

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderahman Mira  
Faculté des Sciences Exactes  
Département de Recherche Opérationnelle

Mémoire Présenté

Pour L'Obtention Du Diplôme De Master

En Recherche Opérationnelle

Option : Modélisation mathématique et techniques de Décision (MMTD)

Par : BENNAI LOUANAS Et : MERAD MOUNIR



## Ordonnancement d'un atelier de type job shop cas de l'entreprise FAGECO

Soutenu à l'Université Abderahmane Mira Bejaia , le 02/07/2016, devant le jury composé de :

M <sup>r</sup> AISSANI DJAMIL	Professeur	Encadreur	à l'UAMB - Bejaia
M <sup>r</sup> ASLI LARBI	M.A.A	Co-Encadreur	à l'UAMB - Bejaia
M <sup>r</sup> LAOUAR ABDELHEK	M.A.A	Président	à l'UAMB - Bejaia.
M <sup>r</sup> ZIANI SOFIANE	M.A.A	Examineur	à l' UAMB - Bejaia.
M <sup>elle</sup> GUERBANE RIMA	M.A.A	Examineur	à l' UAMB - Bejaia.

Année Universitaire 2015 – 2016

# Remerciements

Tout acte de recherche n'est que concrétisation d'un travail collectif.

Avant de présenter ce mémoire ; on tient à remercier très sincèrement le professeur AIS-SANI. D pour l'honneur qu'il nous a fait en assurant la direction et le suivi scientifique du présent mémoire. Outre ses conseils avisés. On tient à exprimer notre profonde gratitude aussi à Monsieur ASLI. L pour avoir Co-encadré ce travail. On eu beaucoup de plaisir de pouvoir travailler avec lui sur les problèmes d'ordonnements d'atelier.

On voudrait remercier le directeur de l'entreprise ENMTP (La Filial FAGECO), pour avoir accueilli notre stage pratique au sein de l'entreprise. On remercie aussi Monsieur ALOUACHE. F, d'avoir assuré le suivi pédagogique de ce travail au sein de l'entreprise, et tous ceux qui nous ont accueilli.

On remercie nos Chers parents de nous avoir soutenu et encouragé toute au long de notre travail.

Nos remerciements sont aussi adressés à la secrétaire de notre département.

On remercie tous les collègues pour leurs aides et soutiens et ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail.

Notre reconnaissance va aux membres du jury pour nous avoir fait l'honneur de Juger notre travail.

Nos plus vifs remerciements sont aussi adressés à tous les enseignants qui ont contribué à notre formation. Ainsi que toutes personnes qui nous ont encouragées de prêt ou de loin.

# Dédicases

Je dédie ce modeste travail

♡ A mes chers parents qui ont beaucoup sacrifiés pour faire de moi ce que je suis.

♡ A mes frères et sœurs.

✠ A tous mes Camarades en Recherche Opérationnelle.

♠ A tous mes amis et amies.

♠ A tous ceux qui se sacrifient pour le savoir et la science.

*Louanas*

\*\*\*\*\*

Je dédie ce modeste travail

♡ A mes chers parents qui ont beaucoup sacrifiés pour faire de moi ce que je suis.

♡ A mes frères et sœurs.

✠ A tous mes Camarades en Recherche Opérationnelle.

♠ A tous mes amis et amies.

♠ A tous ceux qui se sacrifient pour le savoir et la science.

*Mounir*

# Table des matières

<i>Introduction générale</i>	<i>1</i>
<b>1 Présentation de l'entreprise</b>	<b>3</b>
Introduction . . . . .	3
1.1 Historique . . . . .	3
1.1.1 Ses principales missions . . . . .	3
1.1.2 Produits sous-traitance . . . . .	4
1.2 Implantation . . . . .	4
1.3 Forme Juridique . . . . .	4
1.4 Les Clients . . . . .	4
1.5 Principaux fournisseurs . . . . .	5
1.6 Principales matières . . . . .	5
1.7 Organisme d'accueil . . . . .	5
1.8 Présentation du département planning . . . . .	8
1.8.1 Ordonnancement lancement . . . . .	8
Conclusion . . . . .	9
<b>2 Problématique et modélisation</b>	<b>10</b>
Introduction . . . . .	10
2.1 Processus de production . . . . .	10
2.1.1 Description de processus de production . . . . .	10
2.2 Problématique . . . . .	19
2.3 Modélisation . . . . .	19
2.3.1 Description des problèmes d'ordonnancement . . . . .	19
2.3.2 Les classes d'ordonnancement . . . . .	21
2.3.3 Ordonnancement d'ateliers . . . . .	22
2.3.4 Organisations d'ateliers . . . . .	23
2.3.5 Modèle théorique . . . . .	27
Conclusion . . . . .	32
<b>3 Méthodes de résolution des problèmes d'ordonnancement</b>	<b>33</b>
Introduction . . . . .	33
3.1 Représentation des problèmes d'ordonnancement . . . . .	33

3.1.1	Graphe conjonctif-disjonctif . . . . .	33
3.2	Complexité des problèmes d'ordonnancement . . . . .	37
3.3	Méthodes de résolution . . . . .	38
3.3.1	Les méthodes exactes . . . . .	38
3.3.2	Les méthodes approchées . . . . .	39
	Conclusion . . . . .	42
<b>4</b>	<b>Choix de la méthode et implémentation</b>	<b>43</b>
	Introduction . . . . .	43
4.1	Le choix de la méthode . . . . .	43
4.1.1	Les ingrédients de la méthode . . . . .	44
4.2	Implémentation . . . . .	48
4.2.1	Langage de Programmation . . . . .	50
4.2.2	Résultats de la méthode tabou . . . . .	51
4.2.3	Discussion des résultats . . . . .	56
	Conclusion . . . . .	56
	<b>Conclusion générale</b>	<b>57</b>
	<b>Bibliographie.</b>	<b>60</b>

# Liste des tableaux

3.1	Exemple d'instance (J1).	35
3.2	Solution S1 d'instance (J1).	36
3.3	Ordre S1 d'instance (J1).	37
4.1	Tableau des Machines	48
4.2	Tableau des tâches	49
4.3	Temps d'exécution des tâches	51
4.4	La solution initiale	52
4.5	Coût de chaque itération	52
4.6	Coût de chaque itération	53
4.7	La meilleure solution avec le Makespan	54

# Table des figures

1.1	L'organigramme de l'entreprise FAGECO. . . . .	6
1.2	L'organigramme du département planning. . . . .	7
1.3	Le processus de réalisation du dossier de fabrication . . . . .	9
2.1	Schéma du processus de production au sein de FAGECO . . . . .	11
2.2	Parc acier. . . . .	12
2.3	Grenaillage; . . . . .	12
2.4	Sciage; . . . . .	13
2.5	Cintrage; . . . . .	13
2.6	Pliage; . . . . .	14
2.7	Tournage; . . . . .	14
2.8	Fraisage; . . . . .	15
2.9	Perçage; . . . . .	15
2.10	Assemblage; . . . . .	16
2.11	Peinture; . . . . .	17
2.12	Montage; . . . . .	17
2.13	Contrôle; . . . . .	18
2.14	Stockage; . . . . .	18
2.15	Caractéristiques d'une tâche $i$ . . . . .	19
2.16	Typologie par les ressources des problèmes d'ordonnancement . . . . .	20
2.17	Classes des ordonnancements . . . . .	22
2.18	La fonction Ordonnancement dans l'atelier. . . . .	23
2.19	Exemple Job-Shop Simple et hybride. . . . .	24
2.20	Exemple Flow-Shop Simple et hybride. . . . .	25
2.21	Schéma général d'une organisation multi-étages parallèles. . . . .	26
2.22	Schéma montrant les liens entre diverses organisations d'ateliers. . . . .	27
2.23	Les Relations entre les critères d'optimisation . . . . .	30
3.1	Sélection non valide de trois opérations . . . . .	34
3.2	Graphe disjonctif du problème J1 . . . . .	36
3.3	Graphe conjonctif de la solution S1 . . . . .	36
3.4	Graphe conjonctif simplifié de la solution S1 . . . . .	37
4.1	Les variables manipulés par l'algorithme tabou . . . . .	44

4.2	Graphe conjonctif évalué avec le sous graphe critique (en pointillés). . . . .	46
4.3	Voisinage de Nowicki et Smutnicki . . . . .	47
4.4	Les temps d'exécutions des chaque machine . . . . .	50
4.5	Évaluation de la solution courante . . . . .	55

# Introduction générale

La recherche opérationnelle est une discipline scientifique qui s'intéresse à de nombreuses problématiques réelles, elle intervient dans plusieurs domaines de l'industrie, d'administration, . . . .

L'optimisation de nos jours est une discipline très convoitée dans le domaine industriel. Elle permet la fabrication des différents produits avec un coût le plus petit possible. Pour cela, on se doit d'avoir un bon système de conduite, ce système peut être considéré comme bon, s'il s'appuie sur une bonne modélisation permettant une décision optimale qui fait progresser la production.

L'ordonnancement des ateliers constitue une difficulté, majeure dans le pilotage des systèmes à l'intérieur des entreprises. Il existe plusieurs types d'ordonnancement d'ateliers, on trouve : Flow Shop, Job Shop, Open Shop, Ateliers Parallèle, . . . , etc.

Ces problèmes sont très étudiés dans la littérature. Il existe plusieurs travaux et développements pour chacune de ces variantes. La nature NP-difficiles de ces problèmes rend la résolution très difficile voire impossible dans certain cas.

L'aspect combinatoire est la principale difficulté de ce type de problème. Cependant l'obtention d'une solution optimale d'une manière exacte est presque impossible lorsque la taille du problème est importante. Cela nous incite à solliciter les méthodes approchées pour en trouver une bonne solution en un temps raisonnable.

La deuxième moitié du 20<sup>ème</sup> siècle a été terrain de plusieurs développements des méthodes approchées pour le problème d'optimisation. Il s'agit en particulier des métaheuristiques qui sont des schémas de méthodes inspirées à partir des phénomènes naturels et est adaptable à tout type de problème.

Parmi ces dernières approches, la Recherche Tabou est l'une des techniques intensivement utilisée dans la pratique. Elle constitue une méthode qui commence par générer une solution de départ puis la faire évoluer en se basant sur une recherche local, des voisins, et pour ne pas cycloter autour d'un optimum local, elle est dotée d'une mémoire qui sert à interdire au processus de résolution de revenir à une solution visitée auparavant.

L'entreprise ENMTP (filiale FAGECO) pour la fabrication d'engins est l'une des entreprises industrielles où on trouve beaucoup de problèmes d'organisation de travail. Durant notre séjour au sein de l'entreprise nous nous sommes intéressés à la problématique qui est d'organiser le passage des pièces sur différentes machines dans l'optique de minimiser le temps total de fabrication, cette problématique relève de l'ordonnancement de type Job Shop.

Ce mémoire est organisé comme suit :

- ✓ Le premier chapitre est consacré à la présentation de l'organisme d'accueil qui est L'ENMTP, où notre stage de fin d'étude a été effectué.
- ✓ Le deuxième chapitre regroupe les différentes notions qui concernent le système de fonctionnement de l'atelier de production et le rôle de l'ordonnancement au sein de ce système. Ainsi que les différents modèles d'organisation d'ateliers et le modèle théorique qui décrit notre problématique.
- ✓ Le troisième chapitre fait l'objet de l'état de l'art sur les différentes méthodes de résolutions des problèmes d'ordonnancement.
- ✓ L'adaptation de la méthode Tabou pour le problème Job Shop ainsi que les expérimentations numériques font le principal sujet du chapitre 4.
- ✓ Le mémoire s'achève par une conclusion générale mettant l'accent sur les résultats obtenus.

# 1

## Présentation de l'entreprise

Pour bien cerner notre sujet d'étude, nous commencerons par présenter une image synthétique de l'entreprise relative à son champ d'application. Nous allons exposer les éléments essentiels pouvant donner une présentation de l'ENMTP. Ces éléments porteront sur l'historique de la société, nature juridique, objet social, mission et domaines d'activités, potentiel de l'entreprise et structure d'accueil.

### 1.1 Historique

FAGECO est une filiale du groupe ENMTP Créé le 11 Avril 2011. Elle est issue de la filialisation de l'unité grues Bejaia (UGB), elle-même créée en 1973 sous la tutelle de l'entreprise SN METAL et qui fut intégrée en 1983 à l'Entreprise nationale des Matériels de Travaux Publics (ENMTP) par décret N° 83 06 du 01/01/1983 dont le siège social est situé Ain Smara, BP 67, Constantine et dont le capital social est de 15 600 000 000,00 DA. FAGECO occupe une superficie de 78 440  $m^2$  dont 32 880  $m^2$  couverte, son effectif actuel est de 196 travailleurs dont :

- ✓ Cadres supérieurs : 08.
- ✓ Cadres : 32.
- ✓ Cadres Maîtrise : 33.
- ✓ Cadres Exécution : 123.

#### 1.1.1 Ses principales missions

- ✓ Fabrication des grues à montage rapide GMR 20/10, GMR 18/05 et GMR 14/04 ;

- ✓ Fabrication des retro-chargeurs ;
- ✓ Fabrication des épanduses à liant ;
- ✓ Fabrication des bennes à béton ;
- ✓ Fabrication des fourches lèves palettes.

### 1.1.2 Produits sous-traitance

- ✓ Des silos à ciment ;
- ✓ Des passerelles pour avions ;
- ✓ Des passerelles pour bateaux ;
- ✓ Des chariots à bagages pour aéroports ;
- ✓ Des palettes pour NAFTAL ;
- ✓ Divers travaux.

## 1.2 Implantation

- ✓ Siège Social : Route des Aurès BP 90, Bejaia ;
- ✓ Tél : 034 21 16 33 à 36 ;
- ✓ Fax : 034 20 19 44 ;
- ✓ E-Mail : enmtp.fageco@yahoo.fr ;

## 1.3 Forme Juridique

- ✓ Forme juridique : EPE/SPA ;
- ✓ Date de création : 11 Avril 2011 ;
- ✓ Capital social : 976 000 000 DA ;
- ✓ N° immatriculation (RCS) : 06/00-0187095B11 ;

## 1.4 Les Clients

- ✓ Groupe COSIDER.
- ✓ Ministère De La Défense Nationale MDN.
- ✓ Ministère de l'Intérieur et des Collectivités Locales MICL.
- ✓ BATIMETAL REALISATION SPA.
- ✓ Société d'études et de réalisation d'ouvrages d'art de l'est SERO EST BATNA EPE/SPA.
- ✓ HYDROTRAITEMENT : études, réalisation et mise en service et l'exploitation des stations de traitement des eaux.
- ✓ Société de Travaux Routiers Oum El Bouaghi SOTRABE.
- ✓ Entreprise de fabrication mécanique filiale ENMTP SOFAME.
- ✓ Entreprise de fabrication de matériels de terrassement filiale ENMTP SOFARE.

- ✓ Entreprise de fabrication de matériels à béton filiale ENMTP SOMABE.
- ✓ Entreprise portuaire de Béjaïa SOGEPOR de Béjaïa.
- ✓ Aéroport ABANE Ramdane Béjaïa.
- ✓ Air Algérie.
- ✓ NAFTAL.
- ✓ EPLA : Entreprise de Préfabrication Légère et d'Aluminium, Spa.

## 1.5 Principaux fournisseurs

Ses principaux fournisseurs de renommée mondiale sont :

- LEROY SOMER France.
- HUCHEZ France.
- INTAGRA Serbie.
- ROLLIX France, LA LEONESSA Italie, INTAGRA Serbie, ROTEISA Espagne.
- HYDROKIT France.
- SAYA ELECTRONIC Algérie, EDIEL EL ACHOUR Algérie.
- J.EISENBARTH Allemagne, TE.RAVASI Italie.
- CBR FREMICOUR France.
- SONAMIA France, INTERGOMMA Italie, WICKE France, BLICKLE RADER UND ROLLEN Allemagne, RUDIFLEX Italie.

## 1.6 Principales matières

Ses produits ne sont conçus qu'avec des composants de qualité et sont garantis contre tout défaut de fabrication. Les principales matières utilisées sont :

- Motoréducteurs.
- Réducteurs.
- Moteurs à essence.
- Couronnes.
- Composants hydrauliques.
- Armoires électriques.
- Collecteurs et sécurités fin de course.
- Câbles aciers et élingues.
- Pneumatiques.
- Aciers.

## 1.7 Organisme d'accueil

L'entreprise est organisée selon l'hierarchie illustrée par l'organigramme suivant :

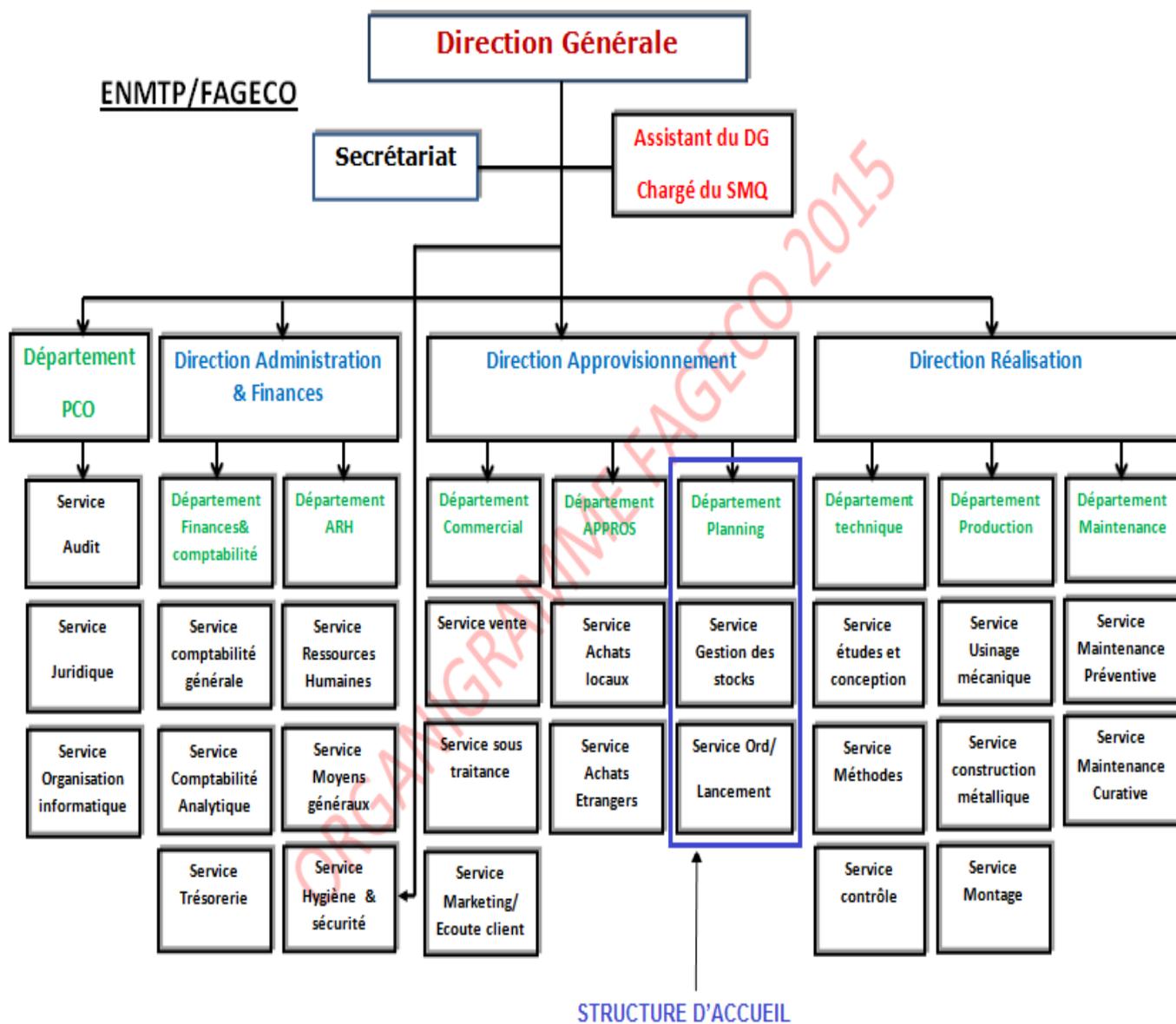


FIGURE 1.1 – L'organigramme de l'entreprise FAGECO.

Notre stage est réalisé au sein du département planning qui est structuré selon l'organigramme ci-dessus

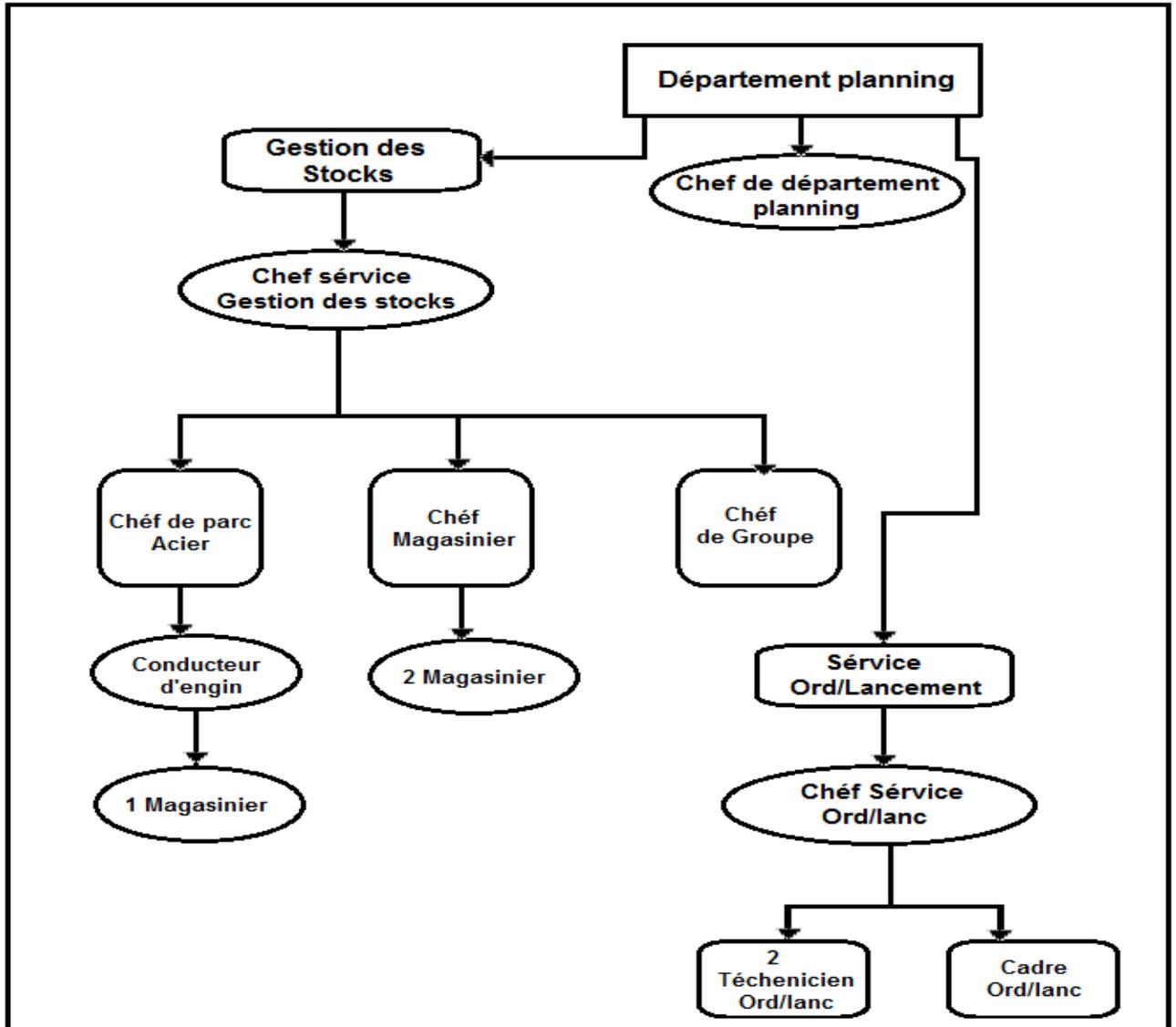


FIGURE 1.2 – L'organigramme du département planning.

## 1.8 Présentation du département planning

### 1.8.1 Ordonnancement lancement

Le service Ordonnancement Lancement est chargé de réaliser le dossier de fabrication et de préparer les ordres de fabrication (OF). On commence par un ordonnancement sur le plan théorique qui est définie comme une technique de régulation et de classement du passage des opérations de fabrication dans l'atelier. La préparation du dossier de fabrication au sein de FAGECO se fait de la manière suivante :

Le dossier lancé par le service ordonnancement sous forme d'un ordre de fabrication (OF) à  $N - 1$  du mois en tenant compte de la mensualisation et la notification du programme. Ce dossier contient les documents suivants :

- **Bon de travail** : Comportant la désignation de l'opération avec le temps alloué de toute la série des pièces à réaliser ainsi la matière première à utiliser.
- **Fiche suiveuse** : Elle comporte toutes les opérations nécessaires ordonnées afin d'assurer l'acheminement du repère envers les postes destinataires.
- **Fiche de fabrication** : Elle comporte tous les repères avec leur temps alloués et leur quantité en matière et pièces entrant dans la fabrication du sous ensemble à réaliser.
- **Bon de sortie matière (B. S. M)** : Document comportant la désignation et la quantité de toutes les matières premières et pièces achetées, à faire servir par le magasinier pour toute la quantité des engins à réaliser.
- **Le dossier technique** : Il est composé du plan technique de sous ensemble et détail de repère.

Pour le suivi de la production au sein de FAGECO, elle utilise les fiches mensuelles de production pour chaque opérateur direct afin de déterminer son temps productif à travers les bon de travaille réalisé et le temps improductif. Les fiches mensuelles doit être remis au service lancement ordonnancement après chaque fin du mois par le département fabrication.

Le schéma suivant résume le processus et les étapes de réalisation du dossier de fabrication et son lancement du département fabrication :

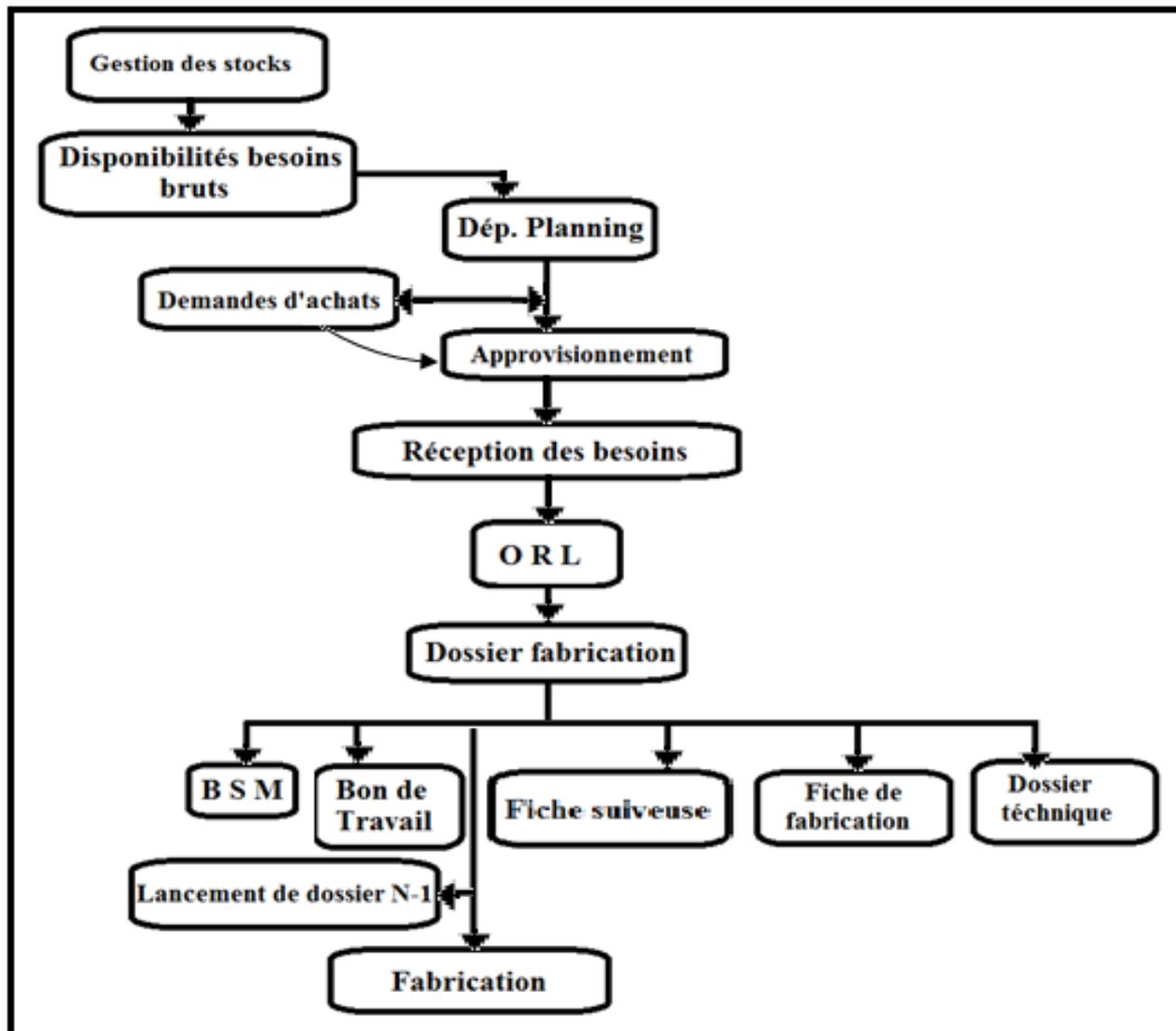


FIGURE 1.3 – Le processus de réalisation du dossier de fabrication

## Conclusion

Tout au long de notre stage, nous sommes arrivés à prendre connaissance aux différentes phases (tâches) de la réalisation des différents produits de FAGECO, qui seront présentées dans les chapitres qui suivent.

# 2

## Problématique et modélisation

### *Introduction*

La gestion de production est l'une des fonctions clés de l'industrie moderne. Bien que d'autres fonctions peuvent être optimisées. Le succès de l'entreprise est fortement lié à la performance des décisions prises au niveau de la production. Plus que jamais, les clients deviennent plus exigeants et leurs besoins changent plus rapidement. Le marché international ou des entreprises sont livrées à une concurrence acharnée, par conséquent, la capacité de produire le bon article au bon moment et au bon prix. La planification et l'ordonnancement sont parmi les fonctions susceptibles de satisfaire un tel objectif. [7]

### **2.1 Processus de production**

Un processus de production ou procédé industriel est un système organisé d'activités qui sont en rapport de façon dynamique et qui sont tournées vers la transformation de certains éléments [17]. Pour ainsi dire, les éléments entrants (les facteurs) deviennent des éléments de sortie (les produits).

#### **2.1.1 Description de processus de production**

Le processus de production chez FAGECO passe par plusieurs étapes qui sont présentées sur la figure FIG. 2.1 et qui serait détaillée par suite.

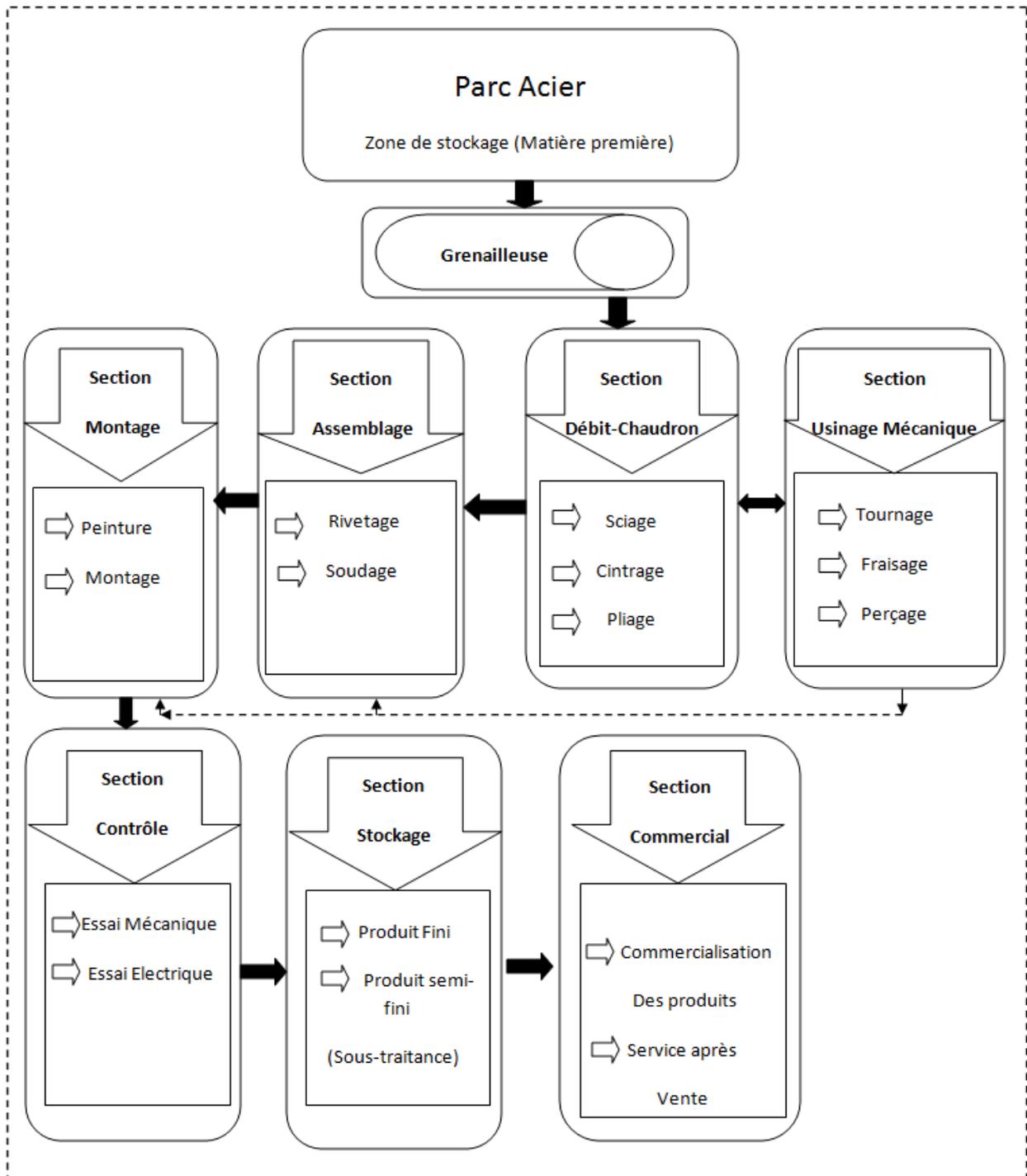


FIGURE 2.1 – Schéma du processus de production au sein de FAGECO

- **Parc Acier** : La zone de stockage de matière première (tube, tôle, barre, carré ...), aussi appelé entreposage est la première phase qui déclenche le processus et il répond à des règles afin de :
  - ✓ Garantir le maintien de la matière première en bon état ;
  - ✓ Optimiser l'espace.



FIGURE 2.2 – Parc acier.

- **Grenailage** : Par une machine grenailleuse qui projette de la grenaille sous forme de micro-billes pour décaper une surface de la matière qui la recouvre.



FIGURE 2.3 – Grenailage ;

- **Débitage et chaudronnerie** : Conduit de façon autonome, l'ensemble des opérations de transformation sous forme de tôles (feuilles, plaques) et de barres (profilés, tubes). À partir des plans, (il/elle) trace, découpe, met en forme, pré-assemble, sous les ateliers suivants :
  - ✓ **Sciage** : Le sciage est un procédé qui permet d'effectuer le débit de profilés (tubes ronds, carrés, plats, ..., etc.), sous l'action d'une lame. La découpe entraînant la séparation de la pièce en deux parties.



FIGURE 2.4 – Sciage ;

- ✓ **Cintrage** : Le cintrage est un procédé mécanique de déformation d'un tube ou d'une barre, suivant un rayon et un angle avec une cintrouse ;



FIGURE 2.5 – Cintrage ;

- ✓ **Pliage** : Le pliage est une opération qui consiste à déformer la matière selon un pli (rectiligne), la machine utilisée (presse plieuse) est dotée d'un poinçon ;



FIGURE 2.6 – Pliage ;

- **Usinage Mécanique** : L'usinage regroupe sous un même terme toutes les étapes de fabrication des pièces mécaniques qui fait appel à de nombreuses techniques et ateliers suivants :
  - ✓ **Tournage** : Le tournage est un procédé d'usinage par enlèvement de copeaux qui consiste à l'obtention de pièces de forme cylindrique ou/et conique à l'aide d'outils coupants sur des machines appelées tours ;



FIGURE 2.7 – Tournage ;

- ✓ **Fraisage** : Le fraisage est une méthode de fabrication où l'enlèvement de matière sous forme de copeaux résulte de la combinaison de deux mouvements ; la rotation de l'outil de coupe, d'une part, et l'avancée de la pièce à usiner d'autre part, l'outil classiquement utilisé est la fraise.



FIGURE 2.8 – Fraisage ;

- ✓ **Perçage** : Le perçage est un procédé de travail qui consiste à exécuter des trous cylindriques à l'aide d'outils appelés forets animés de mouvement de rotation et de translation, ces mouvements sont communiqués par les perceuses et les porte-forets.



FIGURE 2.9 – Perçage ;

- **Assemblage** : L'assemblage est la liaison de différentes pièces d'un ensemble ou produits. Il existe plusieurs techniques d'assemblage qui permettent de fixer les éléments d'un produit les uns aux autres :
  - ✓ **Rivetage** : Le rivetage est un assemblage de pièces à l'aide de rivets. C'est un assemblage définitif, c'est-à-dire non démontable sans destruction de l'attache ;
  - ✓ **Soudage** : Le soudage consiste à faire fondre les pièces à assembler, le métal de base, ainsi qu'un métal d'apport sous forme de fil ou de baguette. Le chauffage peut être assuré par différentes sources d'énergie, les plus courantes étant la flamme (chalumeau) et l'arc électrique (éclair entre une électrode et le métal) ;



FIGURE 2.10 – Assemblage ;

- **Peinture** : La peinture est un revêtement appliqué en une ou plusieurs couches selon différentes méthodes au pinceau, au rouleau et au pistolet, sur les différentes surfaces métalliques.



FIGURE 2.11 – Peinture ;

- **Montage** : L'opérateur de la section montage relie les éléments, pièces, composants d'ensembles mécaniques usinés par les sections précédentes au moyen d'outils et de machines ;



FIGURE 2.12 – Montage ;

- **Contrôle** : Le contrôle est une opération destinée à déterminer avec des moyens appropriés, si le produit y compris, services, documents, code source contrôlé est conforme ou non à ses spécifications ou exigences préétablies et incluant une décision d'acceptation, de rejet ou de retouche, et cela sur des essais :
  - ✓ **Essai Mécanique** : Les essais mécaniques évaluent le comportement d'un équipement dans les conditions réelles d'utilisation en lui appliquant des contraintes mécaniques (chocs, levage, télescopage, translation ...);

- ✓ **Essai Electrique** : Le service essai électrique a pour rôle de réaliser des prestations d'essais fonctionnel électrique (la résistance d'isolement de l'installation électrique, Mesure de la puissance cabine électrique) ;



FIGURE 2.13 – Contrôle ;

- **Stockage** : La section de stockage est importante dans l'entreprise, cette dernière fabrique et stocke ses produits finis et semi fini (sous traitance) avant de les transmettre à ces unités commerciales, sous les conditions de stockage qui peuvent influencer sur la qualité des produits finis et entreposés qui coûte cher ;



FIGURE 2.14 – Stockage ;

- **Commercial** : Cette section chargée de l'exécution des opérations de vente et de la promotion des produits fabriqués, elle est subdivisée en deux service : ventes et après ventes ;

## 2.2 Problématique

L'environnement actuel de productions des engins est caractérisé par des marchés soumis à une forte concurrence. Les exigences et les attentes des clients deviennent de plus en plus croissantes en termes de qualité, de coût et de délais de mise à disposition, pour mieux se placer dans un tel marché surgit par cette forte concurrence l'entreprise FAGECO doit opérer certaine modification sur son système de gestion de production. Un développement d'une nouvelle dimension organisationnelle des différentes ateliers de production au sein de la filiale FAGECO qui est mise en place se repose sur l'ordonnancement optimal de ces ateliers, dont l'objectif est de diminuer la durée du cycle de fabrication.

## 2.3 Modélisation

La modélisation consiste à représenter de la façon la plus fidèle que possible un problème (ou une situation) tel qu'il existe dans la réalité afin de le traiter. C'est une étape cruciale lors de l'étude d'un problème de recherche opérationnelle car une modélisation erronée fausserait toute la résolution du problème.

Il existe plusieurs types de modèles, selon le problème auquel on s'intéresse : les modèles mathématiques, les modèles aléatoires, les représentations graphiques et les modèles visuels.

Pour construire un modèle, on doit passer par plusieurs étapes [19] :

- Identifier une situation problématique de gestion ;
- Formuler la liaison problématique et identifier les objectifs en vu d'une représentation quantitative ;
- Construire le modèle ;

### 2.3.1 Description des problèmes d'ordonnancement

L'ordonnancement consiste à organiser dans le temps la réalisation d'un ensemble de tâches (Jobs), compte tenu de contraintes temporelles (délais, contraintes d'enchaînement, etc.) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises par les tâches. Un ordonnancement décrit l'ordre d'exécution des tâches et l'allocation des ressources au cours du temps, afin de satisfaire un ou plusieurs critères d'optimisation [22].

La description de ce problème se fait par la détermination de ces quatre éléments dits "de base".

1. **Les tâches** : Une tâche ou opération qu'on note " $i$ " est une entité élémentaire de travail localisée dans le temps par une date de début  $t_i$  et une date de fin  $c_i$ , dont la réalisation nécessite une durée  $p_i$  tel que  $p_i = c_i - t_i$  [21].

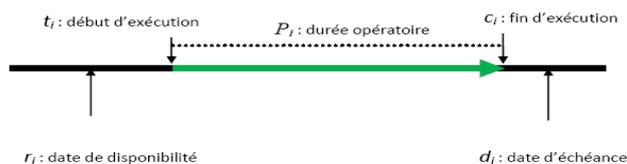


FIGURE 2.15 – Caractéristiques d'une tâche  $i$

2. **Les ressources** : Une ressource  $k$  est un moyen technique ou humain requis pour la réalisation d'une tâche et disponible en quantité limitée [15], de capacité  $A_k$  (supposée constante).

On distingue plusieurs types de ressources :

- **Ressource renouvelable** : Une ressource renouvelable si après avoir être utilisée par une ou plusieurs tâches, elle est à nouveau disponible en même quantité (la main d'œuvre, les machines, l'espace, l'équipement en général) ; les quantités des ressources utilisables à chaque instant sont limitées.
- **Ressource consommable** : Une ressource est dite consommable si elle n'est plus disponible en même quantité après avoir été utilisée par une ou plusieurs tâches, (matière première, budget,..etc.); la consommation globale (ou cumul) au cours du temps est limitée.
- **Ressource doublement contrainte** : Une ressource est doublement contrainte lorsque son utilisation instantanée et sa consommation globale sont toutes les deux limitées (source d'énergie, financement).
- **Ressource disjonctive** : Une ressources disjonctives est dite disjonctive (ou non partageable), si elle ne peut exécuter qu'une tâche à la fois (machine-outil, robot manipulateur).
- **Ressource cumulative** : Une ressources est dite cumulative (ou partageable), si elle peut être utilisée par plusieurs tâches simultanément (équipe d'ouvriers, poste de travail).

La nature de ressources prises en considération permet de dresser une typologie des problèmes d'ordonnement [14] Figure FIG.2.16.

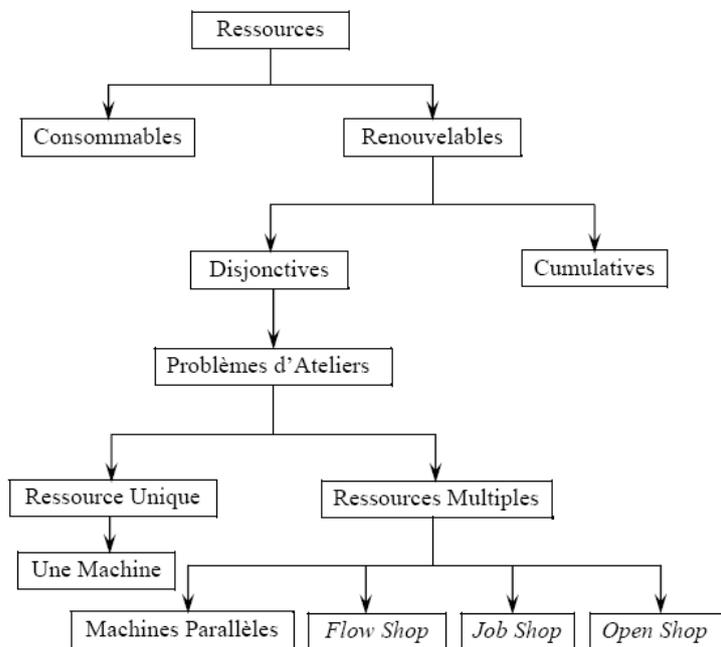


FIGURE 2.16 – Typologie par les ressources des problèmes d'ordonnement

3. **Les contraintes** : Une contrainte exprime des restrictions sur les valeurs que peuvent prendre conjointement les variables représentant les relations reliant les tâches et les ressources.
- Il y a plusieurs types de contraintes [15], parmi celle-ci :
- **Contraintes de ressource** : Une contrainte de ressources représente le fait que les activités utilisent une certaine quantité de ressources. Ces contraintes sont essentiellement soit :
    - Des contraintes d'utilisation des ressources qui expriment la nature, la quantité et les caractéristiques d'utilisation de ces ressources.
    - Des contraintes de disponibilité des ressources qui déterminent les quantités des ressources disponibles au cours du temps.
  - **Contraintes de localisation temporelle** : Sont issues des impératifs de gestion, et relatives aux dates limites des tâches ou du projet entier. On a surtout :
    - La date de disponibilité  $r_i$  : avant laquelle la tâche ne peut pas commencer.
    - La date d'échéance  $d_i$  : avant laquelle la tâche doit être achevée.
  - **Contraintes d'enchaînement** : Une contrainte qui lie le début ou la fin de deux activités par une relation linéaire, ce sont des contraintes imposées généralement par la cohérence technologique (les gammes opératoires dans le cas d'ateliers) qui décrivent des positionnements relatifs.
4. **Les objectifs** : Tout ordonnancement est guidé par un ou plusieurs objectifs qu'il doit chercher leur optimisation. Les objectifs que doit satisfaire un ordonnancement sont variés [21].
- Les objectifs liés au temps : la minimisation du temps total d'exécution, du temps moyen d'achèvement, des durées totales de réglage ou des retards par rapport au dates de livraison, le temps total de flux.
  - Les objectifs liés aux ressources : minimiser la charge d'une ressource ou minimiser le nombre de ressources nécessaires pour réaliser un ensemble de tâches, maximiser le taux de production.
  - Les objectifs liés au coût : minimiser les coûts, de lancement, de production, de stockage (produits finis et semi-finis), de transport, etc.
  - Les objectifs liés à l'énergie ou au débit.

### 2.3.2 Les classes d'ordonnancement

La classification des problèmes d'ordonnancement peut s'établir selon plusieurs critères ; le changement d'une opération, le nombre de machines et leur ordre d'utilisation pour fabriquer un produit et en fonction de l'enchaînement des opérations, on a trois catégories d'ordonnancement [11] :

- **Ordonnancement semi-actif** : Il est impossible d'avancer une opération sans modifier la séquence des opérations sur la ressource, chaque opération est calée, soit sur l'opération qui la précède dans sa gamme, soit sur l'opération qui la précède sur la machine utilisée.
- **Ordonnancement actif** : Il est impossible d'avancer une opération sans reporter le début d'une autre opération.
- **Ordonnancement sans délai ou sans retard** : Aucune opération n'est mise en attente, alors qu'une machine est disponible pour l'exécuter.

Le diagramme d'inclusion des classes d'ordonnancements est présenté sur la figure FIG. 2.17.

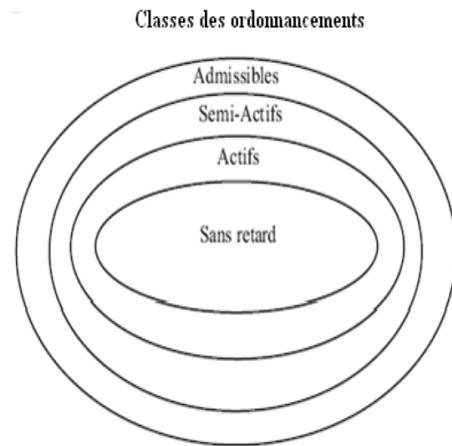


FIGURE 2.17 – Classes des ordonnancements

### 2.3.3 Ordonnement d'ateliers

L'ordonnement d'ateliers couvre un ensemble d'actions qui transforment les décisions de fabrication définies par le programme directeur de production en instructions d'exécution détaillées destinées à piloter et contrôler à court terme l'activité des postes de travail dans l'atelier, la figure ci-dessous représente la fonction Ordonnement dans l'atelier.

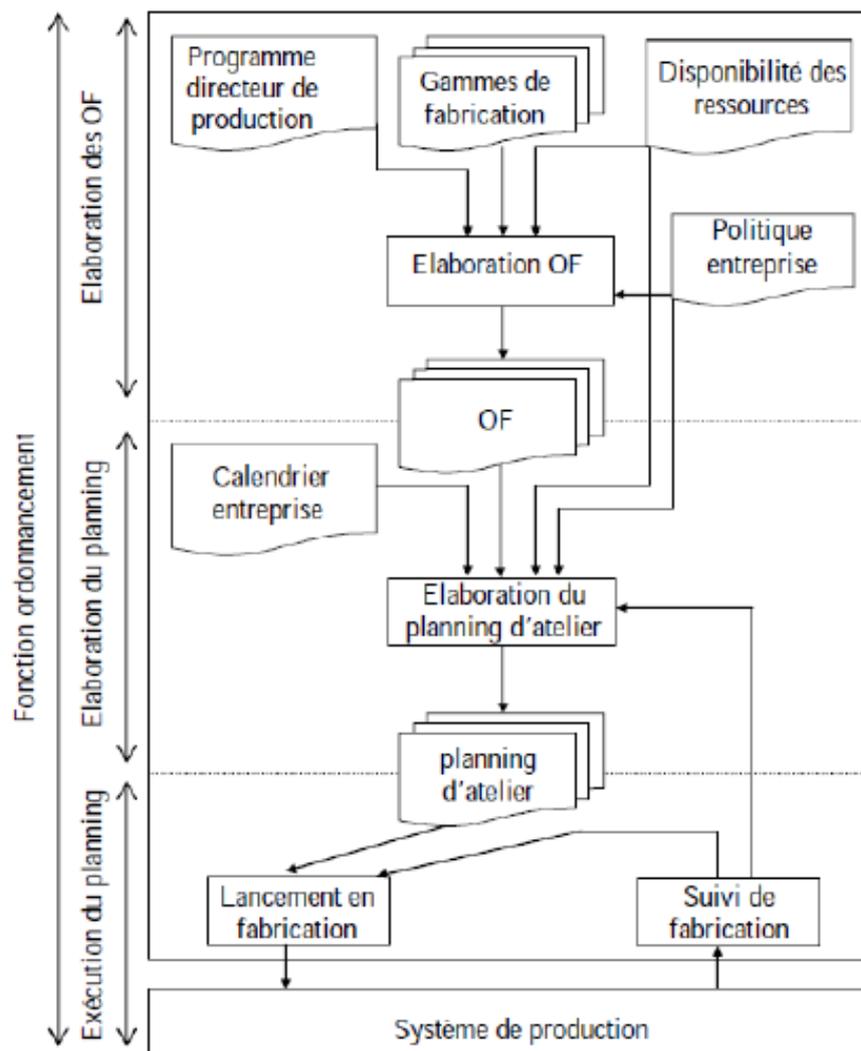


FIGURE 2.18 – La fonction Ordonnancement dans l'atelier.

### 2.3.4 Organisations d'ateliers

Un atelier est formé de machines reliées par des moyens de transport Il est destinée à fabriquer des pièces (dont le modèle théorique est appelé job)[10], elles mêmes se décomposant en diverses tâches, on distingue trois types d'organisations à savoir :

- Machine unique ;
- Machines multiples ;
- Multi-étages parallèles.

1. **Machine unique** : Dans ce cas, l'ensemble des tâches à réaliser est fait par une seule machine. Les tâches sont alors composées d'une seule opération qui nécessite la même machine, l'une des situations intéressantes où on peut rencontrer ce genre de configurations

est le cas où on est devant un système de production comprenant une machine goulot qui influence l'ensemble du processus.

2. **Machine multiples** : Ces organisations se caractérisent par l'existence de plusieurs machines, sur lesquelles seront exécutées plusieurs tâches. Ce sont des organisations énormément répandues dans le domaine de la production industrielle, constituant ainsi les différentes catégories d'ateliers. Trois types ont une importance particulière dans les problèmes d'ordonnancement : le flow-shop, le job-shop et l'open-shop.

**a) Atelier à cheminement multiple : Job-Shop**

Les tâches ne s'exécutent pas sur toutes les machines et dans le même ordre. Chaque tâche emprunte une gamme qui lui est propre. Il existe des contraintes de précédence entre les opérations d'une même tâche, et le nombre d'opérations est plus grand que le nombre de machines.

- Job-Shop Simple : Si pour chaque machine, on dispose d'un et un seul étage, on dit que l'organisation est un job shop simple.
- Job-Shop Hybride : Si pour au moins des machines, on dispose de plus d'un étages, on appelle job-shop hybride.

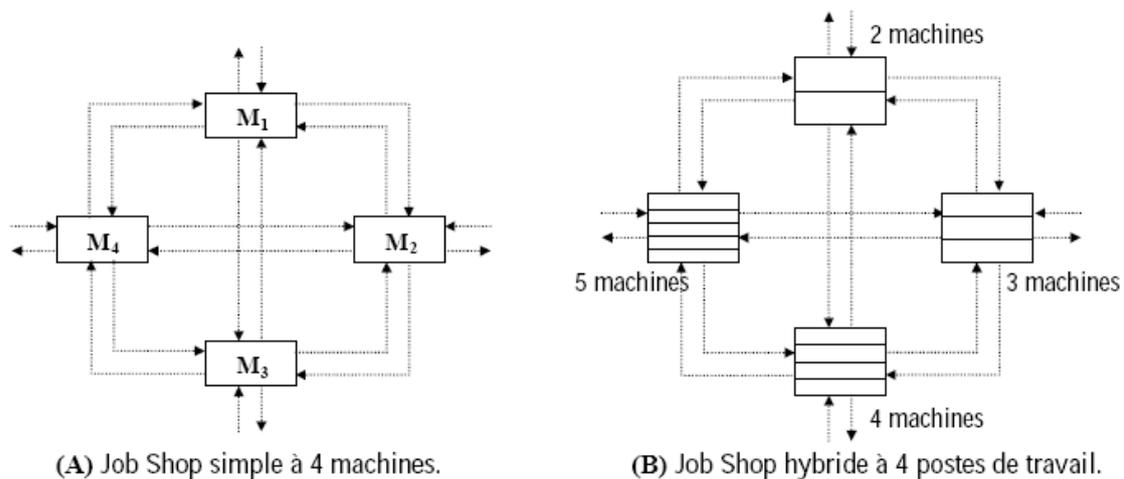
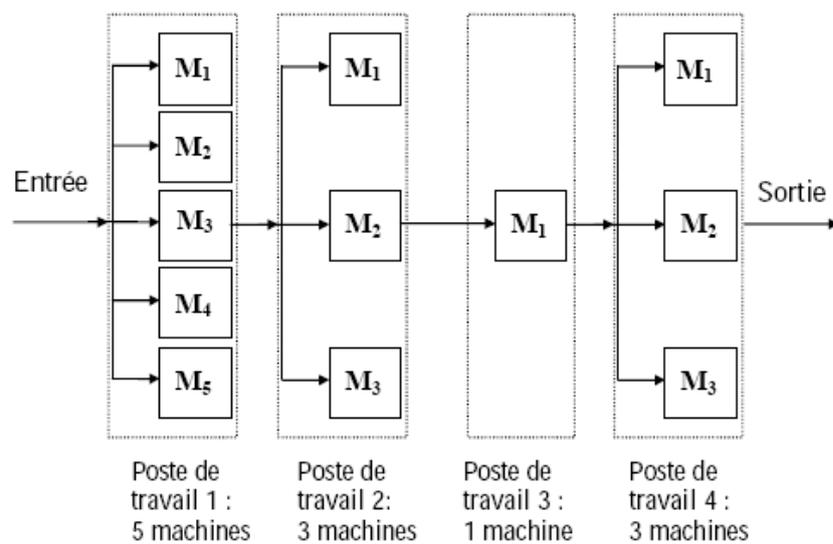
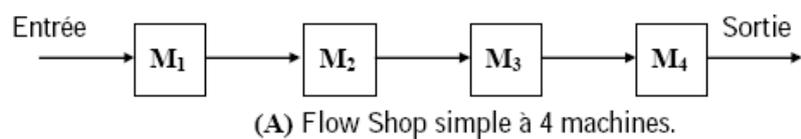


FIGURE 2.19 – Exemple Job-Shop Simple et hybride.

## b) Atelier à cheminement unique : Flow-Shop

- Flow-Shop Simple : Une tâche est constituée d'opérations passant par différentes machines de manière linéaire, le nombre d'opérations de chaque tâche est égal au nombre de machines. Il y a des contraintes de précédence de type chaîne entre les opérations d'une tâche : L'opération  $(i; j + 1)$  ne peut commencer avant la fin de l'opération  $(i; j)$ .
- Flow-Shop Hybride : Le flow-shop hybride est donc une généralisation du flow-shop simple au cas où plusieurs machines sont disponibles sur un ou plusieurs étages pour exécuter les différentes tâches du flow-shop.



(B) Flow Shop hybride à 4 postes de travail.

FIGURE 2.20 – Exemple Flow-Shop Simple et hybride.

**C) Atelier à cheminement libre : open shop**

Dans les organisations de type open-shop, les contraintes de précédence sont relâchées, les opérations nécessaires à la réalisation de chaque tâche peuvent être effectuées dans n'importe quel ordre (les gammes sont libres). Ce cas se présente lorsque chaque produit à fabriquer doit subir une séquence d'opérations, mais dans un ordre totalement libre. Ce type est utilisé rarement dans les entreprises.

3. **Machine multi-étages parallèles** : Les machines sont regroupées en étages distincts. Toutes les machines du même étage sont capables d'effectuer la même opération.

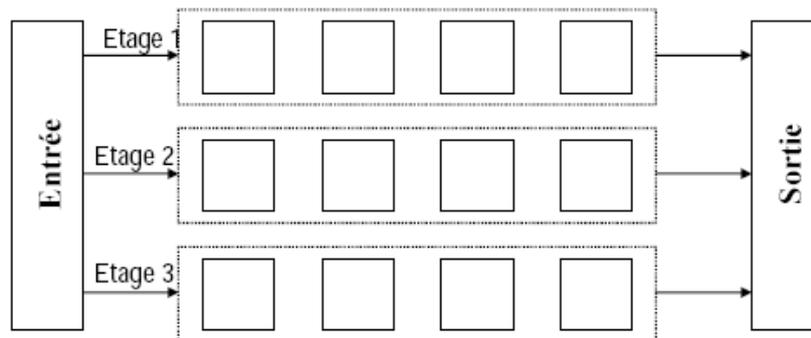


FIGURE 2.21 – Schéma général d'une organisation multi-étages parallèles.

#### 4. Relations entre organisations

La figure ci-dessous (Figure FIG.2.22) illustre schématiquement les différences et les ressemblances qui existent entre les différentes organisations de systèmes de production.

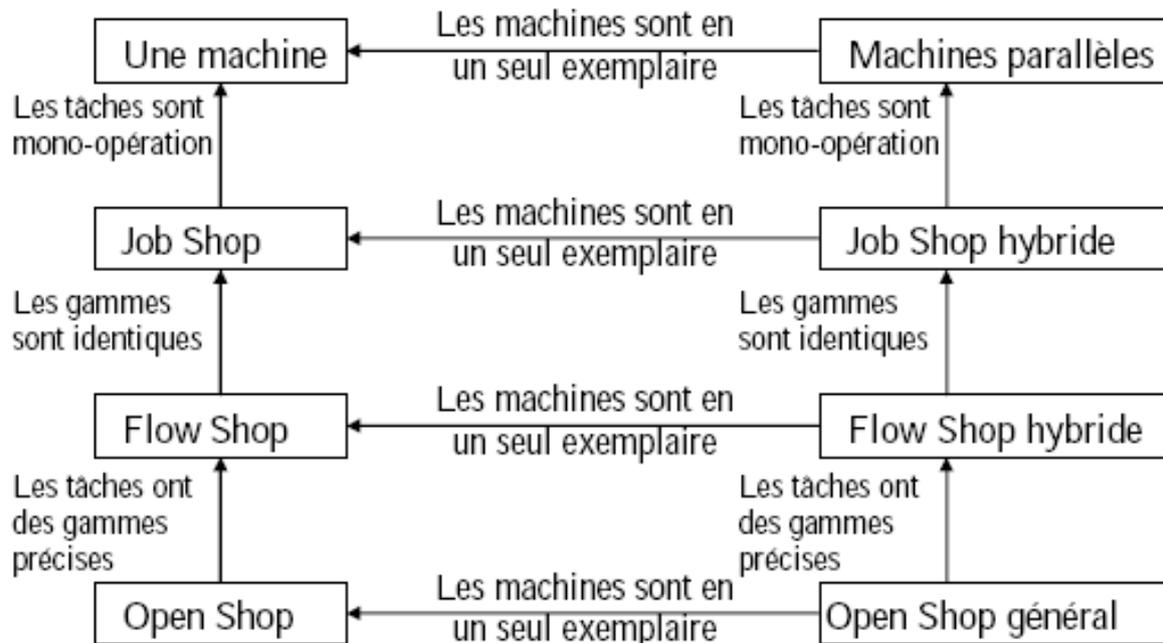


FIGURE 2.22 – Schéma montrant les liens entre diverses organisations d'ateliers.

### 2.3.5 Modèle théorique

#### 1. Les critères d'optimisation

Dans la résolution d'un problème d'ordonnancement, on peut choisir entre deux grands types d'objectifs, visant respectivement à l'optimalité des solutions (trouver la meilleure solution) ou plus simplement à leur admissibilité (trouver une solution).

La recherche de l'optimalité impose à mesurer la qualité d'une solution de répondre aux exigences du problème étudié. Dans ce but, des critères d'optimisation sont définis et utilisés [8].

#### 2. Classification des critères d'optimisation

Le critère peut être vu comme un indicateur de la performance de l'activité de production et peut varier selon la nature du système. Les critères les plus étudiés sont les suivants :

- Les critères liés aux dates de fin et de livraison ;
- Les critères liés aux volumes des encours ;
- Les critères liés à l'utilisation des ressources.

Avant de dénombrer les critères importants de chacune de ces catégories, rappelons-nous les notations suivantes :

- $c_i$  (completion time) : La date d'achèvement de la dernière opération de la tâche  $i$  ;
- $d_i$  (due date) : La date de livraison de  $i$  ;
- $r_i$  (release time) : La date de disponibilité de  $i$  ;
- $p_{ij}$  : La durée de l'opération  $j$  de la tâche  $i$  ;
- $F_i = c_i - r_i$  (low time) : La durée de séjour de la tâche  $i$  depuis sa disponibilité jusqu'à la fin de son exécution ;
- $L_i = c_i - d_i$  (lateness) : Le retard algébrique, qui détermine le retard d'exécution de la tâche  $i$  par rapport à sa date d'échéance ;
- $T_i = \max(L_i, 0)$  (tardiness) : Le retard absolu de la tâche  $i$  ;
- $E_i = \max(-L_i, 0)$  (earliness) : L'avancement de la tâche  $i$ .

### a- Les critères liés aux dates de fin et de livraison

Ce sont des critères liés aux paramètres temporels caractérisant les tâches [20].

- $C_{max} = \max(c_i)$  : (Makespan), est la plus grande date d'achèvement des différentes tâches. C'est le temps nécessaire pour terminer toutes les tâches. Le makespan, est le critère le plus fréquemment utilisé.
- $C_{\Sigma} = \sum(c_i)$  : La somme des dates de fin des tâches.
- $\sum w_i C_i$  : La somme des dates de fin pondérées.
- $\bar{c}$  : La moyenne arithmétique des dates de fin des tâches.
- $F_{max} = \max(F_i)$  : Le maximum des temps de séjour de l'ensemble des tâches.
- $L_{max} = \max(L_i)$  : Le maximum des retards algébriques de l'ensemble des tâches.
- $T_{max} = \max(T_i)$  : Le maximum des retards absolus de l'ensemble des tâches.
- $E_{max} = \max(E_i)$  : Le maximum des temps d'avancement de l'ensemble des tâches.

### b- Les critères liés aux volumes des encours

Soit  $t_0$  la date de début de l'opération  $O$  et  $p_0$  sa durée d'exécution. À chaque instant  $t$ , on peut calculer les mesures suivantes caractérisant les volumes des encours :

- Le nombre de tâches en cours d'exécution :

$$N_p(t) = \sum_{i=1}^n e_i(t); \text{ avec } e_i(t) = \begin{cases} 1 & \text{si } (t_0 \leq t < t_0 + p_0) \\ 0 & \text{sinon.} \end{cases}$$

L'optimisation de l'ordonnancement vise la maximisation de ce critère. Plus la valeur de ce critère est grande, plus le nombre de machines en attente est faible.

- Le nombre de tâches en attente d'exécution :

$$N_w(t) = \sum_{i=1}^n e_i(t); \text{ avec } e_i(t) = \begin{cases} 1 & \text{si } (t_0 + p_0 \leq t < t_{0+1}) \\ 0 & \text{sinon.} \end{cases}$$

La minimisation de ce critère permet de diminuer le nombre de tâches en attente dans les stocks intermédiaires des machines, et par conséquent de réduire la capacité intrinsèque des stocks.

- Le nombre de tâches terminées sur la dernière machine :

$$N_f(t) = \sum_{i=1}^n e_i(t); \text{ avec } e_i(t) = \begin{cases} 1 & \text{si } (C_i \leq t) \\ 0 & \text{sinon.} \end{cases}$$

La maximisation de ce critère permet de répondre dans les meilleurs délais aux demandes des clients.

### c- Les critères liés à l'utilisation des ressources

La gestion des ressources permet de maximiser la charge d'une ressource ou l'utilisation moyenne des ressources, ou maximiser le taux de production, ou encore minimiser le temps d'inactivité de l'ensemble des ressources, ces objectifs sont :

- L'utilisation moyenne des ressources :

$$\bar{U} = \frac{\sum_{i=1}^n \sum_{j=1}^m p_{ij}}{\sum_{j=1}^m M_j \times C_{max}}$$

Ce critère permet de signaler l'utilisation des ressources de l'atelier, en indiquant les périodes pleines et les périodes creuses de la chaîne de fabrication.

- Le temps d'inactivité de l'atelier (Idle time) :

$$I = \sum_{j=1}^m (C_{max} - \sum_{i=1}^n p_{ij})$$

C'est la somme des tranches de temps perdu pendant l'exécution des tâches sur les machines. La minimisation de cet indicateur permet une meilleure exploitation des ressources.

## Propriétés des critères

Deux propriétés importantes des critères de performance qui sont : la régularité des critères et la notion de dominance décrivent un certain nombre de relations entre un critère donné et l'ordonnancement correspondant.

### Critères réguliers

Soient  $C_i(x)$  et  $\delta(x)$ , respectivement la date de fin de la dernière opération de la tâche  $J_i$  et la valeur d'un critère  $\delta$  appliqué à un ordonnancement  $x$ ;  $\delta$  est fonction des dates de fin d'ordonnancement  $C_i$ .

Considérons deux ordonnancement  $x, \hat{x}$ . Si :  $C_i(x) \leq C_i(\hat{x}) \forall i = 1, 2, \dots, n \Rightarrow \delta(x) \leq \delta(\hat{x})$  alors le critère  $\delta$  est dit régulier.

Autrement dit, un critère est régulier, s'il est une fonction décroissante de dates de fin d'exécution des opérations. Donc, on peut le dégrader en avançant l'exécution d'une tâche.

Parmi les critères réguliers :  $C_{max}, T_{max}, L_{max}, \bar{L}, \bar{T}$ . Par contre,  $\bar{E}, E_{max}$ , il sont des critères

non réguliers.

### Notion de dominance

Un sous ensemble de solutions est dit dominant pour l'optimisation d'un critère donné, s'il contient au moins un optimum pour ce critère.

### Relations entre ces critères

Il existe plusieurs relations d'équivalence et de similitude entre les critères d'optimisation. En effet, une solution optimale pour l'un ou l'autre des problèmes pour un critère donné peut s'avérer optimale pour un autre critère [1].

La figure FIG.2.23 donne les liens entre certains critères, et on lit, au début qu'une solution optimale pour le critère  $L_{max}$  est aussi optimale pour le critère  $C_{max}$  (avec  $d_i=0$ ), autrement dit, une méthode qui optimise  $L_{max}$  optimise  $C_{max}$ , la méthode qui optimise le critère  $\sum w_i C_i$  optimise aussi le critère  $\sum C_i$ .

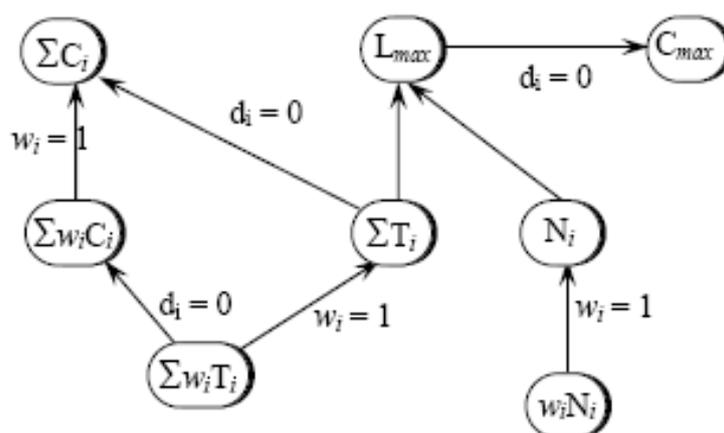


FIGURE 2.23 – Les Relations entre les critères d'optimisation

### Classification des problèmes d'ordonnancement

Les problèmes d'ordonnancement sont décrits par la notation :  $\bullet / \check{S} / \bullet$  [5]

a) Le paramètre  $\bullet$  se compose de deux sous-paramètres  $\bullet_1$  et  $\bullet_2$  :

- $\bullet_1 \in \{\emptyset, P, Q, R, O, F, J\}$  décrit l'organisation de l'atelier :
  - $\bullet_1 = \emptyset$  : Une seule machine.
  - $\bullet_1 = P$  : Machines parallèles identiques. Il s'agit de machines de même cadence, disposées en parallèle pouvant exécuter toutes les tâches.
  - $\bullet_1 = Q$  : Machines parallèles uniformes. Les cadences des machines sont différentes, mais restent indépendantes des tâches.

- <sub>1</sub> =  $R$  : Machines parallèles non liées. Les cadences des machines sont différentes deux à deux et dépendent des tâches exécutées.
  - <sub>1</sub> =  $F$  : Flow-shop.
  - <sub>1</sub> =  $J$  : Job-shop.
  - <sub>1</sub> =  $O$  : Open-shop.
2. •<sub>2</sub> ∈ {∅,  $m$ } : détermine le nombre de machines dans le système.
- <sub>2</sub> = ∅ : Le nombre de machines est variable.
  - <sub>2</sub> =  $m$  : Le nombre de machines est constant et égal à  $m$ .
- b) Le paramètre  $\check{S}$  se compose de cinq sous-paramètres :  $\check{S}_1, \check{S}_2, \check{S}_3, \check{S}_4, \check{S}_5$
1.  $\check{S}_1 \in \{\emptyset, prmp\}$  indique l'autorisation ou non de la préemption  
 $\check{S}_1 = \emptyset$  : La préemption n'est pas autorisée (lorsqu'une opération est commencée, elle doit être terminée avant de pouvoir exécuter une autre opération sur la même machine).  
 $\check{S}_1 = prmp$  : La préemption est autorisée (l'exécution d'une opération peut être interrompue puis reprise sur la même machine ou sur une autre).
  2.  $\check{S}_2$  : Identifie les ressources additionnelles du système.
  3.  $\check{S}_3$  : Décrit les contraintes de précédence.
  4.  $\check{S}_4 \in \{\emptyset, r_i\}$  : Décrit les dates de disponibilité des différentes tâches .
  5.  $\check{S}_5$  : Définit les particularités des durées opératoires s'il y en a.
- c) Le paramètre • représente le critère d'optimisation sur lequel l'ordonnancement est évalué.

## Formulation Mathématique

Dans notre problème il s'agit de trouver un ordre de placer les tâches que vont subir les pièces selon les contraintes de succession qui doivent être respectées afin d'obtenir un temps de fonctionnement minimum.

### Notation :

La notation utilisée dans ce mémoire est résumée dans ce qui suit :

### Indices

$J$  : L'ensemble des tâches ;  
 $M$  : L'ensemble des machines de l'atelier ;  
 $i$  : Indice de la tâche ;  
 $j$  : Indice de l'opération ;  
 $k$  : Indice de machine.

### Paramètres

$n$  : Le nombre de tâches ;  
 $m$  : Le nombre de machines ;  
 $n_i$  : Le nombre d'opérations de la tâche  $i$  ;  
 $O_{i,j}$  : La  $j^{eme}$  opération de la tâche  $i$  ;  
 $t_{i,j,k}$  : La date de début d'exécution de l'opération  $O_{i,j}$  effectuée sur la machine  $k$  ;

$d_{i,j,k}$  : La durée opératoire de l'opération  $O_{i,j}$  effectuée sur la machine  $k$  ;

$C_i$  : La date de fin d'exécution de tâche  $i$  ;

$C_{max}$  : La date de fin d'exécution de la dernière tâche.

### Variable de décision

$$x_{i,j,k} = \begin{cases} 1 & \text{si l'opération } O_{i,j} \text{ est exécutée sur la machine } k ; \\ 0 & \text{sinon.} \end{cases}$$

### Modèle mathématique

$$\left\{ \begin{array}{l} \text{Min}\{C_{max} = \max_i C_i\}; i = 1, 2, \dots, n; \end{array} \right. \quad (1)$$

$$\left\{ \begin{array}{l} t_{i,j,k} x_{i,j,k} + d_{i,j,k} x_{i,j,k} \leq t_{i,j+1,k} x_{i,j+1,k}; i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n_i - 1; k = 1, 2, \dots, m; \end{array} \right. \quad (2)$$

$$\left\{ \begin{array}{l} \sum_{k=1}^m x_{i,j,k} = 1; i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n_i - 1; k = 1, 2, \dots, m \end{array} \right. \quad (3)$$

$$\left\{ \begin{array}{l} t_{i,j,k} - t_{h,l,k} \geq d_{h,l,k} \vee t_{h,l,k} - t_{i,j,k} \geq d_{i,j,k}; i, h = 1, 2, \dots, n; \quad j, l = 1, 2, \dots, n_i; k = 1, 2, \dots, m \end{array} \right. \quad (4)$$

$$\left\{ \begin{array}{l} t_{i,j,k} \geq 0; i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n_i; k = 1, 2, \dots, m \end{array} \right. \quad (5)$$

$$\left\{ \begin{array}{l} x_{i,j,k} \in \{0, 1\}; i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n_i; k = 1, 2, \dots, m \end{array} \right. \quad (6)$$

### Description des équations

Nous pouvons donner la signification de chaque équation comme suit :

(1) : Le critère à minimiser il s'agit de la minimisation du makespan.

(2) : Assure que les opérations appartenant à la même tâche doivent satisfaire les contraintes de précédence.

(3) : Assure que chaque opération est affectée à une seule machine.

(4) : Assure que chaque opération ne doit commencer qu'après l'opération précédente sur la même machine  $k$ .

(5) : Assure que la date de début de chaque opération doit être positive ou nulle.

(6) :  $x_{i,j,k}$  est une variable binaire.

## Conclusion

Nous avons abordé dans ce chapitre l'aspect commun des problèmes d'ordonnancement en production, en exposant notamment le système de fonctionnement de l'atelier de production et le rôle de l'ordonnancement au sein de ce système.

Le deuxième point exposé dans ce chapitre, est la description des problèmes d'ordonnements, les différents modèles d'organisation d'ateliers et le modèle théorique qui décrit notre problématique.

# 3

## Méthodes de résolution des problèmes d'ordonnancement

### Introduction

De manière générale, l'ordonnancement à l'échelle industrielle est un problème de trop grande ampleur pour espérer obtenir la solution optimale dans un temps raisonnable. C'est pour quoi de nombreuses méthodes permettent de rechercher des solutions satisfaisantes en utilisant les méthodes exactes ou les méthodes approchées.

### 3.1 Représentation des problèmes d'ordonnancement

Il existe deux sortes de représentations possibles d'un problème d'ordonnancement : le diagramme de Gantt et le graphe conjonctif-disjonctif.

#### 3.1.1 Graphe conjonctif-disjonctif

Le modèle du graphe conjonctif-disjonctif a été introduit par Roy (1964). Ce modèle utilise un algorithme de plus longs chemins pour déterminer les dates de début des opérations. Ce modèle est utilisé pour représenter les problèmes disjonctifs, c'est-à-dire les problèmes dans lesquels certaines opérations ne peuvent être réalisées en même temps car elles utilisent une ou plusieurs ressources en commun. Dans ce cas, on dit que ces opérations sont "disjointes". On appelle "disjonction" le fait que deux opérations se partagent une même ressource et ne peuvent être traitées que dans des intervalles de temps disjoints. Deux opérations en disjonction doivent être arbitrées, c'est-à-dire que l'ordre de passage sur la ressource commune doit être déterminé. On parle donc de disjonction arbitrée et de disjonction non arbitrée.

Une sélection d'un problème disjonctif consiste à arbitrer toutes ses disjonctions. Si toutes les disjonctions sont arbitrées de manière cohérente, on dit que la sélection est valide (définition ci-dessous) et on obtient un ordre de passage des opérations sur chacune des ressources. Une sélection peut être non valide car elle impose des ordres qui ne sont pas compatibles.

La figure FIG3.1 montre une des permutations de 3 éléments. Or il y a  $2^3 = 8$  sélections possibles (nombre de combinaisons possibles). Il est donc évident que certaines sélections ne sont pas valides, d'ailleurs la sélection de la figure propose un ordre non valide. La sélection de la figure met l'opération 1 avant l'opération 2, et l'opération 2 avant l'opération 3. Il serait judicieux d'arbitrer la disjonction entre 1 et 3 en mettant l'opération 1 avant l'opération 3. Or la sélection propose le contraire.

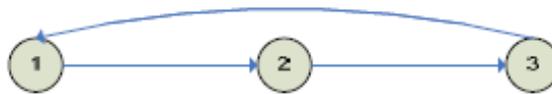


FIGURE 3.1 – Sélection non valide de trois opérations

**Définition :**

Une sélection(ou ordre) valide est une orientation de toutes les disjonctions du problème tel que le graphe des orientations soit acyclique. Une sélection valide permet d'orienter le graphe de manière cohérente. Mais rien n'indique que cette orientation sera associée à une solution. En effet, certaines orientations créent des cycles dans le graphe. Lorsque ces cycles sont de longueur positive, on dit qu'ils sont absorbants (définition ci-dessous). Dans ce cas, il n'y a plus de longs chemins dans le graphe car tout chemin passant par un nœud du cycle absorbant peut être rallongé par autant de tours de cycle que voulu.

**Définition :**

Un cycle (C) est absorbant pour les plus longs chemins si et seulement si : La longueur du cycle C est strictement positive. Il ne faut pas donc confondre les graphes contenant des cycles et ceux qui contiennent des cycles absorbants. A priori ces deux notions sont différentes : on peut calculer les plus longs chemins pour un graphe qui contient un cycle, on ne le peut pas pour un graphe contenant un cycle absorbant. Du point de vue des algorithmes, certains sont spécialisés dans les graphes acycliques (Algorithme de Bellman), d'autres dans les graphes sans cycle absorbant et d'autres capables de détecter les cycles absorbants.

Pour le problème de job shop, tous les arcs sont de longueur strictement positive. Tous les cycles sont donc de longueur strictement positive. Ainsi, pour le job shop, tous les cycles sont absorbants et une solution est associée à chaque graphe conjonctif acyclique. C'est pourquoi les représentations n'envisageant que des sélections valides conduisent systématiquement à des solutions. C'est le cas par exemple de la représentation semi actif. Dans la suite, on détaille la construction des deux graphes : le graphe disjonctif qui modélise le problème et dans lequel a priori aucune disjonction n'est arbitrée et le graphe conjonctif qui modélise une solution,

c'est-à-dire, un ordonnancement semi-actif.

### Graphe disjonctif

On note  $G = (V, A, E)$  un graphe disjonctif, où  $V$  est un ensemble de nœuds,  $A$  est un ensemble d'arcs et  $E$  est un ensemble d'arêtes. Les ensembles  $V$ ,  $A$  et  $E$  sont construits de la manière suivante :

- Pour chaque opération du problème de job shop, on crée un nœud dans  $V$ . Pour faciliter les explications, on pourra donc confondre les nœuds et les opérations. On crée de plus deux nœuds 0 et \* modélisent deux opérations fictives. Le nœud 0 représente une opération fictive réalisée avant toutes les autres opérations, on dit que ce nœud est la source du graphe. Le nœud \* représente une opération fictive réalisée après toutes les autres opérations, on dit que ce nœud est le puits du graphe.
- On crée un arc dans  $A$  entre deux opérations consécutives dans la gamme du job. Ainsi, pour chaque opération de nœud  $i$ , on crée un arc du nœud  $i$  vers le nœud  $i + 1$  où  $i + 1$  est l'opération suivante dans la gamme du job. Cet arc est de longueur  $p_i$  (i.e. la durée de traitement de l'opération  $i$ ).
- On crée une arête dans  $V$  entre deux opérations traitées par la même machine. Cette arête représente la disjonction entre ces deux opérations et devra être orientée pour construire un graphe conjonctif. Il y a donc un ensemble d'arêtes reliant toutes les opérations à réaliser sur la même machine.

**Exemple d'instance (J1) :** L'atelier débit-chaudron est organisé de manière à faire traiter 3 pièces qui constituent la pièce (GABARIT)(p4), en exécutant une ou plusieurs tâches sur chaque pièce primaire. On a 3 pièces primaires : tube (P1), plat (P2), barre (P3) qui passe par 3 machines différentes : Cisaille (M1), Plieuse (M2), Soudage (M3), avec des durées d'exécution différentes. Pour construire la pièce (cabine) (P4).

<b>Job1</b>	O1 = Machine 1, Durée : 4	O2 = Machine 2, Durée : 5	O3 = Machine 3, Durée : 3
<b>Job2</b>	O4 = Machine 2, Durée : 7	O5 = Machine 3, Durée : 3	O6 = Machine 1, Durée : 3
<b>Job3</b>	O7 = Machine 1, Durée : 5	O8 = Machine 3, Durée : 10	O9 = Machine 2, Durée : 8

TABLE 3.1 – Exemple d'instance (J1).

Pour illustrer la construction du graphe disjonctif. La figure FIG3.2 suivante représente le graphe disjonctif associé au problème (J1).

### Graphe conjonctif

À partir d'une représentation semi actif, d'une représentation par ordre total d'opérations, d'une représentation par répétition ou d'une représentation binaire, on peut orienter les arêtes du graphe disjonctif. Orienter une arête entre deux opérations consiste à remplacer cette arête par un arc dans le sens de l'orientation. La longueur de cet arc est la durée de traitement de l'opération de départ de l'arc. Le tableau TAB.3.2 est une sélection valide, correspondant à un ordonnancement S1 de l'instance J1 (Tableau TAB.3.1). Le graphe conjonctif est présenté dans la figure FIG.3.3 suivante.

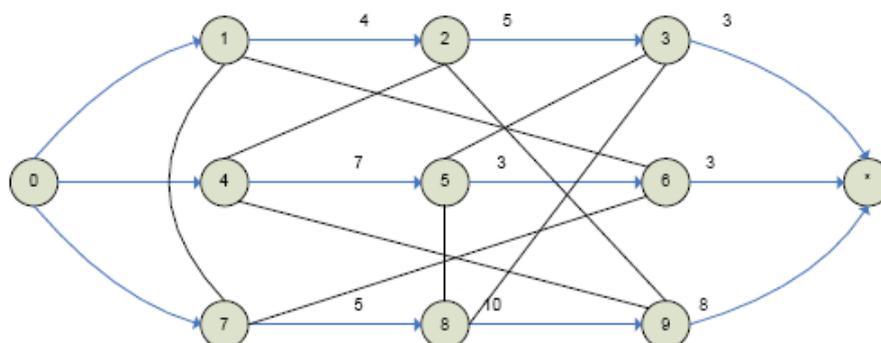


FIGURE 3.2 – Graphe disjonctif du problème J1

<b>M1</b>	$1 < 7$	$7 < 6$	$1 < 6$
<b>M2</b>	$2 < 9$	$4 < 9$	$4 < 2$
<b>M3</b>	$5 < 4$	$5 < 8$	$8 < 3$

TABLE 3.2 – Solution S1 d'instance (J1).

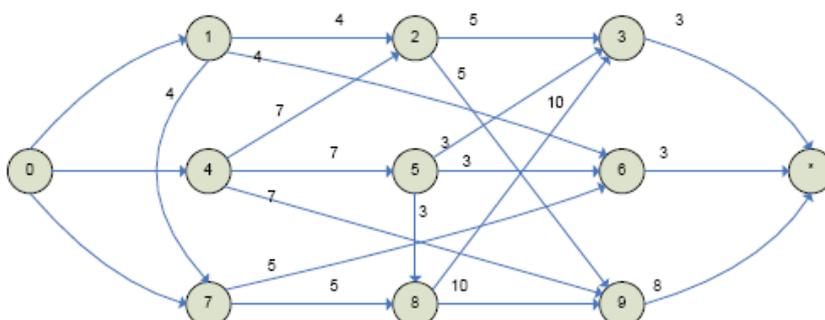


FIGURE 3.3 – Graphe conjonctif de la solution S1

Le graphe conjonctif ainsi obtenu contient de nombreux arcs, car il contient un arc pour chaque paire d'opérations en disjonction.

Il n'est pas nécessaire d'avoir autant d'arcs, comme le schématise la figure suivante, dans cette illustration, on a quatre opérations deux à deux en disjonction, comme il y a plus long chemin qui emprunterait l'arc de 1 vers 3, pourrait être rallongé en passant par 2. On peut donc simplifier le graphe par réduction transitive des arcs conjonctifs, le graphe ainsi obtenu est appelé graphe conjonctif. On peut directement obtenir le graphe conjonctif en transformant la sélection en un ordre des opérations. Puis pour chaque paire d'opérations successives dans cet ordre, on crée un arc allant d'une opération vers la suivante. Pour illustrer cette construction, le tableau TAB.3.3 donne l'ordre des opérations sur les machines, et la figure FIG.3.4 représente le graphe réduit.

<b>M1</b>	1	7	6
<b>M2</b>	2	4	9
<b>M3</b>	5	8	3

TABLE 3.3 – Ordre S1 d'instance (J1).

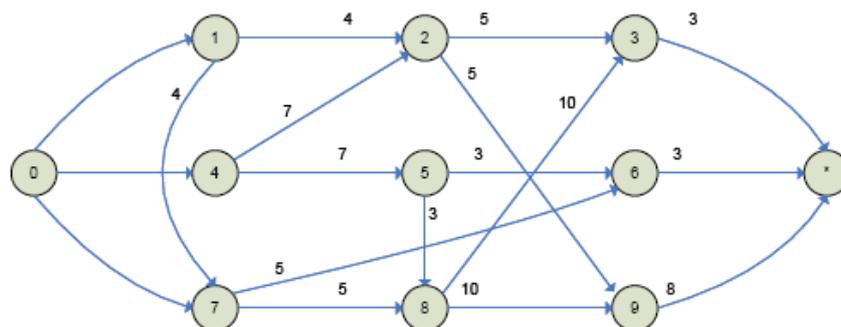


FIGURE 3.4 – Graphe conjonctif simplifié de la solution S1

Pour un ordonnancement donné, certaines notions bien évidentes sur le graphe conjonctif-disjonctif sont intéressantes à évoquer :

1. **L'opération critique** : Une opération est dite critique, si elle provoque nécessairement l'augmentation du makespan de l'ordonnancement, lorsqu'elle est retardée (alors que l'ordre d'exécution des opérations défini par l'ordonnancement ne change pas).
2. **Le chemin critique** : Le chemin critique est une suite d'opérations critiques liées par des relations de précédence. La longueur d'un chemin critique est égale à la somme des durées des opérations qui le composent. Pour un ordonnancement donné, il peut exister plus d'un chemin critique.
3. **Le bloc critique** : est une succession d'opérations critiques s'exécutant sur la même machine. Tout chemin critique peut se décomposer en  $b$  blocs critiques. Pour un problème  $n$ -tâches  $m$ -machines,  $b$  est compris entre 1 et  $n$ .

## 3.2 Complexité des problèmes d'ordonnancement

Les problèmes d'ordonnancement sont en générale NP-difficile même si l'atelier est simple. Plusieurs chercheurs montrent que si l'atelier possède un nombre de tâches ou de machines supérieur ou égale à trois et pour les problèmes à deux machines avec recirculation le problème devient NP-difficile.

Mais pour certain cas particuliers, il existe des algorithmes polynomiaux permettent de les résoudre, parmi les algorithmes les plus connus, on cite :

- Algorithme de Johnson pour le problème de Flow shop à deux machines et  $n$  tâches (pour le critère de makespan).

- Algorithme de Jackson pour le problème de Job shop à deux machines et  $n$  tâches dont chaque tâche comprend au plus deux opérations (pour le critère de makespan).
- Algorithme de Kubiak pour le problème de Job shop à deux machines dont les tâches ont des opérations unitaires (pour le critère de makespan).
- Algorithme de Timkovsky pour le problème de Job shop à deux machines dont les tâches ont des opérations unitaires avec des dates de disponibilité (pour le critère de makespan).

### 3.3 Méthodes de résolution

Les méthodes de résolution des problèmes d'optimisation combinatoire prennent en considération deux facteurs : la qualité des solutions et le temps de résolution. Ainsi ces méthodes peuvent être classées en deux catégories : les méthodes exactes qui garantissent l'optimalité mais avec un temps de calcul très grand ; et les méthodes approchées qui peuvent perdre en optimalité pour gagner en temps d'exécution [12].

Dans cette section, nous allons, tout d'abord, définir brièvement quelques méthodes exactes comme les méthodes de séparation et évaluation progressive, la programmation dynamique et la programmation linéaire. Ensuite nous donnons quelques méthodes approchées comme les algorithmes génétiques, . . . , etc., et à la fin, nous nous intéressons à la présentation de la méthode de "**Recherche Tabou**" qui fait l'objet de ce travail.

#### 3.3.1 Les méthodes exactes

On peut définir une méthode exacte comme une méthode qui garantit l'obtention de la solution optimale pour un problème d'optimisation. L'utilisation de ces méthodes s'avère particulièrement intéressante, mais elles sont souvent limitées au cas des problèmes de petite taille. Parmi ces méthodes on distingue les méthodes de séparation et évaluation progressive (branch and bound), la programmation dynamique, et la programmation linéaire [9].

##### 1. Les méthodes de séparation et évaluation

Les méthodes de séparation et évaluation (SE) consistent à placer progressivement les tâches sur les ressources en explorant un arbre de recherche décrivant toutes les combinaisons possibles. Il s'agit de trouver la meilleure configuration donnée de manière à élaguer les branches de l'arbre qui conduisent à de mauvaises solutions. Ces méthodes effectuent une recherche complète de l'espace des solutions d'un problème donnée, pour trouver la meilleure solution [3].

La démarche de ces méthodes consiste à :

- Diviser l'espace de recherche en sous espaces ;
- Chercher une borne minimale en terme de fonction objectif associée à chaque sous espace de recherche ;
- Éliminer les mauvais sous-espaces ;
- Reproduire les étapes précédentes jusqu'à l'obtention de l'optimum global.

##### 2. La programmation dynamique

Elle se base sur le principe de Bellman (1954) : " Si C est un point qui appartient au chemin optimal entre A et B, alors la portion de ce même chemin allant de A à C est le

chemin optimal entre A et C ". C'est une méthode qui consiste donc à construire d'abord les sous-chemins optimaux et ensuite par récurrence le chemin optimal pour le problème entier. Cette méthode est destinée à résoudre des problèmes d'optimisation à vocation plus générale que la méthode de séparation et d'évaluation (branch and bound) sans permettre pour autant d'aborder des problèmes de tailles importantes.

### 3. La programmation linéaire

C'est l'une des techniques classiques de recherche opérationnelle. Elle permet de modéliser les problèmes dont le critère et les contraintes sont des fonctions linéaires des variables de décision. Parmi les algorithmes les plus importants pour le traitement d'un programme linéaire nous citons la méthode du simplexe.

## 3.3.2 Les méthodes approchées

La taille des problèmes influe de façon importante sur les temps de calcul des méthodes exactes. Ce temps de calcul est raisonnable pour les problèmes de petite taille, mais il devient vite prohibitif si la taille des problèmes augmente, et par conséquent il est utile de développer des méthodes approchées. Ces méthodes qui sans garantir l'optimum absolu, peuvent fournir d'excellentes solutions. En plus d'être beaucoup plus génériques et facilement applicables, ces méthodes possèdent l'immense avantage d'être beaucoup moins contraignantes de sorte qu'elles sont pratiquement toujours applicables. Ainsi, dans de nombreux cas, les méthodes approchées deviennent la seule option performante envisageable [13]. Ce sont en quelque sorte des méthodes de dernier recours.

Il existe deux types de méthodes approchées : les méthodes constructives (les heuristiques) et les méthodes d'amélioration (les métaheuristiques).

### 1. Les méthodes constructives ou heuristiques

Ce sont des méthodes très utilisées pour obtenir très rapidement une première solution pour les problèmes d'ordonnancement. La recherche d'une solution par ces méthodes commence à partir du zéro et à chaque itération une solution pour le problème est élaborée en utilisant, par exemple, des règles de priorité. Ces méthodes sont généralement très rapides.

### 2. Les méthodes d'amélioration ou métaheuristiques

Le principe de ces méthodes n'est plus de construire un ordonnancement à partir des données initiales du problème, mais de modifier le résultat d'une solution admissible en vue d'améliorer la valeur de la fonction objectif. La plupart de ces méthodes utilisent la notion de voisinage de solution, on parle aussi de méthodes par voisinage.

À partir d'une solution initiale (générer aléatoirement ou avec une heuristique), une méthode par voisinage ou encore méthode de recherche locale réalise un processus itératif qui consiste à remplacer la solution courante par l'un de ses voisins en tenant compte de la fonction objectif. La méthode s'arrête et retourne la meilleure solution trouvée. Le critère d'arrêt peut être un nombre d'itération atteint ou une valeur de la fonction objectif atteinte.

#### a) Structure de voisinage et minimum local

Soit  $S$  un ensemble de solution à un problème d'optimisation, et soit  $f$  la fonction objectif. Une structure de voisinage (ou tout simplement un voisinage) est une fonction  $V$  qui associe un sous-ensemble de  $S$  à toute solution  $s \in S$ . Une solution  $\acute{s} \in V(s)$  est dite voisine de  $s$ .

- Une solution  $s \in S$  est un minimum local relativement à la structure de voisinage  $\mathbf{V}$  si :  
 $f(s) \leq f(\acute{s}), \forall \acute{s} \in V(s)$ .
- Une solution  $s \in S$  est un minimum global si :  $f(s) \leq f(\acute{s}), \forall \acute{s} \in S$ .

### b) Classification des métaheuristiques

#### ✂ Les méthodes à base de population

Ces méthodes, comme leur nom l'indique, travaillent sur une population de solution à la fois. Leur principe général consiste à combiner des solutions entre elles pour former de nouvelles, en essayant d'hériter les bonnes caractéristiques des solutions parents. Un tel processus est répété jusqu'à ce qu'un critère d'arrêt soit satisfait.

Parmi les métaheuristiques à base de population, on trouve deux grandes classes : les Algorithmes Génétiques et les Colonies de Fourmis.

**Les algorithmes génétiques :**

Un algorithme génétique est basé sur les mécanismes biologiques tels que les lois de Mendel et sur le principe fondamental (sélection) de Charles Darwin. Holland exposa les principes de ces types d'algorithmes pour permettre aux ordinateurs d'imiter les êtres vivants en évoluant (croisement, mutation, sélection), pour rechercher la solution à un problème. Il expliqua d'abord comment ajouter de l'intelligence dans un programme informatique. Il formalisa ensuite les principes fondamentaux d'un algorithme génétique.

Récemment, un nouveau paradigme génétique fait son apparition, c'est la Programmation Génétique. Le principal promoteur de ce paradigme est J. Koza [13] .

**Les colonies de fourmis :**

L'histoire de l'intelligence remonte à l'étude du comportement des fourmis à la recherche de nourriture au départ de leur nid (Goss, Deneubourg). Les fourmis déplacent du nid à la source de nourriture et vice-versa, d'une façon aléatoire dans un premier temps, puis elles déposent au passage sur le sol une substance odorante appelée phéromone, ce qui est pour effet de créer une piste chimique. Les fourmis peuvent sentir ces phéromones qui ont un rôle de marqueur de chemin, en effet, elles ont une tendance à choisir la piste qui porte la plus forte concentration de phéromone. Cela leur permet de retrouver le chemin vers leur nid lors du retour. Le système de fourmis (Ants System -AS-) est une méthode d'optimisation basée sur ces observations proposées par Dorigo [4].

**✂ Les méthodes de recherche locale**

**Le Recuit Simulé :**

Le recuit simulé est une technique d'optimisation de type Monte-Carlo généralisé à laquelle on introduit un paramètre de température qui sera ajusté pendant la recherche. Elle s'inspire des méthodes de simulation de Metropolis (années 50) en mécanique statistique. L'analogie historique s'inspire du recuit des métaux en métallurgie : un métal refroidi trop vite présente de nombreux défauts microscopiques, c'est l'équivalent d'un optimum local pour un problème d'optimisation combinatoire. Si on le refroidit lentement, les atomes se réarrangent, les défauts disparaissent, et le métal a alors une structure très ordonnée, équivalente à un optimum global.

La méthode du recuit simulé considère une solution initiale et recherche dans son voisinage une autre solution de façon aléatoire. Ce qui implique qu'elle peut se diriger vers une solution voisine de mauvaise qualité avec une probabilité non nulle. Ceci permet d'échapper aux optima locaux. Au début de l'algorithme, un paramètre T (température), est déterminé et décroît tout au long de l'algorithme pour tendre vers 0. La valeur de ce paramètre dépend de la probabilité d'acceptation des solutions. Plus la température T est élevée, plus la probabilité devient forte [2].

**La Recherche Tabou :**

La recherche Tabou a été introduite par F. Glover, le but de l'auteur est de concevoir une méthode de recherche intelligente. En effet, plusieurs notions importantes sont intégrées : l'intensification, la diversification, les critères d'aspiration, la liste taboue, etc. Cette méta-heuristique est très connue, très populaire, et elle rencontre un grand succès auprès

des chercheurs en optimisation combinatoire. Glover et Laguna ont réalisé un ouvrage très détaillé sur cette méta-heuristique.

Le principe de l'algorithme est le suivant : à chaque itération, le voisinage (complet ou sous-ensemble de voisinage) de la solution courante est examiné, la meilleure solution est sélectionnée. En appliquant ce principe, la méthode autorise de remonter vers des solutions qui semblent moins intéressantes mais qui ont peut être un meilleur voisinage. Le risque est de cycler entre deux solutions. Pour éviter les phénomènes de cycler, la méthode a l'interdiction de visiter une solution récemment visitée par le biais d'une liste taboue contenant les attributs des dernières solutions visitées est tenue à jour. Ainsi, tout au long de l'algorithme, la meilleure solution doit être conservée car l'arrêt se fait rarement sur la meilleure solution.

### Algorithme général de la Recherche Tabou

**Algorithme 1 :** Algorithme générale de la Recherche Tabou

#### Entrées :

$x$  : Solution de départ ;

$cx$  : Coût de  $x$  ;

#### 1.Début

2.  $T = \emptyset$  ;

3. **Tant que** (nécessaire) **Faire**

4. **Pour** (tout voisin  $\hat{x}$  de  $x$ ) **Faire**

5. **Si** ( $\hat{x} \in T$ ) **Alors**

6.  $c\hat{x} = c(x)$  ;

7. **Si** ( $c\hat{x} < cx$ ) **Alors**

8.  $cx = c\hat{x}$  ;

9.  $x = \hat{x}$  ;

10. **Fin si**

11. **Fin si**

12. **Fin pour**

13. Mettre à jour la liste  $T$  en insérant le nouveau  $\hat{x}$  ;

14. **Fin tant que**

15.Fin

## Conclusion

Dans ce chapitre nous avons vu la représentation des problèmes d'ordonnancement ainsi que ses différentes méthodes de résolutions. Ces méthodes sont regroupées en deux classes principales : les méthodes exactes et les méthodes approchées.

Pour notre problème Job Shop, nous avons choisi l'une des méthodes approchées (métaheuristique) qui est l'algorithme de la recherche tabou afin d'ordonner les opérations assignées à chaque machine, qui fait le sujet de notre application, et qui sera présenté dans le prochaine chapitre.

# 4

## Choix de la méthode et implémentation

### Introduction

Trouver une des solutions exactes pour les problèmes d'optimisation combinatoire NP-difficiles est une tâche très ardue. Les heuristiques ou les métaheuristiques deviennent l'unique moyen d'obtenir une bonne solution en un temps raisonnable. Ces heuristiques sont principalement basées sur des méthodes de recherche locale (méthodes de recherche tabou, recuit simulé) ou des méthodes évolutives (algorithmes génétique, colonies de fourmis).

Dans le but de résoudre le problème complexe d'ordonnement des ateliers de type Job Shop par la métaheuristique (la recherche tabou), nous présentons dans ce chapitre l'implémentation et la justification du choix adapté pour notre application concernant les techniques, ingrédients et paramètres de la méthode adaptée au problème de Job Shop.

### 4.1 Le choix de la méthode

Le problème que nous considérons est un problème de type job-shop classique. En tenant compte aussi de la taille du problème qui se traduit par le nombre important des variables et des contraintes, nous incitent à l'utilisation des méthodes approchées, où les métaheuristiques présentent de nombreux avantages.

Nous traitons dans ce travail l'approche la méthode de recherche tabou, en utilisant le modèle de graphe conjonctif-disjonctif de Roy (1964). Des mécanismes intelligents sont introduits, le chemin critique, voisinage guidé, la liste tabou et le mécanisme de retour arrière sont les points essentielles qu'on va traiter dans ce chapitre.

La méthode de recherche tabou été largement utilisée dans la littérature, nous citons par exemple l'algorithme tabou Nowiski et smutnicki (1996) qui est l'un des meilleurs algorithmes pour le problème de job shop. Cette méthode appartient aux méthodes de recherche locale déterministe, c'est une méthode itérative qui utilise la notion de voisinage pour explorer l'espace de recherche [16].

### 4.1.1 Les ingrédients de la méthode

#### Définition des variables

1. **Solution initiale  $S_0$  :**

La solution initiale est une solution calculée au départ par un algorithme de complexité simple (linéaire) ou par l'utilisateur si elle est facile à calculer (solution aléatoire). Le but ici n'est pas de trouver une solution optimale mais de fournir une solution quelconque d'une manière naïve, qui serve à faire commencer l'application de l'algorithme (TS).

2. **La solution courante  $S$  :**

La solution courante est une solution calculée par le mécanisme de sélection pour une itération  $k$  de l'algorithme Recherche Tabou (TS), le but est d'explorer l'espace de recherche, afin d'atteindre une solution meilleure.

3. **Fonction objectif  $f()$  :**

La fonction objectif est une fonction qui évalue l'optimalité de la solution (minimiser le Makspan).

4. **La fonction de voisinage  $V()$  :**

La fonction de voisinage  $V$  est définie sur l'espace de recherche,  $\Omega$  et associe à chaque configuration un sous-ensemble  $\Omega$  de (l'espace de solutions voisines à  $s$ ,  $V(s)$ ,  $V(s)=s'$ ).

La configuration  $s'$  voisine est obtenue par un mouvement (permutation de deux tâches s'exécutant sur la même machine), mais les contraintes restent toujours satisfaites.

5. **La solution optimale globale  $S_{globale}$  :**

La solution optimale globale est la meilleure solution trouvée par l'algorithme tabou pendant la recherche, et elle est retournée à la fin de processus à l'utilisateur.  $S_{globale}$  est une solution approchée, il peut exister d'autres solutions meilleures que celle-ci.

6. **Optimum local  $S^*$  [18] :**

Dans un voisinage donné, il existe toujours un optimum local qui représente la meilleure solution dans ce dernier. Il existe des mécanismes pour s'échapper de cette solution parmi eux l'utilisation de la liste taboue.

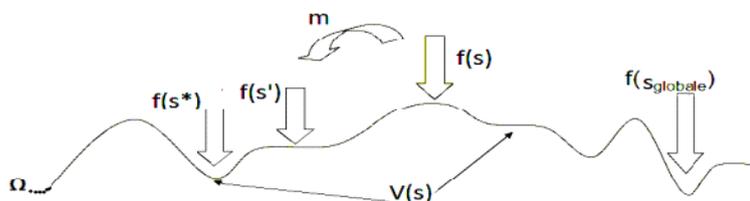


FIGURE 4.1 – Les variables manipulés par l'algorithme tabou

## Définition des termes

### 1. La liste tabou **T** :

L'idée de base de la liste tabou consiste à mémoriser les configurations ou les régions visitées et à introduire des mécanismes permettant d'interdire à la recherche de retourner trop rapidement vers ces configurations, dans le but d'empêcher l'algorithme de tomber dans des cycles à court terme. Ces mécanismes sont des interdictions temporaires de certains mouvements (tabous).

Pour les décrire, il faut spécifier :

- La structure de la liste tabou, la nature des éléments qui la composent ;
- Comment la liste est mise à jour quand un mouvement est effectuée (quels éléments sont ajoutés) ?
- Quels mouvements ou voisins sont interdits, selon le contenu de la liste Taboue ?
- Il faut par ailleurs déterminer le temps pour qu'un élément reste tabou.

Les performances de l'algorithme est de fixer le nombre d'itérations pendant lequel un mouvement reste tabou égale à la longueur de la liste. Une liste **T** avec une grande taille peut devenir très restrictive, par contre si sa taille est limitée peut être devenu inutile et mène à des mouvements cycliques. Mais il a été observé que trop de contraintes (tabous) forcent le programme à visiter des solutions voisines peu alléchantes à la prochaine itération. On parle de stratégie de diversification a court terme.

### 2. Mécanismes et critères d'aspiration [6] :

Un mécanisme d'aspiration détermine un critère selon lequel un mouvement, bien que tabou, peut quand même être acceptée. Ce critère est appelée critère d'aspiration qui sert à autorisé un mouvement interdit, malgré son statut tabou, car dans certains cas, les interdictions occasionnées par la liste taboue peuvent amener la recherche à une bonne solution.

### 3. Critère d'arrêt :

Le critère d'arrêt est un mécanisme qui nous permet d'arrêter la recherche. Par exemple une solution prouvée optimale a été trouvée, une limite a été atteinte en ce qui concerne le nombre d'itérations, le temps de recherche, La recherche semble stagner (nombre d'itérations sans amélioration de la meilleure configuration trouvée).

## L'algorithme Tabou

### Algorithm 2 : Tabou simplifié

**Entrées** : Une instance de problème ;

**Sorties** : Une solution de bonne qualité ;

#### 1. Début

#### 2. Étape1 :

3. Choisir une solution  $s_0$  dans  $\Omega$  ; // l'ensemble des solutions

4. Appliquer :  $s_{global} = s_0 ; k = 0 ; T = 0$  ;

#### 5. Répéter

**6. Étape2 :**

7. Appliquer  $k = k + 1$  ;
8. Générer un sous-ensemble de solutions en  $V\{s, k\}$  pour que :
9. Les mouvements tabous ne soient pas choisis ;
10. Un des critères d'aspiration  $a\{s, m\}$  soit applicable ;

**11.Étape3**

12. Choisir la meilleure solution  $\acute{s}$  parmi l'ensemble de solutions voisines ;
13. Appliquer :  $s = \text{meilleur } \acute{s}$  ;

**14.Étape4**

15. **Si**  $(f(s) \leq f(s_{global}))$  **Alors**

16. Appliquer  $s_{global} = s$  ; // nous avons trouvé une solution meilleure ;

**17.Étape5**

18. Mettre à jour la liste T et les critères d'aspiration ;
19. **Jusqu'à** (une condition d'arrêt soit atteinte) ;

**20.Étape6**

21. Retourner  $s_{global}$  ; // La meilleure solution trouvée ;

**22.Fin****Amélioration de l'algorithme Tabou****1. Amélioration au niveau de voisinage****1.1.Chemin critique :**

Le chemin critique est le chemin le plus long du graphe conjonctif, c'est lui qui définit le makespan de l'ordonnancement correspondant à ce graphe conjonctif. Pour diminuer le makespan d'une solution, il est donc nécessaire de modifier ce chemin. Toute modification de l'ordonnancement qui ne modifie pas le chemin critique ne peut pas améliorer la solution : elle ne peut que faire apparaître un nouveau chemin de longueur supérieure car le plus long chemin du graphe initial reste intact. Inversement, toute modification du chemin critique casse ce chemin, mais rien ne garantit que le nouvel ordonnancement soit égal, meilleur ou pire.

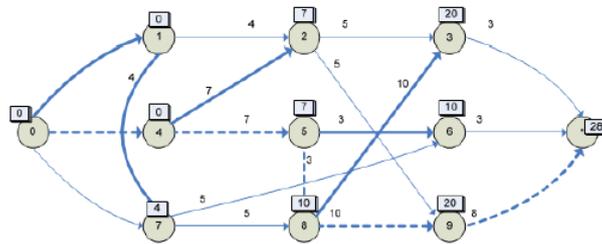


FIGURE 4.2 – Graphe conjonctif évalué avec le sous graphe critique (en pointillés).

Nowicki et Smutnicki (1996) ont proposé un voisinage encore plus guidé qui consiste à permuter les deux premières opérations ou les deux dernières opérations d'un bloc, excepté pour le premier bloc dont on ne permute pas les deux premières opérations et pour le dernier bloc dont

on ne permute pas les deux dernières opérations. Outre la proposition d'un voisinage performant, Nowicki et Smutnicki ont prouvé que leur voisinage dominait le voisinage de Laarhoven. Cela signifie que tout voisin améliorant dans le voisinage de Laarhoven est dans le voisinage de Nowicki et Smutnicki. De plus, ils prouvent que toute solution dont le voisinage est vide (car cela peut arriver) est une solution optimale. Nowicki et Smutnicki (1996) ont utilisé ce voisinage pour réaliser un algorithme tabou. Jusqu'à aujourd'hui leur algorithme est considéré comme étant l'un des algorithmes les plus performants pour le problème de job shop. De plus, Nowicki et Smutnicki (1996) ont montré que ce voisinage était lui aussi faiblement connecté (i.e. qu'il permet toujours d'obtenir l'optimum en un nombre fini d'étapes).

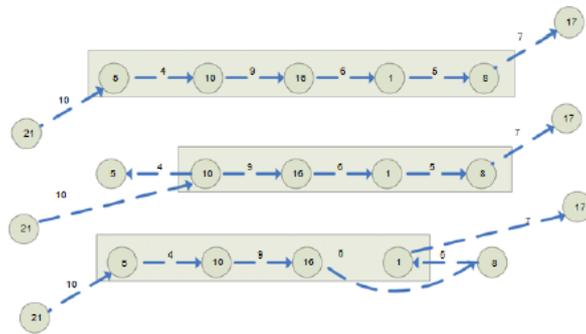


FIGURE 4.3 – Voisinage de Nowicki et Smutnicki

## 4.2 Implémentation

Nous décrivons dans cette partie l'implémentation de l'algorithme sur la machine. Pour cela, nous utilisons le langage de programmation MATLAB. Nous présentons donc les ensembles des fonctions et leurs tâches en donnant quelques détails qui peuvent mettre en relie le fonctionnement de notre application.

En représente l'ordre de passage de 19 pièces sur les 10 machines au niveau de l'atelier de production de FAGECO, chaque pièce est traitée par 10 machines avec un temps d'exécution différent. Le problème qui se pose est : Comment organiser l'exécution de ces pièces avec un temps minimum.

### Tableau des Machines :

<b>Machine</b>	Sciage	Cintrage	Pliage	Tournage	Fraisage
<b>Indice</b>	M 1	M 2	M 3	M 4	M 5
<b>Machine</b>	Perçage	Rivetage	Soudage	Peinture	Montage
<b>Indice</b>	M 6	M 7	M 8	M 9	M 10

TABLE 4.1 – Tableau des Machines

### Tableau des pièces :

On définit 19 pièces (Jobs) différents qui passent par les machines (au-dessus), tel que chaque pièce passe par une gamme qui définit l'ordre de passage sur les machines et le temps d'exécution de chaque opération.

<b>Partie charpente</b>		
<b>Indice</b>	<b>Tâche</b>	<b>Machine : Temps d'exécution (Minutes)</b>
$J_1$	CHASSIS	$M_4 : 6.53 ; M_1 : 3.34 ; M_2 : 0.68 ; M_5 : 2.02 ; M_3 : 1.82 ; M_6 : 3.02 ; M_8 : 15.12 ; M_7 : 6.9 ; M_9 : 2.82 ; M_{10} : 3 ;$
$J_2$	CABINE	$M_3 : 2.69 ; M_5 : 2.84 ; M_1 : 7.33 ; M_4 : 21.64 ; M_8 : 12 ; M_7 : 5 ; M_2 : 1.34 ; M_6 : 6.04 ; M_{10} : 15 ; M_9 : 2.82 ;$
$J_3$	RELEVAGE	$M_1 : 6.67 ; M_3 : 3.5 ; M_4 : 19.13 ; M_2 : 0.89 ; M_7 : 2.22 ; M_8 : 7.31 ; M_5 : 1.81 ; M_6 : 2.04 ; M_9 : 3.11 ; M_{10} : 8.5 ;$
$J_4$	MAT EX	$M_1 : 13.3 ; M_2 : 0.45 ; M_4 : 18.39 ; M_5 : 2.74 ; M_3 : 5.01 ; M_6 : 3.39 ; M_8 : 16.48 ; M_7 : 11.13 ; M_9 : 2.5 ; M_{10} : 8.5 ;$
$J_5$	MAT INT	$M_1 : 6.49 ; M_2 : 0.67 ; M_3 : 1.17 ; M_4 : 5.33 ; M_6 : 1.8 ; M_5 : 0 ; M_7 : 9.3 ; M_8 : 10.2 ; M_9 : 2.5 ; M_{10} : 8.5 ;$
$J_6$	MAT ACC	$M_1 : 5.36 ; M_2 : 0.48 ; M_5 : 2.11 ; M_3 : 4.27 ; M_4 : 11.69 ; M_6 : 0.74 ; M_8 : 2.62 ; M_7 : 0 ; M_9 : 0.91 ; M_{10} : 8.5 ;$
$J_7$	PIED DE FLECHE	$M_2 : 0.6 ; M_4 : 9.98 ; M_1 : 7.39 ; M_3 : 4 ; M_5 : 1.8 ; M_7 : 2.1 ; M_8 : 12.4 ; M_6 : 0.42 ; M_{10} : 8.5 ; M_9 : 2 ;$
$J_8$	TETE DE FLECHE	$M_4 : 6.07 ; M_2 : 0.38 ; M_1 : 6.19 ; M_3 : 2.71 ; M_6 : 1.1 ; M_8 : 6.03 ; M_7 : 5.09 ; M_5 : 0.74 ; M_9 : 1.5 ; M_{10} : 8.5 ;$
$J_9$	FLECHE ACC(chariot)	$M_1 : 1.6 ; M_3 : 0.65 ; M_4 : 5.09 ; M_2 : 0 ; M_6 : 1.16 ; M_7 : 1.10 ; M_8 : 2.68 ; M_5 : 0.89 ; M_{10} : 8.5 ; M_9 : 0.52 ;$
$J_{10}$	TRANSPORT	$M_3 : 3.36 ; M_2 : 0.91 ; M_4 : 7.33 ; M_5 : 0 ; M_6 : 0 ; M_1 : 4.36 ; M_8 : 6.69 ; M_7 : 2.21 ; M_9 : 2.16 ; M_{10} : 2.7 ;$
$J_{11}$	MOUFLE	$M_5 : 0.36 ; M_1 : 2.3 ; M_3 : 0.48 ; M_2 : 0.37 ; M_4 : 4.26 ; M_6 : 0 ; M_7 : 0 ; M_8 : 1 ; M_{10} : 0.34 ; M_9 : 1.23 ;$
<b>Partie mécanique</b>		
<b>Indice</b>	<b>Tâche</b>	<b>Machine : Temps d'exécution (Minutes)</b>
$J_{12}$	CHASSIS	$M_1 : 0 ; M_2 : 0.014 ; M_4 : 2.76 ; M_5 : 5.52 ; M_3 : 0.61 ; M_6 : 0 ; M_7 : 0 ; M_8 : 0.74 ; M_9 : 0 ;$
$J_{13}$	TRANSPORT	$M_2 : 0.59 ; M_3 : 1.34 ; M_1 : 1.1 ; M_5 : 15.14 ; M_4 : 7.3 ; M_7 : 2.4 ; M_6 : 0.5 ; M_8 : 4.36 ; M_9 : 0.5 ;$
$J_{14}$	CABINE	$M_1 : 0.27 ; M_3 : 0.96 ; M_2 : 0 ; M_4 : 3.08 ; M_6 : 0.66 ; M_5 : 5.38 ; M_8 : 0 ; M_7 : 0.47 ; M_9 : 0 ;$
$J_{15}$	TREUIL	$M_4 : 8.08 ; M_3 : 0.08 ; M_2 : 1.42 ; M_5 : 15.61 ; M_1 : 0.21 ; M_6 : 6.85 ; M_7 : 0.47 ; M_8 : 0 ; M_9 : 0 ;$
$J_{16}$	RELEVAGE	$M_1 : 0 ; M_2 : 0 ; M_4 : 9.29 ; M_3 : 2.87 ; M_6 : 0.66 ; M_5 : 10.17 ; M_8 : 2.69 ; M_7 : 1.2 ; M_9 : 0 ;$
$J_{17}$	MATS	$M_2 : 0 ; M_1 : 1.32 ; M_4 : 11.8 ; M_3 : 0.94 ; M_5 : 13.78 ; M_6 : 1.94 ; M_7 : 0 ; M_8 : 1.23 ; M_9 : 0.09 ;$
$J_{18}$	FLECHES	$M_3 : 0 ; M_4 : 4.52 ; M_5 : 8.99 ; M_1 : 0 ; M_7 : 0.31 ; M_8 : 1.13 ; M_6 : 1.17 ; M_2 : 0 ; M_9 : 0 ;$
$J_{19}$	POULIE	$M_1 : 0 ; M_2 : 0 ; M_3 : 0 ; M_4 : 0.95 ; M_5 : 27.16 ; M_6 : 0.2 ; M_7 : 0 ; M_8 : 0 ; M_9 : 2.9 ;$

TABLE 4.2 — Tableau des tâches

Pour bien illustrer les temps d'exécutions des chaque machine, nous avons fait appel à EXCEL afin de calculer les temps d'utilisation de chaque machine  $k$  pour l'ensemble des tâches  $J$ .

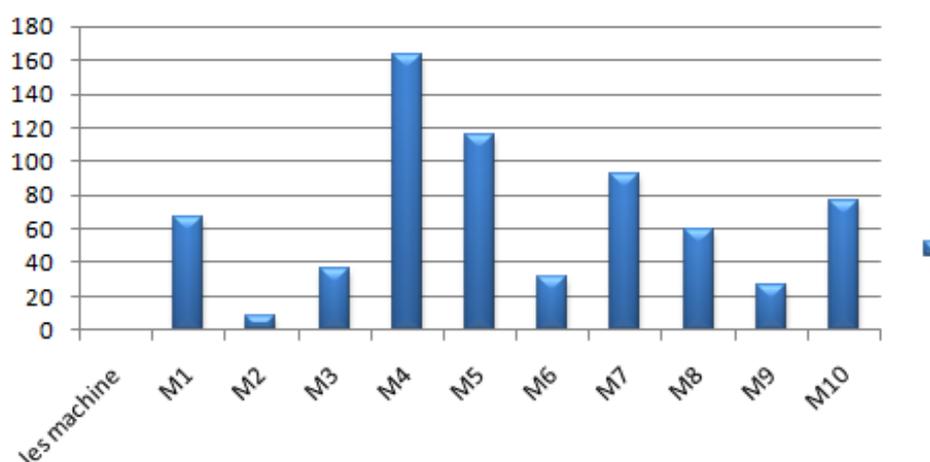


FIGURE 4.4 – Les temps d'exécutions des chaque machine

## 4.2.1 Langage de Programmation

### Description de l'environnement MATLAB :

MAT(rix) LAB (oratory) est un logiciel puissant doté à la fois d'un langage de programmation haut niveau et d'outils dédiés au calcul numérique et à la visualisation numérique. Développé en C par la société Mathworks, MATLAB était initialement destiné à faire du calcul matriciel simplement. Actuellement, MATLAB recouvre d'autres domaines d'applications tels que :

- Résolution des problème non-linéaire ;
- Optimisation et contrôle ;
- Résolution des systèmes linéaires ;
- Graphique 2D et 3D.

### Justification du choix de l'environnement de programmation :

Le choix s'est porté a l'emploi du logiciel MATLAB, en raison que ce dernier a des applications destinées au domaine d'optimisation qui nécessite le calcule matriciel avec une grande rapidité d'exécution. MATLAB comprend en standard plusieurs outils utiles à la résolution de problèmes d'optimisation.

Pour les besoins plus complexes, la boîte à outils Optimization Toolbox complète les fonctions de base avec de nombreux algorithmes spécialisés.

## 4.2.2 Résultats de la méthode tabou

Bien que la méthode recherche tabou a montré son efficacité pour la résolution de plusieurs problèmes difficiles, elle reste néanmoins, difficile à adapter au problème job shop comme dans notre cas. Ceci est surtout dû au nombre considérable des paramètres à définir (solution initiale, fonction voisinage, évaluation de la solution courante, taille de la liste tabou).

Les données qu'on a introduit à notre application sont présentées comme suit :

- Nombre de machines :  $m = 10$ .
- Nombre de tâches :  $n = 19$ .
- Nombmre maximal d'itérations = 100.

Indice	Tâche	Temps d'exécution (Minutes)
$J_1$	CHASSIS	78.91
$J_2$	CABINE	76.7
$J_3$	RELEVAGE	55.21
$J_4$	MAT EX	81.89
$J_5$	MAT INT	45.96
$J_6$	MAT ACC	36.68
$J_7$	PIED DE FLECHE	49.19
$J_8$	TETE DE FLECHE	38.31
$J_9$	FLECHE ACC(chariot)	22.19
$J_{10}$	TRANSPORT	29.72
$J_{11}$	MOUFLE	10.34
$J_{12}$	CHASSIS	9.77
$J_{13}$	TRANSPORT	33.23
$J_{14}$	CABINE	10.82
$J_{15}$	TREUIL	32.72
$J_{16}$	RELEVAGE	26.88
$J_{17}$	MATS	31.1
$J_{18}$	FLECHES	16.12
$J_{19}$	POULIE	31.21

TABLE 4.3 – Temps d'exécution des tâches

Après plusieurs exécutions de l'algorithme et adaptation des différents paramètre à notre problème sous MATLAB, on a obtenus les résultats suivants :

Indice	Tâche	Le passage dans la machine
$J_1$	CHASSIS	2
$J_2$	CABINE	3
$J_3$	RELEVAGE	4
$J_4$	MAT EX	5
$J_5$	MAT INT	6
$J_6$	MAT ACC	8
$J_7$	PIED DE FLECHE	7
$J_8$	TETE DE FLECHE	9
$J_9$	FLECHE ACC(chariot)	10
$J_{10}$	TRANSPORT	1
$J_{11}$	MOUFLE	2
$J_{12}$	CHASSIS	3
$J_{13}$	TRANSPORT	4
$J_{14}$	CABINE	5
$J_{15}$	TREUIL	6
$J_{16}$	RELEVAGE	7
$J_{17}$	MATS	9
$J_{18}$	FLECHES	8
$J_{19}$	POULIE	10

TABLE 4.4 – La solution initiale

Itération	Coût
1	222.97
2	202.19
3	168.96
4	167.76
5	164.12
6	158.62
7	144.46
8	141.46
9	141.63
10	125.90

TABLE 4.5 – Coût de chaque itération

Itération	Coût
11	117.22
12	116.13
13	111.51
14	105.44
15	105.31
16	93.52
17	92.71
18	86.02
19	85.87
20	82.09.90
21	81.89
22	81.89
23	81.89
24	81.89
25	81.89

TABLE 4.6 – Coût de chaque itération

On remarque que le coût de chaque itération reste fixe a partir de 21eme itération jusqu'a la 100eme itération

Indice	Tâche	Le passage dans la machine
$J_1$	CHASSIS	7
$J_2$	CABINE	6
$J_3$	RELEVAGE	8
$J_4$	MAT EX	4
$J_5$	MAT INT	9
$J_6$	MAT ACC	3
$J_7$	PIED DE FLECHE	10
$J_8$	TETE DE FLECHE	5
$J_9$	FLECHE ACC(chariot)	9
$J_{10}$	TRANSPORT	3
$J_{11}$	MOUFLE	2
$J_{12}$	CHASSIS	3
$J_{13}$	TRANSPORT	2
$J_{14}$	CABINE	5
$J_{15}$	TREUIL	2
$J_{16}$	RELEVAGE	5
$J_{17}$	MATS	1
$J_{18}$	FLECHES	5
$J_{19}$	POULIE	9
Makespan	$C_{max}$	503.8798 Minutes

TABLE 4.7 – La meilleure solution avec le Makespan

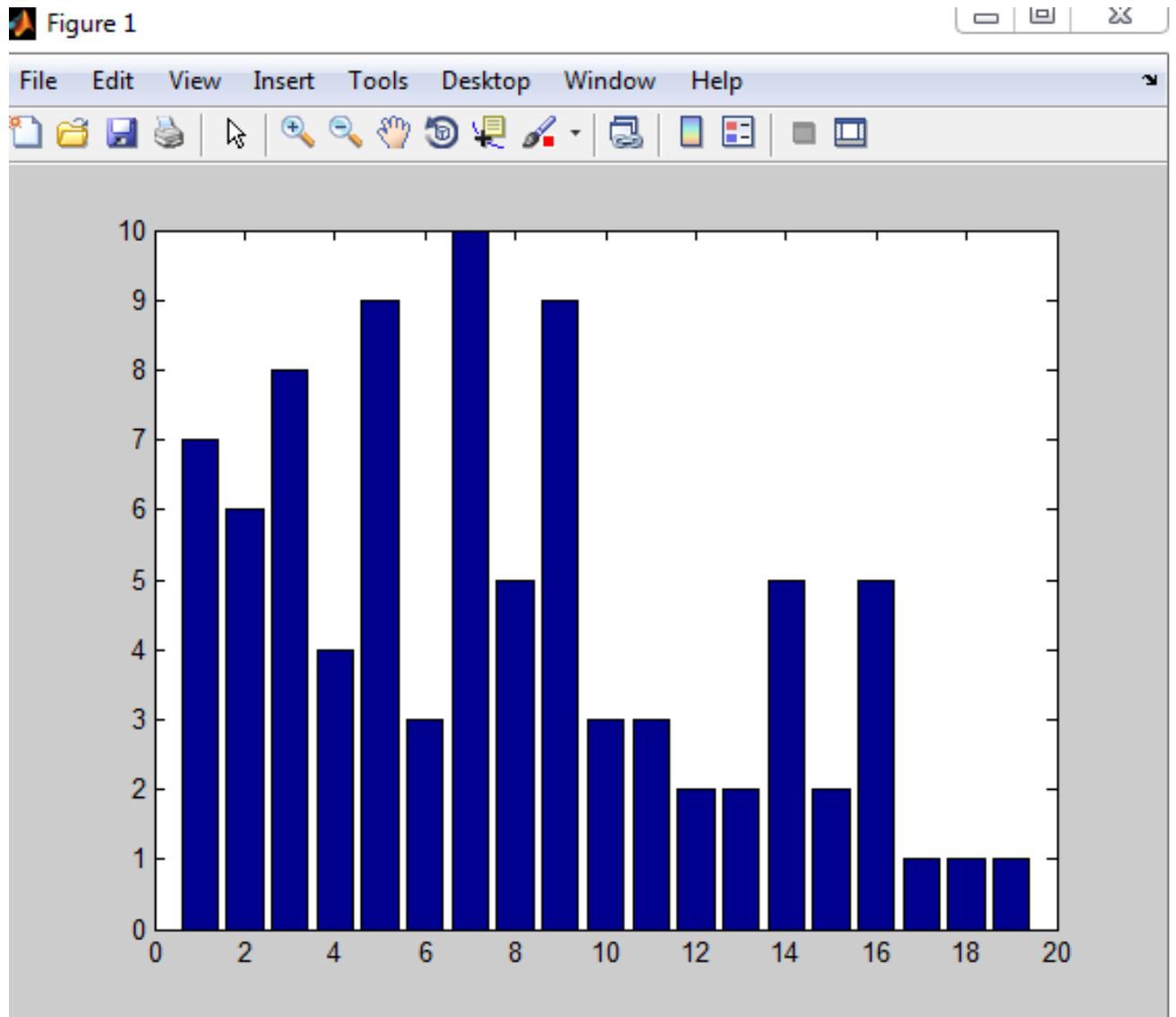


FIGURE 4.5 – Evaluation de la solution courante

### 4.2.3 Discussion des résultats

Après avoir lancer l'algorithme de recherche tabou dans MATLAB, on remarque que le temps d'exécution de 100 itérations augmente au fur et à mesure qu'on augmente le nombre de voisins. La complexité d'une approche de résolution basée sur la recherche tabou dépend de la taille du voisinage de solution courante et de la méthode d'évaluation de chacun de ses voisins afin de déterminer celui qui minimise le makespan.

Il a été démontré que près 90% du temps de résolution est pris par l'évaluation des voisinages. Par conséquent, il est intéressant d'utiliser des voisinages aussi restreints que possible afin de diminuer au maximum la complexité de résolution .

On peut conclure que la méthode recherche tabou :

- Incorpore de la mémoire dans la stratégie de recherche.
- Offre des économies dans la stratégie de recherche.
- la solution générée en général n'est pas optimale.
- Une liste tabou trop longue peut être restrictive, par contre une liste trop courte risque de s'avérer inutile.

On a pu aussi conclure que la méthode de recherche tabou donne la meilleure solution d'achèvement (makespan)  $C_{max} = 503.8798$  Minutes par rapport au temps de réalisation au niveau de l'entreprise ( $C_{max} = 716.31$  Minutes)

## Conclusion

Dans ce chapitre, nous avons présenté la solution fournie par la méthode recherche Tabou. Comme le travail d'un chercheur opérationnel se termine à ce stade (proposition des différentes solutions au décideur), le décideur prend la responsabilité (décision) de trancher sur le plan de production adéquat à lancer.

# Conclusion générale

Les systèmes de production d'une entreprise sont au centre des préoccupations du monde industriel et c'est devenu un enjeu économique et social incontournable. En effet, une meilleure prise en compte du temps de réalisation (temps d'achèvement d'une tâche) d'un produit dans la planification des travaux permet d'améliorer les processus de fabrication ainsi que le profit de l'entreprise.

Dans ce mémoire, nous avons présenté, dans un premier temps, le processus de production. Puis, nous avons décrit le problème d'Ordonnement d'atelier d'une manière détaillée. Après, Nous avons proposé une approche de modélisation du problème de passage des tâche en tant que problème d'ordonnement .

En second ordre on décrit les problèmes d'ordonnement et leur représentation ainsi que les ressources et les contraintes qui sont liée à ces derniers. Puis on parlé de l'organisation des ateliers, dans notre cas pratique c'est une organisation de type job shop. En suite on a donné le modèle théorique ainsi que la formulation mathématique.

Enfin, et après avoir donné quelques notions sur les méthodes de résolutions, nous avons fait une approche par la méthode de recherche Tabou et nous avons pu élaborer un ordonnancement pseudo optimal du processus de fabrication d'un produit FAGECO(une grue GMR 20/10 dans notre étude), nous avons présenté une solution qui minimise le temps de réalisation , et cela en appliquant l'algorithme Tabou que nous avons développer sous MATLAB qui permet d'avoir un ordre optimale de 19 pièce sur les 10 machine de la chaîne de production, ce qui nous a permet également de mieux gérer les ressources. Pour un meilleur suivi de l'évolution des opérations, nous avons fait recours à l'EXCEL, que nous avons utilisé pour illustré le temps d'exécution des machin sous forme d'un diagramme. De ce fait, on a estimé la durée minimale du processus à 503 minute.

Il reste à dire qu'on est arrivé a de bon résultats en appliquant la méthode de recherche Tabou sur les valeur réelles de l'entreprise dont on a effectué notre stage.

En termes de perspectives :

Même si nous considérons que cette méthaheuristique (Recherche Tabou) est performante pour les besoins industriels, nous pensons qu'une amélioration de son degré de performance est possible en faisant appel à d'autre méthaheuristique, entre autre les colonies de fourmis ou le recuit simulé.

Une autre perspective est de considérer une modélisation déférente afin d'amélioré la solution final à ce problème d'ordonnancement, comme par exemple, le fait de considérer un ordonnancement en flow shop peut s'avérer intéressant dans la pratique.

D'une autre part on pense que l'entreprise peut minimiser le temps d'exécution des tachés en faisant appelle a d'autre méthode, comme par exemple doté le département planning et la chaine de production d'une pointeuse informatisé et d'une application qui pourra transmettre les OF(Ordre de fabrication) en un temps minimum.

# Bibliographie

- [1] A.LETOUZEY. *Ordonnancement interactif basé sur des indicateurs*. Thèse doctorat, Institut National Polytechnique de Toulouse, 2001.
- [2] A.MERANI. *stat of the art review of job shop scheduling,1999*.
- [3] B.APTISTE. *Constraint baded branch and bound,algorithme for job shop shceduling*. 1996.
- [4] B.HELAB. *optimisation de problème job shop par l'algorithem génétique Tunisi 03/07/2009*.
- [5] D.DUVIER. *Etude de l'hybridation des métaheuristique :Application à un problème d'ordonnancement de type Job Shop*. Thèse doctorat,Université de Litoral côte d'opale,LIL, 2002.
- [6] D.JOHAN.ET.A.PETROWSKI. *Métaheuristique pour l'optimistion difficile*.
- [7] ET M.PILLET, A. *Gestion de production*. Editions d'Organisation, 4<sup>ème</sup> édition, 2002.
- [8] F.TANGOURI. *L'Ordonnancement dynamique dans les industrie agroalimentaires*. Thèse doctorat,Autoumatiqueet Informatique Industrielle, 2007.
- [9] H.BOUKEF. *Ordonnancement d'atelier job shop flexible et flow shop en indestriepharmaceutique*. 16 mars 2011.
- [10] H.HENTOUS. *Contribution au pilotage des systèmes de production de type Job Shop*. Thèse de doctorat ,INSA Lyon, 1999.
- [11] J.CARLIER.ET.P.CHRETIENNE. *Problèmes d'Ordonnancement,Modélisation,Complexité Algorithmes*. Eddition Masson,Paris, 1988.
- [12] K.MEBAREK. *Utilisation des stratégies Métaheuristique de l'atelier type Job Shop*. Thèse Magister,Université de Batna, 2008.
- [13] N.BENRABAH. *Recuit simulé pour le problème d'ordonnancement de type job shop-flexible*. USTHB.
- [14] P.LOPEZ.ET.F.ROUBELLAT. *Ordonnancement de la production*. Hermés Sciences Paris,Productique, 2001.
- [15] P.LOPEZ.ET.P.ESQUIROL. *L'Ordonnancement*. Econommica, 1999.
- [16] P.MIRIELLE. *Construction d'une métaheuristique au problème de projet à moyens limite*. laboratoire d'informatique,Avignon,France, 2001.
- [17] P.NORIGEON. *Cours processus de production*. université de Nice Shopia-Antipolis, 2008.
- [18] S.BOUKEROUCHA. *La recherche tabou pour la résolution d'un problème d'ordonnancement de type job shop*. UMBB, 2013.
- [19] S.BOULAKEROUN.ET.A.ABEDELLI. *Planification et Ordonnancement de la production au niveau de l'entreprise Cevital*. Mémoire de fin d'étude pour,Département de Recherche Opérationnelle Université de Bejaia, 2002.

- [20] T.JENSEN. *Robust and flexible scheduling with evolutionnary computation*. University of Arhes,Danmark, 2011.
- [21] V.T'KINDT.AND.J.BILLOUT. *Multicritéria scheduling theory,Models and Algorithm*e. H.Scott Edition,Springer-Verlag Berlin, 2006.
- [22] Y.HARRATH. *Contribution a l'ordonnancement conjointes de la production et de la maintenance : cas Job Shop*. Thèse doctorat Automatique et Robotique de l'Université de Franche-Comté, 2003.