

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira - Béjaia



Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire de fin de cycle
En vue de l'obtention du diplôme de Master
en Mathématiques Appliquées
Spécialité : Modélisation Mathématique et Techniques de Décision

Thème

Gestion de la production à l'entreprise Meriplast



Réalisé par :

Mr MAKHLOUFI Yacine et Mlle MEHANI Assia

Soutenu le 03 juillet 2017, devant le jury composé de :

| | | | |
|--------------------|------------------|-----------------|--------------|
| Mr N. KHIMOUM | MAA | Univ. de Béjaia | Président |
| Mr D. AISSANI | Professeur | Univ. de Béjaia | Promoteur |
| Mlle Z. AOUDIA | MAA | Univ. de Béjaia | Co-promoteur |
| Mr K. MEZIANI | Doctorant "LMD" | Univ. de Béjaia | Examinateur |
| Mlle A. OUTAMAZIRT | Doctorante "LMD" | Univ. de Béjaia | Examinatrice |

Année Universitaire : 2016/2017

Remerciements

On remercie Dieu le tout puissant de nous avoir accordé le courage et la volonté d'achever ce modeste travail.

Nous remercions vivement le professeur D. AISSANI pour l'honneur qu'il nous a fait en dirigeant ce présent mémoire et assurer son suivi scientifique, en plus de ses spécieux conseils et ses orientations. Nous tenons aussi, a remercier Mlle Z. AOUDIA d'avoir co-dirigé ce travail, et pour sa gentillesse, ses conseils et sa disponibilité.

Nous remercions Mr L. MIHOUBI chef du service méthode de l'entreprise MERIPLAST pour son accueil et ses explications. Et nous remercions aussi Mme N. BELAMRI responsable du service recherche et développement pour son aide, son écoute, sa disponibilité et aussi sa gentillesse.

Nous saluons aussi l'ensemble du service méthode, le service commercial, le personnel de la DRH et le service qualité pour la sympathie et leurs conseils, et tous ceux qui nous ont accueilli à Meriplast.

On remercie également Mr N. KHIMOUM d'avoir accepté de présider le jury, ainsi que Mr K. MEZIANI et Mlle A. OUTAMAZIRT pour avoir accepté d'examiner ce modeste travail.

Nous remercions nos chers parents pour leurs encouragements, leurs soutiens et leurs aides durant tout notre cursus et pour lesquels nous devons notre réussite.

Nos plus vifs remerciements sont aussi adressés à tous les étudiants de notre promotion et tous les enseignants du département de la Recherche Opérationnelle.

Dédicaces

Je tiens a dédier ce modeste travail :

A mes chers parents qui m'ont beaucoup aidé, soutenu et encouragé,

A mes frères et sœurs,

A mon *Binôme*, ainsi qu'à toute sa famille,

A mes oncles et leurs femmes,

A mes tantes et leurs maris,

A la mémoire de mes grand parents,

A toutes mes amies qui m'ont beaucoup aidé,

Assia

Dédicaces

Je tiens a dédié ce modeste travail :

A mes chers parents qui m'ont beaucoup aidé, soutenu et encouragé,

A mes sœurs,

A ma *Binôme*, ainsi qu'à toute sa famille,

A mes oncles et leurs femmes,

A mes tantes et leurs maris,

A la mémoire de mon grand père,

A tous mes amis qui m'ont beaucoup aidé,

A mes collègues de la Recherche Opérationnelle,

Yacine

Table des matières

| | |
|--|-------------|
| Table des matières | v |
| Table des figures | vii |
| Liste des tableaux | viii |
| Introduction générale | 1 |
| 1 Présentation de l'entreprise | 3 |
| 1.1 Présentation | 3 |
| 1.1.1 Historique | 3 |
| 1.1.2 La gamme de produits MERIPLAST | 4 |
| 1.1.3 Organigramme général | 4 |
| 1.2 Processus de production | 7 |
| 1.2.1 Equipement | 7 |
| 1.2.2 Les différents process de production | 7 |
| 1.3 Principe d'extrusion [24] | 8 |
| 1.4 Impression par flexographie | 9 |
| 1.5 Procédés de complexage | 10 |
| 1.6 Montée en cadence de la production | 10 |
| 1.7 Position du problème | 11 |
| 2 Notions de base sur l'optimisation combinatoire et l'ordonnancement | 12 |
| 2.1 Optimisation Combinatoire | 12 |
| 2.2 Complexité des problèmes d'optimisation combinatoire | 13 |
| 2.2.1 Classification des problèmes d'optimisation combinatoire | 13 |
| 2.3 Le problème d'ordonnancement | 14 |
| 2.4 Concepts de base en ordonnancement [20] | 14 |
| 2.4.1 Les tâches | 14 |
| 2.4.2 Les ressources | 15 |
| 2.4.3 Modélisation | 16 |
| 2.5 Ordonnancement en atelier | 18 |
| 2.5.1 Problème à une machine | 18 |
| 2.5.2 Problème à machines parallèles | 19 |
| 2.5.3 Atelier à cheminement unique (<i>flow shop</i>) | 19 |
| 2.5.4 Atelier à cheminement multiple (<i>job shop</i>) | 19 |
| 2.5.5 Atelier à cheminement libre (<i>Open shop</i>) | 20 |
| 2.6 Classification des problèmes d'ordonnancement [29] | 20 |

| | | |
|----------|--|-----------|
| 2.7 | Gestion de production | 21 |
| 2.7.1 | Les classes de décision en gestion de production | 21 |
| 2.7.2 | Les niveaux de décision en gestion de production | 22 |
| 2.8 | Méthodes de résolution [20] | 23 |
| 2.8.1 | Les méthodes exactes | 23 |
| 2.8.2 | Les méthodes approchées ou métaheuristiques | 25 |
| 2.9 | Conclusion | 32 |
| 3 | Modélisation du problème | 33 |
| 3.1 | L'approche générale | 33 |
| 3.2 | Collectes des données | 33 |
| 3.2.1 | Les jobs | 33 |
| 3.2.2 | Les opérations | 34 |
| 3.2.3 | Les différentes machines de chaque atelier | 35 |
| 3.3 | Modélisation du problème | 38 |
| 3.3.1 | Construction du modèle | 38 |
| 3.4 | Conclusion | 41 |
| 4 | Résolution du problème | 42 |
| 4.1 | Choix de la méthode | 42 |
| 4.2 | La méthode de Recherche Tabou | 42 |
| 4.2.1 | Algorithme de la recherche Tabou pour Job Shop Flexible | 43 |
| 4.3 | Application de l'algorithme génétique | 45 |
| 4.3.1 | L'algorithme génétique adapté au problème du Job Shop Flexible | 46 |
| 4.4 | Conclusion | 48 |
| 5 | Tests et comparaison des résultats | 49 |
| 5.1 | Présentation de Matlab | 49 |
| 5.2 | Résultats et tests | 49 |
| 5.2.1 | Données d'entrée | 50 |
| 5.2.2 | Les résultats obtenus par l'algorithme recherche Tabou | 50 |
| 5.2.3 | Les résultats obtenus par l'algorithme génétique | 51 |
| 5.2.4 | Comparaison des résultats | 53 |
| 5.3 | Evaluation des performances de la production | 53 |
| 5.3.1 | Production par 24H avec la recherche Tabou | 53 |
| 5.3.2 | Production réelle | 54 |
| 5.3.3 | Variation de la demande | 54 |
| 5.3.4 | Interprétation des résultats | 56 |
| 5.4 | Conclusion | 56 |
| | Conclusion générale | 57 |
| | Bibliographie | 59 |

Table des figures

| | | |
|-----|---|----|
| 1.1 | Organigramme de l'entreprise | 6 |
| 1.2 | Principe d'une ligne d'extrusion gonflage | 9 |
| 1.3 | Photo d'une ligne d'extrusion gonflage | 9 |
| 1.4 | L'évolution du chiffre d'affaire | 10 |
| 2.1 | Organigramme de classification des méthodes d'optimisation combinatoire | 25 |
| 2.2 | L'opérateur de croisement à 1 point | 27 |
| 2.3 | L'opérateur de croisement à 2 points | 27 |
| 2.4 | Opérateur de mutation dans le cas de codage binaire | 28 |
| 2.5 | Organigramme de l'algorithme génétique | 29 |
| 2.6 | Algorithme général de la méthode recherche tabou | 30 |
| 2.7 | Algorithme relatif au fonctionnement général du recuit simulé | 32 |
| 3.1 | Schéma du processus de production de MERIPLAST | 37 |

Liste des tableaux

| | | |
|-----|--|----|
| 2.1 | Quelques valeurs de α_1 | 20 |
| 2.2 | Quelques valeurs du champ β | 20 |
| 2.3 | Quelques valeurs du champ γ | 21 |
| 3.1 | Opérations de chaque job | 35 |
| 3.2 | Opérations s'effectuant sur chaque machine | 36 |
| 5.1 | Les données d'entrée | 50 |
| 5.2 | Les résultats de la recherche Tabou | 51 |
| 5.3 | Les résultats de l'algorithme génétique | 52 |
| 5.4 | Les résultats de la production avec la méthode recherche Tabou | 53 |
| 5.5 | Les pourcentages d'utilisation des machines | 54 |
| 5.6 | Augmentation de la demande de 10% | 55 |
| 5.7 | Augmentation de la demande de 25% | 55 |
| 5.8 | Augmentation de la demande de 40% | 56 |

Introduction générale

Depuis plusieurs années, le contexte de mondialisation et l'accroissement de la concurrence ont défini un nouvel ordre économique et industriel pour les entreprises de production de biens. En effet, ces entreprises sont quotidiennement confrontées à la maîtrise et à l'amélioration des performances de l'ensemble de leurs processus afin de garantir leur pérennité et leur compétitivité [4].

L'entreprise Meriplast, tout comme toute autre entreprise industrielle, tente de faire face à l'augmentation de la demande en conservant les critères de qualités des services fournis. Pour ce faire, elle tente d'appliquer les approches scientifiques et les méthodes de la recherche opérationnelle.

Au début de son lancement, l'entreprise disposait d'un seul site de production. Puis, après plusieurs années de labeurs, elle a fait l'acquisition d'un nouveau site de production qui lui permet d'augmenter considérablement sa production. La question qui se pose : est-ce que les capacités de production des deux unités peuvent faire face à l'augmentation de la demande ?.

L'organisation et la gestion de la production conditionnent le succès des projets du monde de l'entreprise et de la recherche. Dans ce processus, la fonction ordonnancement vise à organiser l'utilisation des ressources technologiques ou humaines pour répondre à une demande ou satisfaire un plan de production préparé par la fonction planification. Les problèmes d'ordonnancement apparaissent dans de nombreux domaines : l'industrie (atelier, gestion de production), la construction (suivi de projets), mais aussi l'informatique (gestion des processus) et l'administration (emplois du temps) [21].

Plusieurs travaux scientifiques concernant les problèmes d'ordonnancement considèrent ces derniers comme étant déterministes (les données du problème sont connues) et s'articulent essentiellement autour d'approches de résolution exactes ou bien approchées. De plus, la plupart de ces approches sont résolues de manière centralisée, et la solution proposée est donnée sous forme d'un ordonnancement prévisionnel unique satisfaisant les contraintes du système et ayant les performances attendues. Cependant, dans la réalité, l'environnement d'un système de production est de nature à la fois non déterministe et perturbé. En effet, les données du problème ne sont pas toujours connues à l'avance. De plus, de nouvelles informations peuvent surgir pendant l'exécution du plan initial qui devient par la même occasion obsolète lors de sa réalisation [6].

La résolution d'un problème d'ordonnancement passe par une phase d'identification et de modélisation et par une phase de recherche de la méthode de résolution adéquate. Dans la première phase, les contraintes à respecter et les critères à optimiser, lors de la résolution, sont identifiés et décrits [18].

Dans ce travail, nous évaluons les performances du système de production de l'entreprise Meriplast. Nous analysons la capacité de l'entreprise à faire face à l'augmentation de la demande. Pour ce faire, nous avons modélisé le système de production de l'entreprise sous forme d'un modèle mathématique qui est un ordonnancement de type Job Shop Flexible où nous avons proposé deux méthodes de résolution, à savoir les algorithmes génétiques et la recherche Tabou. Puis, nous procédons à l'évaluation du système de production en tenant compte des résultats obtenus par la recherche Tabou. Les résultats de l'évaluation montrent que le système de production de l'entreprise peut faire face à une augmentation de la demande allant jusqu'à 40%.

Ce mémoire est organisé sur cinq chapitres. Le premier chapitre est consacré à la présentation de l'entreprise ainsi qu'à la position du problème. Le second chapitre fournit des notions en optimisation combinatoire et en ordonnancement. Dans le troisième chapitre, nous modélisons le problème, où nous l'avons classé comme un problème de Job Shop Flexible. Le chapitre quatre porte sur les approches utilisées pour la résolution du problème à l'aide d'une adaptation des algorithmes génétiques et la recherche Tabou. Les tests et résultats ainsi que l'évaluation des performances du système de production sont résumés dans le cinquième chapitre. Enfin, nous terminons par une conclusion.

Chapitre 1

Présentation de l'entreprise

La SARL MERIPLAST (de : Meridji Plastique) est spécialisée dans le développement, la fabrication et la commercialisation d'emballages flexibles : films thermo-rétractables neutres et imprimés, films complexés, films agricoles, gaines, housses et sacherie.

En activité depuis 1987, la société offre une large gamme de produits destinés à une clientèle dans divers secteurs d'activité tels que l'agroalimentaire, les matériaux de construction, l'agriculture, le textile,...

1.1 Présentation

MERIPLAST est une société de droit algérien, créée le 13 décembre 1999. Elle est constituée juridiquement en SARL (Société à Responsabilité Limitée), au capital social de **454.130.000,00 DZD**, dont les parts sociales sont détenues pour 88.8% par Mr Abdelhakim MERIDJI, gérant de la société.

MERIPLAST dispose de deux unités de production :

- Le site 1 se situe sur la route de l'arrière port de Béjaia.
- Le site 2 se situe à proximité de l'entrée de la ville de Béjaia (Oued-Ghir).

1.1.1 Historique

MERIPLAST est une entreprise familiale en activité depuis 1987, spécialisée dans la fabrication de films en plastiques, installée sur un site de production de $2700 m^2$. Elle dispose de plusieurs lignes de production entièrement automatisées ; d'une capacité de production de 2000 tonnes par mois.

Dans le cadre de son développement, un deuxième site a été réalisé au niveau de la zone d'activité d'Ibourassen, commune de Oued-Ghir sur une superficie de $1600 m^2$. Ce qui permettra à MERIPLAST d'augmenter ces capacités de production à 3500 tonnes par mois.

En septembre 2015, l'entreprise MERIPLAST a été certifiée ISO 9001/2008.

1.1.2 La gamme de produits MERIPLAST

La gamme de produits MERIPLAST est constituée actuellement de :

- Film Thermo rétractable Neutre (FTN),
- Film Thermo rétractable Imprimé (FTI),
- Film Etirable (FE),
- Film Agricole Stabilisé trois saisons (FAS),
- Film Poly Ethylène tri couches Neutre et Imprimé (FPEtN/FPEtI),
- Gaine plastique,
- Housse thermo rétractable et étirable (HN/HI),
- Gaine cover pall,
- Sacherie Neutre, Imprimée et personnalisée toutes dimensions (SN/SI),
- Film Complexe Neutre et Imprimé (FCN/FCI).

1.1.3 Organigramme général

MERIPLAST emploie 241 agents au 31/03/2017, répartis par catégorie socioprofessionnelle comme suit :

- Cadres : 29
- Agents de maîtrise : 76
- Exécution : 136

1. Direction Générale

Dirigée par un directeur général qui assure et applique les décisions des différents conseils administration, coordonne les travaux entre les différents services.

2. Audit Contrôle Gestion

- Audit.
- Coût et prix de revient.
- Contrôle mouvement des stocks.

3. Recherche et Développement

- Développement des nouveaux produits.

4. Direction des Ressources Humaines (DRH)

- Administration et paie.
- Recrutement, formation et Gestion des Carrières.
- Moyens généraux.

5. Direction Finance et Comptabilité (DFC)

- Comptabilité générale.
- Trésorerie.
- Comptabilité matières.

6. Production

- Méthodes O/L.
- Extrusion.
- Impression.
- Finition.

7. Maintenance

- Méthodes maintenance.
- Maintenance préventive.
- Maintenance curative.

8. Management Qualité

- Suivi des procédures de gestion suivant les normes.

9. Contrôle Qualité

- Soutien à la production.
- Contrôle libérateur.
- Traitement Non-Conformité.
- Développement nouveaux produits.

10. Approvisionnements Gestion Stocks

- Achats étrangers.
- Achats locaux.
- Magasinage.
- Suivi des mouvements des stocks.
- Transfert matières.

11. Commercial

- Administration ventes.
- Gestion clientèle.
- Gestion des commandes.

12. Assistante Direction

13. Réseau Informatique

- Assurer le bon fonctionnement du réseau.
- Remédier au différents problèmes du réseau interne.

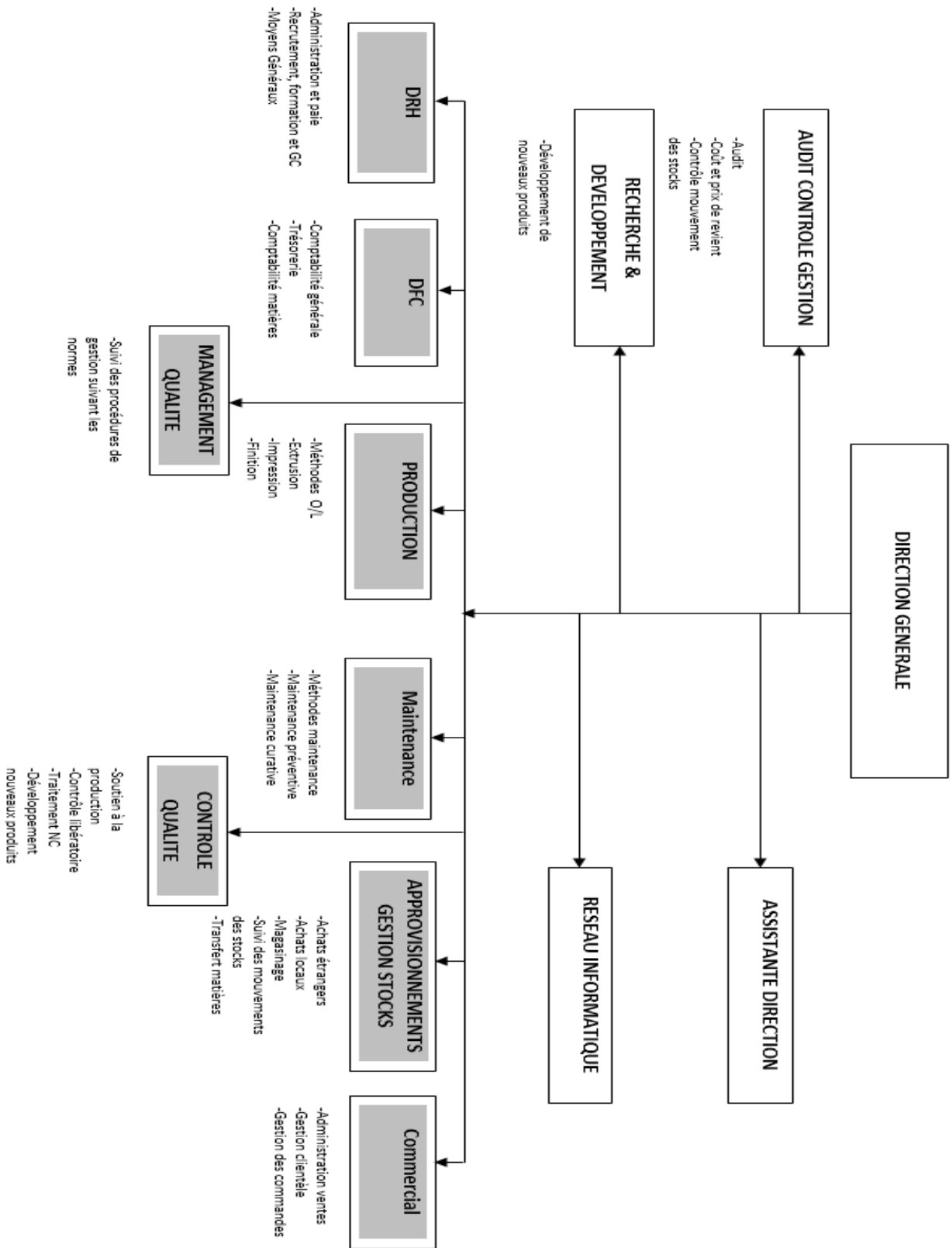


FIGURE 1.1 – Organigramme de l'entreprise

1.2 Processus de production

MERIPLAST travaille 24h/24h, avec des lignes de production de dernières technologies, qui lui assurent des capacités de production frôlant 2500 tonnes/mois, de films plastiques, toutes qualités confondues, et ce, au choix du client pour la laize (largeur) et l'épaisseur.

1.2.1 Equipement

Pour atteindre ses objectifs, MERIPLAST dispose de différentes machines réparties sur les deux sites comme suit :

Site 1 :

- 04 Extrudeuses monocouche.
- 01 Extrudeuse tricouche.
- 01 Extrudeuse OPTIMEX (dernière génération).
- 01 Imprimante flexographie avec une unité hélios (08) couleurs.
- 02 Découpeuses refendeuses.
- 01 Soudeuse.

Site 2 :

- 01 Extrudeuse MACCHI (dernière génération).
- 01 Imprimante flexographie avec une unité hélios (08) couleurs.
- 01 Contre-colleuse (Complexeuse).
- 01 Découpeuse refendeuse.

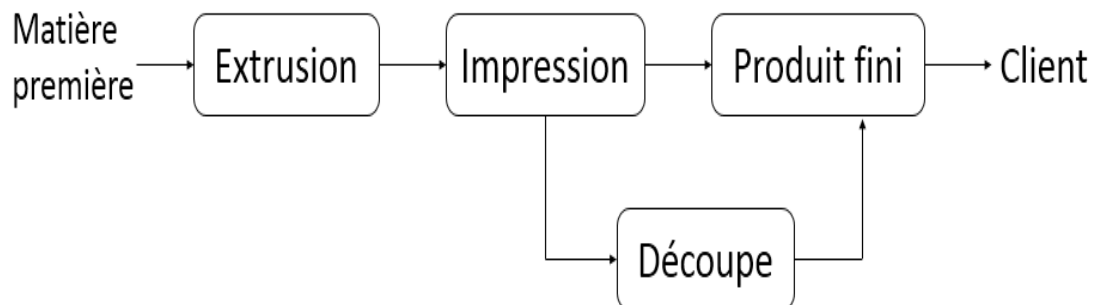
1.2.2 Les différents process de production

Selon la nature du film plastique, le processus de production passe par les différentes étapes principales, à savoir l'extrusion, l'impression, le soudage, le complexage et la découpe.

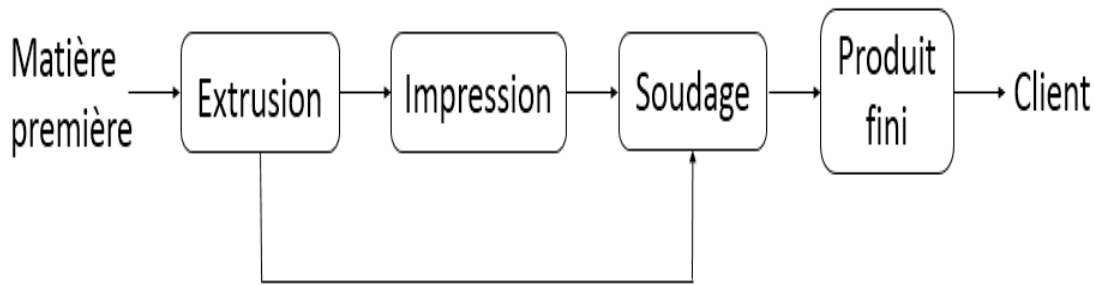
Process 1 : Films neutres



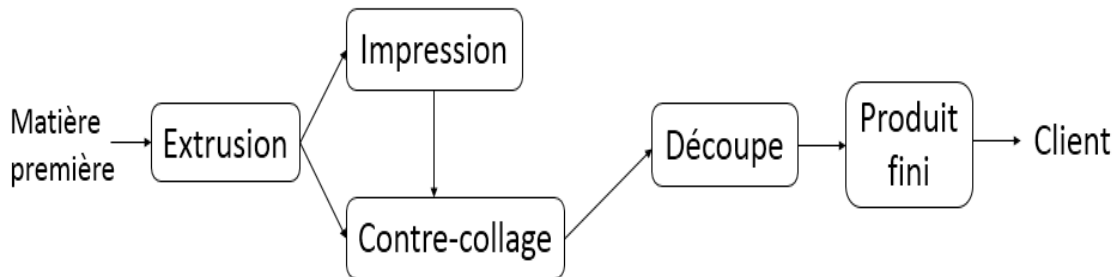
Process 2 : Films imprimés



Process 3 : Sacherie (neutre/imprimé)



Process 4 : Films complexes (neutre/imprimé)



1.3 Principe d'extrusion [24]

L'extrusion gonflage est un des procédés permettant la fabrication de films polymère. Les matières polymères transformées par ce mode d'extrusion sont essentiellement le polyéthylène (90%), le polypropylène, le PVC souple et le polyamide. Cette technique est utilisée en sachetterie car elle peut être implémentée directement sur la chaîne de conditionnement automatique. Le principe de fonctionnement d'une ligne d'extrusion gonflage est présenté en figure 1.2 et une ligne réelle illustrée en figure 1.3.

Les filières sont, dans les installations de soufflage de gaines, positionnées verticalement, ce pour assurer une action homogène de la pesanteur sur le film très fin finalement obtenu. La sortie du polymère fondu se fait vers le haut car il va falloir beaucoup de place pour souffler et refroidir la gaine, ce qui est plus facilement gérable (et à un coût moindre) avec de hauts plafonds qu'en creusant le sol. Le polymère fondu est extrudé en continu à travers une filière annulaire donnant un "tube" épais. Cette matière encore fondue va subir deux déformations simultanées.

La première est due aux rouleaux d'entraînements situés en haut de la ligne de gonflage. En entraînant la gaine, ils l'étirent verticalement. Les rouleaux ont un second rôle, celui de refermer le "tube" sur lui-même : l'espace entre les deux rouleaux est si faible qu'il permet juste au film cylindrique de passer une fois replié sur lui-même. Du coup, le film y est refermé de façon quasi

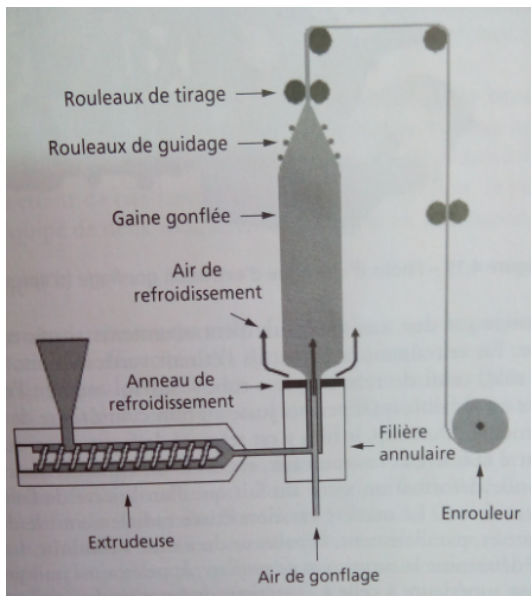


FIGURE 1.2 – Principe d'une ligne d'extrusion gonflage



FIGURE 1.3 – Photo d'une ligne d'extrusion gonflage

étanche. L'intérieur du film, entre la filière et ces rouleaux, est donc un espace fermé. La seconde déformation vient du fait que l'on injecte de l'air sous pression dans cet espace fermé. La matière est alors étirée radialement : le diamètre de la poche augmente et, parallèlement, l'épaisseur du "tube" diminue. La quantité d'air ainsi insufflé détermine la surpression de gonflage, appelée ainsi puisque la pression dans la poche est supérieure à celle à l'extérieur de cette poche (qui est égale à la pression atmosphérique). On l'augmente progressivement jusqu'à ce que le film présente l'épaisseur requise.

Le tube extrudé subit donc un biétirage. Il devient une gaine très fine de forme galbée que l'on appelle aussi une bulle. Ses caractéristiques dimensionnelles vont dépendre des dimensions initiales du tube extrudé, de la vitesse de rotation des rouleaux d'entraînement ainsi que de la surpression de gonflage.

Le refroidissement de la gaine s'effectue généralement par soufflage d'air au niveau d'un anneau extérieur, situé au niveau ou au-dessus de la filière. La gaine se refroidit donc progressivement lors de sa progression vers le haut. Une fois la gaine solidifiée, sa forme n'évolue plus ; elle est entraînée, repliée sur elle-même (par une tireuse de mise à plat à rouleaux) et bobinée (dans le poste d'enroulement).

1.4 Impression par flexographie

C'est un procédé d'impression directe par une forme imprimante en relief, mais dans ce procédé, celle-ci est "flexible", réalisée en caoutchouc vulcanisé ou en matière plastique souple. Ce procédé, particulièrement adapté à l'impression sur pellicules plastiques destinées à l'emballage, permet à l'aide de rotatives à bobines, de réaliser des impressions de qualité en couleur, dans la mesure où les documents originaux ont été conçus pour ce procédé. Longtemps réservée à des impressions grossières sur cartons ou papiers Kraft, la flexographie doit son essor aux dé-

veloppements technologiques conjugués des machines rotatives, des plaques photopolymères, et des encres que les chimistes adaptent au fur et à mesure que l'on invente de nouveaux supports synthétiques d'emballage [2].

1.5 Procédés de complexage

Les procédés de complexage (ou lamination) permettent de réunir 2 couches et plus de matériaux en un seul. Il y a 2 principaux procédés de complexage :

- Lamination à froid.
- Lamination à chaud.

Cette entreprise adopte la lamination à froid, qui est adaptée lorsque l'un des composants est sensible aux températures élevées. En premier, un des films est enduit d'une fine couche de colle à froid ou température modérée. En appliquant une pression entre le film enduit et le film à complexer, on obtient un seul et même matériau épais. Les colles utilisées sont en fonction des produits à complexer à base de solvants ou à l'eau. Les colles sont généralement activées dans un tunnel de séchage thermique ou à UV [1].

1.6 Montée en cadence de la production

Depuis l'entrée en exploitation en l'an 2000, la production a connu une montée en cadence régulière pour atteindre 1500 tonnes par mois.

En parallèle, les parts du marché de MERIPLAST connaissent un développement croissant et son chiffre d'affaire progresse régulièrement.

| Année | 2010 | 2011 | 2012 | 2013 | 2014 |
|-------------------------------------|--------|------|--------|--------|--------|
| Chiffre d'Affaire (Millions de DZD) | 1616,4 | 2211 | 2431,2 | 2897,9 | 3264,6 |

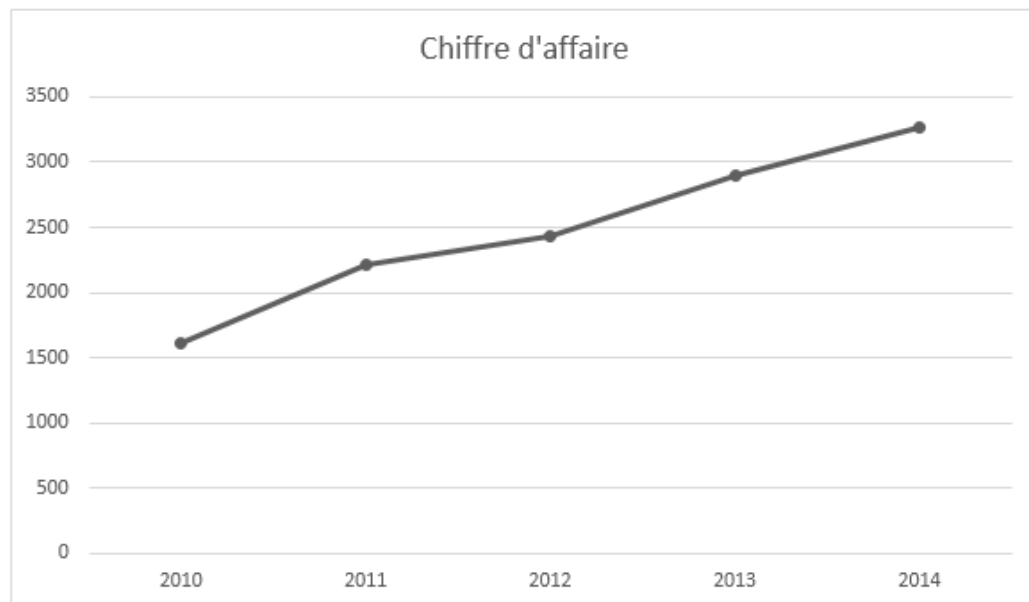


FIGURE 1.4 – L'évolution du chiffre d'affaire

1.7 Position du problème

L'objectif des dirigeants de MERIPLAST est d'améliorer la productivité et la qualité des produits finis, afin de satisfaire ses clients en respectant les délais de livraison et en leur proposant des produits de meilleure qualité à des coûts plus raisonnables. En effet, dans le domaine industriel, les clients deviennent de plus en plus exigeants. Ce qui incite les entreprises à être compétitives, pour gagner des parts de marché en satisfaisant au mieux la clientèle et ce, en trouvant un compromis entre les coûts, la qualité et surtout les délais de livraison des produits finis.

La question qui se pose est la suivante : est-ce que les capacités de production des deux unités de Béjaia et Oued Ghir peuvent faire face à l'augmentation de la demande ?. Dans un premier temps, nous nous intéressons à la gestion de la production du système actuel. Puis, nous aborderons l'évaluation des performances dans le cas où la demande augmente.

L'ordonnancement est une fonction clé dans la gestion de la production. Elle vise à planifier dans le temps et d'organiser l'utilisation des ressources présentes dans les ateliers de production de l'entreprise afin d'atteindre les objectifs tracés auparavant (qualité, coût et délais).

Le bureau méthode O/L, qui est rattaché à la direction production, organise le plan de production quotidiennement en fonction des commandes exprimées par la direction générale et la direction commerciale. Son rôle est d'organiser l'ensemble des travaux sur les différentes machines disponibles, sachant que certains travaux nécessitent le passage par une seule machine alors que d'autres nécessitent deux, voire trois machines. Donc, on est en présence d'un problème d'ordonnancement de type Job Shop Flexible.

Pour la résolution de tels problèmes (de classe NP-Difficile), on ne dispose pas de méthodes exactes susceptibles d'exploiter en un temps raisonnable l'espace des solutions réalisables. D'où la nécessité de faire appel aux métaheuristiques (algorithmes génétiques, recuit simulé, recherche Tabou, ...).

Chapitre 2

Notions de base sur l'optimisation combinatoire et l'ordonnancement

La fonction ordonnancement vise à organiser l'utilisation des ressources technologiques et humaines présentes dans les ateliers ou les services de l'entreprise pour satisfaire soit directement les demandes des clients, soit les demandes issues d'un plan de production préparé par la fonction de planification de l'entreprise.

L'optimisation combinatoire définit un cadre formel pour de nombreux problèmes de l'industrie. Elle permet de modéliser, d'analyser et de résoudre les problèmes consistant à déterminer les solutions satisfaisant un objectif en respectant les contraintes.

2.1 Optimisation Combinatoire

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données [16].

Definition 2.1.1. [16]

Une instance I d'un problème de minimisation (respectivement maximisation) est un couple (X, f) où $X \subseteq S$ est un ensemble fini de solutions admissibles et f une fonction à minimiser

$$f : X \rightarrow \mathbb{R}$$

Le problème considéré consiste de trouver $s^ \in X$ et $f(s^*) \leq f(s)$, (respectivement $f(s^*) \geq f(s)$), $\forall s \in X$.*

On cite comme principaux problèmes d'optimisation combinatoire [3] :

- problème du sac à dos,
- problème du plus court chemin,
- problème d'affectation,
- problème du voyageur de commerce,
- problème de localisation des dépôts,
- problème du postier,
- problème de l'arbre de poids minimum,
- problème de mise en boîtes,
- problème d'ordonnancement (ordonnancement de projet, d'atelier, de la production).

2.2 Complexité des problèmes d'optimisation combinatoire

Les premières applications scientifiques des ordinateurs portaient sur des données de taille réduite ; il n'était pas donc de se préoccuper du nombre d'opérations des algorithmes. Depuis, le champ d'application s'est considérablement élargi : on cherche à résoudre des problèmes hautement combinatoire en recherche opérationnelle (emploi du temps, problèmes de tournées, ordonnancement,...). En effet, on pourrait penser que la complexité des algorithmes ne peut pas être évaluée indépendamment de l'ordinateur [8].

La durée d'exécution est une mesure imparfaite pour les algorithmes, car elle dépend trop de la machine, du langage de programmation et des compilateurs utilisés (constructeur, version, option de compilation,...), et elle dépend aussi de la taille des données [8].

Definition 2.2.1. [28]

*Un algorithme est dit **polynomial**, si sa complexité temporelle (dans le pire des cas) est en $O(n^p)$ pour un certain p .*

2.2.1 Classification des problèmes d'optimisation combinatoire

Les problèmes peuvent être classés selon les propriétés de l'ensemble des solutions [17] :

Un problème de décision :

C'est un problème dont la réponse est tout simplement OUI ou NON.

Un problème polynomiale réductible :

On a P1 et P2 deux problèmes de décision, on dit que P1 est polynomiale réductible à P2 s'il existe un algorithme polynomiale qui convertit chaque instance d'entrée de P1 à une autre instance de P2.

Un problème de classe P :

Un problème est dit polynomial, s'il existe un algorithme de complexité polynomiale pour le résoudre. On dit que les problèmes de classe P sont faciles.

Un problème de classe NP :

Un problème appartient à la classe NP, s'il peut être résolu en temps polynomial par un algorithme non déterministe.

Un problème de la classe NP-complet :

Un problème est NP-complet, si tout problème NP s'y réduit en temps polynomial. Autrement, les problèmes NP les plus difficiles sont NP-complet.

Un problème de la classe NP-difficile :

Un problème d'optimisation est NP-difficile si le problème de décision qui lui correspond est NP-complet. On dit qu'un problème est dans la classe NP-difficile, si chaque autre problème dans NP est réductible d'une manière polynomiale dans ce dernier.

2.3 Le problème d'ordonnancement

L'ordonnancement est la programmation dans le temps de l'exécution d'une série de tâches (ou activités, opérations) sur un ensemble de ressources physiques (humaines et techniques), cherchant à optimiser certains critères, financiers ou technologiques, et en respectant les contraintes de fabrication et d'organisation [15].

Un ordonnancement constitue une solution au problème d'ordonnancement. Il décrit l'exécution des tâches et l'allocation des ressources au cours du temps et vise à satisfaire un ou plusieurs objectifs. Plus précisément on parle de problème d'ordonnancement lorsqu'on doit déterminer dans le temps les dates du début et de la fin des tâches, alors qu'on réserve le terme de problème de séquençement au cas où l'on cherche seulement à fixer un ordre d'exécution des tâches qui peuvent être en conflit pour l'utilisation des ressources. Un ordonnancement induit nécessairement un ensemble unique de relations de séquençement. Par contre détermination d'un séquençement des tâches recouvre toute une famille d'ordonnements (éventuellement une infinité si le domaine de variation des dates est non borné ou réel) [8].

2.4 Concepts de base en ordonnancement [20]

2.4.1 Les tâches

Une tâche est une entité élémentaire de travail localisée dans le temps par une date de début t_i ou de fin c_i , dont la réalisation est caractérisée par une durée p_i (on a $c_i = t_i + p_i$) et par l'intensité α_i^k avec laquelle elle consomme certains moyens k , ou ressources.

Dans certains problèmes, les tâches peuvent être subdivisées en sous tâches pour laisser le moins possibles les ressources inactives ; dans d'autres au contraire, on ne peut subdiviser une tâche et on ne peut l'interrompre une fois commencée. On parle alors respectivement de problèmes préemptifs et non préemptifs.

En ordonnancement d'atelier on préférera de remplacer le terme tâche par opération. En production manufacturière, on distingue souvent plusieurs phases dans l'exécution des opérations : préparation, phase principale, finition, transport. Ces dernières peuvent fortement influencer - selon les conditions de fabrication - sur le temps de séjour d'un produit dans un atelier.

2.4.2 Les ressources

Une ressource k est un moyen technique ou humain requis pour la réalisation d'une tâche et disponible en quantité limitée, sa capacité A_k (supposée connue). On distingue plusieurs types de ressources :

A- Ressources renouvelables

Une ressource est renouvelable si après avoir été utilisée par une ou plusieurs tâche, elle est à nouveau disponible en même quantité (les hommes, les machines, l'espace, l'équipement en général, etc.) ; la quantité de ressource utilisée à chaque instant est limitée.

B- Ressources consommables

Une ressource est dite consommable si elle n'est plus disponible en même quantité après avoir été utilisée par une ou plusieurs tâche (matière première, budget, etc.) ; la consommation globale (ou cumul) au cours du temps est limité.

C- Ressources doublement contrainte

Une ressource est doublement contrainte lorsque son utilisation instantanée et sa consommation globale sont toutes deux limitées (source d'énergie, financement, etc.).

D- Ressources disjonctives

Une ressource est dite disjonctive (ou non partageable) si elle ne peut exécuter qu'une seule tâche à la fois (machine-outil, robot manipulateur).

E- Ressources cumulatives

Une ressource est dite cumulative (ou partageable) si elle peut être utilisée par plusieurs tâche simultanément (équipe d'ouvriers, poste de travail).

2.4.3 Modélisation

La modélisation consiste à représenter les principaux aspects de la réalité par un ensemble de formules, le plus souvent mathématiques, qui mettent en jeu les variables de décisions concernées et leurs interactions.

Dans un problème d'ordonnancement, et d'une manière générale, les variables de décisions que l'on rencontrera concernent des décisions sur le temps ou sur les ressources. Une contrainte exprime des restrictions sur les valeurs que peuvent prendre conjointement les variables de décision. On distingue les contraintes temporelles et les contraintes de ressources.

1- Contraintes temporelles

Elles englobent :

- Les contraintes liées au temps alloué, issues généralement d'impératifs de gestion et relatives aux dates limites des tâches (délais de livraison, disponibilités des approvisionnements) ou à la durée totale d'un projet ; pour une tâche on notera r_i sa date de début (avant la quelle elle ne peut commencer) et d_i sa date de fin (avant la quelle elle doit être achevée).
- Les contraintes de succession ou d'antériorité et plus généralement les contraintes de cohérence technologique, qui décrivent le positionnement relatif de certaines tâches par rapport à d'autres (par exemple : contraintes de gammes dans le cas de problème d'atelier).
- Les contraintes de calendriers liées aux respects d'horaires de travail, etc.

Ces contraintes peuvent toutes s'exprimer à l'aide d'inégalités de potentiels, qui imposent une distance minimale b_{ij} entre deux instants particuliers associés aux tâches (le plus souvent les dates de début) :

$$x_i - x_j \geq b_{ij}$$

Cette contrainte primitive peut être utilisée lors de l'expression de quelques-unes des contraintes temporelles présentées plus haut.

a- Contraintes exprimables à l'aide d'une inégalité de potentiel :

La contrainte de succession entre deux tâches i et j , notée symboliquement $i < j$ (i précède j), est représentable par la relation numérique $t_j - t_i \geq p_i$.

Une contrainte de date limite peut aussi s'exprimer à l'aide de la contrainte primitive, en introduisant une tâche fictive 0 de durée nulle, dont le début fixe l'origine des temps ($t_0 = 0$) ; par exemple, la contrainte $t_i \geq r_i$ se réécrit sous la forme $t_i - t_0 \geq r_i$; pour la contrainte $c_i \leq d_i$, on obtient la forme $t_0 - c_i \geq -d_i$.

b- Contraintes exprimables à l'aide d'une conjonction d'inégalités de potentiels :

C'est le cas des contraintes de localisation relatives telle que le chevauchement entre deux tâches i et j . Il s'exprime par une contrainte nécessitant deux inégalités de potentiels : $(c_j - t_i \geq 0) \wedge (c_i - t_j \geq 0)$. Ce type d'expression logique conjonctive est utilisé pour exprimer qu'une contrainte temporelle est satisfaite si toutes les contraintes membre qui la composent sont simultanément satisfaites.

c- Contraintes exprimables à l'aide d'une disjonction d'inégalités de potentiels :

Ce type d'expression fait intervenir le connecteur \vee (ou logique). La contrainte est satisfaite si au moins un des membres de la disjonction est une contrainte satisfaite. Soit l'exemple suivant : Soit un produit pouvant se réaliser selon deux gammes opératoires différentes :

- Gamme 1 : opération i puis l'opération j .
- Gamme 2 : opération j , attente minimale de durée p_{att} (pour refroidissement, séchage de la pièce, etc.), puis opération i .

La contrainte temporelle globale qui traduit l'existence de ces deux gammes peut s'écrire sous la forme : $(t_j - c_i \geq 0) \vee (t_i - c_j \geq p_{att})$.

2- Contraintes de ressources et problème de séquençement

Ces contraintes traduisent le fait que les ressources sont disponibles en quantité limitée (leur capacité) ; on parle également de contraintes de partage. Soit, pour un ordonnancement donné, l'ensemble des tâches qui consomment la ressource k à l'instant t :

$T_k(t) = \{i \in 1, \dots, n / t \in [t_i, t_i + p_i]\}$ (on peut noter que les tâches qui se terminent juste à l'instant t , ne sont pas comptées dans $T_k(t)$). La limitation de la capacité de la ressource k se traduit par la contrainte suivante :

$$\forall t, \sum_{i \in T_k(t)} \alpha_i \leq A_k$$

Pour satisfaire les contraintes de ressources, on doit éviter le chevauchement temporel de certains sous-ensembles de tâches, d'où la nécessité d'ordonner un nombre suffisant de paires de tâches, donc à ajouter de nouvelles contraintes de succession à l'ensemble initial de contraintes. Ce type de décision est appelé par la suite décision de séquençement ou arbitrage. L'ensemble de décisions de séquençement caractérise le problème de séquençement induit par le problème d'ordonnement initial. On distingue deux types de contraintes de ressources, liées à la nature disjonctives ou cumulatives des ressources.

Ressources disjonctives

Il s'agit des ressources qui ne peuvent être utilisées que par une tâche à la fois. Dans un problème d'atelier, il s'agit des machines. A tout couple de tâche (i, j) utilisant la même ressource, est associée la paire de disjonction $(i < j) \vee (j < i)$ qui se traduit sous la forme d'une

disjonction entre deux inégalités de potentiels :

$$(t_j - t_i \geq p_i) \vee (t_i - t_j \geq p_j)$$

Pour satisfaire ces contraintes, il est nécessaire et suffisant de résoudre toutes les paires de disjonction relatives à chaque ressource disjonctive, ce qui conduit à un séquençement total des tâches qui utilisent une même ressource disjonctive.

Ressources cumulatives

Lorsque la capacité et les intensités sont quelconques, les ensembles de tâches non réalisables simultanément sont de cardinalité quelconque. Le respect des contraintes de ressources ne conduit qu'à un séquençement partiel des tâches utilisant la même ressource.

3- Objectifs et critères d'évaluation

Lorsque l'on aborde la résolution d'un problème d'ordonnement, deux cas se présentent, soit on vise à trouver des solutions optimales par rapport à un ou plusieurs critères, soit on se contente de trouver des solutions admissibles vis-à-vis des contraintes. L'approche par optimisation suppose que les solutions candidates à un problème puissent être ordonnées de manière rationnelle selon un ou plusieurs critères d'évaluation numériques permettant d'apprécier la qualité des solutions. On cherchera à minimiser ou maximiser de tels critères. On note par exemple ceux :

- Liés aux ressources : la quantité-maximale, moyenne pondérée de ressources nécessaires pour réaliser un ensemble de tâches ; la charge de chaque ressource.
- Liés aux coûts de lancement, de production, de transport, de stockage, etc.
- Liés au temps : la minimisation du temps total d'exécution (makespan) qui revient à maximiser la productivité, la minimisation du temps total de séjour qui est équivalent à minimiser le temps d'attente et les encours, et enfin la minimisation de la somme des retards qui est un critère particulièrement important pour le respect des délais. Certaines caractéristiques particulières mais fréquentes dans les entreprises telles que les temps de changement d'outils ou des dates d'arrivées différentes des tâches sont aussi d'autres critères d'évaluation.

2.5 Ordonnement en atelier

Suivant la manière dont la gamme d'un produit est organisée, plusieurs types de problèmes d'ateliers apparaissent.

2.5.1 Problème à une machine

L'étude des environnements à machine unique peut apparaître comme un cas d'école. Ceux-ci peuvent toutefois servir de cadre de base pour élaborer des raisonnements multi machines,

plus réalistes mais plus complexes. En pratique en effet, il peut s'avérer que les difficultés d'ordonnancement d'un atelier proviennent d'une machine dominante sur laquelle se rencontre les conflits, les attentes... etc.

Généralement, les hypothèses propres au problème à une machine sont :

- $m = 1$ et la machine est toujours disponible.
- Les " n " travaux comprennent chacun une seule opération à laquelle ils ont assimilés.
- Initialement, il n'existe pas de contraintes de précédence entre les différentes opérations.
- Les temps de préparation sont indépendants de la séquence des opérations et sont inclus dans leurs durées [30].

2.5.2 Problème à machines parallèles

Le problème à machine parallèles se caractérise par le fait que plusieurs machines sont possibles pour l'exécution d'un travail qui n'en nécessite qu'une. D'un point de vue théorique, ce problème est une généralisation du problème à une machine et un cas particulier de problème d'atelier multi machine. Pratiquement, c'est un cas fréquemment rencontré dans des applications réelles, en particulier, dans l'ordonnancement de traitements informatique [13].

2.5.3 Atelier à cheminement unique (*flow shop*)

Dans le cas du flow shop, tout travail visite chaque machine de l'atelier et l'ordre de passage de chaque travail sur les différentes machines est le même pour tous les travaux (flux unidirectionnel). Cet ordre, ou gamme, unique est donnée du problème. Cette particularité se rencontre très fréquemment en pratique, elle correspond par exemple à une chaîne de traitement ou de montage [20].

2.5.4 Atelier à cheminement multiple (*job shop*)

Un problème de type job shop est une modélisation d'une unité de production disposant de moyen polyvalents utilisés suivant des séquences différentes en fonction des produits. L'objectif consiste à ordonner la réalisation des produits de manière à minimiser le coût global, en respectant un certain nombre de contraintes sur les machines utilisées pour effectuer chaque opération élémentaire entrant dans la fabrication des différents produits [11].

Job shop flexible

Le problème du job shop flexible représente une généralisation du problème job shop présenté précédemment, du fait qu'une opération donnée peut être réalisée une ou plusieurs ressources et possède une durée de traitement dépendant de la ressource utilisée. L'ensemble des contraintes de ce problème est aussi composé des contraintes de précédence, contraintes temporelles et contraintes disjonctives. Une solution au problème Job Shop flexible ne consiste plus uniquement à trouver une séquence des opérations sur chacune des ressources et à leur fixer une date de début mais également à déterminer une affectation de chaque opération à l'une de ses ressources potentielles de telle sorte que la durée totale de l'ordonnancement est minimisée. Ce problème est NP-difficile puisque le problème classique job shop est aussi NP-difficile [12].

2.5.5 Atelier à cheminement libre (*Open shop*)

Ce type d'atelier est moins contraint que celui de type flow-shop ou de type job-shop. Ainsi, l'ordre des opérations n'est pas fixé a priori ; le problème d'ordonnement consiste, d'une part, à déterminer le cheminement de chaque produit et, d'autre part, à ordonner les produits en tenant compte des gammes trouvées, ces deux problèmes pouvant être résolus simultanément. Comparé aux autres modèles d'ateliers, l'open-shop n'est pas couramment utilisé dans les entreprises [25].

2.6 Classification des problèmes d'ordonnement [29]

La notation $\alpha/\beta/\gamma$ permet de caractériser un problème d'ordonnement de manière précise. Le champ α décrit la structure du problème et se décompose généralement en deux sous champs α_1 et α_2 . Le sous champs α_1 indique la nature du problème (job shop, flow shop, etc.) et le sous champs α_2 précise le nombre de machines. Le champ β décrit les types de contraintes prises en compte. Enfin, le champ γ indique la fonction objectif considérée.

À titre indicatif, quelques valeurs classiques de α_1 , β et γ sont décrites dans les tableaux suivants :

| Valeur | Description |
|--------|-------------------------------|
| ϕ | machine unique |
| P | machine parallèles identiques |
| F | flow shop |
| J | job shop |
| O | open shop |
| FH | flow shop hybride |
| JG | job shop généralisé |
| OG | open shop généralisé |

TABLE 2.1 – Quelques valeurs de α_1

| Valeur | Description |
|---------|--|
| $prec$ | il existe des contraintes de précédence générale entre les opérations |
| r_i | une date de début au plus tôt r_i est associée à chaque travail i |
| d_i | une date d'échéance préférentielle d_i est associée à chaque travail i |
| $p = 1$ | les durées opératoires sont unitaires |
| $pmtn$ | la préemption des opérations est autorisée |

TABLE 2.2 – Quelques valeurs du champ β

| Valeur | Description |
|------------------|--|
| C_{max} | la durée totale (makespan) |
| L_{max} | le plus grand retard algébrique |
| T_{max} | le plus grand retard vrai |
| ΣU_i | le nombre de travaux en retard |
| $\Sigma[w_i]C_i$ | la durée moyenne ou pondérée des travaux |

TABLE 2.3 – Quelques valeurs du champ γ

2.7 Gestion de production

Pour survivre et garder leur place dans un marché toujours en pleine extension ne s'offrent qu'aux meilleurs, les entreprises doivent s'adapter et faire évoluer leurs modes de gestion (gestion des connaissances, ingénierie simultanée, prise en compte de la chaîne logistique, automatisation et informatisation...). L'automatisation et l'informatisation ont permis d'améliorer certaines fonctions de l'entreprise : automatisation des machines, informatisation de la comptabilité, de la conception... Certaines fonctions, comme la gestion de production, restent encore insuffisamment informatisées [14].

En effet, les problèmes à résoudre en gestion de production sont des problèmes d'optimisation très difficiles et les contraintes à respecter sont nombreuses : respect des délais, minimisation des stocks et encours, minimisation des coûts,...

Le développement de l'informatique a mis, à la disposition des entreprises, des outils d'aide à la décision, à savoir les modules MRP (Material Requirement Planning) permettant de calculer des besoins en composants, et la simulation des flux permettant de tester les scénarios choisis par l'expert planificateur. Le point faible de ces outils réside dans le fait qu'ils ne peuvent pas proposer des solutions aux problèmes qui se posent au décideur.

La gestion de la production est la fonction de gestion qui a pour objectif d'assurer le respect des délais, l'optimisation de l'utilisation des ressources et la minimisation des coûts, par le biais de la planification et de l'ordonnement dans le cadre de la stratégie de l'entreprise.

2.7.1 Les classes de décision en gestion de production

Pour situer les différents problèmes rencontrés en gestion de production, on peut classer les décisions de la gestion en trois classes :

Les décisions stratégiques

Il s'agit de la formulation de la politique à long terme pour l'entreprise, parmi ces décisions on cite : définition du portefeuille, décisions d'investissement, de licenciement,.... Ce genre de décisions se prennent au niveau de toute l'entreprise et par la direction générale.

Les décisions tactiques

Ce sont les décisions à moyen terme parmi lesquelles on trouve la planification de la production sur 18 mois. Ces décisions se prennent au niveau de l'usine et par les cadres.

Les décisions opérationnelles

Elles regroupent les décisions de gestion quotidienne pour faire face à la demande au jour, dans le respect des décisions tactiques. Parmi ces décisions, on trouve : gestion des stocks, gestion de la main d'œuvre et la gestion des équipements. Ces décisions se prennent au niveau des ateliers et par les agents de maîtrise.

2.7.2 Les niveaux de décision en gestion de production

On distingue cinq niveaux de décisions en gestion de production :

Le programme directeur de production

Le programme directeur de production correspond à une détermination plus précise des quantités prévues par le plan de production, il concerne la fonction production et on y incorpore les commandes fermes.

Le plan de production

Il reflète la cohérence entre l'activité industrielle, le plan marketing et la politique financière de l'entreprise. Il permet d'adapter la capacité aux variations de la demande. La première entrée du plan de production est constituée des prévisions de ventes mois par mois.

Le plan de besoins en matières

À partir du programme directeur de production est effectué un calcul des besoins bruts et des besoins nets par éclatement de nomenclature, par le biais de la méthode MRP. La méthode MRP est une technique d'ordonnancement à moyen terme. Elle permet donc de déterminer les OF (ordres de fabrication) et les OA (ordre d'achat).

L'ordonnancement

C'est la détermination de l'ordre de passage de l'ensemble des travaux à réaliser pour la production d'un bien, afin d'optimiser les ressources en fonction des objectifs fixés, en quantité et en délais. Il consiste à transformer les ordres de fabrication suggérés par le plan de besoins en matières en ordres de fabrication fermes pour chaque ressource de l'atelier, à partir de contraintes techniques (gamme opératoire, temps d'opération...).

Le suivi de production

Pour suivre l'évolution des travaux dans l'atelier, il faut disposer des informations sur l'état de l'activité des ateliers : travaux non commencés, travaux en cours et ceux achevés. Pour cela, on utilise un dossier de fabrication : Bon de travaux, Bon de sortie matière, Fiche suiveuse, ...). La plupart du temps le responsable de production ne peut pas réaliser tout ce qui a été planifié

lors de la phase d'ordonnement à cause des aléas de production (panne machine, avenant à une commande, . . .). Il doit effectuer des re-ordonnements de la production pour pallier à ces aléas et assurer l'atteinte des objectifs.

2.8 Méthodes de résolution [20]

En recherche opérationnelle, on distingue deux types de méthodes de résolution, à savoir les méthodes exactes et les méthodes approchées.

Une méthode est dite exacte si elle garantit l'optimalité des solutions trouvées. Sinon, elle sera dite approchées, ou heuristique, lorsqu'on observe empiriquement qu'elle fournit de bonnes solutions.

On note que le coût de calcul est un facteur primordial dans le choix d'une méthode de résolution automatique. Bien que la puissance croissante des ordinateurs puisse limiter la taille des problèmes combinatoires abordables par les méthodes exactes, leurs résolution est souvent d'un coût élevé (temps de calcul, espace mémoire), ce qui explique le recours aux méthodes approchées qui elles fournissent de bonnes solutions en un temps de calcul raisonnable.

Pour ce qui est des problèmes d'ordonnement, plusieurs méthodes de résolution (exactes et approchées) sont rencontrées, on cite :

- Les procédures de séparation et d'évaluation (branch and bound).
- Les algorithmes génétiques, le recuit simulé, la recherche tabou.
- Les méthodes hybrides (procédures de séparation et d'évaluation avec algorithme génétique) etc.

2.8.1 Les méthodes exactes

On peut définir une méthode exacte comme étant une méthode qui fournit une solution optimale pour un problème d'optimisation. L'utilisation de ce type de méthodes s'avère particulièrement intéressante dans les cas des problèmes de petites tailles. La méthode par séparation et évaluation (branch and bound) constituent certainement celles qui sont les plus utilisées pour résoudre les problèmes d'optimisation multi-objectifs. D'autres méthodes telles que la programmation linéaire ou la programmation dynamique, sont aussi utilisées couramment.

Toutes ces méthodes examinent d'une manière implicite, la totalité de l'espace de recherche pour produire la solution optimale [25].

a- La méthode de Branch and Bound

L'algorithme Branch and Bound consiste 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.

L'algorithme branch and bound effectue une recherche complète de l'espace des solutions d'un problème donné, pour trouver la meilleure solution.

La démarche de l'algorithme Branch and Bound consiste à [10] :

- 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.

b- La programmation dynamique

Elle se base sur le principe de Bellman [5] : " 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.

c- La programmation linéaire

C'est l'une des techniques classiques de recherche opérationnelle. Elle repose sur la méthode du simplexe et les algorithmes de points intérieurs de Karmarkar [26].

Elle consiste à minimiser une fonction coût en respectant des contraintes, le critère et les contraintes étant des fonctions linéaires des variables du problème [22].

Un programme linéaire a la forme suivante :

$$(PL) \begin{cases} \min z = cx, \\ Ax \leq b, \\ x \geq 0. \end{cases}$$

La programmation linéaire en nombres entiers

Un programme en nombre entier de dimension $(m \times n)$ a la forme suivante :

$$(PLNE) \begin{cases} \min z = cx, \\ Ax \leq b, \\ x_j \in N. \end{cases}$$

Où A est une matrice $(m \times n)$, b un vecteur de dimension m , c un vecteur de dimension n et x est un vecteur inconnu de dimension n . Dans le cas particulier, où les contraintes $x_j \in N$ sont remplacées par $x_j \in \{0, 1\}$, on dit qu'on a un programme linéaire en 0,1.

2.8.2 Les méthodes approchées ou métaheuristiques

Malgré l'évolution permanente de l'informatique, il existe toujours, pour un problème polynomial, une taille critique au-dessus de laquelle une énumération, même partielle, des solutions admissibles devient prohibitive.

Les métaheuristiques sont ainsi des méthodes de recherche générales, dédiées aux problèmes d'optimisation difficile. Elles sont, en général, présentées sous forme de concepts. Les principales métaheuristiques sont celles basées sur la recherche locale, telles que le recuit simulé et la recherche Tabou, et celles basées sur les algorithmes évolutionnistes telles que les algorithmes génétiques ainsi que les algorithmes basés sur la recherche globale tels que les algorithmes de colonies de fourmis et les algorithmes reposant sur la méthode d'optimisation par essaim particulaire [25].

Voici un organigramme qui nous présente les différentes classifications des méthodes d'optimisation combinatoire.

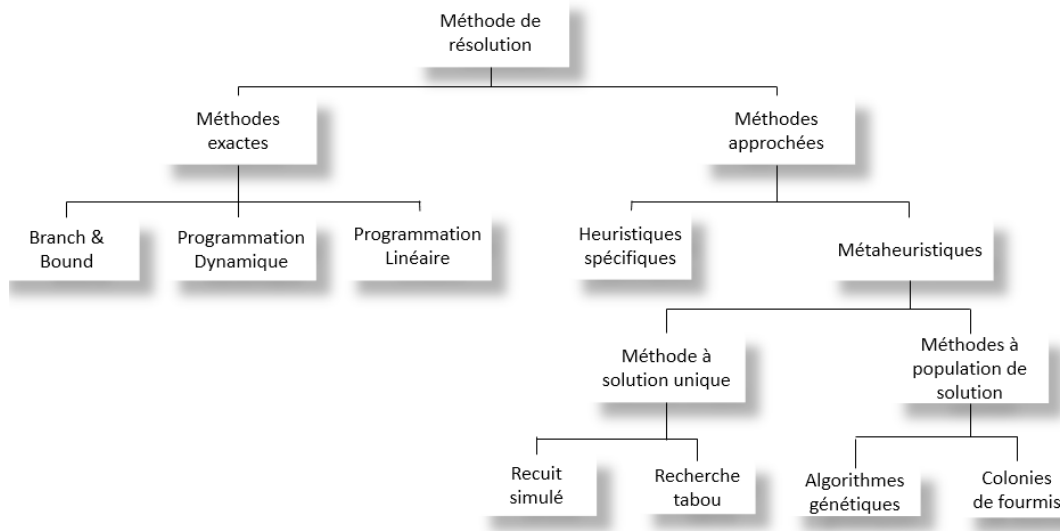


FIGURE 2.1 – Organigramme de classification des méthodes d'optimisation combinatoire

a- Les heuristiques

Les heuristiques sont des méthodes empiriques qui donnent généralement de bons résultats sans pour autant être démontrables. Elles se basent sur des règles simplifiées pour optimiser un ou plusieurs critères. Le principe général de cette catégorie de méthodes est d'intégrer des stratégies de décision pour construire une solution proche de celle optimale tout en cherchant à avoir un temps de calcul raisonnable. Parmi ces stratégies, nous distinguons :

- FIFO (First In First Out) où la première tâche arrivée est la première à être ordonnancée ;
- SPT (Shortest Processing Time) où la tâche ayant le temps opératoire le plus court est

- traitée en premier ;
- LPT (Longest Processing Time) où la tâche ayant le temps opératoire le plus important est traitée en premier ;
- EDD (Earliest Due Date) où la tâche ayant la date due la plus petite est la plus prioritaire [25].

b- Algorithmes génétiques

Les algorithmes génétiques ont été mise au point par John Holland dans les années 1960 [32]. Ils sont des algorithmes itératifs dont le but est d'optimiser une fonction prédéfinie, appelée fitness. Pour réaliser cet objectif, l'algorithme travaille sur un ensemble de points, appelés population d'individus. Chaque individu ou chromosome représente une solution possible du problème donné. Il est constitué d'éléments, appelés gènes, dont les valeurs sont appelées allèles. L'utilisation d'un algorithme génétique nécessite la définition, au préalable, d'un espace de recherche dont les éléments de base sont les chromosomes et d'une fonction définie sur cet espace (fonction fitness) dont la valeur optimale est évaluée en rapport avec les opérateurs de croisement et de mutation choisis, la figure 2.5 (page 29) représente le fonctionnement général d'un algorithme génétique [25].

Les algorithmes génétiques sont basés sur les principes suivants [17] :

1- Codage du chromosome : Le choix du codage des données dépend de la spécificité du problème traité. Il conditionne fortement l'efficacité de l'algorithme génétique. Un chromosome (une solution particulière) à différentes manières d'être codé et le codage le plus utilisé est le binaire (0-1).

2- Génération de la population initiale : Dans les problèmes d'optimisation, une connaissance de candidat de bonne qualité comme point d'initialisation conditionne la rapidité de la convergence vers l'optimum dans l'ensemble des solutions réalisables est totalement inconnue, il est naturel de générer aléatoirement des individus. Les tirages sont réalisés en respectant les contraintes et d'une manière uniforme dans chacun des domaines associés aux composantes de l'ensemble des solutions. Des connaissances a priori sur le problème traité permettent de générer les individus dans un domaine particulier afin d'accélérer la convergence de l'algorithme génétique. Les opérations de croisements et de mutation permettent d'entretenir la diversité d'une population non homogène au cours de générations afin de parcourir l'ensemble de solutions le plus largement possible [27].

3- Méthode de sélection : La sélection permet d'identifier les individus susceptibles d'être croisés dans une population. Nous trouvons dans la littérature plusieurs principes de sélection :

- Sélection par rang ;
- Sélection par roulette ;
- Sélection aléatoire ;
- Sélection par tournoi.

4- Opérateur de croisement :

Le croisement a pour but d'enrichir la diversité de la population en manipulant les composantes des chromosomes. Classiquement les croisements sont envisagés avec deux parents et génèrent deux enfants. Il est appliqué avec une probabilité P_c , communément appelée probabilité de croisement. Les plus anciens opérateurs de croisement sont l'opérateur de croisement à un point et à deux points sur deux chromosomes à codage binaire. Ils constituent la base des opérateurs de croisement [27].

- **Opérateur de croisement à un point :** L'opérateur à un point de croisement consiste à diviser chacun des deux parents en deux parties à la même position choisi au hasard. L'enfant1 est composé de la première partie du premier parent et de deuxième parent alors que l'enfant2 est constitué de la première partie du deuxième parent et de la deuxième partie du premier parent.

L'opérateur à un point de croisement est illustré dans la figure 2.2.

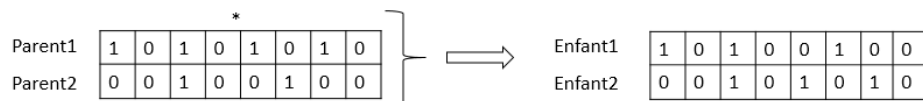


FIGURE 2.2 – L'opérateur de croisement à 1 point

- **Opérateur de croisement à deux points :** L'opérateur à deux points de croisement consiste à fixer deux positions. L'enfant1 sera la copie du parent1 en remplaçant sa partie entre les deux positions par celle du parent2. On effectuera la même opération pour déterminer l'enfant en intervertissant les rôles des parent1 et parent2.

L'opérateur à deux points de croisement est illustré dans la figure 2.3.

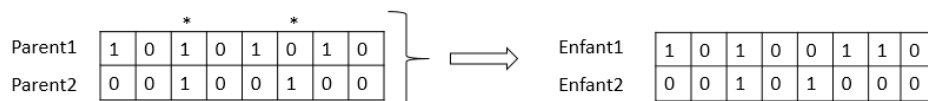


FIGURE 2.3 – L'opérateur de croisement à 2 points

5- Opérateur de mutation

L'opérateur de mutation apporte aux algorithmes génétiques l'aléa nécessaire à une exploration efficace de l'espace. Cet opérateur nous garantit que l'algorithme génétique sera susceptible d'atteindre la plupart des points de domaine réalisable [23].

Cet opérateur de mutation est utilisé avec une probabilité P_m nommée probabilité de mutation. Dans les algorithmes génétiques à codage binaire, cette probabilité s'effectuait sur les gènes

en échangeant sa valeur de 0 à 1 ou de 1 à 0 et non sur le chromosome tout entier. Mais avec un algorithme génétique codé autrement, on applique cette probabilité par rapport à l'individu tout entier et non sur les gènes [27].

L'opérateur de mutation est illustré dans la figure 2.4 suivante :

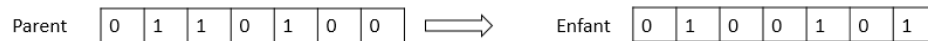


FIGURE 2.4 – Opérateur de mutation dans le cas de codage binaire

6- Méthode d'insertion

Après l'étape de mutation, on utilise une méthode d'insertion pour générer une nouvelle population. Lors de la construction de cette population, on se trouve devant un vrai problème : faut-il garder des enfants ou des parents ou bien un certain pourcentage d'entre eux en respectant que la taille de la population (N) reste constante ?.

7- Critère d'arrêt

Le test d'arrêt joue un rôle primordial dans le jugement de la qualité des individus. Son but est de nous assurer l'optimalité de la solution finale obtenue par l'algorithme génétique. Les critères d'arrêt sont de deux natures :

1. Arrêt après un nombre fixé a priori de générations.
2. Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment. Ce test d'arrêt reste le plus objectif et le plus utilisé.

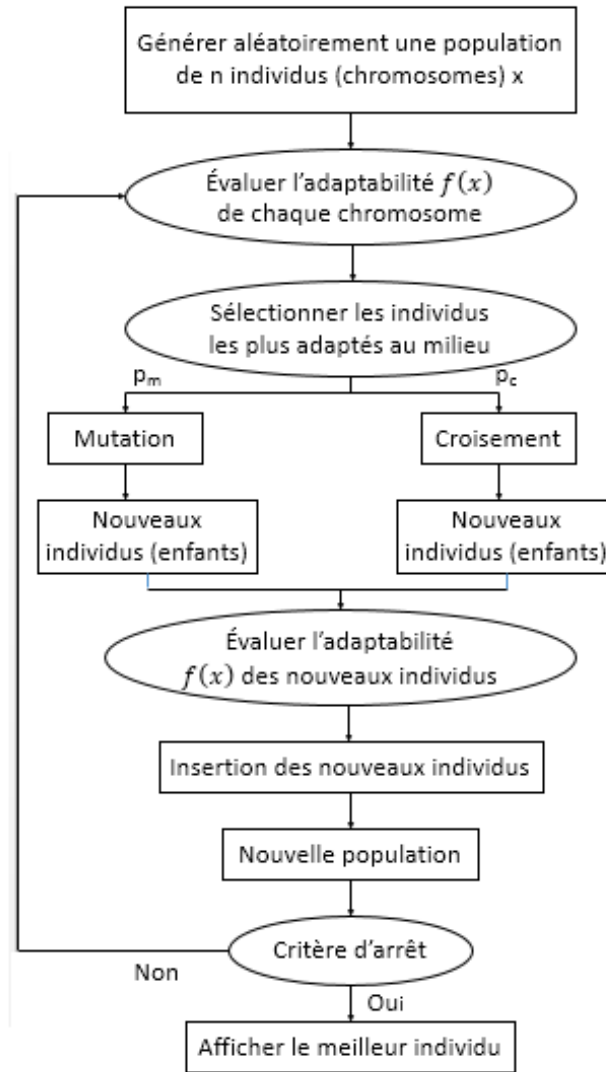


FIGURE 2.5 – Organigramme de l'algorithme génétique

c- La recherche Tabou [25]

Bien que son origine remonte à 1977, la recherche tabou n'est proposée qu'au milieu des années 80 par Fred. Cette méthode, développée pour résoudre des problèmes combinatoires, la plupart NP-difficiles, propose de surmonter le problème des optima locaux par l'utilisation d'une mémoire.

La méthode tabou est une procédure itérative qui, partant d'une solution initiale, tente de converger vers la solution optimale en exécutant, à chaque pas, un mouvement dans l'espace de recherche. Chaque pas consiste d'abord à engendrer un ensemble de solutions voisines de la solution courante pour ensuite en choisir la meilleure, même si ce choix entraîne une augmentation de la fonction objectif à minimiser.

Les domaines d'application de la recherche tabou sont vastes et variés, ils passent de l'ordonnancement à la robotique, au problème du voyageur de commerce, à l'électronique voire même aux applications médicales,...

En tenant compte des notations suivantes, les étapes d'évolution de l'algorithme sont présentées dans la figure .

- s_0 : la solution initiale,
- s^* : la meilleure solution actuelle,
- $f(x)$: la fonction à minimiser,
- $f(s^*)$: la valeur de la fonction à minimiser pour obtenir la meilleure solution,
- T : la liste tabou.

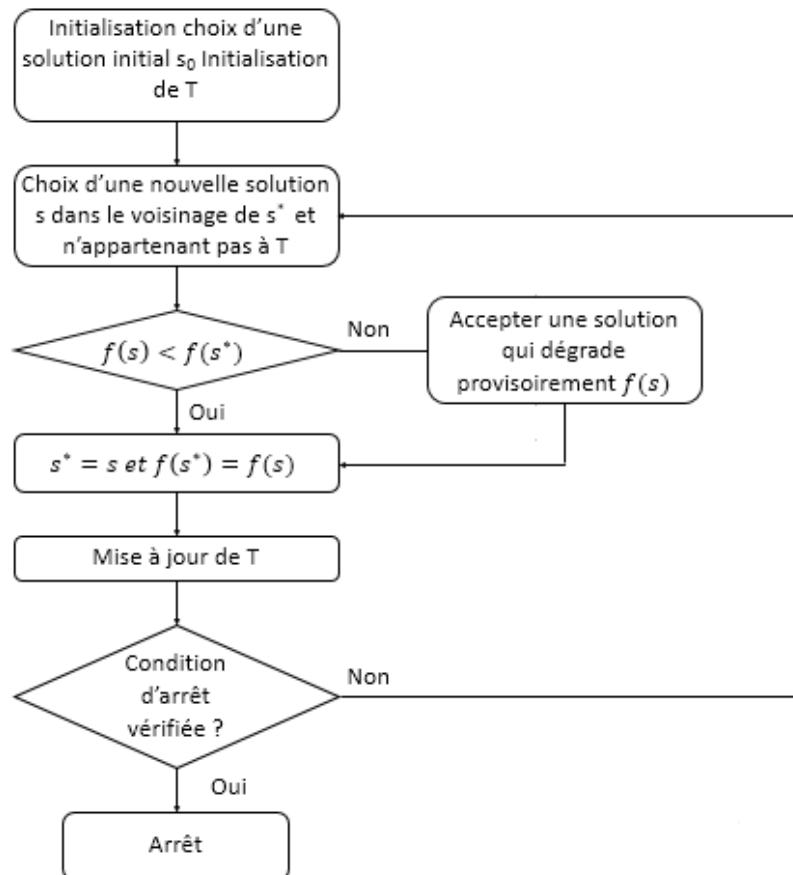


FIGURE 2.6 – Algorithme général de la méthode recherche tabou

d- Le recuit simulé

Inspiré du recuit physique, ce processus est utilisé en métallurgie pour améliorer la qualité d'un solide et cherche un état d'énergie minimale qui correspond à une structure stable du solide. Ainsi, pour qu'un métal retrouve une structure proche du cristal parfait, on porte celui-ci à une température élevée, puis on le laisse refroidir lentement de manière à ce que les atomes aient le temps de s'ordonner régulièrement. L'algorithme du recuit simulé, permet de résoudre les problèmes de minima locaux. En effet, une nouvelle solution de coût supérieur à celui de la solution courante ne sera pas forcément rejetée, son acceptation sera déterminée aléatoirement en tenant compte de la différence entre les coûts ainsi que du facteur température T . Ce paramètre, sert à prendre en compte le fait que plus le processus d'optimisation est avancé, moins on est prêts à accepter une solution plus coûteuse. Par contre, l'acceptation de solutions fortement coûteuses permet, au début, de mieux explorer l'espace des solutions possibles et ainsi, d'accroître les chances d'approcher le minimum global [31].

A chaque itération, une seule solution voisine est générée. Celle-ci est acceptée si elle est meilleure que la solution courante. Dans le cas contraire, la nouvelle solution est acceptée avec une certaine probabilité qui dépend de l'importance de la détérioration et du paramètre T correspondant à la température. En règle générale, la température est diminuée par paliers, à chaque fois qu'un certain nombre d'itérations est effectué. La meilleure solution trouvée est mémorisée. L'algorithme est interrompu lorsqu'aucune solution voisine n'a été acceptée pendant un cycle complet d'itérations à température constante [31].

Un algorithme d'optimisation par recuit simulé se décompose selon les étapes suivantes [19] :

- choix d'une fonction à optimiser,
- adoption d'un schéma de recuit dans lequel sont précisés la température initiale, le nombre de configurations générées à chaque température et le schéma de décroissance du critère,
- génération stochastique de configurations voisines, correspondant aux transitions,
- choix d'un critère d'acceptation.

Cet algorithme est présenté dans la figure 2.7.

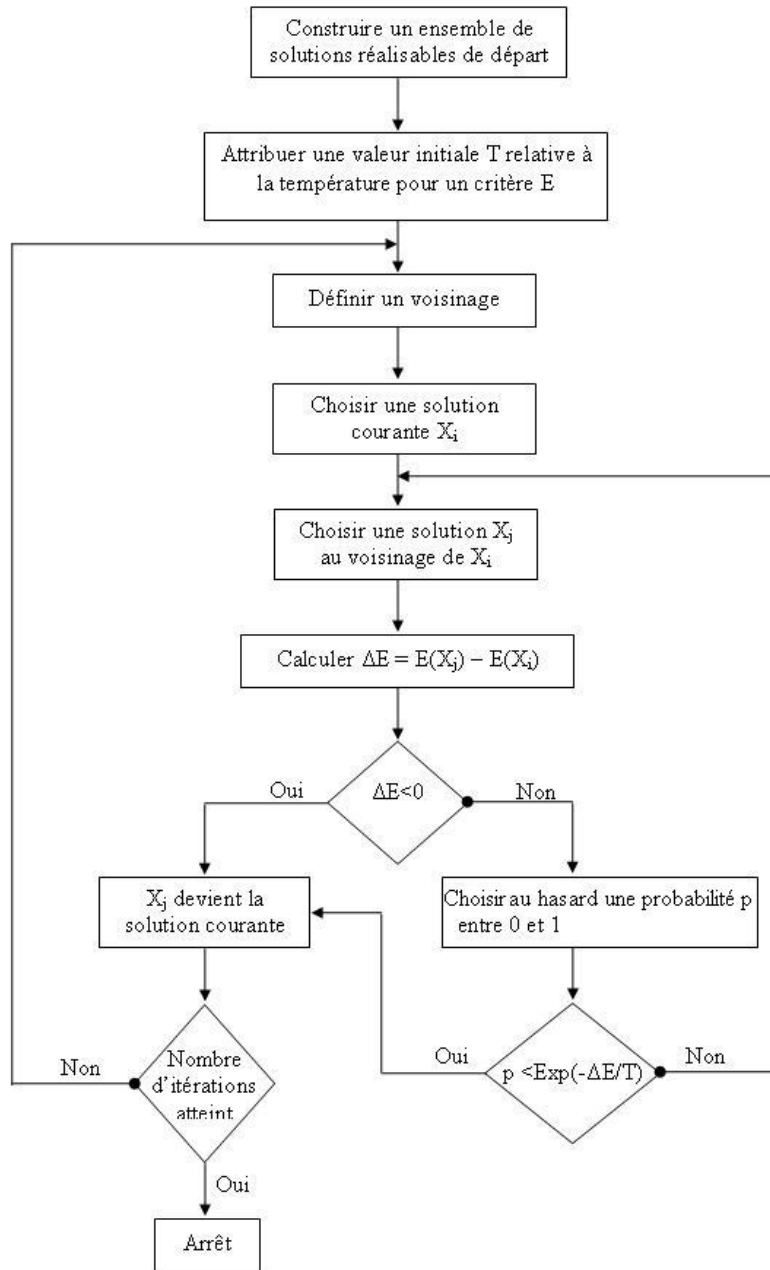


FIGURE 2.7 – Algorithme relatif au fonctionnement général du recuit simulé

2.9 Conclusion

Dans ce chapitre, nous avons donné quelques rappels théoriques sur l'optimisation combinatoire et l'ordonnancement, ainsi que les méthodes de résolution.

Chapitre 3

Modélisation du problème

Dans ce chapitre, nous allons élaborer un modèle mathématique capable d'illustrer la problématique posé dans le premier chapitre.

3.1 L'approche générale

La modélisation d'un problème est une étape cruciale lors de l'étude d'une situation réelle, puisque une mauvaise modélisation induit des résultats erronés.

La méthodologie de la modélisation est la suivante :

1. Bien définir le but à atteindre (la problématique) ;
2. Élaborer un plan d'exploitation (de réalisation) du projet ;
3. Position de problème : c'est l'étape la plus importantes (définition concrète du problème, définition des variables et des paramètres du système) ;
4. Construction du modèle ;
5. Élaboration d'une méthode de calcul ;
6. Exploitation du problème technique sur machine ;
7. Récolte des données ;
8. Validation du modèle ;
9. Interprétation des résultats.

3.2 Collectes des données

3.2.1 Les jobs

Pour la modélisation, nous avons pris en considération 12 produits seulement qui seront les jobs. $J = \{1, \dots, 12\}$ est l'ensemble des jobs, décrit par :

1. production de FTN.
2. production de FTI.
3. production de FE.
4. production de FAS.
5. production de FPEtN.

6. production de FPEtI.
7. production de HN.
8. production de HI.
9. production de SN.
10. production de SI.
11. production de FCN.
12. production de FCI.

3.2.2 Les opérations

Chaque job possède un ordre de passage sur les différents ateliers de l'usine : extrusion, impression, sacherie, complexage et découpe.

Les opérations d'extrusion

1. extrusion de FTN.
2. extrusion de FTI.
3. extrusion de FE.
4. extrusion de FAS.
5. extrusion de FPEtN.
6. extrusion de FPEtI.
7. extrusion de HN.
8. extrusion de HI.
9. extrusion de SN.
10. extrusion de SI.

Les opérations d'impression

11. impression de FTI.
12. impression de FPEtI.
13. impression de HI.
14. impression de SI.
15. impression de FCI.

Les opérations de la sacherie

16. soudage de HN.
17. soudage de HI.
18. soudage de SN.
19. soudage de SI.

Les opérations de complexage

20. complexage de FCN.
21. complexage de FCI.

Les opérations de découpage

22. découpage de FTI.
23. découpage de FPETI.
24. découpage de FCN.
25. découpage de FCI.

Les opérations constituant chaque job sont présentées dans le tableau suivant, où O_{ij} représente la i^{eme} opération du job $j \in J$:

| Jobs | Opération nécessaires | nombre d'opérations ($nbop_j$) |
|------|--------------------------------|----------------------------------|
| 1 | $O_{1,1}$ | 1 |
| 2 | $O_{1,2}, O_{2,2}, O_{3,2}$ | 3 |
| 3 | $O_{1,3}$ | 1 |
| 4 | $O_{1,4}$ | 1 |
| 5 | $O_{1,5}$ | 1 |
| 6 | $O_{1,6}, O_{2,6}, O_{3,6}$ | 3 |
| 7 | $O_{1,7}, O_{2,7}$ | 2 |
| 8 | $O_{1,8}, O_{2,8}, O_{3,8}$ | 3 |
| 9 | $O_{1,9}, O_{2,9}$ | 2 |
| 10 | $O_{1,10}, O_{2,10}, O_{3,10}$ | 3 |
| 11 | $O_{1,11}, O_{2,11}$ | 2 |
| 12 | $O_{1,12}, O_{2,12}, O_{3,12}$ | 3 |

TABLE 3.1 – Opérations de chaque job

3.2.3 Les différentes machines de chaque atelier

Au niveau de l'extrusion

1. une machine de capacité de 1,7 tonnes par jour,
2. une machine de capacité de 2 tonnes par jour,
3. une machine de capacité de 0.85 tonnes par jour,
4. une machine de capacité de 7 tonnes par jour,
5. une machine de capacité de 5.5 tonnes par jour,
6. une machine de capacité de 12 tonnes par jour,
7. une machine de capacité de 25 tonnes par jour.

Au niveau de l'impression

- 8. une imprimante de capacité de 12 tonnes par jour,
- 9. une imprimante de capacité de 12 tonnes par jour.

Au niveau de la sacherie

- 10. une soudeuse de capacité de 7 tonnes par jour.

Au niveau de la découpe

- 11. une découpeuse de capacité de 15 tonnes par jour,
- 12. une découpeuse de capacité de 41 tonnes par jour,
- 13. une découpeuse de capacité de 55 tonnes par jour.

Au niveau du complexage

- 14. une contre-colleuse de capacité de 27.5 tonnes par jour.

Le tableau ci-dessous résume les opérations qui peuvent être effectuées sur chaque machine :

| Machines (M_k) | Opérations qui peuvent être exécutées par M_k |
|--------------------|---|
| M_1 | $O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4}, O_{1,7}, O_{1,8}, O_{1,9}, O_{1,10}$ |
| M_2 | $O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4}, O_{1,7}, O_{1,8}, O_{1,9}, O_{1,10}$ |
| M_3 | $O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4}, O_{1,7}, O_{1,8}, O_{1,9}, O_{1,10}$ |
| M_4 | $O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4}, O_{1,7}, O_{1,8}, O_{1,9}, O_{1,10}$ |
| M_5 | $O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4}, O_{1,5}, O_{1,6}, O_{1,7}, O_{1,8}, O_{1,9}, O_{1,10}$ |
| M_6 | $O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4}, O_{1,5}, O_{1,6}, O_{1,7}, O_{1,8}, O_{1,9}, O_{1,10}$ |
| M_7 | $O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4}, O_{1,5}, O_{1,6}, O_{1,7}, O_{1,8}, O_{1,9}, O_{1,10}$ |
| M_8 | $O_{2,2}, O_{2,6}, O_{2,8}, O_{2,10}, O_{1,12}$ |
| M_9 | $O_{2,2}, O_{2,6}, O_{2,8}, O_{2,10}, O_{1,12}$ |
| M_{10} | $O_{2,7}, O_{2,9}, O_{3,8}, O_{3,10}$ |
| M_{11} | $O_{2,11}, O_{3,2}, O_{3,6}, O_{3,12}$ |
| M_{12} | $O_{2,11}, O_{3,2}, O_{3,6}, O_{3,12}$ |
| M_{13} | $O_{2,11}, O_{3,2}, O_{3,6}, O_{3,12}$ |
| M_{14} | $O_{1,11}, O_{2,12}$ |

TABLE 3.2 – Opérations s'effectuant sur chaque machine

La figure suivant schématise le système de production totale de l'entreprise qu'on a modélisé. Le schéma montre toutes les machines des deux unités et leurs répartitions ainsi que le passage des différents produits.

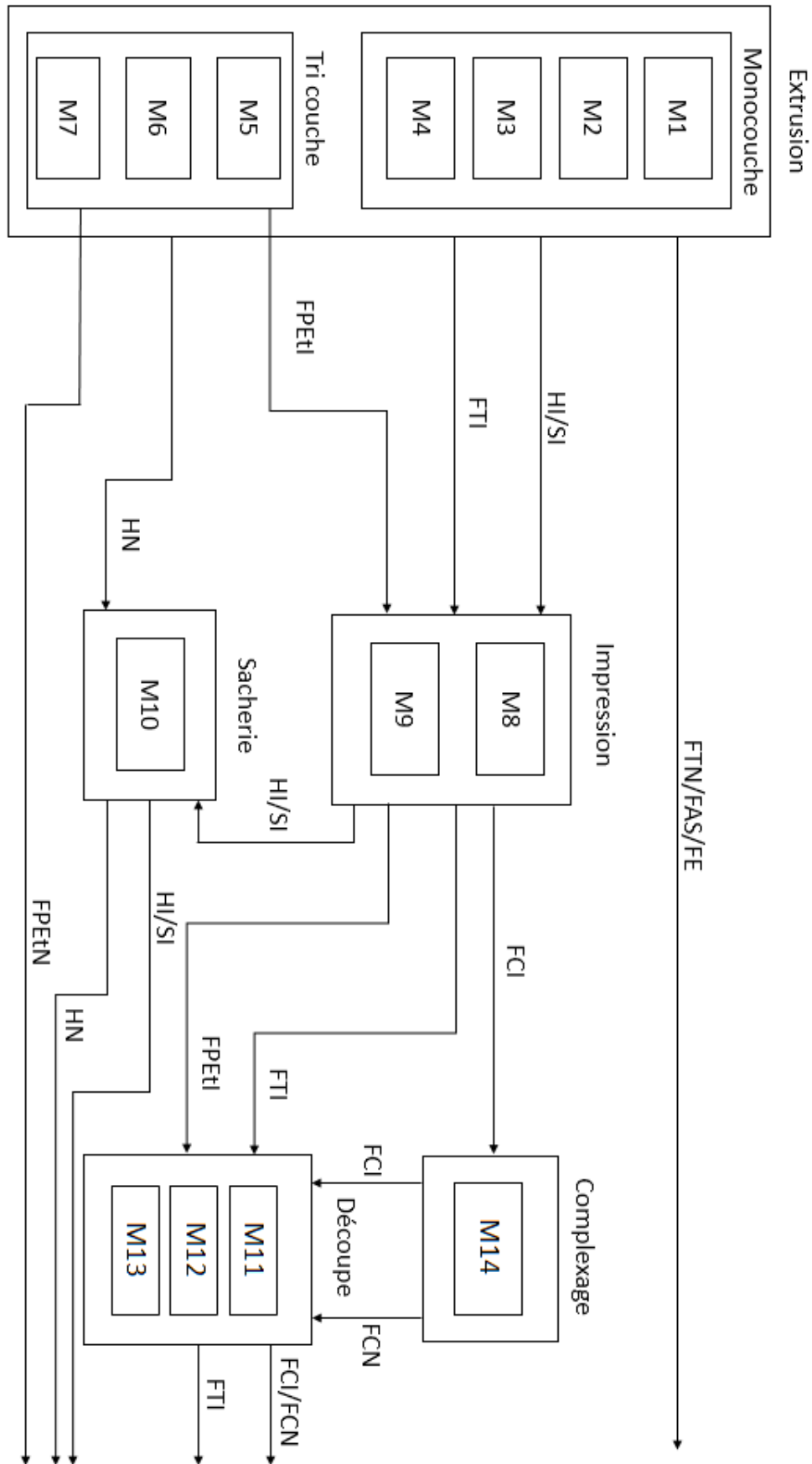


FIGURE 3.1 – Schéma du processus de production de MERIPLAST

3.3 Modélisation du problème

La modélisation mathématique est la manière la plus efficace de modéliser les problèmes d'ordonnancement car elle permet d'élaborer des méthodes de calcul exactes ou approchées lorsqu'on se trouve en présence d'un problème de taille importante. On utilise souvent la programmation linéaire.

3.3.1 Construction du modèle

Les notations :

Par la suite, on adopte les notations suivantes :

- $J = \{1, \dots, 12\}$: l'ensemble des jobs à réaliser ;
- $O_{i,j}$: est la i^{eme} opération du job j ;
- $nbop_j$ est le nombre des opérations du job j ;
- $N = \sum_{j=1}^{12} nbop_j$, correspond au nombre total d'opérations ;
- $O_{i_{der},j}$: désigne la dernière opération du job j ;
- $d_{i,j,k}$: désigne la durée de l'opération $O_{i,j}$ sur la machine k ;
- E_{ij} : l'ensemble des machines pouvant effectuer l'opération $O_{i,j}$;
- t_{ij} : la date de début de l'opération $O_{i,j}$;
- $t_{i_{der},j}$: la date de début de la dernière opération du job j ;
- C_j : la date d'achèvement du job j ;
- C_{max} : la plus grande date d'achèvement des jobs, avec $C_{max} = Max_{j \in J} C_j$

Facteurs non contrôlables :

- nombre et type de job,
- nombre et type d'opérations constituant chaque job,
- nombre, capacités de chaque machine,
- durées des opérations,
- succession des opérations de chaque job.

Facteurs contrôlables :

- l'affectation des opérations aux machines,
- date de début de chaque opération,
- le séquençement des opérations sur machine.

Les hypothèses :

Dans notre problème, on se fixe les hypothèses suivantes :

- chaque machine est disponible pendant toute la période de l'ordonnancement, c'est-à-dire les pannes des machines ne sont pas prises en compte dans notre étude,
- à l'instant $t = 0$, toutes les machines sont disponibles et prêtes à l'utilisation,
- une machine ne peut exécuter qu'une seule opération, donc elle devient disponible à la fin de l'opération en cours d'exécution,
- une opération en cours d'exécution ne peut pas être interrompue,

- les jobs sont indépendants les uns des autres.

Variabes de décision :

Soient les variables de décisions définies comme suit :

$$X_{i,j,k} = \begin{cases} 1 & \text{si l'opération } O_{i,j} \text{ est affectée à la machine } k, \\ 0 & \text{sinon.} \end{cases}$$

$$j = 1, \dots, 12, \quad i = 1, \dots, nbop_j, \quad k \in E_{ij}.$$

$$Y_{ij,i'j',k} = \begin{cases} 1 & \text{si l'opération } O_{i',j'} \text{ succède à l'opération } O_{i,j} \text{ sur la machine } k, \\ 0 & \text{sinon.} \end{cases}$$

$$j = 1, \dots, 12, \quad i = 1, \dots, nbop_j, \quad j' = 1, \dots, 12, \quad i' = 1, \dots, nbop_{j'}, \quad k \in E_{ij}.$$

Objectif :

L'objectif est de minimiser la plus grande date d'achèvement des travaux C_{max} :

$$\min \max_{j \in J} C_j$$

Où la date d'achèvement C_j d'un job j est égale à la date de début de sa dernière opération $t_{i_{der}j}$ plus la durée de cette dernière sur la machine qui l'effectuera :

$$C_j = t_{i_{der}j} + \sum_{k \in E_{i_{der}j}} d_{i_{der},j,k} \cdot X_{i_{der},j,k}, \quad \forall j = 1, \dots, 12.$$

Les contraintes :

1. une opération $O_{i,j}$ n'est affectée qu'à une seule machine :

$$\sum_{k \in E_{ij}} X_{i,j,k} = 1, \quad \forall j = \overline{1, 12}, \quad i = \overline{1, nbop_j}.$$

2. la succession des opérations d'un même job doit être respectée :

$$t_{ij} \geq t_{i-1j} + \sum_{k \in E_{ij}} d_{i-1,j,k} X_{i-1,j,k}, \quad \forall j = \overline{1, 12}, \quad i = \overline{2, nbop_j}.$$

3. deux opérations ne peuvent s'effectuer simultanément sur une machine :

$$Y_{ij,i'j',k} \left[t_{ij} + \sum_{k \in E_{ij}} d_{i,j,k} \cdot X_{i,j,k} \right] \leq t_{i'j'}, \quad \forall k, \quad j = \overline{1, 12}, \quad i = \overline{1, nbop_j}, \quad j' = \overline{1, 12}, \\ i' = \overline{1, nbop_{j'}}, \quad \text{avec } O_{i,j} \neq O_{i',j'}.$$

4. chaque opération précède une et une seule opération $O_{i',j'}$

$$\sum_{j'=1}^{12} \sum_{i'=1}^{nbop_{j'}} \sum_{k \in E_{ij} \cap E_{i'j'}} Y_{ij,i'j',k} = 1, \quad \forall j = \overline{1, 12}, \quad i = \overline{1, nbop_j}.$$

5. contraintes de positivité et de binarité :

$$\begin{aligned}
 C_j &\geq 0 & \forall j = \overline{1,12} \\
 t_{ij} &\geq 0 & \forall j = \overline{1,12}, i = \overline{1, nbop_j} \\
 d_{i,j,k} &\geq 0 & \forall k \in E_{ij}, j = \overline{1,12}, i = \overline{1, nbop_j} \\
 X_{i,j,k} &\in \{0,1\} & \forall k \in E_{ij}, j = \overline{1,12}, i = \overline{1, nbop_j} \\
 Y_{ij,i'j',k} &\in \{0,1\} & \forall k \in E_{ij}, j = \overline{1,12}, i = \overline{1, nbop_j} \\
 & & j' = \overline{1,12}, i' = \overline{1, nbop_{j'}}
 \end{aligned}$$

Le modèle mathématique final s'écrit donc, comme suit :

$$\begin{aligned}
 & \text{Min Max}_{j \in \{1, \dots, 12\}} C_j \\
 & \left\{ \begin{array}{ll}
 \sum_{k \in E_{ij}} X_{i,j,k} = 1, & \forall j = \overline{1,12}, i = \overline{1, nbop_j}. \\
 t_{ij} \geq t_{i-1j} + \sum_{k \in E_{ij}} d_{i-1,j,k} X_{i-1,j,k} & \forall j = \overline{1,12}, i = \overline{2, nbop_j}. \\
 Y_{ij,i'j',k} \cdot \left[t_{ij} + \sum_{k \in E_{ij}} d_{i,j,k} \cdot X_{i,j,k} \right] \leq t_{i'j'} & \forall k, j = \overline{1,12}, i = \overline{1, nbop_j}, j' = \overline{1,12}, \\
 & i' = \overline{1, nbop_{j'}}, \text{ avec } O_{i,j} \neq O_{i',j'}. \\
 \sum_{j'=1}^{12} \sum_{i'=1}^{nbop_{j'}} \sum_{k \in E_{ij} \cap E_{i'j'}} Y_{ij,i'j',k} = 1, & \forall j = \overline{1,12}, i = \overline{1, nbop_j}. \\
 C_j \geq 0 & \forall j = \overline{1,12} \\
 t_{ij} \geq 0 & \forall j = \overline{1,12}, i = \overline{1, nbop_j} \\
 d_{i,j,k} \geq 0 & \forall k \in E_{ij}, j = \overline{1,12}, i = \overline{1, nbop_j} \\
 X_{i,j,k} \in \{0,1\} & \forall k \in E_{ij}, j = \overline{1,12}, i = \overline{1, nbop_j} \\
 Y_{ij,i'j',k} \in \{0,1\} & \forall k \in E_{ij}, j = \overline{1,12}, i = \overline{1, nbop_j} \\
 & j' = \overline{1,12}, i' = \overline{1, nbop_{j'}}
 \end{array} \right.
 \end{aligned}$$

3.4 Conclusion

Dans ce chapitre, nous avons proposé un modèle mathématique capable de décrire au mieux notre problème. Nous l'avons identifié comme étant un problème d'ordonnancement d'ateliers de production de type Job Shop Flexible. Il s'agit d'ordonner les différents jobs de l'unité de production de MERIPLAST. Chaque job étant un produit, où chacun de ces jobs se compose d'opérations bien définie. Ainsi, le modèle construit englobe l'ensemble des contraintes du Job Shop Flexible en considérant comme objectif la minimisation de la plus grande date d'achèvement des jobs (Makespan). Le nombre important de variables et de contraintes présent en considération, le problème est classé NP-difficile [12]. Donc, sa résolution par une approche exacte est très complexe, d'où la nécessité de recourir à des méthodes approchées (recherche Tabou, algorithme génétique) que nous allons aborder dans le prochain chapitre.

Chapitre 4

Résolution du problème

Les problèmes d'optimisation combinatoire sont très nombreux dans la vie quotidienne des entreprises. Ils sont souvent faciles à définir mais sont difficiles à résoudre. En tenant compte de l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle. Ces méthodes peuvent être exactes ou approximatives.

Le problème d'ordonnement Job Shop est parmi les problèmes les plus étudiés dans la littérature. Il est classé parmi les problèmes NP-difficiles au sens fort [12], et ne possède pas de solutions algorithmiques efficaces valables pour toutes les données. Donc les métaheuristiques deviennent l'unique moyen d'obtenir une bonne solution en un temps raisonnable.

4.1 Choix de la méthode

Dans le cas des problèmes NP-difficile, deux possibilités sont offertes. Si le problème est de petite taille, alors les algorithmes exacts permettant de trouver la solution optimale peuvent être utilisés, sinon, on sera contraint d'appliquer les métaheuristiques.

Dans ce travail, on cherche à résoudre un problème d'ordonnement de type Job Shop Flexible. Ce problème est NP-difficile et vu le nombre de variables et de contraintes prise en considération dans le modèle mathématique, il est impossible de le résoudre par une méthode exacte qui fournisse une solution optimale, d'où il nous reste juste les métaheuristiques qui aboutissent à des solutions satisfaisant les objectifs considérés.

Notre choix est alors : les algorithmes génétiques, et la recherche Tabou.

4.2 La méthode de Recherche Tabou

La méthode recherche tabou a été créée par F.Glover en 1986. Depuis, plusieurs recherches ont été effectuées dans le but d'élargir son application dans le domaine de la recherche opérationnelle et intelligence artificielle.

La méthode de recherche tabou est une métaheuristique fondée sur le principe de la recherche locale. Ce principe consiste à explorer l'espace de recherche composé de toutes les solutions réa-

lisables dans le but d'aboutir à la solution optimale [12].

Le concept de cette méthode est que pour chaque solution x de l'ensemble des solutions réalisables X (ou espace de recherche) est associé un ensemble de voisins $V(x) \subset X$ appelé voisinage de x .

Cette approche utilise une solution initiale réalisable et une structure de voisinage à chaque itération de la méthode. Le voisinage, $V(x)$, de x est généré en effectuant un mouvement, appelé aussi une simple perturbation de x .

A l'itération n , l'ensemble des voisins de la solution courante, x_n , est généré, puis, ils sont examinés et le meilleur voisins, $x' \in V(x)$, de x est sélectionné. On peut remarquer qu'il est possible d'avoir le voisin de moins bonne qualité que la solution courante. Cette méthode est répertoriée parmi les méthodes avec mémoire car elle garde en mémoire les solutions déjà visité et ceci permet d'éviter les cycles. Cette mémoire est appelée "*liste tabou*".

4.2.1 Algorithme de la recherche Tabou pour Job Shop Flexible

Bien que la méthode recherche tabou a montré son efficacité pour la résolution de nombreux problèmes difficiles, elle reste difficile à adapter au problème du job shop flexible. Ceci est dû au nombre important de paramètres à définir [12]. Le schéma général de l'algorithme recherche tabou est le suivant :

1. Paramètres initiaux ;
2. Solution initiale ;
3. Recherche du voisinage ;
4. Evaluation du voisinage ;
5. Liste tabou ;
6. Critère d'arrêt.

1. Paramètres initiaux

Notre algorithme recherche Tabou a besoin de quelques paramètres afin qu'il s'exécute à savoir :

- Le nombre total d'itérations ;
- La taille de la liste tabou ;
- Le nombre de voisins à générer.

2. Solution initiale

Pour trouver la solution initiale de notre algorithme, on commence par générer aléatoirement une séquence de 25 chiffres distincts suivant une loi uniforme, puis on fait passer la séquence générée par un autre programme qui l'ordonne selon les contraintes de précédence, par exemple : l'opération $O_{1,2}$ précède $O_{2,2}$ qu'elle même précède $O_{3,2}$ dans le job 2. A la fin, on aura une séquence ordonnée et qui respecte le séquençement des opérations.

3. Recherche du voisinage

Pour la génération du voisinage, on procède à ce qu'on appelle la perturbation simple. Son principe est illustré dans l'algorithme suivant :

Algorithme 1 : L'algorithme recherche du voisinage

Entrée : L : séquence ordonnée initiale,
Sortie : L : séquence après permutation,
générer $a, b \in \{1, \dots, 25\}$;
Tant que a et b ne sont pas permutés **faire**
 générer $a, b \in \{1, \dots, 25\}$;
 Si a et b appartiennent au même job **alors**
 générer $a, b \in \{1, \dots, 25\}$;
 Sinon
 Si a et b vérifient la précédence avec la permutation **alors**
 On permute a et b dans L ;
 Sinon
 générer $a, b \in \{1, \dots, 25\}$;
 FinSi ;
 FinSi ;
FinTantque ;

4. Evaluation du voisinage

A cette étape, on évalue tous les voisins de la solution courante, ainsi le meilleur voisin non tabou sera sélectionné pour la prochaine itération. L'évaluation d'un voisin, $F(x)$, est la plus grande date d'achèvement des jobs C_{max} . Donc plus $F(x)$ est petit, meilleur est le voisin.

5. La liste Tabou

Pour éviter le piège des optimums locaux dans lequel le processus de recherche peut être facilement absorbé, le recherche tabou utilise une mémoire temporaire où elle sauvegarde les dernières solutions visitées. La liste Tabou est étendue par la meilleure solution trouvée, mais sa taille est fixe, donc si la liste Tabou est pleine, on remplace l'ancienne solution par la nouvelle.

6. Critère d'arrêt

L'algorithme de recherche tabou ainsi adapté peut être arrêté soit en fonction de la valeur de la solution courante, c-à-d, si l'algorithme n'améliore pas assez la valeur de la fonction, soit le nombre d'itérations est atteint.

Algorithme recherche Tabou

Nous adoptons les notations suivantes afin d'élaborer notre algorithme :

- S_{meil} : la meilleure solution trouvée et $EVL(S_{meil})$,
- S_c : la solution courante,
- S_v : la solution voisine de S_c ,
- S_{mv} : le meilleur voisin de S_c ,
- nb_{max} : le nombre d'itérations maximum sans amélioration,
- nbr : le nombre d'itérations sans amélioration.

Algorithme 2 : L'algorithme de la méthode recherche Tabou

Entrée : N : nombre d'itérations, Tl : taille de la liste Tabou, nv : nombre de voisin à générer à chaque itération.

Sortie : $S_{meil}, EVL(S_{meil})$

S_c = la solution initiale ;

$nbr = 0$;

$k = 1$;

$EVL(S_{meil}) = \infty$;

Tant que ($nbr < nb_{max}$) et ($k \leq N$) **faire**

$S_{mv} = \emptyset$; $EVL(S_{mv}) = \infty$;

Pour chaque voisin S_v de S_c non tabou **faire**

Si $F(S_v) < EVL(S_{mv})$ **alors**

$S_{mv} = S_v$;

$EVL(S_{mv}) = F(S_v)$;

Finsi ;

Fin pour ;

$S_c = S_{mv}$;

$nbr = nbr + 1$;

Si $EVL(S_{mv}) < EVL(S_{meil})$ **alors**

$S_{meil} = S_{mv}$;

$EVL(S_{meil}) = EVL(S_{mv})$;

Finsi ;

Mettre à jour la liste tabou

$k = k + 1$;

Fin Tant que ;

4.3 Application de l'algorithme génétique

Les algorithmes génétiques font partie de la famille des algorithmes évolutionnaires. Ils s'inspirent de l'évolution naturelle des espèces. Avec ce type de méthodes, il ne s'agit pas de trouver une solution analytique exacte mais de trouver une bonne solution satisfaisante dans un temps de calcul raisonnable. La première description du processus des algorithmes génétiques a été donnée par Holland en 1975, puis Goldberg (1989). Ils les ont utilisés pour résoudre des problèmes concrets d'optimisation [9].

Le but de ces algorithmes génétiques est d'optimiser une fonction prédéfinie, appelée fonction objectif, ou *fitness* ; ils travaillent sur un ensemble de solutions candidates, appelé "*population*" d'individus ou chromosomes (on utilisera indifféremment individu ou chromosome). Ces derniers sont constitués d'un ensemble d'éléments, appelés "*gènes*", qui peuvent prendre plusieurs valeurs, appelées "*allèles*"[9].

4.3.1 L'algorithme génétique adapté au problème du Job Shop Flexible

Les étapes principales de l'algorithme génétique adapté à notre problème du job shop flexible sont les suivantes :

1. Codage de la solution ;
2. Population initiale ;
3. Evaluation de la population ;
4. Selection ;
5. Croisement ;
6. Mutation ;
7. Insertion ;
8. Critère d'arrêt.

1. Codage de la solution

Le premier pas dans l'implantation des algorithmes génétiques est de créer une population d'individus initiaux. Chaque individu de la population est codé par un chromosome. Une population est donc un ensemble de chromosomes. Chaque chromosome code un point de l'espace de recherche. L'efficacité de l'algorithme génétique va donc dépendre du choix du codage d'un chromosome [9].

Deux types de codages ont été présentés dans la littérature : le codage direct et le codage indirect. Si le chromosome représente implicitement la solution, le codage est direct. Par contre, si le chromosome contient un ensemble d'informations spécifiques à un problème donné et la solution du problème est obtenue après une transformation, le codage est indirect. Dans notre cas, on a opté pour le codage direct, c-à-d, le chromosome est une chaîne de 25 éléments correspondant aux différentes tâches qu'on a. Un exemple de notre codage est :

| | | | | | | | | | | | | |
|----|----|---|----|---|---|----|----|----|----|----|----|-----|
| 21 | 23 | 2 | 16 | 3 | 7 | 13 | 24 | 25 | 14 | 11 | 18 | ... |
|----|----|---|----|---|---|----|----|----|----|----|----|-----|

| | | | | | | | | | | | | | |
|-----|----|---|----|----|---|---|----|---|---|----|----|----|---|
| ... | 17 | 1 | 22 | 19 | 8 | 5 | 20 | 4 | 9 | 12 | 10 | 15 | 6 |
|-----|----|---|----|----|---|---|----|---|---|----|----|----|---|

2. La population initiale

Il existe plusieurs mécanismes de génération de la population initiale. Le problème principal dans cette étape est le choix de la taille de la population. Si la taille de la population est trop grande, le temps de calcul augmente et demande un espace mémoire important. Par contre, une population de taille très petite, la solution obtenue n'est pas satisfaisante. Il faut donc trouver le bon compromis.

Dans notre cas, pour générer un individu de la population on fait appel au même programme utilisé pour la génération de la solution initiale de la recherche tabou (vu à la section 4.2.1). On génère alors N individus qui est la taille de la population.

3. Evaluation des individus

Chaque individus de la population se voit attribuer une évaluation qui représente sa force. l'évaluation d'un individu, dans notre cas, c'est la plus grande date d'achèvement des travaux. Donc, plus l'évaluation est petite, plus l'individu est fort.

4. Sélection

La sélection permet d'identifier les individus susceptibles d'être croisés dans une population. Il existe plusieurs techniques de sélection (vu à la section 2.7.2 partie b). Nous avons opté pour la sélection par rang, où on trie les individus par ordre croissant de leurs force ainsi on sélectionne les k premiers, qui est la taille de la selection (avec k paire).

5. Croisement

Le croisement permet d'enrichir la population en manipulant les composantes des chromosomes. Un croisement est envisagé avec deux parents et génère un ou deux enfants. Il est appliqué avec une probabilité P_c , appelée probabilité de croisement. Il existe plusieurs types de croisement (section 2.7.2 partie b). On a utilisé le croisement à deux points. Après la sélection de deux individus, nous générons un nombre aléatoire $a \in [0, 1]$, si $a \leq P_c$ alors on applique le croisement sur les deux parents choisis, sinon, on copie les parents dans les enfants.

6. Mutation

La mutation apporte l'aléa nécessaire à une exploration efficace de l'espace. Elle permet de quitter les extremas locaux [9]. Il existe aussi plusieurs type de mutation, nous avons donc utilisé où on permute entre deux tâches. On l'applique sur les deux enfants générés par le croisement. Cet opérateur de mutation est utilisé avec une probabilité P_m . On génère $b \in [0, 1]$, si $b \leq P_m$ alors on applique la mutation sur l'enfant 1, sinon, on l'applique sur l'enfant 2.

7. L'insertion

Après l'étape de mutation, on utilise une méthode d'insertion pour générer une nouvelle population. Il existe plusieurs stratégies dans la littérature [9]. Nous, on a choisit parmi la population initiale et la population générée, par croisement et mutation, les N meilleurs individus.

8. Le critère d'arrêt

Le test d'arrêt joue un rôle très important dans le jugement de la qualité des individus. Les critères d'arrêt sont de deux types :

- arrêt après un nombre fixé a priori de générations ;
- arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment.

L'algorithme de la méthode

Algorithme 3 : Algorithme génétique

Entrée : N, P_c, P_m, I_{max} : le nombre d'itérations maximum.

Sortie : ID_{best} : meilleur individu, $EVL(ID_{best})$: son évaluation.

Générer la population initiale ;

Calculer son évaluation ;

$i=0$;

Tant que $i \leq I_{max}$ **faire**

Pour k allant de 1 jusqu'à $N/2$ **faire**

 selectionner deux parents parmi les meilleurs ;

 générer $\alpha \in [0, 1]$;

Si $\alpha \leq P_c$ **alors**

 On fait le croisement des deux parents ;

Sinon

 On copie les deux parents dans les enfants ;

Finsi ;

 générer $\beta \in [0, 1]$;

Si $\beta \leq P_m$ **alors**

 On fait muter l'enfant 1 ;

Sinon

 On fait muter l'enfant 2 ;

Finsi ;

Fin pour ;

 Insertion des meilleurs individus ;

ID_{best} = meilleur individu trouver ;

$EVL(ID_{best})$ = l'évaluation du meilleur individu ;

$i = i + 1$;

Fin Tant que ;

4.4 Conclusion

Dans ce chapitre, nous avons présenté les deux métaheuristiques choisies pour la résolution du problème posé. Nous avons adapté l'algorithme génétique à notre situation où nous avons présenté de façon précise les différentes étapes de la méthode et les opérateurs appliqués afin de créer les nouvelles solutions. Nous avons également adapté la méthode recherche Taabou. Les résultats des deux méthodes ainsi qu'une comparaison seront présentés au prochain chapitre.

Chapitre 5

Tests et comparaison des résultats

La taille des problèmes d'ordonnancement et des modèles mathématiques rend toute tentative de résolution manuelle presque impossible. Par contre, en industrie, les entreprises préfèrent gagner du temps dans la résolution de ces problèmes en se contentant d'une solution quasi-optimale. Donc, la résolution par des métaheuristiques s'avèrent bénéfique pour les entreprises.

Au vu de l'importance de la taille de notre problème, nous avons eu recours à l'outil informatique. Nous avons implémenté les deux métaheuristiques considérées dans le chapitre précédent.

5.1 Présentation de Matlab

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

5.2 Résultats et tests

Nous allons expérimenter dans cette section les deux méthodes que nous avons élaboré auparavant à savoir : le recherche Tabou et l'algorithme génétique.

L'entreprise Meriplast travaille selon un plan de production issu directement des commandes journalières qu'elle reçoit. Pour tester notre programme, nous prenons comme exemple les données journalières fournit par l'entreprise.

5.2.1 Données d'entrée

Comme nous n'avons pas les données concernant la demande mais les données de production journalière du mois de mai 2017, nous les avons considéré comme demande et nous avons eu le tableau suivant :

| Produit | Quantité (tonnes) |
|---------|-------------------|
| FTN | 10 |
| FTI | 6 |
| FE | 6 |
| FAS | 8 |
| FPEtN | 5 |
| FPEtI | 8.5 |
| HN | 2 |
| HI | 1.5 |
| SN | 2 |
| SI | 2.5 |
| FCN | 4 |
| FCI | 5 |

TABLE 5.1 – Les données d'entrée

5.2.2 Les résultats obtenus par l'algorithme recherche Tabou

Nous avons remarqué qu'il est préférable d'exécuter plusieurs fois l'algorithme avec les mêmes paramètres. La variation des paramètres influence fortement la qualité des solutions et le temps d'exécution. En effet, si le nombre d'itérations dépasse 200, l'algorithme n'améliore pas la solution.

Après de nombreuses exécutions de l'algorithme, nous présentons dans le tableau suivant une des meilleures solutions avec les paramètres suivant :

- taille de la liste tabou : 9
- nombre de voisins : 6
- nombre d'itérations : 80

| Machine | Produit | date de début | date de fin |
|--------------|---------|-------------------|--------------------------|
| Extrudeuse 1 | HN | 0j : 00 : 00 : 00 | 1j : 04 : 14 : 07 |
| Extrudeuse 2 | SI | 0j : 00 : 00 : 00 | 1j : 06 : 00 : 00 |
| Extrudeuse 3 | HI | 0j : 00 : 00 : 00 | 1j : 18 : 21 : 10 |
| Extrudeuse 4 | SN | 0j : 00 : 00 : 00 | 0j : 06 : 51 : 25 |
| | FTN | 0j : 07 : 06 : 25 | 1j : 17 : 23 : 34 |
| Extrudeuse 5 | FE | 0j : 00 : 00 : 00 | 1j : 02 : 10 : 54 |
| Extrudeuse 6 | FPEtN | 0j : 00 : 00 : 00 | 0j : 10 : 00 : 00 |
| | FAS | 0j : 10 : 15 : 00 | 1j : 02 : 15 : 00 |
| Extrudeuse 7 | FTI | 0j : 00 : 00 : 00 | 0j : 05 : 45 : 36 |
| | FPEtI | 0j : 06 : 00 : 36 | 0j : 14 : 10 : 12 |
| Imprimante 1 | FCI | 0j : 00 : 00 : 00 | 0j : 10 : 00 : 00 |
| | FPEtI | 0j : 14 : 10 : 12 | 1j : 07 : 10 : 12 |
| | FTI | 1j : 07 : 25 : 12 | 1j : 19 : 25 : 12 |
| Imprimante 2 | SI | 1j : 06 : 00 : 00 | 1j : 11 : 00 : 00 |
| | HI | 1j : 18 : 21 : 10 | 1j : 21 : 21 : 10 |
| Sacherie | SN | 0j : 06 : 51 : 25 | 0j : 13 : 42 : 51 |
| | HN | 1j : 04 : 14 : 07 | 1j : 11 : 05 : 32 |
| | SI | 1j : 11 : 20 : 32 | 1j : 19 : 54 : 50 |
| | HI | 1j : 21 : 21 : 10 | 2j : 02 : 29 : 45 |
| Complexeuse | FCN | 0j : 00 : 00 : 00 | 0j : 03 : 29 : 27 |
| | FCI | 0j : 10 : 00 : 00 | 0j : 14 : 21 : 48 |
| Découpeuse 1 | FCI | 0j : 14 : 21 : 48 | 0j : 22 : 21 : 48 |
| Découpeuse 2 | FCN | 0j : 03 : 29 : 27 | 0j : 05 : 49 : 56 |
| | FTI | 1j : 19 : 25 : 12 | 1j : 22 : 55 : 55 |
| Découpeuse 3 | FPEtI | 1j : 07 : 10 : 12 | 1j : 10 : 52 : 44 |

TABLE 5.2 – Les résultats de la recherche Tabou

Le tableau précédent résume l'ensemble des jobs et leurs affectation aux différentes machines ainsi que leurs dates de début et de fin obtenus avec la recherche tabou. Par exemple, le FTI passe sur l'extrudeuse 7 à 0j : 00 : 00 : 00 et se termine à 0j : 05 : 45 : 36, ensuite, passe à l'imprimante 1 à 1j : 07 : 25 : 12 jusqu'à 1j : 19 : 25 : 12. Enfin, passe à la découpeuse 2 à 1j : 19 : 25 : 12 pour finir à 1j : 22 : 55 : 55. On remarque aussi que la plus grande date d'achèvement (Makespan) $C_{max} = 2j : 02 : 29 : 45$ qui correspond à la fabrication du HI.

5.2.3 Les résultats obtenus par l'algorithme génétique

À l'exécution de l'algorithme génétique, on remarque qu'il faut l'exécuter plusieurs fois pour obtenir de bonnes solutions. Tout comme pour la recherche tabou, les paramètres d'entrées de l'algorithme influent sur la qualité de la solution trouvée et le temps d'exécution. En effet, pour le nombre d'itérations, on est aller jusqu'à 2500 itérations sans amélioration de l'objectif à partir de 150. De même, pour les probabilités de croisement et de mutation, pour $P_m < 0.4$, l'algorithme fournit de bonnes solutions.

Une des meilleures solutions obtenues par l'algorithme, avec les paramètres suivants :

- taille de la population : 25,
- nombre d'itérations de l'algorithme : 80,
- probabilité de croisement P_c : 0.7,
- probabilité de mutation P_m : 0.3.

est illustrée dans le tableau suivant :

| Machine | Produit | date de début | date de fin |
|--------------|---------|-------------------|--------------------------|
| Extrudeuse 1 | SN | 0j : 00 : 00 : 00 | 1j : 04 : 14 : 07 |
| Extrudeuse 2 | SI | 0j : 00 : 00 : 00 | 1j : 06 : 00 : 00 |
| Extrudeuse 3 | HI | 0j : 00 : 00 : 00 | 1j : 18 : 21 : 10 |
| Extrudeuse 4 | FTN | 0j : 00 : 00 : 00 | 1j : 10 : 17 : 08 |
| Extrudeuse 5 | FPEtN | 0j : 00 : 00 : 00 | 0j : 21 : 49 : 05 |
| Extrudeuse 6 | FTI | 0j : 00 : 00 : 00 | 0j : 12 : 00 : 00 |
| | FE | 0j : 12 : 15 : 00 | 1j : 00 : 15 : 00 |
| Extrudeuse 7 | FPEtI | 0j : 00 : 00 : 00 | 0j : 08 : 09 : 36 |
| | HN | 0j : 08 : 24 : 36 | 0j : 10 : 19 : 48 |
| | FAS | 0j : 10 : 34 : 48 | 0j : 18 : 15 : 36 |
| Imprimante 1 | FPEtI | 0j : 08 : 09 : 36 | 1j : 01 : 09 : 36 |
| | FCI | 1j : 01 : 24 : 36 | 1j : 11 : 24 : 36 |
| | FTI | 1j : 11 : 39 : 36 | 1j : 23 : 39 : 36 |
| Imprimante 2 | SI | 1j : 06 : 00 : 00 | 1j : 11 : 00 : 00 |
| | HI | 1j : 18 : 21 : 10 | 1j : 21 : 21 : 10 |
| Sacherie | HN | 0j : 10 : 19 : 48 | 0j : 17 : 11 : 13 |
| | SN | 1j : 04 : 14 : 07 | 1j : 11 : 05 : 32 |
| | SI | 1j : 11 : 20 : 32 | 1j : 19 : 54 : 50 |
| | HI | 1j : 21 : 21 : 10 | 2j : 02 : 29 : 44 |
| Complexeuse | FCN | 0j : 00 : 00 : 00 | 0j : 03 : 29 : 27 |
| | FCI | 1j : 11 : 24 : 36 | 1j : 15 : 46 : 24 |
| Découpeuse 1 | FPEtI | 1j : 01 : 09 : 36 | 1j : 14 : 45 : 36 |
| Découpeuse 2 | FCN | 0j : 03 : 29 : 27 | 0j : 05 : 49 : 56 |
| | FTI | 1j : 23 : 39 : 36 | 2j : 03 : 10 : 19 |
| Découpeuse 3 | FCI | 1j : 15 : 46 : 24 | 1j : 17 : 57 : 19 |

TABLE 5.3 – Les résultats de l'algorithme génétique

Dans le tableau précédent, on a représenté l'un des meilleurs ordonnancements trouvé par l'algorithme génétique avec les dates de début et de fin pour chaque job ainsi que l'affectation des machines. Par exemple, le FPEtI passe par l'extrudeuse 7 à la date 0j : 00 : 00 : 00 et se termine à la date 0j : 08 : 09 : 36, ensuite, il passe par l'imprimante 1 à la date 0j : 08 : 09 : 36 et s'achève à la date 1j : 01 : 09 : 36. Enfin, à la date 1j : 01 : 09 : 36 passe à la découpeuse 1 pour finir à la date 1j : 14 : 45 : 36. Nous remarquons que la plus grande date d'achèvement $C_{max} = 2j : 03 : 10 : 19$ qui correspond à la production du FTI.

5.2.4 Comparaison des résultats

Le tableau suivant résume les résultats obtenus par les deux méthodes :

| | Makespan C_{max} |
|----------------------|--------------------|
| Recherche Tabou | 2j : 02 : 29 : 45 |
| Algorithme génétique | 2j : 03 : 10 : 19 |

En exécutant, plusieurs fois, les deux méthodes avec les mêmes données d'entrée (tableau 5.1), nous avons constaté que la méthode recherche Tabou produit de meilleurs résultats comparés à ceux de l'algorithme génétique. Mais, on remarque une certaine similitude des résultats à cause du caractère aléatoire des deux méthodes.

5.3 Evaluation des performances de la production

Afin d'évaluer les performances, du système de production de l'entreprise Meriplast, on fait varier la demande, puis on suit l'évolution du système. Pour cela, on va appliquer la méthode recherche tabou avec les même paramètres (décrit à la section 5.2.2). Dans un premier temps, on va comparer les quantités produites avec la méthode tabou sur 24H avec les capacités de production théorique. Puis, en second lieu, on va comparer les quantités produites par les deux unités au mois de mai 2017 (données fournies par l'entreprise). Enfin, nous prendrons les données du tableau 5.1 comme demande actuelle et nous les ferons varier.

5.3.1 Production par 24H avec la recherche Tabou

En utilisant l'ordonnancement trouvé par la méthode recherche tabou (tableau 5.2), nous allons extraire les quantités produites par 24H. Le résultat est dans le tableau suivant :

| Production | 0-24H (Tonnes) | 24H-48H (Tonnes) | Capacités (Tonnes/24H) |
|------------|----------------|-------------------|------------------------|
| M1 | 1.7 | 0.3 | 1.7 |
| M2 | 2 | 0.5 | 2 |
| M3 | 0.85 | 0.65 | 0.85 |
| M4 | 6.92 | 5.08 | 7 |
| M5 | 5.5 | 0.5 | 5.5 |
| M6 | 11.875 | 1.125 | 12 |
| M7 | 14.5 | 0 | 25 |
| M8 | 9.915 | 9.385 | 12 |
| M9 | 0 | 4 | 12 |
| M10 | 2 | 4.32 (reste 0.68) | 7 |
| M11 | 5 | 0 | 15 |
| M12 | 4 | 6 | 41 |
| M13 | 0 | 8.5 | 55 |
| M14 | 9 | 0 | 27.5 |

TABLE 5.4 – Les résultats de la production avec la méthode recherche Tabou

D'après le tableau précédent, on remarque que toutes les machines d'extrusion atteignent leurs capacités de production sauf la machine M7. Par contre, les autres machines n'ont pas encore atteint leurs capacités, à savoir les découpeuses, la complexeuse et les imprimantes.

5.3.2 Production réelle

Avec les données de production du mois de mai 2017 de l'entreprise, et les capacités des machines, on peut calculer les pourcentages d'utilisation des machines, comme le montre le tableau suivant (données du 7 mai 2017) :

| Machine | Quantité produite (Tonne) | Capacité (Tonne/24H) | Pourcentage d'utilisation |
|---------|---------------------------|----------------------|---------------------------|
| M1 | 1.866 | 1.7 | 109 |
| M2 | 1.543 | 2 | 77.15 |
| M3 | 0 | 0.85 | 0 |
| M4 | 5.544 | 7 | 79.2 |
| M5 | 4.62 | 5.5 | 84 |
| M6 | 11.301 | 12 | 94.17 |
| M7 | 10.677 | 25 | 42.70 |
| M8 | 6.907 | 12 | 57.55 |
| M9 | 10.907 | 12 | 90.89 |
| M10 | 2.48 | 7 | 35.42 |
| M11 | 0.459 | 15 | 3.06 |
| M12 | 4.63 | 41 | 11.29 |
| M13 | 10.572 | 55 | 19.22 |
| M14 | 3.954 | 27.5 | 14.37 |

TABLE 5.5 – Les pourcentages d'utilisation des machines

L'analyse de ce tableau montre que les anciennes machines (avec faibles capacités) sont utilisées au maximum de leurs capacités, mais les nouvelles machines (capacités élevées) leurs utilisation n'a pas encore atteint le maximum des capacités de production.

Une petite comparaison entre les deux tableaux 5.4 et 5.5, fait apparaitre que les résultats obtenus par la méthode Tabou sont supérieur, à ceux de la production réelle, ce qui peut être expliqué par le fait qu'on a négligé certains paramètres comme les délais et nous avons pris en considération les gammes de produits et non pas les produits.

5.3.3 Variation de la demande

Ici, nous allons faire varier la demande journalière totale et nous allons comparer la quantités produisent par rapport aux capacités de production des deux unités et nous nous arrêterons lorsque les capacités seront atteintes. La demande considérée est celle du tableau 5.1. Et les taux d'augmentation sont 10%, 25% et 40%. Les résultats sont résumés dans les tableaux suivants :

| Production | 0-24H (Tonnes) | 24H-48H (Tonnes) | Capacités (Tonnes/24H) |
|------------|----------------|-------------------|------------------------|
| M1 | 1.7 | 0.5 | 1.7 |
| M2 | 2 | 0.2 | 2 |
| M3 | 0.85 | 0.8 | 0.85 |
| M4 | 6.92 | 6.83 | 7 |
| M5 | 5.5 | 3.3 | 5.5 |
| M6 | 9.35 | 0 | 12 |
| M7 | 18.7 | 0 | 25 |
| M8 | 6.6 | 0.7(reste 0.95) | 12 |
| M9 | 10.9 | 6.7 | 12 |
| M10 | 2.11 | 5.04 (reste 1.65) | 7 |
| M11 | 0 | 6.6 | 15 |
| M12 | 0 | 9.35 | 41 |
| M13 | 9.9 | 0 | 55 |
| M14 | 9.9 | 0 | 27.5 |

TABLE 5.6 – Augmentation de la demande de 10%

| Production | 0-24H (T) | 24H-48H (T) | Capacités(T/24H) |
|------------|-----------|--------------------|------------------|
| M1 | 1.7 | 0.8 | 1.7 |
| M2 | 2 | 1.125 | 2 |
| M3 | 0.85 | 0.85 (reste 0.175) | 0.85 |
| M4 | 6.92 | 7 (reste 1.08) | 7 |
| M5 | 5.5 | 1.25 | 5.5 |
| M6 | 10.625 | 0 | 12 |
| M7 | 24.48 | 0.52 | 25 |
| M8 | 7.5 | 3.125 | 12 |
| M9 | 7.625 | 9.25 (reste 1.875) | 12 |
| M10 | 2.5 | 3.65 (reste 3.85) | 7 |
| M11 | 3.77 | 6.6 | 15 |
| M12 | 3.075 | 9.425 | 41 |
| M13 | 0 | 10.625 | 55 |
| M14 | 11.25 | 0 | 27.5 |

TABLE 5.7 – Augmentation de la demande de 25%

| Production | 0-24H (T) | 24H-48H (T) | 48H-72H (T) | Capacités (T/24H) |
|------------|-----------|-------------|-------------|-------------------|
| M1 | 1.7 | 1.1 | 0 | 1.7 |
| M2 | 2 | 1.5 | 0 | 2 |
| M3 | 0.85 | 0.85 | 0.4 | 0.85 |
| M4 | 6.92 | 7 | 2.88 | 7 |
| M5 | 5.5 | 1.5 | 0 | 5.5 |
| M6 | 11.9 | 0 | 0 | 12 |
| M7 | 24.48 | 3.52 | 0 | 25 |
| M8 | 7.97 | 0.43 | 0 | 12 |
| M9 | 7 | 11.9 | 2.1 | 12 |
| M10 | 2.8 | 2.47 | 5.93 | 7 |
| M11 | 2.43 | 4.56 | 0 | 15 |
| M12 | 0 | 14 | 0 | 41 |
| M13 | 0 | 0.9 | 11 | 55 |
| M14 | 11.17 | 1.42 | 0 | 27.5 |

TABLE 5.8 – Augmentation de la demande de 40%

5.3.4 Interprétation des résultats

Pour une augmentation de la demande de 10%, tableau 5.6, on remarque que les capacités de production des deux unités peuvent faire face. Les machines M1, M2, M3, M4 et M5 ont atteint leurs capacités. Par contre, les autres machines ne les ont pas encore atteint. Même constat si la demande augmente de 25%.

Enfin, en augmentant la demande totale jusqu'à 40%, nous remarquons que les extrudeuses et la sacherie ont atteint leurs capacités. Ce n'est pas le cas des autres machines. En particulier les découpeuses qui sont, largement, moins utilisées. Ce fait peut être expliqué par :

Les découpeuses ne fonctionnent pas toutes seules. Elles sont rattachées soit à la complexeuse, soit aux imprimantes. Puis, les imprimantes aussi sont utilisées soit après les extrudeuses ou après la sacherie. Et comme les extrudeuses et la sacherie ont atteint leurs capacités, donc on ne pourra pas atteindre les capacités des autres machines. C'est pour ça, qu'on arrête la variation de la demande.

5.4 Conclusion

Dans ce chapitre, nous avons, dans un premier temps, testé nos deux programmes, puis, nous avons comparé les résultats obtenus. Ensuite, nous sommes passés à l'évaluation des performances de la production, où nous avons comparé la production réelle par rapport aux résultats qu'on a trouvé. Enfin, nous avons déterminé le taux d'augmentation de la demande qui nécessitera des mesures spécifiques.

Conclusion générale

L'objectif de notre travail a été l'étude d'un problème de gestion de production relevé, durant notre stage, au sein de l'entreprise Meriplast.

Dans un premier lieu, nous nous sommes attachés à la description du problème, qui sera ensuite modélisé sous forme d'un modèle mathématique. Le modèle ainsi obtenu est un problème d'ordonnancement de type job shop flexible en considérant l'objectif de minimisation de la plus grande date d'achèvement des travaux. Comme ce type de problème appartient à la classe NP-Difficile, sa résolution par des méthodes exactes s'avère très complexe. Ceci nous a conduit aux méthodes approchées. Deux méthodes approchées ont été proposées : recherche Tabou et algorithme génétique, qui sont implémentées dans le logiciel MATLAB.

Dans un second lieu, nous avons testé les capacités des deux unités de production en faisant varier la demande totale de l'entreprise, où nous avons constaté qu'une augmentation de 40% de la demande engendrera une saturation des capacités des extrudeuses et de la sacherie, mais pas les capacités des autres machines.

Cette étude nous a permis de bien comprendre les lois internes de fonctionnement du système de production. Sur la base des résultats obtenus, les gestionnaires de l'entreprise Meriplast pourront tester plusieurs variantes de production pour faire face à l'augmentation de la demande. A titre d'exemple : réduire le cycle de production (au niveau des deux unités de production) et envisager l'implantation d'une autre unité de production plus rapprochée de certains clients et voir si cela a une incidence sur les coûts de distribution. De même, nous pouvons prendre en considération les délais et les coûts.

Bibliographie

- [1] Garnissages de rouleaux, complexage et enduction. http://www.hannecard.com/files/uploads/files/APP_LAM_Lamination_Coating_fr.pdf.
- [2] Les procédés d'impression. <http://perso.wanadoo.fr/elisabeth.fays/cours/>.
- [3] Z. Aoudia. *Cours d'optimisation combinatoire*, Université de Béjaia, 2015.
- [4] L. Azzabi. *Contribution à l'amélioration d'un système de production : intégration de la méthode six sigma et approche multicritère d'aide à la décision dans sidelec internationale*. PhD thesis, Ecole Doctorale De SFAX, 2010.
- [5] R.E. Bellman. *The Bellman continuum*. Editions Robert S. Roth, 1986.
- [6] M.L. Berrandjia. *Approche multi-agents pour l'ordonnancement par coopération multi-ressources avec délais de production*. PhD thesis, Ecole doctorale STIC, Sciences et Technologies de l'Information et de la Communication, 2015.
- [7] S. Boukari and H. Saigh. *Ordonnancement multicritère d'un atelier de production de type job shop flexible : cas de Cevital*, Mémoire d'ingénieur en Recherche Opérationnelle, Université de Béjaia, 2004.
- [8] J. Carlier and P. Chrétienne. *Problème d'ordonnancement*. Masson, Paris, 1988.
- [9] T. Chaari. *Un algorithme génétique pour l'ordonnancement robuste : application au problème du fow shop hybride*. PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2010.
- [10] Y. Collette and P. Siarry. *Optimisation Multiobjectif*. Editions Eyrolles, Paris, 2002.
- [11] D. Duvivier. *Etude de l'hybridation de méta-heuristiques, application à un problème d'ordonnancement de type jobshop*. PhD thesis, Université du littoral, côte d'opale, France, 2000.
- [12] M. Ennigrou and K. Ghedira. *Approche multi-agents basée sur la recherche tabou pour le job shop flexible*. PhD thesis, Institut préparatoire aux études d'Ingénieurs-ElManar, Tunisie, 2003.
- [13] P. Esquirol and P. Lopez. *L'ordonnancement*. Economica, 1999.
- [14] T. Fengal and Z. Ramdani. *Ordonnancement d'un atelier de production de type job shop flexible multicritère : cas de Danone Djurdjura Algérie*, Mémoire d'ingénieur en Recherche Opérationnelle, Université de Béjaia, 2005/2006.
- [15] Gotha. *Flexibilité et robustesse en ordonnancement*. Le bulletin de la ROADEF, 2002.
- [16] H. Jin-Kao, P. Galinier, and M. Habib. *Méthahéuristiques pour l'optimisation combinatoire et l'affectation sous contraintes*. *Revue d'Intelligence Artificielle*, 1999.

- [17] S. Kaci and K. Zidane. *Modélisation sous forme d'un Job shop Multi-ressources Bi-critères de l'Atelier de production Margarinerie-Cevital*, Mémoire de fin de cycle en Recherche Opérationnelle, Université de Béjaia, 2010.
- [18] A. Karray. *Contribution à l'ordonnancement d'ateliers agroalimentaires utilisant des méthodes d'optimisation hybrides*. PhD thesis, Ecole Centrale de Lille et l'Ecole Nationale d'Ingénieurs de Tunis, 2011.
- [19] C. Laquerbe, L. Pibouleau, P. Floquet, and S. Domenech. *Procédures stochastiques en génie des procédés : Méthode du recuit simulé et algorithmes génétiques*. 6^{ème} Colloque Maghrébin sur les Modèles Numériques de l'Ingénieur, Tunis, 1998.
- [20] P. Lopez and F. Roubellat. *Ordonnancement de la production*. HERMES Science Europe, 2001.
- [21] A. Malapert. *Techniques d'ordonnancement d'atelier et de fournées basées sur la programmation par contraintes*. PhD thesis, Ecole Nationale Supérieure des Techniques Industrielles et des Mines de Nantes, 2011.
- [22] K. Mellouli, A. El Kamel, and P. Borne. *Programmation linéaire et applications*. Editions Technip, 2004.
- [23] K. Merhoum and M. Djeghaba. *Algorithme génétique pour le problème d'ordonnancement de type job-shop*, Université Badji Mokhtar, Annaba, 2007.
- [24] S. Nigen. *Technologie de l'extrusion : Appareillages. Procédés. Défauts d'extrusion*. DUNOD, 2006.
- [25] H. Boukef Ben Othman. *Sur l'ordonnancement d'ateliers job-shop flexibles et flow-shop en industries pharmaceutiques Optimisation par algorithmes génétiques et essais particuliers*. PhD thesis, l'École Centrale de Lille et l'École Nationale d'Ingénieurs de Tunis, 2009.
- [26] M. Sakarovitch. *Graphes et Programmation Linéaire*. Edition Hermann, Paris, 1984.
- [27] E.G. Talbi, C. Cotta, and E. Alba. *Parallel hybrid metaheuristics, in parallel metaheuristics : A new class of algorithms*, Wiley-Interscience, 2005.
- [28] S. Tison. *Algorithmique Avancée et complexité*, Notes de cours, Université Lille1, 2009/2010.
- [29] L.H. Trung. *Utilisation d'ordres partiels pour la caractérisation de solutions robustes en ordonnancement*. PhD thesis, INSA, Toulouse, 2005.
- [30] Unknown. *Ordonnancement et systèmes multi-agents*. 2000.
- [31] M. Widmer. *Les Métaheuristiques : Des outils performants pour les problèmes industriels*. 3^{ème} Conférence Francophone de Modélisation et Simulation MOSIM'01, Troyes, 2001.
- [32] Q. Zhang. *Contribution à l'ordonnancement d'ateliers avec ressources de transports*. PhD thesis, Université de Technologie de Belfort-Montbéliard, 2012.

Résumé

Le but de ce travail est d'analyser la gestion de la production au sein de l'entreprise MERIPLAST.

Pour cela, nous avons ordonnancé les opérations des différents jobs et affecté les lignes de productions pour les différents produits de l'entreprise, en indiquant la date de début de chaque opération. Le problème a été modélisé sous forme d'un Job Shop Flexible en considérant comme objectif la minimisation de la plus grande date d'achèvement des jobs. Ce problème a été classé parmi les problèmes NP-difficiles. Pour cela, nous avons adapté deux métaheuristiques : les algorithmes génétiques et la recherche Tabou. Puis, nous avons évalué les capacités de production de l'entreprise.

Les résultats obtenus par les deux méthodes montrent que la méthode recherche Tabou donne de meilleurs résultats par rapport à l'algorithme génétique. En plus de la performance du système actuel, nous avons déterminé le taux d'augmentation de la demande qui nécessitera des mesures spécifiques.

Mots clés : *Entreprise Meriplast, gestion de production, Ordonnancement, Optimisation combinatoire, Job Shop Flexible, métaheuristiques, algorithme génétique, recherche Tabou, ...*

Abstract

The aim of this work is to analyze the management of production within the MERIPLAST company.

To do this, we have arranged the operations of the different jobs and assigned the production lines for the different products of the company, indicating the start date of each operation. The problem was modeled as a Flexible Job Shop with the goal of minimizing the biggest job completion date. This problem has been classified as NP-difficult. For this, we have adapted two metaheuristics : genetic algorithms and taboo search. Then we evaluated the production capacities of the company.

The results obtained by the two methods show that the tabu search method gives better results compared to the genetic algorithm. In addition to the performance of the current system, we have determined the rate of increase in demand that will require specific measures.

Keys words : *Meriplast company, production management, Scheduling, combinatorial optimization, Flexible Job Shop, metaheuristics, genetic algorithm, taboo search, ...*
