductible. Entre eux, les problèmes de la classe NP-Complet sont aussi difficiles. La classe NP-Difficile regroupe les problèmes (pas forcément dans la classe NP) tels que n'importe quel problème de la classe NP leur est polynomialement réductible [21].

## **4 Conclusion**

Dans ce chapitre, nous avons parlé sur les problèmes d'optimisation combinatoires après nous avons résumé les différentes notions liées à la gestion de production, on a mentionné ses différents éléments et ses caractéristiques générales, Et nous avons parlé brièvement sur ses activités, ainsi que nous avons présenté les systèmes de production et nous avons défini les sous systèmes avec la typologie soit en fonction de l'organisation des ressources soit en fonction du pilotage de la production et nous expliqué ses différents types de décision Parmi ces types, nous avons parlé de la décision opérationnels qui inclut l'ordonnancement qui fait parmi les problèmes de systèmes de production et les problèmes d'optimisation combinatoire, le prochain chapitre va traiter ce type des problèmes.

**Chapitre 02**  
**Les problèmes**  
**d'ordonnancement**

---

## 1 Introduction

Un problème d'ordonnancement consiste à organiser dans le temps la réalisation de tâches ou les opérations, compte tenu d'un certain nombre de contraintes afin d'atteindre un certain objectif appelé fonction économique.

L'ordonnancement joue un rôle essentiel dans de nombreux secteurs d'activités : la conception (de bâtiments, de produits, de systèmes, . . .), l'administration (gestion d'emplois du temps, gestion du personnel), l'industrie (gestion de la production), l'informatique (ordonnancement de processus, ordonnancement de réseaux). Les méthodes d'ordonnancement foisonnent dans la littérature. Elles se différencient par la nature du problème considéré (nombre de ressources, structure particulière du problème, . . .), la nature des contraintes prises en compte, les objectifs à satisfaire (minimisation des coûts, de la durée totale de mise en œuvre, . . .) et la nature de l'approche de résolution adoptée (heuristiques, méthodes exactes, métaheuristiques, approches par contraintes, . . .).

La maîtrise de l'ordonnancement est d'un intérêt primordial pour les entreprises qui sont confrontées à des contraintes de productivité, de réactivité et de flexibilité. L'étude de l'ordonnancement est également d'un intérêt théorique toujours renouvelé pour les chercheurs puisqu'il n'existe pas encore de méthodes à la fois générale et de faible complexité algorithmique à cause de la nature fortement combinatoire des problèmes d'ordonnancement.

## 2 Définitions générales des problèmes d'ordonnements

« *Ordonner, c'est programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution* » [10].

L'ordonnancement est une branche de la recherche opérationnelle et de la gestion de la production qui vise à améliorer l'efficacité d'une entreprise en termes de coûts de production et de délais de livraison. Les problèmes d'ordonnancement sont présents dans tous les secteurs d'activités de l'économie, depuis l'industrie manufacturière jusqu'à l'informatique [11].

Ordonnancer le fonctionnement d'un système industriel de production consiste à gérer l'allocation des ressources au cours du temps, tout en optimisant au mieux un ensemble de critères .C'est aussi programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution [8]

Un problème d'ordonnancement consiste à affecter des tâches à des ressources à des instants donnés pour répondre au mieux aux besoins exprimés par un client, au meilleur coût et dans les meilleurs délais, tout en tenant compte des contraintes.

### 3 Les éléments d'un problème d'ordonnancement

L'ordonnancement est décrit généralement par les Quatre éléments sont retenus pour décrire d'une façon explicite un ordonnancement.

- **Les tâches**

Dans le cadre des problèmes d'ordonnancement, parmi les informations utiles, on trouve le temps nécessaire pour la réalisation d'une tâche, la(les) ressource(s) qu'elle requiert (une ressource étant renouvelable ou non), une date de début au plus tôt, une date de fin souhaitée et une date de fin impérative. Des contraintes de précédence conditionnent l'exécution d'une tâche par rapport à la fin d'exécution d'autres tâches [35].

Une tâche est alors considérée comme l'exécution d'une opération sur une certaine machine. Plusieurs tâches peuvent être regroupées pour constituer un « job » pour lequel on donne la gamme de production. Il est également nécessaire de connaître la nature de la tâche, par exemple savoir si une tâche est interruptible ou non, c'est-à-dire de savoir si le traitement d'une tâche sur une ressource donnée peut s'interrompre pendant une certaine durée et de le poursuivre ensuite, cela se traduit par l'autorisation de la préemption des tâches ou non[8][37]

#### **Notation**

Il ya des notations [tableau1] sont introduites pour décrire une opération. Chaque variable représentera une information particulière se rapportent à la réalisation de l'opération [36].

- ri :** la date de disponibilité de la tâche  $i$  .
- ti :** la date de début de la tâche  $i$  (release date) .
- ci :** la date de fin d'exécution de la tâche  $i$  (START date) .
- di :** la date d'échéance de la tâche  $i$  (completion time) .
- pi :** la durée opératoire de la tâche  $i$  (due date) .
- F = ci - ri :** la durée de séjour de l'opération  $i$  sur la machine avant qu'elle redevienne disponible (processing time date) .
- Li = ci - di :** le retard algébrique (lateness) entre la fin d'exécution de la tâche  $i$  par rapport à sa date d'échéance .
- Ti = max(Li, 0) :** le retard absolu de la tâche  $i$  (tardiness) .
- Ei = max(- Li, 0) :** l'avancement (earliness) de la tâche  $i$  .

La Figure 3 montre les relations entre les différentes variables représentant une tâche.

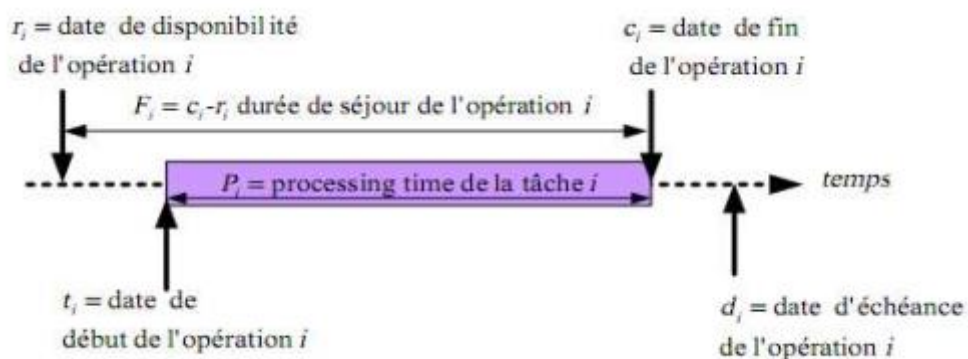


Figure 3: Les caractéristiques d'une tâche

■

- **Les ressources**

Pour l'exécution des tâches, ces dernières requièrent certaines ressources telles que des machines, la main d'œuvre, les moyens financiers, etc. Une ressource  $k$  est donc, un moyen humain ou technique qui est utilisé dans la réalisation d'une tâche. Elle est disponible en quantité limitée, sa capacité  $A_k$ .

Deux familles de ressources existent : les ressources renouvelables et les ressources consommable [11][38].

*Les ressources renouvelables* si elles sont utilisées par une ou plusieurs tâches, elles seront à nouveau disponibles en même quantités (les hommes, les machines, l'espace, l'équipement en général, ...etc.), les quantités des ressources utilisables à chaque instant sont limitées.

*Les ressources consommables* utilisée pour la réalisation d'une ou plusieurs tâches à la fois elles ne sont pas disponible telle que (matière première, budget,...etc), la consommation globale (ou cumul) au cours du temps est limitée.

- **Les contraintes**

Selon [12] les contraintes expriment les restrictions que peuvent prendre conjointement les variables de décision. Donc les contraintes nous renseignent sur les limites imposées par l'environnement. On distingue plusieurs types de contraintes, qui sont [13], [12]:

Les contraintes de positionnement temporelles, Pour la réussite d'un projet ou d'un plan de production, on doit se soumettre aux impératifs de production (réalisation) traduits par le respect d'un planning fixé avant. Au niveau global on parlera d'une date de lancement d'une tâche et d'une date de livraison. A un autre niveau de détail plus fin, pour une tâche ou une opération. Ce dernier niveau se traduit par la définition des dates suivante : (i) date de disponibilité :  $r_i$  ; (ii) date d'échéance :  $d_i$ .

Les contraintes de précédence, une contrainte qui lie le début d'une activité à la fin d'un autre est appelée contrainte de succession ou de précédence. Les gammes opératoires sont un exemple de ces contraintes, d'autres contraintes sont imposées dans certain cas telles que les contraintes de synchronisation, de simultanéité, ou de recouvrements etc. [12].

Les contraintes de ressources, Les contraintes de ressources représentent le fait que les activités requièrent tout au long de leur exécution une certaine quantité de ressources [13]. Les contraintes de ressources concernent les deux points suivants: (i) l'utilisation de ces ressources, (leur nature, la quantité nécessaire, et les caractéristiques d'utilisation) ; (ii) la disponibilité et la quantité de ces ressources. [12]

#### 4 Les typologies des problèmes d'ordonnancement

Selon la nature des variables mises en jeu, la nature des contraintes, ou encore la structure particulière du graphe non-conjonctif associé à un problème, plusieurs classifications des problèmes d'ordonnancement sont proposées dans la littérature où leurs complexité est généralement de l'ordre NP-difficile sauf quelques cas bien particuliers [37].

Les problèmes d'ordonnancement de projet (Problème central de l'ordonnancement) a une complexité polynomiale ainsi que certains cas de l'ordonnancement d'atelier (qu'on verra ultérieurement).

Une typologie des problèmes d'ordonnancement peut être dressée dans l'architecture suivante des figures 5 et 6:

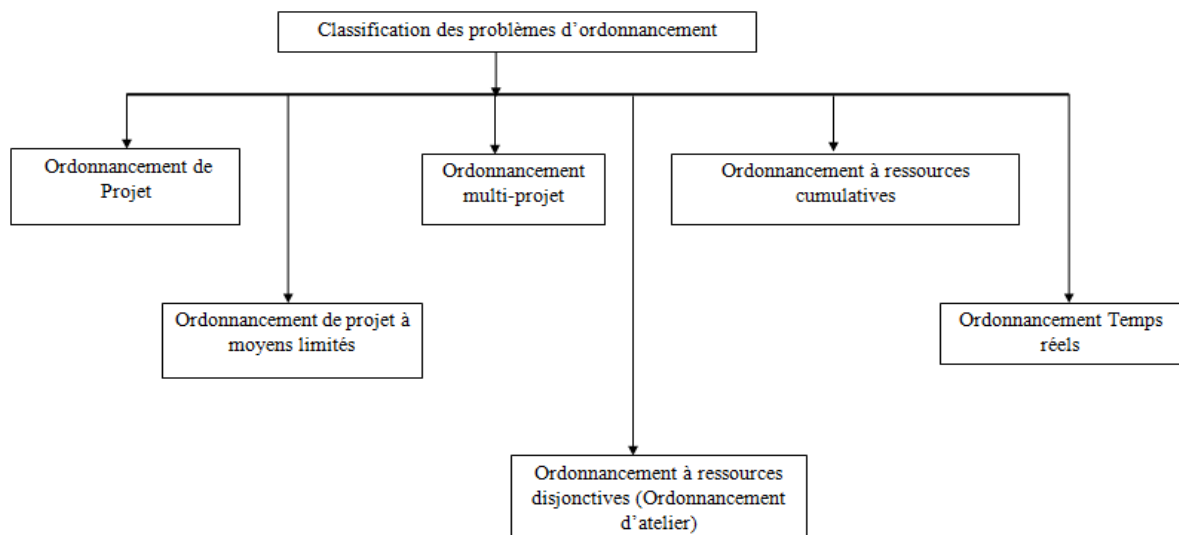


Figure 4: Typologie des problèmes d'ordonnancement

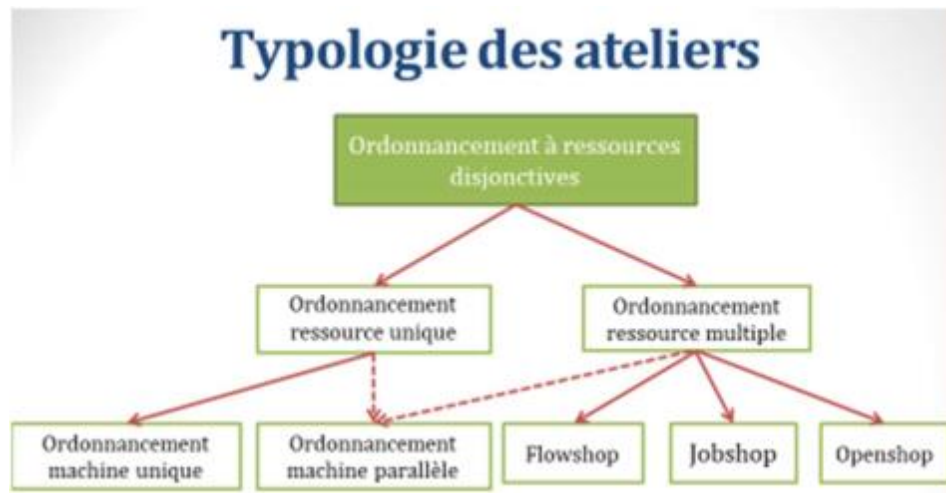


Figure 5: Typologie des problèmes d'ordonnancement par les ressources

Nous nous intéressons au Job Shop parce qu'il est classique dans la littérature et car il s'agit d'un problème très général parmi les problèmes classiques d'ordonnancement d'atelier.

## 5 Les méthodes de résolution

Au fil des années, de nombreuses méthodes de résolution de problèmes ont été proposées. Ainsi, une grande variété de concept et principe, de la stratégie et des performances ont été discernées. Cette variété et ces différences ont permis de regrouper les différentes méthodes de résolution de problèmes d'ordonnancement en deux classes principales : la classe de méthodes exactes et la classe des méthodes approchées [38].

### 5.1 Les méthodes exactes

Elles recherchent un ordonnancement optimal qui minimise ou maximise un des critères présentés ou une combinaison de plusieurs critères. Les techniques utilisées sont les méthodes par séparation et évaluation, la programmation dynamique, la déduction mathématique, la théorie des jeux, la théorie des graphes etc. Cette technique est coûteuse en temps de calcul [14].

## 5.2 Les méthodes approchées

Elles recherchent une solution réalisable de la fonction objectif, mais sans garantir d'optimalité. Elles sont plus pratiques pour la résolution de problèmes difficiles ou NP difficiles.

Ces méthodes sont souvent classées en deux catégories : des méthodes heuristiques et des méthodes méta-heuristiques.

### 5.2.1 Les méthodes heuristiques

Une heuristique est une règle empirique simple basée sur l'expérience. C'est une technique de calcul approchée qui exploite d'une façon satisfaisant le problème à optimiser et qui fournit une solution admissible, non nécessairement exacte, dans un temps polynomial pour un problème d'optimisation difficile. En d'autres termes, une heuristique est un algorithme qui sacrifie en partie, la qualité de la solution au sens purement mathématique pour accélérer le processus de résolution. Elle est également une stratégie de bon sens pour se déplacer intelligemment dans l'espace des solutions, afin d'obtenir une solution approchée, la meilleure possible, dans un délai de temps raisonnable.

En manière simple, Elles sont basées sur des méthodes empiriques, elles utilisent des règles simples pour optimiser un ou plusieurs critères. Le principe général de cette catégorie de méthodes est d'intégrer des stratégies de décisions pour construire une solution proche de l'optimal tout en essayant d'avoir un temps de calcul raisonnable [8]

Elle est là, à la place d'une méthode exacte qui donne la solution dans un temps de résolution exponentiel. Elle est dédiée, généralement, à un problème bien particulier.

Exemples d'heuristiques :

- **FIFO** (First In First Out) : la première tâche qui vient est la première tâche ordonnancée,
- **SPT** (Shortest Processing Time) : la tâche ayant le temps opératoire le plus court est traitée en premier lieu,
- **LPT** (Longest Processing Time) : la tâche ayant le temps opératoire le plus important est ordonnancée en premier lieu,
- **EDD** (Earliest Due Date) : cet algorithme choisit parmi les tâches exécutables celle dont le délai est échu le plus tôt. Si aucune tâche n'est disponible, alors un temps libre est généré,

- **SRPT** (Shortest Remaining Processing Time) : cette règle, servant à lancer la tâche ayant la plus courte durée de travail restant à exécuter, est très utilisée pour minimiser les encours et dans le cas des problèmes d'ordonnement préemptifs,
- **ST** (Slack Time) : à chaque point de décision, l'opération ayant la plus petite marge temporelle est prioritaire. Faute de disponibilité des ressources de production, cette marge peut devenir négative.

### 5.2.2 Les méta-heuristiques

Le mot méta-heuristique est composé de deux mots ; le mot méta qui est un préfixe signifiant "au-delà" ou bien "dans un niveau supérieur"; et le mot heuristique qui vient du verbe heuriskein et qui signifie 'trouver' [8].

On cite quelques définitions de la notion 'méta-heuristiques' :

Définition (1) : Une métaheuristique est formellement définie comme un processus de génération itératif qui guide une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter l'espace de recherche, des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions quasi optimales [8].

Définition (2) : Une métaheuristique est un ensemble de concepts qui peuvent être utilisés pour définir des méthodes heuristiques pouvant être appliquées à un large éventail de problèmes différents. En d'autres termes, une métaheuristique peut être considérée comme un cadre algorithmique général qui peut être appliqué à différents problèmes d'optimisation avec relativement peu de modifications pour les adapter à un problème spécifique. Des exemples de métaheuristiques incluent le recuit simulé (SA), la recherche tabou (TS), la recherche locale itérée (ILS), les algorithmes évolutionnaires (EC) et l'optimisation des colonies de fourmis (ACO) [8].

Les méta-heuristiques à base de solution unique débutent la recherche avec une seule solution initiale. Elles se basent sur la notion du voisinage pour améliorer la qualité de la solution courante. En fait, la solution initiale subit une série de modifications en fonction de son voisinage. Le but de ces modifications locales est d'explorer le voisinage de la solution actuelle afin d'améliorer progressivement sa qualité au cours des différentes itérations [11].

Les méta-heuristiques sont souvent inspirées de processus naturels. Il existe un grand nombre de méta-heuristiques différentes citons :

### 5.2.2.1 Le recuit simulé

*Le recuit simulé* : Cette technique provient de l'observation de la formation d'une structure cristalline quand un métal refroidit. Le recuit simulé est une technique permettant de trouver, au bout d'un temps raisonnable, une solution pour des problèmes de complexité élevée. Elle est utilisée pour la détermination de la répartition statique des tâches. Elle est considérée comme technique stochastique et ne garantit pas de solution optimale [14].

### 5.2.2.2 L'optimisation par colonies de fourmis

L'optimisation par colonies de fourmis ou Ant Colony Optimization (ACO) en anglais est une méthode d'optimisation basée sur la manipulation d'un ensemble de solutions ; cette méthode représente toute une classe des métaheuristiques qui reposent sur la notion de l'intelligence collective. La solution finale est plus complexe que celle d'un composant simple. Plusieurs mots apparaissent dans ce domaine : l'auto-organisation, émergence et autres [8].

L'algorithme de la colonie de fourmis a été proposée pour résoudre le problème des navetteurs commerciaux .Il est basé sur trois étapes de base :

- Construis le chemin de chaque fourmi.
- Distribuez des phéromones sur chaque chemin de fourmis.
- Évaporation des voies de phéromone

L'optimisation par colonies de fourmis est très efficace pour le problème de voyageur de commerce .Nous présentons les différentes étapes d'un algorithme de colonies de fourmis pour un problème de voyageur de commerce (TSP) qui sont illustrées par l'algorithme 1

- **Tant que le critère d'arrêt n'est pas atteint faire**
- **Pour  $k=1$  à  $m$  faire**
- **Choisir une ville au hasard**
- **Pour chaque ville non visitée  $i$  faire**
- **Choisir une ville  $j$ , dans la liste des villes restantes selon (F-1)**
- **Fin Pour**
- **Déposer une piste sur le trajet (t) conformément à (F-2)**
- **Fin Pour**
- **Évaporer les pistes selon (F-3)**
- **Fin Tant que**

Algorithme 1: Colonies de fourmis pour le problème du voyageur de commerce (TSP) [A]

## 6 Les algorithmes génétiques

L'algorithme génétique est choisit pour résoudre notre problème Job shop avec des contraintes de transport parce qu'il est adaptable à plusieurs types des problèmes, Robustes, Faciles à implémenter, Faciles à hybrider et Faciles à paralléliser.

Les algorithmes génétiques se basent sur une population initiale d'individus, de taille bien déterminée, de solutions admissibles. Les solutions sont aussi appelées individus ou chromosomes. Par la suite, la structure de la population change en appliquant des évolutions progressives sur plusieurs générations. Pendant chaque génération, des nouveaux individus sont générés par les opérateurs de sélection, de croisement et de mutation. Les différentes étapes d'un algorithme génétique sont illustrées par l'algorithme 2 [B].

---

### Algorithme génétique

---

#### Begin

Génération d'une population initiale de  $k$  individus ;

#### Repeat

1. Evaluation de la fonction objectif  $f$  de chaque individu ;
2. Sélection des meilleurs individus ;
3. Croisement entre 2 individus: génération 2 enfants issus de deux parents sélectionnés ;
4. Mutation : transformation aléatoire des gènes de certains individus de la population ;

**Until** la satisfaction de la condition d'arrêt.

Retourner la meilleure solution.

#### End.

---

Algorithme 2: Algorithme générale de l'algorithme génétique

Donc à partir de la structure générale, L'application d'un algorithme génétique à un problème particulier nécessite cinq éléments suivants [8] :

- **Le codage des solutions (individus)**

Associe à chacun des points de l'espace de solution une structure de données. Elle vient généralement après une phase de modélisation mathématique du problème traité. La qualité du codage conditionne le succès de l'algorithme.

Une population est présentée par un ensemble de chromosomes.

- **Une méthode de génération de la population initiale**

La population initiale qui servira de base pour les générations futures doit être non homogène.

- **Une fonction à optimiser**

Cette fonction sert à évaluer chaque individu, elle retourne une valeur réelle nommée : « fitness », qui sera utilisée dans le calcul de probabilité de sélection d'un individu [15].

- **Les opérateurs génétiques**

La puissance des algorithmes génétique réside dans leurs opérateurs génétiques (croisement, mutation, sélection). En effet, ces opérateurs permettent d'établir un balayage partiel de l'espace de recherche, ce qui réduit son temps de réponse.

## 6.1 Les opérateurs génétique

### 6.1.1 Le croisement

L'opérateur de croisement consiste à choisir une paire d'individus afin de recombinaison les deux chromosomes sélectionnés dans le but de produire deux nouveaux chromosomes enfants portants les meilleures caractéristiques des deux chromosomes parents.

### 6.1.2 Mutation

L'opérateur de mutation, assure le brassage et la recombinaison des gènes parentaux, permettant de diversifier la future population afin d'éviter une convergence rapide vers une solution optimale localement. Étant aléatoire, cet opérateur agit selon une probabilité  $P_c$  fixée par l'utilisateur en fonction du problème à optimiser [16].

### 6.1.3 Sélection

L'opérateur de sélection consiste à choisir des individus les mieux adaptés qui vont pouvoir survivre et/ou se reproduire pour transmettre leurs caractéristiques à la génération suivante.

## 7 Représentation des solutions

Les problèmes d'ordonnancement sont définis par la donnée d'un certain nombre d'opérations (les tâches) et des contraintes de succession entre ces tâches, ainsi que les durées de ces tâches. Plusieurs méthodes de modélisation existent actuellement. On peut citer entre autres: le diagramme de Gantt, la méthode des potentiels et la méthode PERT [27].

### 7.1 Le diagramme de Gantt

Ce diagramme a été créé par Henry Gantt en 1910, est le plus courant pour l'ordonnancement, on peut avoir deux représentations de ce diagramme :

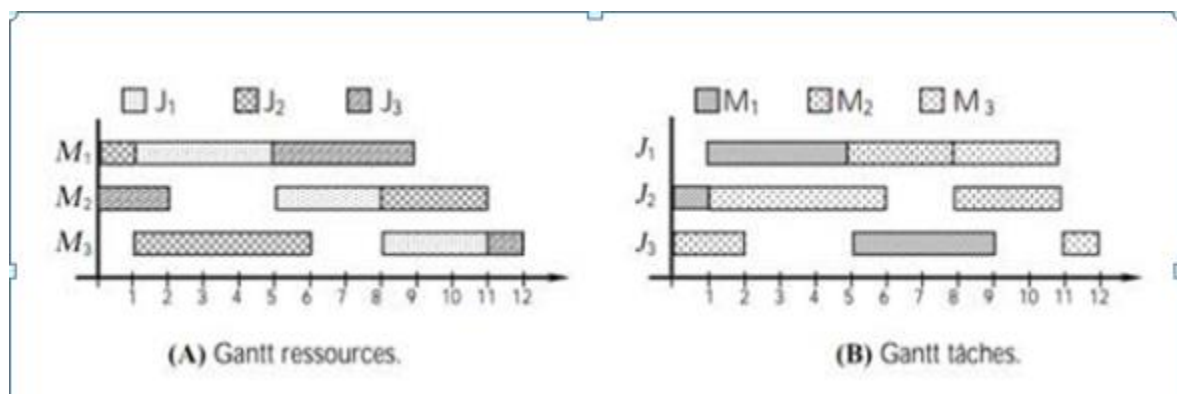


Figure 6:Le diagramme de Gantt ressource et Le Gantt tâche

Pour les problèmes d'ordonnancement avec contraintes cumulatives, la représentation par diagramme de Gantt peut encore être élargie en considérant une représentation hybride entre le diagramme à barres et la courbe de charge. Ce mode de représentation, qui visualise l'exécution temporelle des tâches sur une ressource ainsi que leur consommation de cette ressource, sert de support pour l'illustration de caractéristiques générales des ordonnancements [40].

---

**8 Conclusion**

Nous avons présenté, dans ce chapitre, les différentes notions et les concepts liés aux problèmes d'ordonnements ainsi que les méthodes de résolution utilisées dans la littérature. Ces méthodes sont regroupées en deux classes : les méthodes exactes et les méthodes approchées et nous avons résumé quelques des informations importantes liées à notre méthodes de résolution qui est l'algorithme génétique avec la représentation diagramme de Gantt.

# **Chapitre 03**

## **Les problème Job-Shop**

## 1 Introduction

La classification des problèmes d'ordonnement dans un atelier peut se faire selon le nombre de machines et leur ordre d'utilisation pour fabriquer un produit, qui dépend de la nature de l'atelier considéré. Un atelier est caractérisé par le nombre de machines qu'il contient et par son type. Il y a trois types d'ateliers : flow-shop, job-shop et open-shop, avec des extensions possibles pour chacun d'eux.

Avant les problèmes de job shop sont considérés comme des problèmes simples dont les déplacements des pièces d'une machine à une autre n'étaient pas pris en compte dans le calcul de l'ordonnement ou bien sont considérés intégrés dans la tâche elle-même. Or, cette hypothèse est souvent non justifiée en pratique. En effet, le transport des tâches doit être intégré aux données du problème. Ce présent chapitre est consacré la présentation de ce type de problème.

## 2 $\alpha/\beta/\gamma$

La notation  $\alpha/\beta/\gamma$  Elle permet en effet de caractériser un problème d'ordonnement de manière précise.

- a) Le champ  $\alpha$  : la structure de problème et se décompose en deux sous champ  $\alpha 1$  et  $\alpha 2$ , et  $\alpha 1$  indique la nature de problème (job shop, flow-shop, etc.),  $\alpha 2$  précisant le nombre de machines ou de pools.
- b) Le champ  $\beta$  : les types de contraintes prises en compte.
- c) le champ  $\gamma$  : fonction objectif ou la description des critères.

## 3 L'atelier Job shop

Appelés ateliers à cheminement multiple, l'ordre de passage des jobs sur les machines peut être différent d'un job à l'autre. Il y a des possibles de passage d'un job sur une machine plusieurs fois. Et on même temp il y a des possibles de passage pas d'un job sur tous les machines.

Si, pour chaque machine, on dispose d'un et un seul exemplaire, on dit que l'organisation est un Job Shop simple. Si, au contraire, pour un au moins des machines (postes de travail), on dispose de plus d'un exemplaire, on l'appelle Job Shop Hybride.[14]

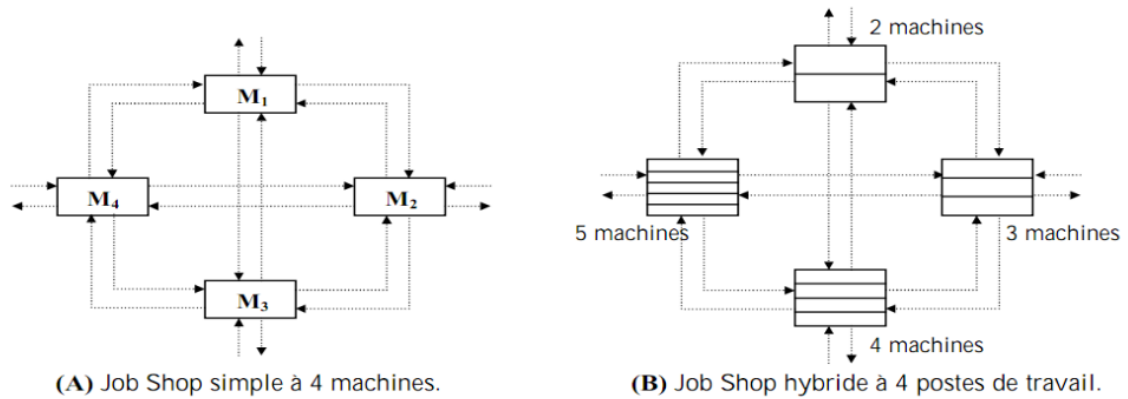


Figure 7: Job shop simple et hybride

## 4 Complexité

Les problèmes de job shop sont en général NP-complets même si l'atelier est simple. En effet, Lenstra et Rinnooy Kan ont montré que les ateliers possédant plus de trois machines, ou un nombre de tâches supérieur ou égal à trois, sont NP-difficiles même si la préemption est permise. De même, pour les problèmes à deux machines, dès qu'il y a recirculation, ils deviennent fortement NP-difficiles. En revanche, si la fonction objective est la minimisation du makespan [26].

Le tableau suivant résume les principales complexités en fonction des caractéristiques de l'atelier [26].

Type d'atelier	Complexité
J2   $n_i \leq 2$   $C_{\max}$	$O(n \log n)$
J2   $p_{ij} = 1$   $C_{\max}$	$O(n \log(nr))$
J2   $p_{ij} = 1$ ; $r_i$   $C_{\max}$	$O(n^2)$
J2   $n = k$   $C_{\max}$	$O(r^{2k})$
J2   $p_{ij} \in \{1, 2\}$   $C_{\max}$	NP-difficile
J2   chains ; $p_{ij} = 1$   $C_{\max}$	NP-difficile
J2   prmtn   $C_{\max}$	NP-difficile
J2   $n = 3$ ; prmtn   $C_{\max}$	NP-difficile
J3   $p_{ij} = 1$   $C_{\max}$	NP-difficile
J3   $n = 3$   $C_{\max}$	NP-difficile

Tableau 1: Récapitulatif des principales complexités d'un job shop

## 5 Types de job shop

### 5.1 Job shop classique

Le problème d'ordonnancement de type job shop consiste à réaliser un ensemble de  $n$  jobs sur un ensemble de  $m$  machines pour atteindre certains objectifs. Chaque job  $J$  est composé d'une suite de  $n$  opérations devant être exécutées sur les différentes machines  $m$  selon un ordre préalablement défini.

#### 5.1.1 Les contraintes liées aux les machines et aux tâches

Les contraintes concernant les machines et les tâches [14], Les machines sont indépendantes les unes des autres.

- Une machine ne peut exécuter qu'une seule opération à la fois à un instant donné.
- Les machines sont disponibles pendant toute la durée de l'ordonnancement.
- Les jobs sont indépendants les uns des autres. En particulier, il n'existe aucun ordre de priorité attaché aux jobs.

La figure (7) représente un problème type de job shop classique composé de 4 jobs et 6 machines, les gammes opératoires sont les suivantes:

J1 en Orange : Entrée → Machine1-Machine2-Machine5-Machine6→ Sortie

J2 en Bleu : Entrée → Machine1-Machine4-Machine3→ Sortie

J2 en Vert : Entrée → Machine2-Machine3-Machine5 → Sortie

J4 en Rouge : Entrée→ Machine1-Machine2-Machine5-Machine6→ sortie

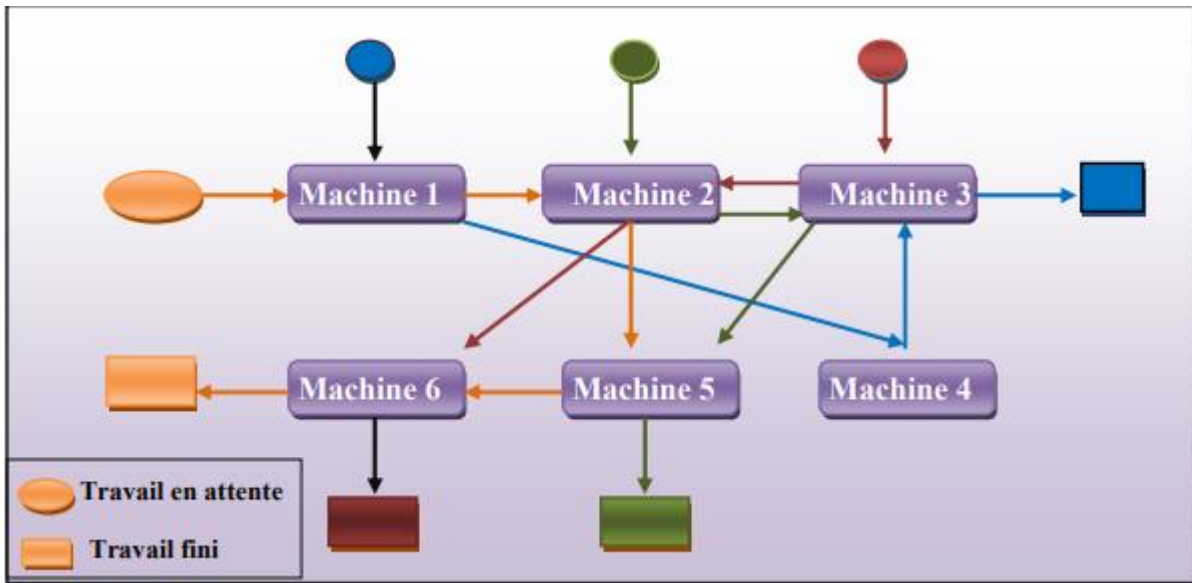


Figure 8 : Job shop classique

## 5.2 Job shop flexible

Le problème d'ordonnancement d'atelier flexible (FJSP) est une généralisation du JSP classique qui permet de traiter des opérations sur une machine à partir d'un ensemble de machines alternatives. Par conséquent, le FJSP est plus difficile à calculer que le JSP. De plus, le problème d'ordonnancement des opérations, le FJSP présente une difficulté supplémentaire causée par le problème d'affectation des opérations à un ensemble de machines disponibles. Ce problème est connu pour être fortement NP-Dur même si chaque tâche comporte au plus trois opérations et qu'il y a deux machines [17].

Dans ce modèle, les machines qui effectuent la même opération sont groupées dans un même étage

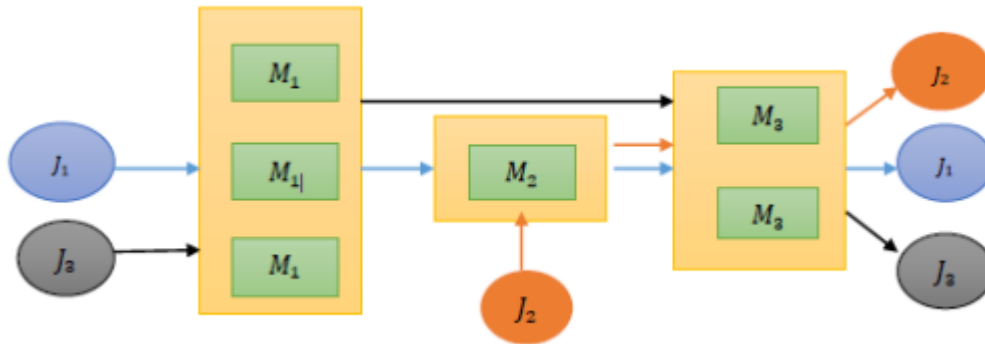


Figure 9 : Job shop flexible

Dans les lignes suivantes, nous présentons quelques problèmes d'ordonnancement spécifiques au le problème Job-Shop.

## 6 Le problème d'ordonnancement job shop avec contrainte de blocage

Ce problème du job shop avec des machines ayant un espace de stockage limité ou inexistant. Ceci fait intervenir une nouvelle contrainte au problème du job shop classique qui est la contrainte de blocage. Celle-ci signifie que chaque opération reste sur la machine en cours, même si elle a terminé son traitement, jusqu'à ce que la machine suivante sur laquelle elle va s'exécuter soit disponible. La contrainte de blocage, apparaît dans le cas où les machines ne disposent pas d'espace de stockage. L'absence de celui-ci a pour effet qu'un job ne peut quitter sa machine que si la prochaine sur laquelle il va s'exécuter soit libre.

Ce problème est connu dans la littérature comme étant NP\_difficile, cette complexité est due à l'explosion combinatoire du nombre de solutions qui croît exponentiellement avec la taille du problème.

Le problème du job shop avec la contrainte de bocage est défini comme suit: soit  $n$  jobs, composé chacun de  $k_i$  ( $i=1, \dots, n$ ) opérations, qui doivent être traitées, selon un ordre donné, sur  $m$  machines différentes avec un espace de stockage limité ou inexistant. L'atelier de production est soumis aux contraintes suivantes :

- l'ordonnancement est non préemptif.
- l'exécution d'une opération ne peut être interrompue.
- Chaque machine ne peut traiter qu'une seule opération à la fois.
- Le temps de traitement (processing time) de chaque opération est connu au préalable.
- Le cyclage n'est pas permis (un job ne passe pas sur la même machine deux fois dans la même gamme).

L'objectif étant de minimiser le temps total d'exécution (makespan).

Dans [23] proposent de résoudre ce problème par un algorithme basé sur la métaheuristique Recherche Taboue.

L'apport réside dans la technique de choix du voisinage que nous avons proposée qui est très importante pour la fiabilité de la méthode. Ils utilisent une technique qui opère sur le chemin critique trouvé dans le graphe alternatif modélisant le problème. Cette technique construit toujours des solutions réalisables. De plus, il est implémenté une stratégie d'évaluation de voisinage qui vise à calculer uniquement les dates de début d'un sous-ensemble d'opérations concernées par le mouvement et non pas toutes les opérations. Ceci a permis de réduire les temps d'exécution d'une manière considérable tout en améliorant les qualités des solutions trouvées.

## 6.1 Modélisation par un graphe alternatif

Un graphe alternatif est une généralisation des graphes disjonctifs. Notons  $\sigma(O_i)$  l'opération qui suit  $O_i$  dans le travail. Nous distinguons l'un des types de contraintes disjonctifs entre chaque couple d'opérations  $(O_i, O_j)$  qui s'exécutent sur la même machine  $M$  [24]:

## 6.2 Arcs disjonctifs de blocage (arcs alternatifs)

Ce type de contraintes apparaît lorsque la machine  $M_k$  ne contient pas un espace de stockage. Dans ce cas, les deux opérations  $O_i$  et  $O_j$  sont reliés par deux arcs alternatifs de  $\sigma(O_i)$  vers  $O_j$  et de  $\sigma(O_j)$  vers  $O_i$  ce que signifie que l'opération  $O_j$  ou  $O_i$  ne peut commencer qu'après le début de l'opération  $\sigma(O_i)$  ou  $\sigma(O_j)$  ce qui assure que la machine  $M_k$  est libre.

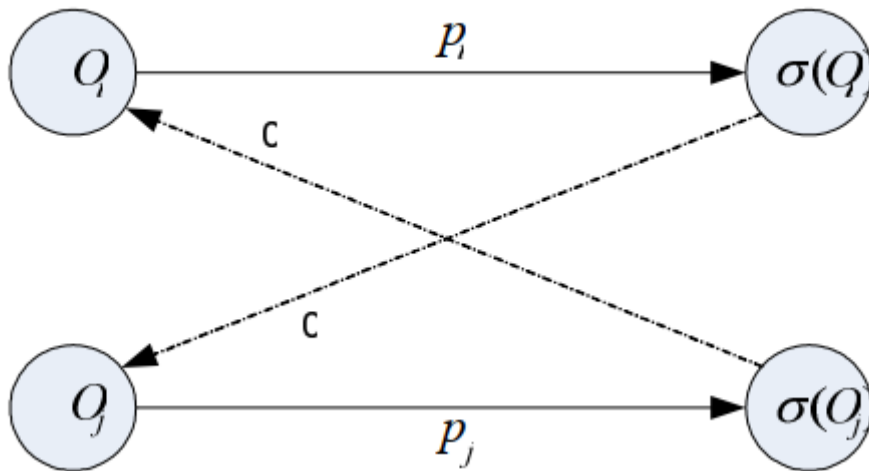


Figure 10: arcs croisés dans un job shop avec contrainte de blocage

### 6.3 Exemple d'application :

Soit trois jobs ayant les gammes opératoires suivantes [23]:

Les Jobs	La gamme des Jobs	Les durées
J1	M1 , M2,M3	5,3,8
J2	M2	6
J3	M2,M1,M3	8,2,7

Tableau 2 : Exemple d'un job shop avec blocage à trois jobs et trois machines

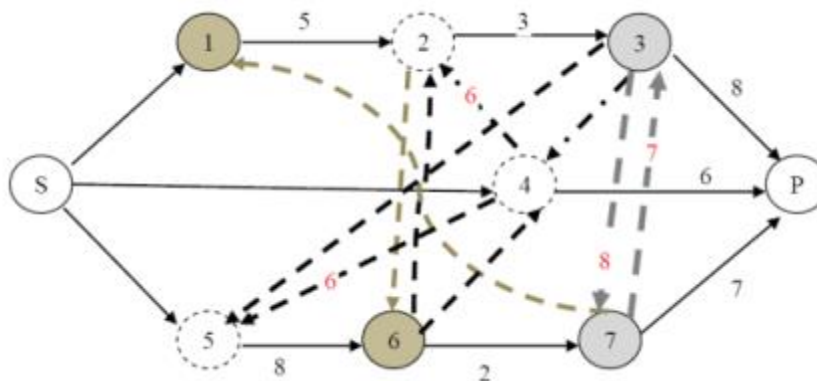


Figure 11: Graphe alternatif correspondant à l'exemple du tableau 1

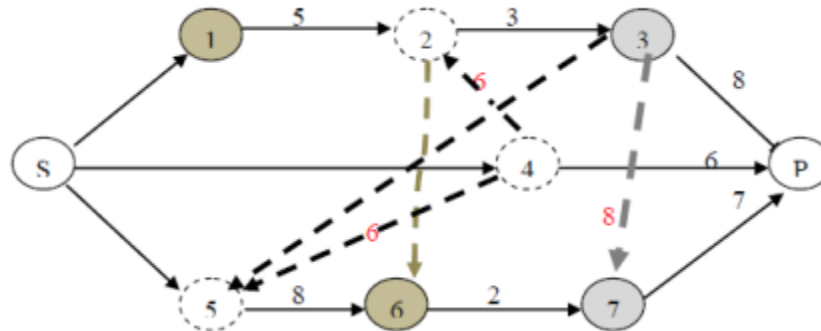


Figure 12: Graphe représentant une solution

## 7 Ordonnancement d'ateliers de type Job shop avec deux opérateurs

L'ordonnancement d'ateliers de type job shop étudiés en présence d'opérateurs dont l'affectation change selon un mode libre, où l'ordre de tâches est connu au départ et l'objectif est la minimisation du retard algébrique maximum. Les opérateurs ont des niveaux différents de performance [25].

### 7.1 Description du problème

Dans un problème de job shop,  $n$  tâches doivent être traitées par  $m$  machines. Chaque tâche  $j, j = 1, \dots, n$  doit être traitée par chacune des machines  $i, i = 1, \dots, m$  et ce dans un ordre fixe qui lui est propre.

Les opérations nécessitent la présence de l'un des  $k$  opérateurs ( $k < m$ ) pour toute la durée de son traitement. L'impact de la présence d'un nombre d'opérateurs inférieur au nombre de machines  $y$  est ainsi modélisé par des temps d'attente durant lesquels une machine restera inactive même si elle a des opérations en attente. Les paramètres  $p_{ij}$ ,  $d_{ij}$  et  $C_{ij}$  représentent respectivement le temps de traitement, la date d'échéance et la date de fin de traitement de la tâche  $j$  sur la  $i^{\text{ème}}$  machine [25].

Un opérateur de type  $h$  a une cadence  $vh$ . Selon le mode libre, le changement d'affectation des opérateurs peut se produire à tout instant. Un opérateur pourra alors interrompre le traitement d'une tâche pour s'occuper d'une autre tâche. Cette tâche pourra par

la suite être reprise par ce même opérateur ou être confiée à un autre opérateur. L'objectif est de minimiser le retard algébrique maximum,  $L_{\max} = \max_{j=1, \dots, n} L_{mj}$  où  $L_{ij} = C_{ij} - d_{ij}$ .

## 7.2 La Formulation de problème

Notons le problème  $Jm/res Lkh1, S, vh, f/L_{\max}$ . Les symboles  $S$  pour indiquer que l'affectation se fait sous un ordre de tâches donné  $S, vh$  qui indique que les opérateurs ont des cadences différentes et  $f$  qui indique que le mode de changement d'affectation est libre.

Tel que  $L$  : les types d'opérateurs

Le problème  $Jm/res 2kh1, S, vh, f/L_{\max}$  est résoluble en temps polynomial. La preuve se fera via la méthode de restriction. Les contraintes de passage des opérations sur les machines et celles générées par  $S$ , forment un graphe de précedence. Ainsi, le problème peut être vu comme un problème de deux machines parallèles uniformes avec un graphe de précedence et la possibilité de préemption, noté par  $Q2/prec, prmp/L_{\max}$ , où les machines, les tâches et le graphe de précedence représentent respectivement les 2 opérateurs, les  $m * n$  opérations, et les relations de précedence entre elles. La possibilité de préemption modélise le mode de changement d'affectation libre. Ce problème est résoluble en  $O(n^2)$  [25].

## 8 Job shop à deux machines et algorithme de Johnson

Cet algorithme proposé par Jackson, repose sur la réduction du problème  $J2//C_{\max}$  dont les tâches ont au plus deux opérations [26].

L'algorithme de Jackson s'applique à un job shop à deux machines sur lesquelles doivent être exécutés  $n$  tâches, qui ont au plus deux opérations. Le problème job shop à deux machines ( $J2//C_{\max}$ ) admet une solution connue et qui est de complexité polynomiale. On le résout par l'algorithme de Johnson. L'algorithme est optimal pour le cas de job shop à deux machines.

Chaque tâche nécessite pour son exécution le passage sur deux machines : les machines. Soient  $t_{AJ}$  et  $t_{BJ}$ , les temps d'exécution de la tâche  $i$  sur les machines A et B respectivement.

Comme vu, le critère d'ordonnancement est la minimisation du temps total d'exécution des tâches  $C_{\max}$  sur les deux machines.

1. Tant qu'il reste des tâches non placées dans la séquence,
  - 1.1. Trouver la tâche pour lequel  $\min_j \{t_{Aj}, t_{Bj}\}$
  - 1.2. Si le temps d'exécution minimale de la tâche est celui de la machine de départ alors cette tâche est mise au début de la séquence.
  - 1.3. Si le temps d'exécution minimale de la tâche est celui de la seconde machine alors cette tâche est mise à la fin de la séquence.
2. FinTantQue

Algorithme 3: L'algorithme de Johnson

Un algorithme - dont une mise en œuvre est possible avec une complexité en  $O(n \log n)$  - suit le schéma suivant:

- 1 Constituer les groupes de travaux  $U = \{i / t_{Aj}\}$ ;  $V = \{i / t_{Aj} \geq t_{Bj}\}$ .
- 2 Ordonner U suivant l'ordre SPT sur la première machine ( $t_{Aj}$ ) et V suivant l'ordre LPT sur la deuxième machine ( $t_{Bj}$ ).

### 8.1 Exemple d'application

Le Tableau résume les temps d'exécution des cinq tâches sur les machines A et B

Tâches T	T1	T2	T3	T4	T5
Temp d'exécution $t_{Aj}$	40	140	70	210	20
Temp d'exécution $t_{Bj}$	50	40	140	60	210

Tableau 3: Les temps d'exécution des tâches

En appliquant l'algorithme de Johnson vu précédemment, l'ordonnancement optimal est [T5, T1, T3, T4, T2].

La tâche pour laquelle  $\min(t_{Aj}, t_{Bj})$  est la tâche numéro cinq T5, le temps d'exécution minimale de la tâche est celui de la machine de départ donc T5 est mise au début de la séquence. Puis, T1 est mise en deuxième position. Puis T2 est mise en dernière position. Puis T4 est mise en avant dernière position. Enfin, T3 est mise à la dernière place disponible.

## **9 Conclusion**

Plusieurs travaux de recherche ont été faits pour trouver une solution au problème job shop avec contrainte de transport dont ils ont appliqué des méthodes exactes et des méthodes approchées, leur objectif est toujours de rapprocher plus à la réalité.

Dans le chapitre suivant nous allons, de notre part, de donner notre proposition sur la résolution de ce problème à partir de créer un logiciel permet de calculer le Makespan en utilisant les algorithmes génétiques.

# **Chapitre 04**

## **Etude Pratique**

---

## 1 Introduction

Dans ce chapitre nous allons présenter notre problème d'ordonnancement ainsi que la démarche de résolution proposée.

L'objectif du travail est de trouver, en utilisant la méthode méta-heuristique qui est l'algorithme génétique, les meilleures solutions possibles pour un ordonnancement sous la contrainte de transport, tout en minimisant le temps total d'exécution (Makespan).

## 2 L'environnement de développement

Notre travail est écrit dans L'environnement de développement Matlab version 7.9 sous Windows 10 Professionnel. Et nous avons utilisé comme élément matériel un ordinateur TOSHIBA qui possède les propriétés suivant :

### Informations système générales

#### Édition Windows

Windows 10 Professionnel

© 2019 Microsoft Corporation. Tous droits réservés.

#### Système

Processeur : Intel(R) Pentium(R) CPU B950 @ 2.10GHz 2.10 GHz

Mémoire installée (RAM) : 2,00 Go (1,89 Go utilisable)

Type du système : Système d'exploitation 64 bits, processeur x64

Figure 13: Les propriétés de notre PC

### 2.1 Matlab « Mat (matrix) et laboratory (lab) »

Est un langage de programmation de quatrième génération et un environnement de développement ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. Les utilisateurs de MATLAB (environ un million en 2004) sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Matlab peut s'utiliser seul ou bien avec des *toolbox* (« boîte à outils »).

### 3 Formulation du problème

#### 3.1 Description du System à étudier

L'objectif de notre travail est de trouver, en utilisant l'algorithme génétique les meilleures solutions pour un ordonnancement des produits tout en minimisant le temps total d'exécution (Makespan) qui respect les contraintes imposés.

Le problème est défini donc par  $J4/\beta1/Cmax$  selon la notation la plus utilisée en ordonnancement.

#### 3.2 Fonction objectif

Minimisation de Makespan ( $Cmax$ )

#### 3.3 Les contraintes

- 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 recirculation (un produit ne peut exécuter sur une machine qu'une seule fois).
- $\beta1 = \emptyset$  : Pas de préemption (le produit doit terminer toute l'opération avant de quitter la machine).
- Les produits passent dans seule direction
- Les produits sont disponibles à  $t=0$ .

### 4 Démarche de résolution

Nous avons utilisé ces algorithmes génétiques qui nous ont aidés à résoudre ce problème.

#### 4.1 Codage

##### 4.1.1 Le codage basé sur les jobs

Ce codage représente une liste ordonnée des Jobs dont la longueur du chromosome est équivalente à  $n$  gènes ou  $n$  Jobs, chaque gène représente un Job et l'ordonnement de tous les Jobs se traduit par le lancement des opérations appartenant au premier Job (premier gène) selon l'ordre décrit dans la gamme opératoire du premier Job.



Figure 14: Codage de chromosome

Le choix de ce type de codage est basé sur les points suivants :

- Un seul point d'entrée et un seul point de sortie pour tous les produits.
- Le transport des produits s'effectue dans seule direction (Job shop).

## 4.2 Création de Population initiale

1. Début
2. Enter la taille de la population.
3. Enter le nombre de job  $n$ .
4. Tant que  $ch. \leq$  la taille de population faire  
Créer nouvelle chromosome  $chi$   
Si  $chi = ch.$   
Créer un nouvelle  $chi$  différent à  $ch.$   
FinSi  
Fin tant que
5. Fin

## 4.3 Le croisement

### 4.3.1 Le croisement de l'ordre des jobs

La figure 15 ci-dessous explique plus clairement ce qu'est ce type et comment il marche

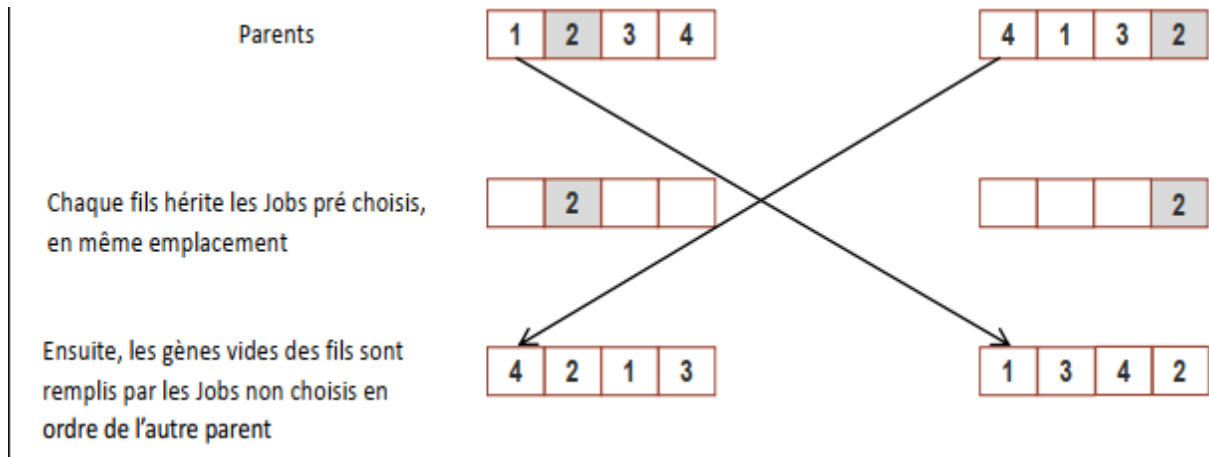


Figure 15:Le croisement par des jobs

## Algorithme de ce croisement

1. Début
2. Pour tout les chromosomes  $ch$ . faire
  - Choisir les bons chromosomes
  - $[x \ y]=size(chcroi)$ .
  - Pour deux chromosomes choisis  $ch1, ch2$ 
    - Fixer une valeur aléatoire dans l'intervalle  $[1, y]$
    - Remplacer les valeurs non fixé de  $ch1$  vers  $ch2$  et l'inverse
  - Fin Pour
3. Fin Début

## 4.4 Mutation

### 4.4.1 Mutation par position

Le principe de la mutation de position consiste à choisir deux positions sur le chromosome choisis, puis permuter les deux gènes. Voilà un exemple (figure 16)



Figure 16:Mutation entre 1 et 4

## Algorithme de cette mutation

1. Début
2. Choisir les chromosomes  $chmu$  ayant probabilité  $0.1 \leq pm \leq 0.2$
3. Si le chromosome est choisis
  - Deux gènes est choisis aléatoirement
  - On inverse les position de ces deux gènes
  - Fin Si
4. Fin Début

## 4.5 Selection

### 4.5.1 La sélection par la roue de fortune

C'est une roue décomposée en plusieurs secteurs, où chacun correspond à un chromosome de population. La superficie de chaque secteur est relative à la valeur de la fonction objective. Donc, plus la force d'adaptation de l'individu est importante, plus la superficie de son secteur est importante. L'évaluation permet de calculer la fitness d'un individu. En cas de problème de minimisation, la finesse croit inversement avec la fonction objective. La fitness d'un individu  $x_i$  est mesurée par :

$f(x_i) = 1 / C(x_i)$  où  $C(x_i)$  est la valeur de la fonction objectif obtenu avec  $x_i$   
 La probabilité de sélection d'un individu  $x_i$ , notée  $prob(x_i)$ , est proportionnelle à sa fitness  $f(x_i)$  et telle que  $\sum prob(x_i) = 1$ . Elle est définie :

$$prob(x_i) = \frac{f(x_i)}{\sum_1^n f(x_i)}$$

## Algorithme de Selection

1. Début
2. Classer les individus de la population selen leur valeurs de fitness (ordre croissant).
3. Pour  $i=1$  à  $n$  faire
  - Calculer la probabilité de Sélection  $p_i$
  - Générer un nombre  $x$  appartenant l'intervalle  $[0,1]$
  - Si  $x < p_i$  alors
  - Sélectionner l'individu  $i$ .
  - Fin Si
- Fin Pour
4. Fin

## 4.6 Evaluation des individus

L'évaluation des individus peut se faire selon le critère le Makespan, Sur la base de ce critère, la *fitness* de chaque individu est estimée.

Généralement, l'évaluation de fitness des chromosomes se fait en trois étapes :

- La construction de l'ordonnancement correspondant au chromosome.
- Le calcul de fonctions objectives : Makespan
- Le calcul de fitness en fonction de la fonction objectif.

## 5 Condition d'arrêt

La condition d'arrêt qui est choisie pour notre résolution. C'est : Arrêt après un nombre de générations fixé a priori. Un nombre de générations est produit signifie qu'un pourcentage de l'espace des solutions est exploré.

## 6 Les données utilisés

Les trois tableaux suivants représentent successivement Les temps des opérations des jobs sur les machines, le routage de job et les temps de transport entre les machines.

	Machine 1	Machine 2	Machine 3	Machine 4
Job 1	50	56	70	39
Job 2	0	15	75	23
Job 3	40	0	25	30
Job 4	60	10	35	20

Tableau 4 : Les temps des opérations des jobs sur les machines

	Machine 1	Machine 2	Machine 3	Machine 4
Job 1	M1	M2	M3	M4
Job 2	M2	M3	M4	/
Job 3	M1	M3	M4	/
Job 4	M1	M4	M2	M3

Tableau 5: La routage des jobs

	Machine 1	Machine 2	Machine 3	Machine 4
Machine 1	1.5	5	3	5
Machine 2	0	6.5	3	5
Machine 3	1.5	0	8	5
Machine 4	1.5	8	3	13

Tableau 6 : Les temps de transport entre les machines

## 7 Le programme :

### 7.1 Comment ça marche

Le contenu de code du programme consiste un ensemble des M-files que nous appelons dans le code général M-file « AlgoJobShop.m ».

Les étapes de l'appel sont les suivantes :

- Nous appelons « population.m » pour écrire la population initiale de vingt chromosomes avec quatre gènes en utilisant la fonction de Matlab « randperm » qui permet de créer les chromosomes à partir de 1 à 4 avec permutation chaque fois. Nous avons déjà dit que les gènes est le codage des jobs.
- Nous appelons « selection.m » qui évalue les individus à partir de calculer le fitness de chaque individu de la population courante.
- Nous appelons « croisement.m » pour appliquer le croisement par des jobs sur chaque deux parent choisis.
- Nous appelons « mutation.m » pour appliquer la mutation par valeurs sur un chromosome choisis chaque fois.
- Nous appelons Guntt.m pour afficher le diagramme de Gantt.

- Après setp itérations, il sera obtenu les résultats et affiche le Cmax avec les meilleurs solutions et le diagramme d Gantt.

La figure suivante représente les ensembles des M-files qui l'on utilisés sous Matlab.

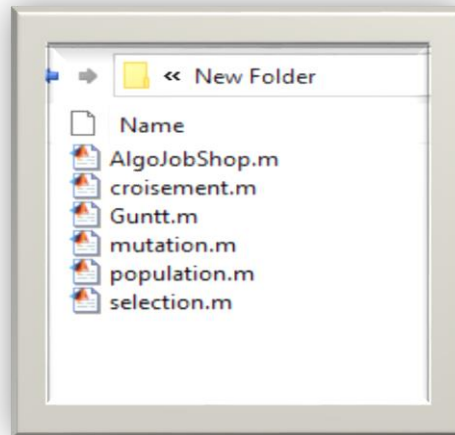


Figure 17: Les ensembles des M-file

## 7.2 Le schéma de résolution du problème

Le schéma montre les étapes de la démarche de résolution utilisée dans notre travail.

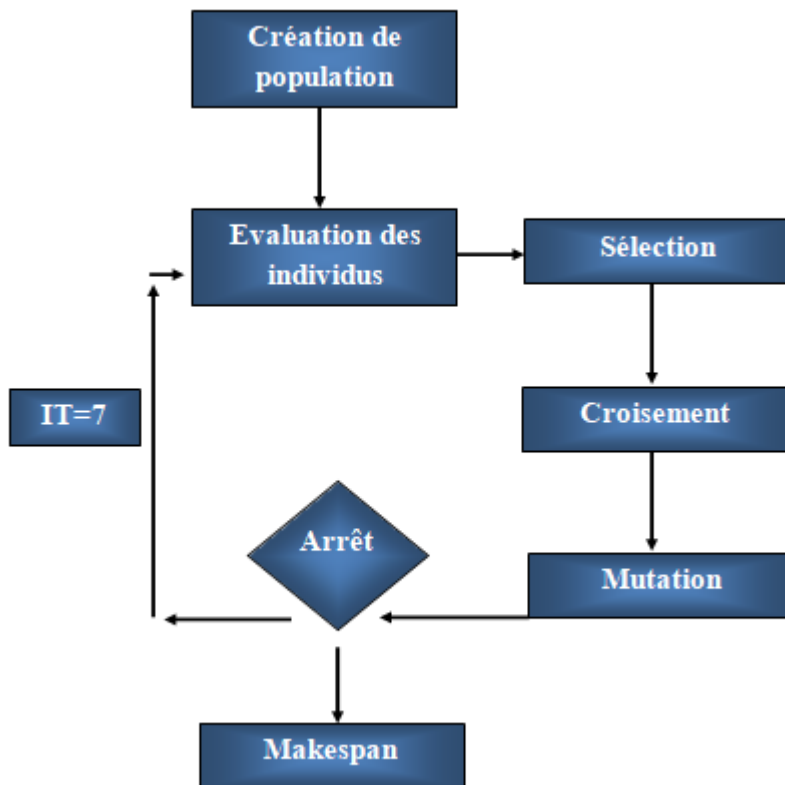


Figure 18:Le schéma de résolution du problème

### 7.3 Illustration de logiciel

Notre Logiciel se compose des fenêtres qui permet de saisir le nombre des jobs et le nombre des machines et importer les temps d'opérations de chaque tâche sur les machines, la routage des jobs et les temps de transport puis des fenêtres représente les résultats automatiquement et par le diagramme de Gantt .

Maintenant, nous vous montrons un ensemble des figures qui affichent le contenu de notre programme et seront présentées de manière ordonnée .Sachant que nous avons passer l'interface « login.m » à l'interface « pageinitiale.m ».

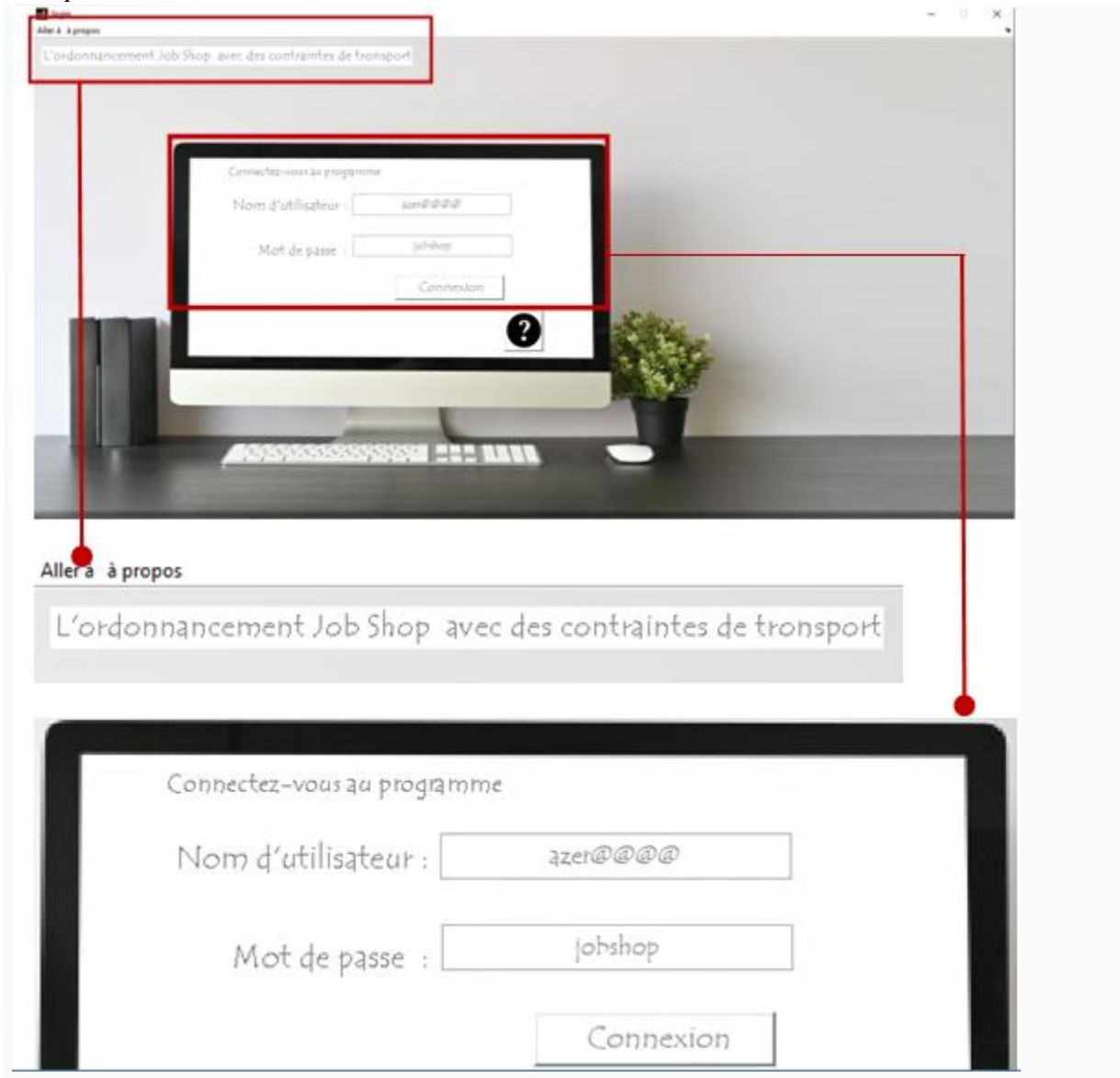


Figure 19: la page de Connexion au programme



Figure 20: La page d'entrée de logiciel

Lorsque 'on clique sur 'Nouveau calcul' il sera afficher la démarche de saisir notre données comme il est montré dans la figure 21 et 22.

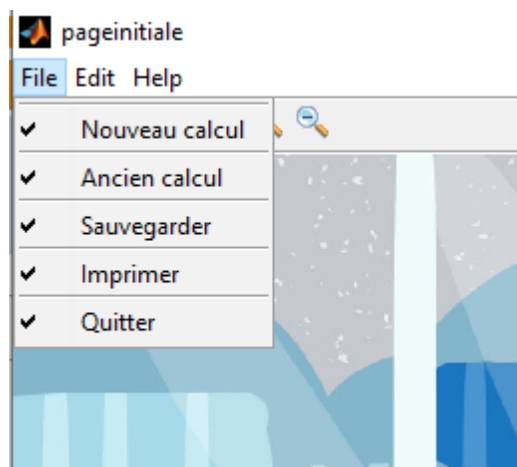


Figure 21: Le bouton "Nouveau calcul"

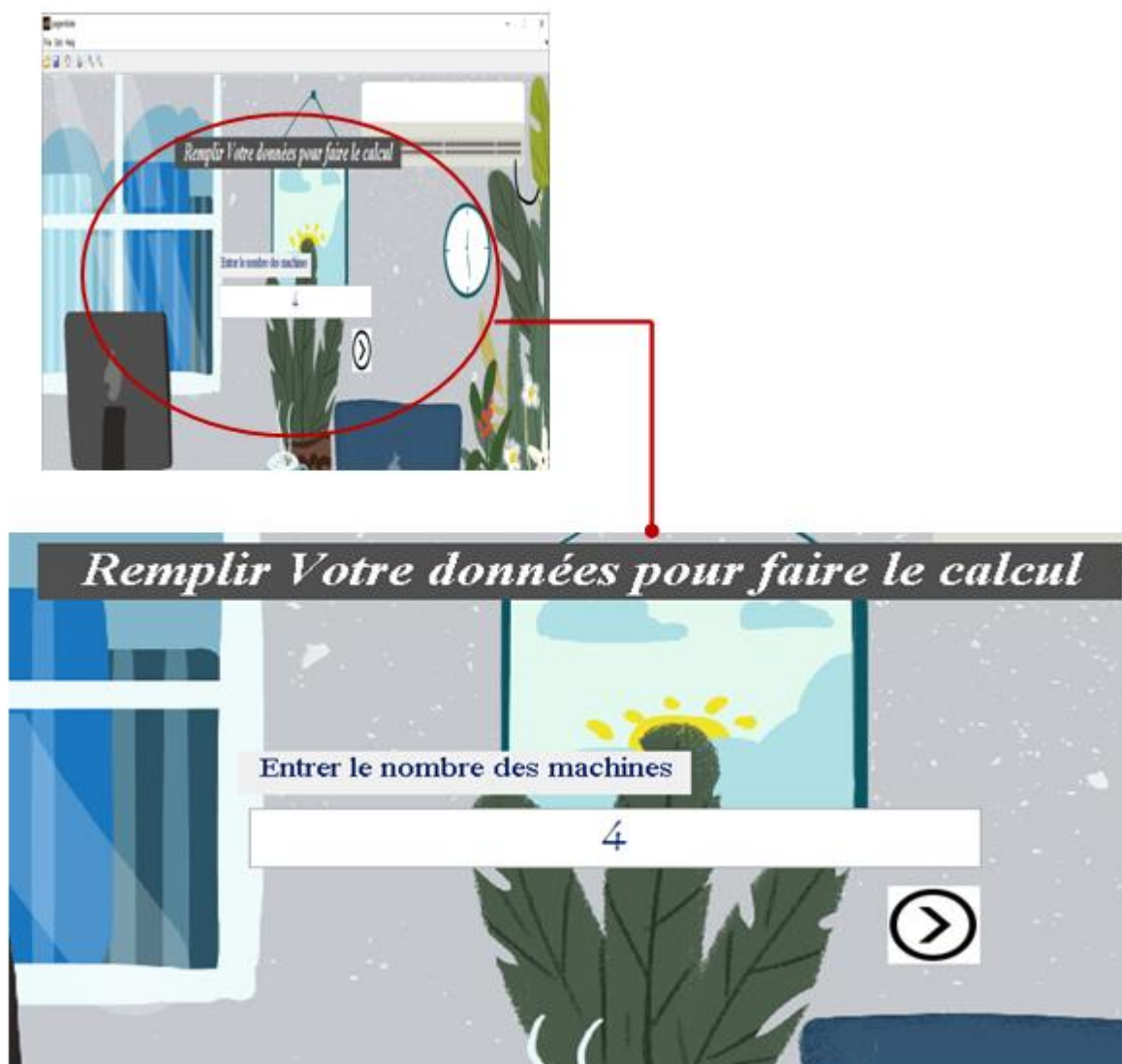


Figure 22: la page qui permet de choisir le nombre des machines



Figure 23: La page qui permet de choisir le nombre des jobs



Figure 24 : La page qui permet de saisir la taille de population

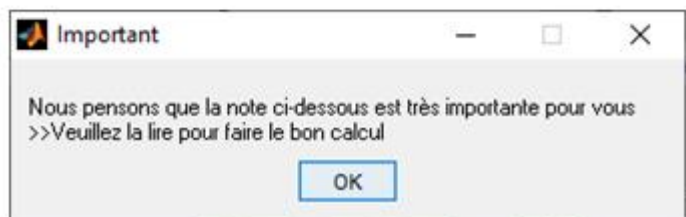


Figure 25: après saisi la taille de population

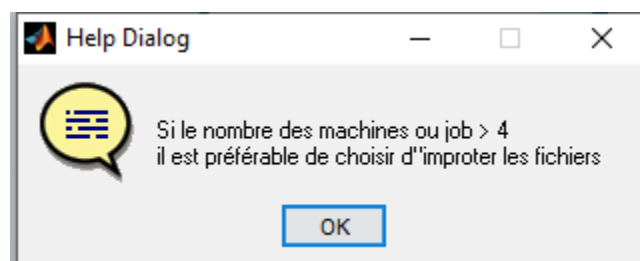


Figure 26: Le contenu de la note



La figure 25 illustre que nous avons d'abord cliqué sur « exécuter » pour faire le calcul puis « afficher » pour afficher les résultats illustrés et il ya commande « Diagramme de Gantt » qui affiche le diagramme de Gantt pour représenter notre solutions.

### Résultat obtenu représenté par le diagramme de Gantt

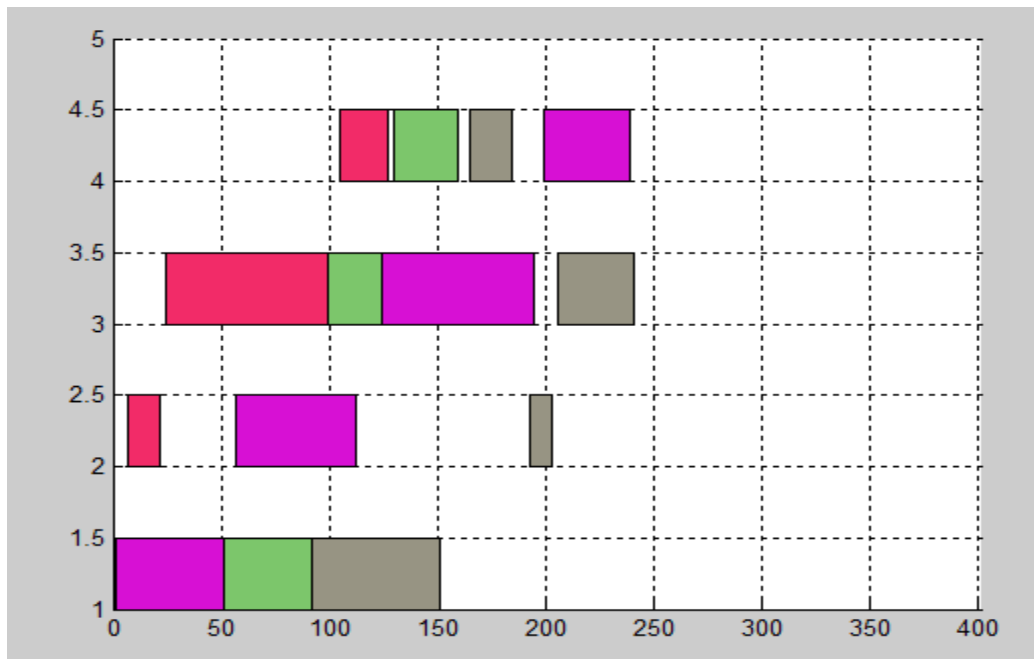


Figure 29: Cmax obtenu par le Diagramme de Gantt

### 7.3.1 Illustration des barre de menus et Tools



Figure 30: Le contenu de "Edit"

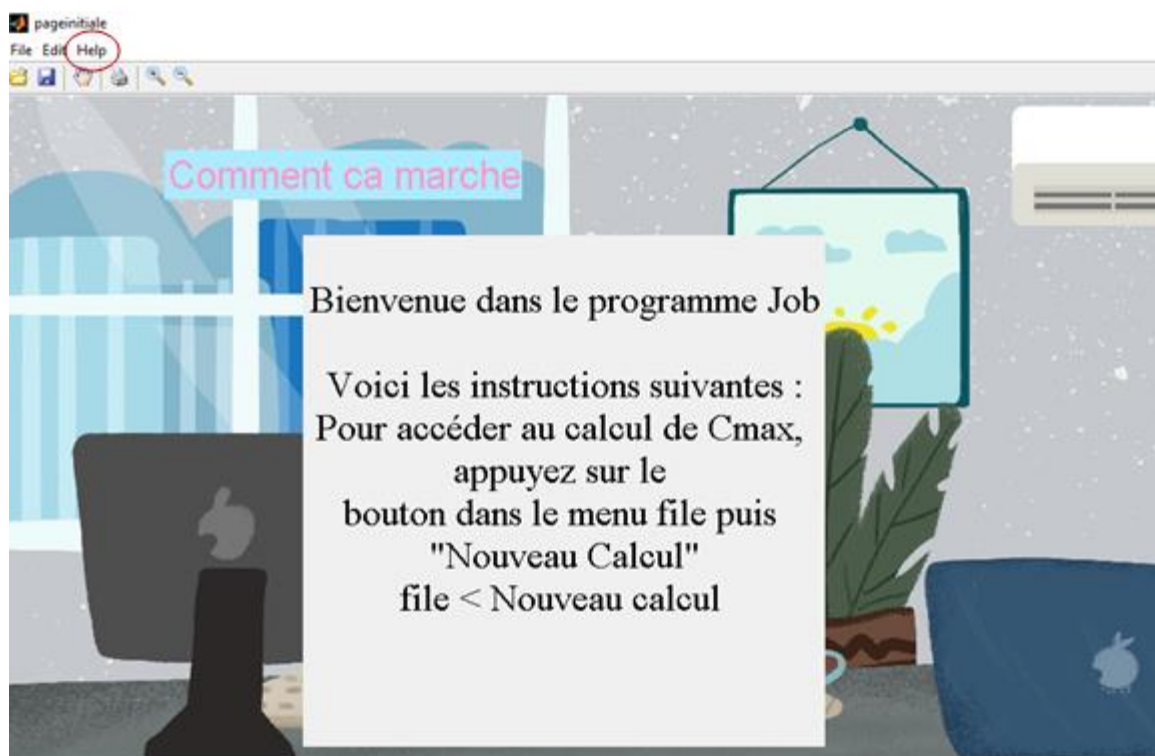


Figure 31 : Le contenu de "Help"

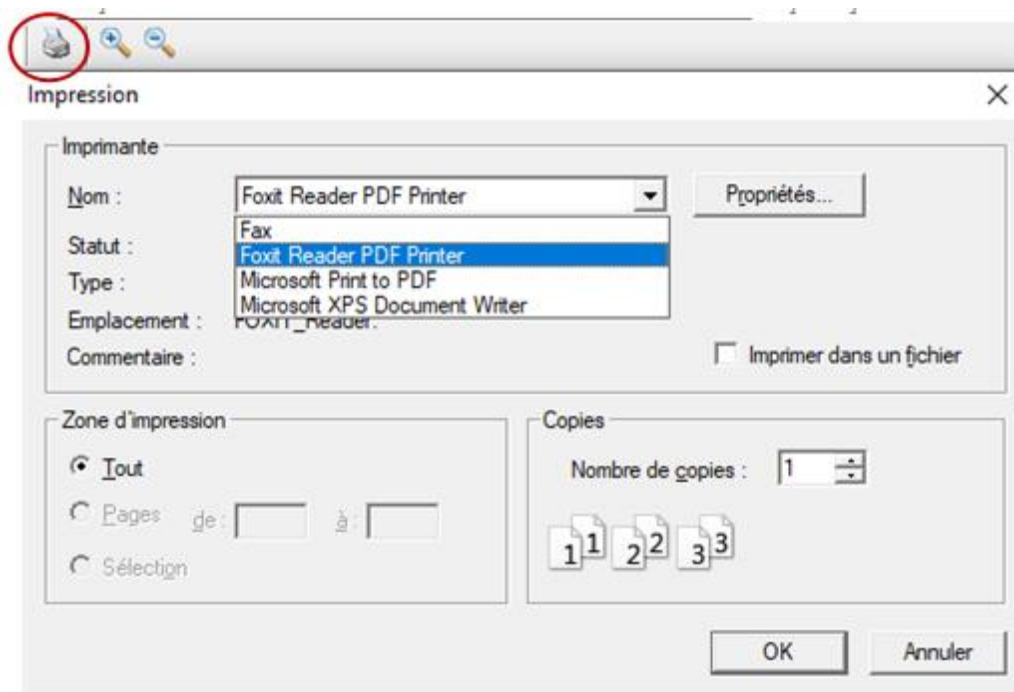


Figure 32: Le contenu de L'icone imprimante

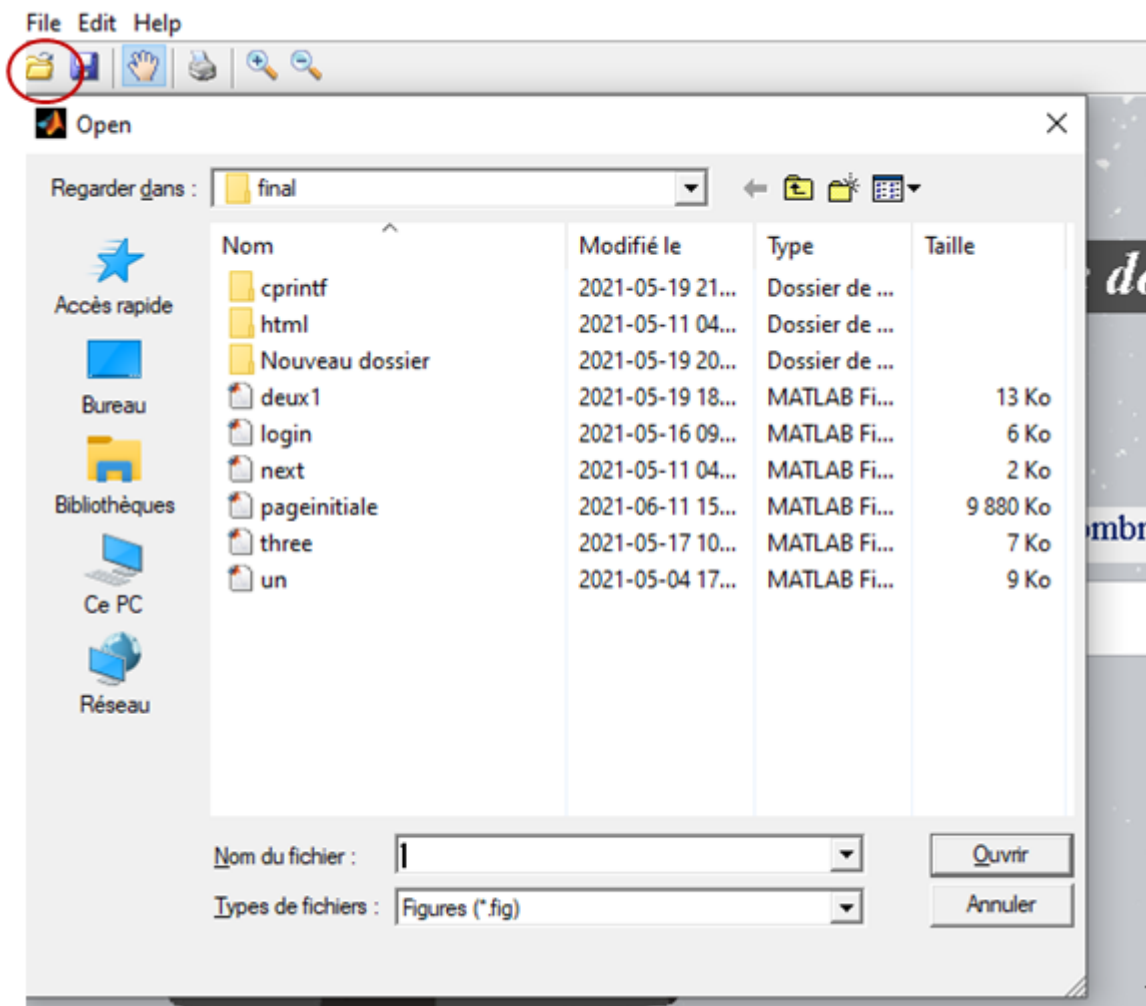


Figure 33: Le Contenu de l'icone "Ouvrir fichier"

Pour ce problème de quatre produits nous avons remarqué que le programme nous permet de trouver la meilleure solution avec un pourcentage presque de 100%. Les résultats obtenus sont très motivants.

## 8 Conclusion

Nous avons présenté dans ce chapitre le programme développé, le langage et ses outils utilisés pour résoudre le problème que nous avons couvert dans ce mémoire.

Ensuite, nous avons présenté les algorithmes que nous avons adoptés pour la résolution du problème de job shop avec contrainte de transport où les produits sont transportés dans un sens unidirectionnel et on a présenté l'implémentation de notre programme et nous avons considéré un exemple avec les résultats obtenus.

## Conclusion générale

Le problème d'ordonnancement des ateliers de type Job Shop est l'un des problèmes d'ordonnancement intensivement étudiés. C'est un problème extrêmement complexe. Il est classé parmi les problèmes combinatoires difficiles au sens fort. Cette complexité est due à l'explosion combinatoire du nombre de solutions qui croît exponentiellement avec la taille du problème. L'utilisation de méthodes exactes en vue de l'obtention de solutions optimales semble non réaliste. Le recours à des méthodes approchées comme les heuristiques est devenu incontournable.

Dans ce mémoire, nous nous sommes intéressés aux problèmes d'ordonnancement des ateliers de production. Plus particulièrement, les problèmes de job-shop avec contrainte de transport en utilisant l'algorithme génétique pour la résolution de cette problématique.

Ensuite, nous avons réalisé un programme informatique permet de résoudre et nous nous donne le makespan avec les meilleurs ordonnancements. Donc les méthodes approchées ne présentent dans certains cas des meilleures solutions.

Pour enrichir ce travail, nous pouvons finalement ajouter des perspectives telles que :

- Notre travail nécessite des améliorations dans l'algorithme génétique
- La proposition de méthodes exactes pour la résolution de ce type de problème.
- Développée un autre programme traite toute les cas des problèmes d'ordonnancement job shop ou bien toute les cas des problèmes d'ordonnancement job shop avec contraintes de transport.

## Références bibliographiques

- [1] VACHER J. Ph, *Un système adaptatif par agents avec utilisation des algorithmes génétiques multi-objectifs : Application à l'ordonnancement d'atelier de type job-shop  $N \times M$* , Thèse de Doctorat, Université de Havre, 2000.
- [2] BLONDEL F, *Gestion de la production*, 3ème édition, DUNOD, Paris 2004.
- [3] KEBABLA Mebarek, *Utilisation des stratégies Métaheuristiques pour l'ordonnancement des ateliers de type Job Shop*, Thèse de Magister, Université de Batna, 2008.
- [4] JAVEL G, *Organisation et Gestion de production*, 3ème édition, DUNOD, Paris, 2004
- [5] GIARD V, *Gestion de la Production*, 2ème édition, ECONOMICA, Paris, 1988.
- [6] FONTANILI F, *Intégration d'outils de simulation et d'optimisation pour le pilotage d'une ligne d'assemblage multi produit à transfert asynchrone*, Thèse de doctorat, Université de Paris XIII, 1999.
- [7] LETOUZEYA A, *Ordonnancement interactif basé sur des indicateurs : Applications à la gestion de commandes incertaines et à l'affectation des opérateurs*, Thèse de Doctorat, Institut National Polytechnique de Toulouse, 2001.
- [8] HANNACHE Aboubakr et LEMMOUIS Abdelhamid. *Résolution d'un problème d'ordonnancement de type job shop avec contrainte de transport*. Thèse de doctorat. Université de Tlemcen, 2016.
- [9] CARLIER J et CHRETIENNE P, *Problèmes d'ordonnancement : modélisation / complexité /algorithmes*, Edition Masson, 1988.
- [10] PINEDO M, *Scheduling: Theory, Algorithms and systems*, Prentice-Hall, Englewood Clis, New Jersey, 1955.
- [11] NADIR Ramla. *Un problème d'ordonnancement de type Job Shop dans un environnement dynamique*. Diss. Université de Msila, 2019.
- [12] LOPEZ P et ESQUIROL P, *L'ordonnancement*, ECONOMICA, Paris, 1999.
- [13] BAPTISTE P, *une étude théorique et expérimentale de la propagation des contraintes de ressources*. Thèse de doctorat, université Technologique de Compiègne, 1998.
- [14] BEN AHMED Razika et KHERROUBI Zakiya. *Optimisation d'un problème d'ordonnancement de type job shop avec contrainte de transport*. Diss.
- [15] DUMMOND R, BRESINA M, J et SWANSON K, *Just-In-Case Scheduling* .Twelfth National Conference on Artificial Intelligence (AAAI-1994), Seattle, USA, 1994.
- [16] MESGHOUNI K, *Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de la production*, Thèse de Doctorat, Université des Sciences et Technologies de Lille 1, 1999.

- [17] NOURI Housseem Eddine, OLFA Belkahla Driss, and GHEDIRA Khaled, *Solving the flexible job shop problem by hybrid metaheuristics-based multiagent model*, Journal of Industrial Engineering International 14.1 (2018): 1-14.
- [18] LARABI Mohand. *Le problème de job-shop avec transport: modélisation et optimisation*. Diss. Université Blaise Pascal-Clermont-Ferrand II, 2010.
- [19] DOLGUI Alexandre, and Proth J.M, *Les systèmes de production modernes. Volume 1: Conception, gestion et optimisation*. Lavoisier, Paris, 2006.
- [20] BELHOUL Lyes, *Résolution de problèmes d'optimisation combinatoire mono et multi-objectifs par énumération ordonnée*. Diss. Université Paris Dauphine-Paris IX, 2014.
- [21] MOUHOUB Nasser Eddine. *Algorithmes de construction de graphes dans les problèmes d'ordonnancement de projet*, Thèse de Doctorat, Sétif, 2011.
- [22] BAPTISTE Pierre, *Gestion de production et ressources humaines: méthodes de planification dans les systèmes productifs*, Presses inter Polytechnique, 2005.
- [23] DEBBAH Adel et and AITZAI Abdelhakim, *Recherche Taboue pour le problème d'ordonnancement job shop avec contrainte de blocage*.
- [24] BENMEDJDOUB Brahim. *Job shop avec contraintes de blocage et de sans attente*. Diss. 2010.
- [25] BENKALAI Imène, REBAINE Djamel et BAPTISTE Pierre, *Ordonnancement d'ateliers de type job shop avec opérateurs*.
- [26] RUHLMANN Carine. *Étude du problème de job shop avec un convoyeur*. Université du Québec à Chicoutimi, 2007.
- [27] MOUHOUB Nasser Eddie, BELOUADAH Hocine, et BOUBETRA Abdelhak, *Algorithme de construction de graphe PERT à partir d'un graphe des potentiels donné, Informatica* , Thèse de Doctorat ,2006 .
- [28] DILMI Ramzi, *Application de la programmation par contraintes au problème d'optimisation du plan du test logiciel*, Diss, Université de M'sila, 2017.
- [29] BOUNIF Mohamed Elhadi, *Optimisation à base de simulation pour le développement des systèmes décisionnels*, Diss, Université de M'sila, 2015
- [30] GAHGAH Hadjer, *Problème d'emploi de temps: proposition un algorithme bio-inspiré*. Diss. Université de M'sila, 2018.
- [31] RAHNEB Asma, *Etude comparative sur l'application des métaheuristiques à un problème d'ordonnancement d'atelier de type job shop*, Diss, Université de M'sila, 2018.
- [32] LANNAK Hanane, *Adaptation de la méthode coucou pour la résolution du problème job shop*, Diss, Université de M'sila, 2018.
- [33] ZAITER Meriem, *Insertion temps réel d'une production dans un atelier de type job shop*, Diss, Université de M'sila, 2018.

- [34] HEMMAK Allaoua et BOUDERAH Brahim, *Hybrid algorithm for optimization problems applied to single machine scheduling*, *International Journal of Computer Applications* 66.24, Université de M'sila, 2013.
- [35] SELT Omar, *Numerical solution for solving scheduling problem*, *Global Journal of Pure and Applied Mathematics* 12.5, Université de M'sila, 2016.
- [36] KAHLOULA Tarek Yassine, *Constructive Algorithms for Manufacturing Scheduling Problems*, Diss, Université de M'sila, 2017.
- [37] KRAM Zakaria, *Insertion d'une opération dans un problème d'ordonnement a machines parallèles*, Diss, Université de M'sila, 2019.
- [38] BELLACHE Nour El Islam, *Développement et implémentation d'un solveur bio inspiré pour la résolution d'un problème d'ordonnement d'atelier*, Diss, Université de M'sila, 2017.
- [39] GUERNA Abderrahim, *Applicatin s'une approche bio-inspire au problème d'ordnnancement des instructions*, Diss, Université de M'Sila, 2012.
- [40] HEMMAK Allaoua, *Support de cours d'optimisation combinatoire Focus sur les méthodes de résolution approchée*, Université de M'sila, 2017.

Les sites web :

- [A] slideplayer <https://slideplayer.fr/slide/1149714/>
- [B] dspace <http://dspace.univ-msila.dz:8080/xmlui/handle/123456789/20561>

## ملخص

تتمحور هذه المذكرة حول حل مشكلة جدولة العمل مع قيود النقل. في هذا النوع من المشاكل لا توجد طرق دقيقة تقوم بإعطاء حلول دقيقة. هنا تقوم الطرق التقريبية بدورها بتولي حل هذه المشكلات من الدرجة الصعبة و اخترنا منها الخوارزمية الجينية لاستخراج كل الحلول الممكنة لجدولة الاعمال و استخراج وقت الأقصى لتنفيذ العمليات .

الكلمات المفتاحية : الجدولة - متجر العمل - قيود النقل - الخوارزمية الجينية -الوقت الأقصى لتنفيذ العمليات

## Résumé

Cette mémoire concerne la résolution du problème d'ordonnancement du travail avec contraintes de transport. Dans ce type de problème il n'y a pas de méthodes exactes qui donnent des solutions exactes. Ici, les méthodes approchées résolvent à leur tour ces problèmes d'un degré NP difficile et nous avons choisi parmi elles l'algorithme génétique pour extraire toutes les solutions possibles et extraire le Makespan.

Mots clés : Ordonnancement – Atelier job shop - Contraintes de transport - Algorithme génétique – Makespan

## Abstract

This dissertation is about solving the scheduling problem job shop with constraints of transport. In this type of problem there are no exact methods that give exact solutions. Here, the approximate methods in turn solve these problems of a difficult degree and we choose from them the genetic algorithm to extract all possible solutions for scheduling problem and extract the maximum time to execute the operations.

Keywords: Scheduling – Job shop - Transport constraints - Genetic algorithm – Makespan

---