

الجمهورية الجزائرية الديمقراطية الشعبية
Republique Algerienne Democratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة تلمسان – أبي بكر بلقايد –
Université de Tlemcen – Aboubakr Belkaïd –
كلية التكنولوجيا
Faculté de Technologie
قسم الهندسة الصناعية
Département de Génie Industriel



Memoire de Projet de fin d'Etudes

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

En : Génie industriel, chaîne logistiques

Ordonnancement d'un atelier flow-shop en industries pharmaceutiques

Présenté Par : Benaïssa Ayoub

Soutenu le 29/06/2025 devant le jury composé de :

Presidente :	Mme Kouloughli Sihem	Professeur	Université de Tlemcen
Examineur :	Mr Hadri Abdelkader	M.C.B	Université de Tlemcen
Examinatrice :	Mme Houbad Yamina	M.C.B	Université de Tlemcen
Encadrant :	Mme.Laribi Imane	M.C.B	Université de Tlemcen

Année universitaire 2024/2025

REMERCEMENTS

D’abord, nous tenons à remercier Allah, notre créateur, pour nous avoir donné le courage, la santé, les moyens et la patience nécessaires pour accomplir ce modeste travail.

Je tiens à exprimer ma chaleureuse gratitude envers mon encadrante, Mme. Laribi Imane pour sa précieux encadrement, ses conseils éclairés, ses qualités humaines et ses compétences de chercheuses. Sa patience m’a permis de mener à bien cette mémoire. Je tiens à lui exprimer toute ma gratitude pour la confiance qu’elle m’a accordée.

Je souhaite également remercier les membres du jury, M. Hadri, Mme. Kouloughli, Mme. Houbad, pour leur intérêt et leur acceptation d’évaluer ce mémoire, ainsi que pour leurs remarques constructives qui m’ont permis d’améliorer mon travail.

Je remercie du fond du coeur ma famille pour leur soutien inconditionnel et leur patience tout au long de ces années d’études. Leur confiance et leurs encouragements ont été une source de motivation constante.

Je tiens également à exprimer ma reconnaissance envers mes amis et collègues de promotion pour leurs échanges intellectuels stimulants et leur camaraderie. Leur présence a considérablement enrichi mon expérience universitaire.

Enfin, je souhaite exprimer mes sincères remerciements et mon respect à tous les professeurs et au personnel administratif de Faculté de Technologie, Département Génie Industriel, pour leur dévouement et leur aide précieuse. Leur contribution a grandement facilité mon parcours académique.

À tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire, je vous adresse mes plus sincères remerciements.

الاهداء

وآخر دعواهم أن الحمد لله رب العالمين،

فالحمد لله حبًا وشكرًا وامتنانًا على البدء والختام.

من قال أنا لها نالها، وإن أبت رغمًا عنها أتيت بها، وها أنا أرى مرحلتي الدراسية قد شارفت على الانتهاء، عانقت اليوم مجدًا عظيمًا، فعلتها بعد تعب ومشقة دامت سنين في سبيل الحلم، حملت في طياتها آمانيات الليالي، وأصبح عنائي اليوم للعين قرّة، ها أنا اليوم أقف على عتبة تخرجني، أقطف ثمار تعبتي وأرفع قبعتي بكل فخر، فاللهم لك الحمد قبل أن ترضى، ولك الحمد إذا رضيت، ولك الحمد بعد الرضا.

وبكل حب أهدي ثمرة تخرجي ونجاحي:

إلى من جعل الله الجنة تحت قدميها، واحتضنني قلبها قبل يديها، وسهلت لي الشدائد بدعائها، من كانت الداعمة الأولى والأبدية، معلمتي الأولى، أمي الغالية حفظها الله.

إلى خيرة أيامي وصفوتها، ومن بوجودهم أكتسب قوة ومحبة لا حدود لها، إلى ضلعي الثابت، أماني أيامي، إخوتي: إسماعيل وأحمد خليل.

إلى رفقاء قلبي وأصدقاء الأيام المرة والحلوة، إلى من راهنوا على نجاحي وذكروني بمدى قوتي واستطاعتي، إخوتي الذين أنجبتهم الأيام.

إلى كل عائلتي، إلى الذين يبهجهم نجاحي، وكل من كان عونًا وسندًا في هذا الطريق، لأصدقاء ورفقاء السنين.

ما سلكنا البدايات إلا بتيسيره، وما بلغنا النهايات إلا بتوقيفه، وما حققنا الغايات إلا بفضلته، فالحمد لله.

L'ordonnancement de la production en environnement flow shop constitue un enjeu majeur dans l'industrie pharmaceutique, en raison des contraintes strictes de stérilisation et de compatibilité entre produits. Ce mémoire propose une approche combinée visant à minimiser le makespan tout en intégrant les spécificités de ce secteur.

Le problème est formalisé à l'aide d'un modèle mathématique de type programme linéaire, résolu avec CPLEX. En parallèle, des méthodes heuristiques sont développées, dont une adaptation de l'algorithme NEH et une heuristique dédiée, G-NEH-S, exploitant les regroupements de lots compatibles.

Les résultats expérimentaux mettent en évidence l'efficacité de l'heuristique G-NEH-S pour améliorer les performances de production, tout en garantissant le respect des contraintes de qualité et de sécurité imposées par l'industrie pharmaceutique.

Mots clé : Ordonnancement , Flow Shop , Stérilisation , Compatibilité , Heuristiques , Algorithme NEH , Algorithme G-NEH-S

Scheduling production in a flow shop environment is a major challenge in the pharmaceutical industry, due to strict sterilization and product compatibility constraints. This thesis proposes a combined approach aiming to minimize the makespan while addressing the specific requirements of this sector.

The problem is formulated as a Linear Program and solved using CPLEX. In parallel, heuristic methods are developed, including an adaptation of the NEH algorithm and a customized heuristic called G-NEH-S, which incorporates compatibility groupings.

Experimental results highlight the efficiency of the G-NEH-S heuristic in improving production performance, while ensuring compliance with the quality and safety standards specific to the pharmaceutical industry.

Keywords : Scheduling , Flow Shop , Sterilization , Compatibility , Heuristics , NEH Algorithm , G-NEH-S Algorithm

TABLE DES MATIÈRES

Remerciements	i
Dédicaces	ii
Résumé	iii
Abstract	iv
Introduction générale	x
1 Chapitre 01	1
Revue de littérature	1
1.1 Introduction	1
1.2 Ordonnancement de production	1
1.2.1 Ordonnancement flow shop	2
1.2.2 Caractéristiques du flow shop	2
1.2.3 Propriétés des problèmes d’ordonnancement flow shop	2
1.2.4 Typologie des variantes du Flow Shop : standard, avec temps de réglage, et avec blocage . . .	3
1.3 Contraintes spécifiques à l’industrie pharmaceutique	4
1.3.1 La contamination croisée : enjeux et risques	4
1.3.2 Processus de nettoyage et de stérilisation des équipements	4
1.3.3 Bonnes pratiques organisationnelles pour prévenir la contamination croisée	5

1.3.4	Définition des temps de stérilisation fixes et variables	6
1.3.5	Impact sur l'ordonnancement	7
1.3.6	Revue des approches prenant en compte la stérilisation et les incompatibilités de produits . . .	7
1.4	Algorithmes d'optimisation	7
1.4.1	Méthodes exactes	7
1.4.2	Heuristiques	8
1.4.3	Métaheuristiques	9
1.5	Motivation	10
1.6	Conclusion	11
2	Chapitre 02	12
	Modélisation et méthodes de résolution	12
2.1	Introduction	12
2.2	Une instance de problème	12
2.3	Modélisation mathématique	14
2.3.1	Paramètres et indices	14
2.3.2	Variables	14
2.3.3	Modèle	14
2.3.4	Signification des équations	15
2.4	Adaptation de NEH classique	16
2.4.1	Présentation de l'algorithme NEH	16
2.4.2	Adaptation pour le cas avec stérilisation	16
2.4.3	Exemple explicatif	16
2.5	Heuristique proposée G-NEH-S	18
2.5.1	Modélisation des compatibilités sous forme de graphe	19
2.5.2	Évaluation et tri des groupes	19
2.5.3	Ordonnancement inter-groupe, séquençement groupe par groupe	19
2.5.4	Ordonnancement intra-groupe : séquençement des lots compatibles	20
2.5.5	Optimisation globale par insertion	20

2.6	Mise en œuvre de l’heuristique G-NEH-S sur une instance du problème	21
2.6.1	Présentation de l’instance	21
2.6.2	Regroupement des lots compatibles	21
2.6.3	Évaluation et tri des groupes	22
2.6.4	Ordonnancement inter-groupe	22
2.6.5	Ordonnancement intra-groupe	23
2.6.6	Construction de la séquence finale	24
2.6.7	Optimisation globale par insertion	24
2.7	Conclusion	25
3	Chapitre 03	26
	Expérimentations et résultats	26
3.1	Introduction	26
3.2	Paramètre expérimentaux	26
3.2.1	Benchmark	26
3.2.2	Implémentation des méthodes	27
3.3	Résultats expérimentaux	27
3.4	Comparaison des performances	29
3.4.1	Comparaison sur les petites instances	29
3.4.2	Limites du solveur MILP (CPLEX)	29
3.4.3	Comparaison G-NEH-S vs NEH sur les grandes instances	29
3.4.4	Impact du taux d’incompatibilité	29
3.4.5	Comparaison des temps de calcul	30
3.5	Conclusion	30
	Conclusion générale	31
	Références	32

LIST OF FIGURES

2.1	Diagramme de Gantt d'une séquence aléatoire	13
2.2	Diagramme de Gantt de la séquence finale	18
2.3	Graphe de compatibilité des lots	22
2.4	Diagramme de Gantt de la séquence G_1G_2	23
2.5	Diagramme de Gantt de la séquence G_2G_1	23
2.6	Diagramme de Gantt de la meilleure séquence du G_1	23
2.7	Diagramme de Gantt de la meilleure séquence du G_2	24
2.8	Diagramme de Gantt de la séquence finale avec G-NEH-S	24

LIST OF TABLES

1.1	Méthodes de stérilisation couramment utilisées	5
2.1	Temps de traitement des lots sur chaque machine	13
2.2	Matrice d'incompatibilité entre les lots	13
2.3	Matrice d'incompatibilité entre les lots	21
3.1	Tableaux de résultats	28

INTRODUCTION GÉNÉRALE

L'ordonnancement de la production est une étape cruciales dans toute industrie manufacturière, et plus particulièrement dans l'industrie pharmaceutique, où les contraintes réglementaires, les exigences de qualité et les conditions de fabrication spécifiques rendent le problème particulièrement complexe. Dans ce contexte, l'atelier de type Flow Shop est une configuration fréquente, dans laquelle les lots suivent un enchaînement d'opérations identiques sur des machines alignées.

Le présent mémoire s'intéresse à l'ordonnancement d'un atelier Flow Shop en environnement pharmaceutique, en tenant compte de deux contraintes majeures : le respect des temps de stérilisation indispensables pour garantir la sécurité des produits, et la compatibilité des lots, visant à éviter les risques de contamination croisée. L'objectif principal est la minimisation du makespan (C_{max}), soit la durée totale nécessaire à la production de tous les lots.

Pour répondre à cette problématique, une approche mixte est adoptée, combinant une méthode exacte ainsi que des heuristiques développées sous MATLAB.

Ce travail est structuré comme suit :

Le premier chapitre propose une revue de la littérature sur les problèmes d'ordonnancement, les contraintes spécifiques à l'industrie pharmaceutique et les approches classiques de résolution des problèmes d'optimisation.

Le deuxième chapitre présente la modélisation formelle du problème étudié, l'adaptation d'une méthode existante et la proposition d'une heuristique dédiée prenant en compte les contraintes spécifiques.

Le troisième chapitre est consacré à l'expérimentation, à l'analyse comparative des méthodes et à l'évaluation des performances sur des instances.

Ce mémoire contribue à une meilleure compréhension et résolution des problèmes d'ordonnancement en environnement flow shop pharmaceutique, où les contraintes de stérilisation et de compatibilité rendent l'optimisation complexe, et où l'efficacité opérationnelle doit impérativement s'articuler avec les exigences de sécurité.

1.1 Introduction

L'ordonnancement de la production est un levier stratégique pour améliorer la performance industrielle, notamment dans les systèmes de type Flow Shop, où chaque lot suit le même enchaînement de machines. Ce type d'organisation est particulièrement présent dans l'industrie pharmaceutique, où les exigences de traçabilité, de stérilisation et de compatibilité entre produits imposent des contraintes supplémentaires à la planification.

Dans ce contexte, le respect des règles de stérilisation entre lots incompatibles devient incontournable afin de garantir la sécurité sanitaire. Ces exigences, combinées à des contraintes classiques comme les temps de traitement et la séquence imposée des opérations, rendent le problème d'ordonnancement plus complexe et difficile à résoudre par des approches naïves ou manuelles.

Face à ces défis, les méthodes d'optimisation, qu'elles soient exactes ou heuristiques, offrent des outils efficaces pour rechercher des solutions de qualité en un temps raisonnable. Leur application permet non seulement d'améliorer l'efficacité de la production, mais aussi de garantir le respect des normes réglementaires strictes.

Ce chapitre pose ainsi les bases du problème étudié : il présente le contexte de l'ordonnancement en environnement Flow Shop, les contraintes spécifiques à l'industrie pharmaceutique, ainsi que les principales approches méthodologiques mobilisées pour y répondre.

1.2 Ordonnancement de production

L'ordonnancement de production constitue un enjeu stratégique au sein de l'ensemble des processus de l'industrie. C'est la recherche de l'ordre optimal dans lequel les tâches doivent être effectuées sur un grand nombre de ressources disponibles (machines, postes de travail, lignes de production) dans le but d'optimiser un ou plusieurs critères tels que le délai, le coût ou encore le temps total de production (makespan).

Dans le cadre d'une industrie moderne, qui fait face à une forte variabilité de la demande, à des contraintes de production hétérogènes ainsi qu'à une intensification de la compétition sur les délais, l'optimisation de l'ordonnancement est un facteur clé important pour accroître la performance globale d'une entreprise. C'est le cas d'un optimiseur du

temps d'attente, d'utilisation des machines, de respect des délais clients, mais également des inactivités productives lors des changements de séries.

L'ordonnancement peut s'appliquer sur toutes les strates d'une organisation, à partir de la planification stratégique jusqu'à l'exécution opérationnelle au niveau de l'atelier. Il s'effectue tout en demeurant dans diverses limites : capacité restrictive, machine hors service pour un certain temps, chapiteau, maintenance, ou même limites particulières au type de produit tels que dans le secteur pharmaceutique ou agroalimentaire.

1.2.1 Ordonnancement flow shop

L'ordonnancement Flow Shop, ou atelier à flux linéaire, se présente comme un modèle classique de production d'une entreprise, dans lequel un lot de n tâches (ou lots) doit s'effectuer dans un certain ordre sur m machines ou postes de travail successifs. Ce modèle d'organisation est très répandu dans les lignes automatiques ou semi-automatiques, comme celles que l'on trouve dans les industries de transformation, chimie, pharmacie.

Dans un Flow Shop classique, chacune des tâches accomplies sur les machines est effectuée dans le même ordre par toutes les machines. Cette organisation reflète bien les processus de production de masse, où les opérations sont le plus souvent séquentielles et répétitives par nature.

L'accent principal en ce qui concerne l'ordonnancement dans ce contexte particulier serait plus souvent tourné vers la réduction du makespan, qui est le temps total nécessaire pour accomplir toutes les tâches. Cependant, d'autres facteurs, tels que le retard total, le temps d'attente moyen, l'utilisation moyenne des machines, etc., pourraient également être pris en considération.

1.2.2 Caractéristiques du flow shop

Les principales caractéristiques de ce modèle sont : Chaque tâche suit un chemin identique à travers les machines. Les temps de traitement sont généralement connus à l'avance. Les machines ne peuvent traiter qu'une seule tâche à la fois. Les tâches ne peuvent pas être interrompues une fois commencées (non préemptive).

Ce modèle, bien que simplifié par rapport à la réalité industrielle, reste un cadre de référence fondamental pour la recherche en ordonnancement et permet le développement et l'analyse d'approches d'optimisation efficaces.

1.2.3 Propriétés des problèmes d'ordonnancement flow shop

Les problèmes d'ordonnancement, en particulier ceux de type Flow Shop, appartiennent à une classe de problèmes connus pour être NP-difficiles (Non-polynômial temps-difficiles) pour plus de deux machines. Cela indique qu'à l'heure actuelle, aucun algorithme n'est connu pour résoudre ces problèmes en temps polynomial pour tous les cas possibles dans un scénario donné. Cela signifie qu'à mesure que la taille du problème augmente, par exemple, le nombre de tâches, de machines ou de contraintes, l'explosion combinatoire associée à la résolution du problème s'intensifie.

Cela capture la nature du problème, qui entraîne simultanément un nombre significatif de permutations devant être testées pour trouver la solution optimale. Dans ce cas, l'objectif est de minimiser le makespan. Dans un problème de permutation Flow Shop avec n tâches, par exemple, les permutations possibles sont $n!$. Lorsque n est égal à 10, les séquences requises seraient 3 628 800. De plus, des contraintes supplémentaires telles que les temps de préparation, les incompatibilités et les timings de stérilisation compliquent encore davantage le problème déjà complexe. Le cadre de complexité du problème de Flow Shop avec un temps de résolution augmentant exponentiellement en fonction de la taille du problème.

1.2.4 Typologie des variantes du Flow Shop : standard, avec temps de réglage, et avec blocage

Malgré la simplicité structurelle du modèle de base du Flow Shop, les exigences industrielles engendrent fréquemment des contraintes additionnelles qui conduisent à divers dérivés du problème.

Chacun d'eux complexifie le traitement tout en augmentant la pertinence du modèle pour son utilisation en pratique. On identifie généralement les variantes suivantes :

1.2.4.1 Flow shop standard

C'est la forme la plus basique, où chaque travail est effectué sur tous les équipements dans une séquence définie, sans délais de préparation spécifiques ni contraintes spéciales.

On suppose que les tâches peuvent être transférées directement d'un appareil à l'autre, sans problème lié au stockage temporaire.

1.2.4.2 Flow shop avec setup times

Dans plusieurs secteurs industriels, la transition de production entre deux lots consécutifs nécessite un temps de préparation (ou de nettoyage) sur certaines machines.

On peut avoir les temps de configuration suivants :

- Constantes indépendantes de la séquence ;
- Dépendants de la série, c'est-à-dire que le délai de préparation est conditionné par la nature du lot précédemment géré (situation courante dans les secteurs de la pharmacie, des peintures ou de l'agroalimentaire).

Cette version rend le problème plus conforme aux conditions réelles, mais extrêmement plus ardu à résoudre.

1.2.4.3 Blocking flow shop

Dans certains contextes, il n'y a pas de zones de stockage temporaires entre les équipements (absence de tampons). Si la prochaine machine n'est pas accessible, la tâche actuellement en cours immobilise la machine en place jusqu'à ce que la suivante soit disponible.

Cette entrave a un impact significatif sur la productivité générale et l'organisation des tâches devient alors essentielle pour prévenir les engorgements.

1.2.4.4 No-wait flow shop

Dans ce scénario, il est impératif qu'il n'y ait pas de pause entre deux opérations consécutives : une fois qu'une tâche débute, elle doit être exécutée de manière continue sans rupture.

Ce modèle est caractéristique des processus où le facteur temps est crucial, comme dans le cas des réactions en chaîne en chimie ou de certaines procédures stériles en pharmacie.

Ces formes du Flow Shop offrent une modélisation plus précise des restrictions opérationnelles dans divers domaines industriels. Concernant l'industrie pharmaceutique, certaines de ces contraintes, en particulier les temps de préparation et les interdictions de stockage, revêtent une importance spécifique. Nous le démontrerons dans la prochaine section.

1.3 Contraintes spécifiques à l'industrie pharmaceutique

En industrie pharmaceutique, la qualité des produits fabriqués est soumise à des normes très strictes, notamment en ce qui concerne la stérilité des équipements de production. La contamination croisée, définie comme le transfert non désiré de substances actives ou de micro-organismes entre différents lots ou produits, représente un risque majeur pouvant compromettre la sécurité et l'efficacité des médicaments. Ce risque est particulièrement critique lorsque plusieurs types de produits, tels que des antibiotiques et des vaccins, sont fabriqués successivement sur les mêmes équipements.

L'objectif de cette section est de présenter les méthodes de stérilisation appliquées aux machines industrielles pharmaceutiques ainsi que les bonnes pratiques organisationnelles et techniques permettant d'éviter la contamination croisée.

1.3.1 La contamination croisée : enjeux et risques

La contamination croisée peut avoir plusieurs origines :

- Résidus de substances actives ou excipients d'un lot précédemment fabriqué,
- Présence de micro-organismes ou endotoxines,
- Transfert de particules ou poussières entre zones de production.

Les conséquences potentielles incluent :

- Altération de la qualité et de la pureté des produits,
- Risques sanitaires pour le patient,
- Non-conformité réglementaire pouvant mener à des retraits de lots,
- Pertes économiques importantes pour l'entreprise.

1.3.2 Processus de nettoyage et de stérilisation des équipements

1.3.2.1 Nettoyage

Le nettoyage consiste à éliminer toute trace visible de matières organiques, de poudres, ou de résidus chimiques sur les surfaces des équipements. Il s'agit d'une étape primordiale car la stérilisation ne peut être efficace qu'après un nettoyage minutieux.

Techniques de nettoyage :

- Nettoyage manuel,
- Nettoyage en place (CIP – Clean-In-Place), particulièrement adapté aux circuits fermés (tuyauteries, cuves),

- Utilisation de détergents spécifiques, compatibles avec les matériaux et les produits manipulés.
- Rinçage à l'eau purifiée ou à l'eau pour injection (WFI).

1.3.2.2 Désinfection

Après nettoyage, la désinfection vise à réduire la charge microbienne résiduelle en utilisant des agents chimiques (e.g., solutions à base d'alcool, peroxyde d'hydrogène) ou thermiques. Elle prépare les surfaces à la stérilisation.

1.3.2.3 Stérilisation

La stérilisation est l'étape clé qui détruit tous les micro-organismes viables, incluant les spores bactériennes.

Méthodes couramment utilisées :

Méthode	Description	Application typique
Stérilisation à la vapeur saturée (autoclavage / SIP – Steam-In-Place)	Exposition à de la vapeur d'eau sous pression (121°C pendant 15 à 30 minutes). Adaptée aux matériaux résistants (inox).	Cuves, tuyauteries, filtres et équipements métalliques
Chaleur sèche	Chauffage prolongé à haute température (160-180°C) dans un four dédié.	Verrerie, instruments métalliques
Stérilisation chimique par peroxyde d'hydrogène vaporisé	Diffusion de peroxyde d'hydrogène sous forme de vapeur ou brouillard pour désinfection et stérilisation.	Isolateurs, capteurs, surfaces en plastique
Stérilisation par rayonnement (UV, gamma)	Utilisée principalement pour emballages ou dispositifs médicaux. Usage limité sur équipements industriels.	Emballages, dispositifs médicaux à usage unique

Table 1.1: Méthodes de stérilisation couramment utilisées

1.3.3 Bonnes pratiques organisationnelles pour prévenir la contamination croisée

Au-delà des méthodes techniques, des mesures organisationnelles doivent être strictement respectées pour assurer la maîtrise de la contamination :

- Production par campagnes : regrouper la fabrication de lots similaires afin de minimiser les changements de produits et réduire les risques de contamination.
- Zonage et flux unidirectionnels : séparation physique des zones selon le degré de contamination (zones propres, zones sales) et gestion des flux.
- Validation des procédures de nettoyage et stérilisation : protocoles validés par analyses microbiologiques et chimiques.
- Traçabilité rigoureuse : registres précis des opérations de nettoyage, désinfection et stérilisation.
- Systèmes fermés et isolateurs : pour limiter l'exposition des produits et équipements à l'environnement.

1.3.3.1 Contrôles et surveillance microbiologique

Le maintien d'un environnement stérile est contrôlé par des analyses régulières :

- Tests microbiologiques : prélèvements sur surfaces, équipements et air.
- Tests d'endotoxines : vérification de l'absence de toxines bactériennes.
- Analyses de résidus chimiques : garantir l'absence de contamination chimique croisée.

Ces contrôles valident la conformité aux bonnes pratiques de fabrication (BPF / GMP).

En résumé, la stérilisation des machines en industrie pharmaceutique est un processus complexe, combinant techniques physiques et chimiques adaptées, avec des mesures organisationnelles rigoureuses. La prévention de la contamination croisée nécessite un nettoyage préalable efficace, une stérilisation adaptée, des bonnes pratiques rigoureuses et une surveillance microbiologique constante. La maîtrise de ces éléments est indispensable pour assurer la qualité et la sécurité des produits pharmaceutiques.

1.3.4 Définition des temps de stérilisation fixes et variables

Dans le secteur pharmaceutique, la procédure de stérilisation des matériels entre deux fabrications est cruciale pour prévenir toute contamination croisée et assurer la qualité des produits réalisés. Ce processus nécessite des interruptions durant lesquelles les équipements ou les chaînes de production sont stérilisés, et ces interruptions peuvent fluctuer selon divers éléments. On peut les diviser en deux grandes catégories : les temps de stérilisation fixes et les temps de stérilisation variables.

1.3.4.1 Temps de stérilisation fixes

Les temps de stérilisation fixes correspondent à des intervalles préétablis et fixes nécessaires pour réaliser le nettoyage et la désinfection des équipements, indépendamment des conditions de production ou des articles manipulés. Ces délais sont habituellement déterminés en se basant sur les protocoles standards prescrits par les bonnes pratiques de fabrication (BPF), et ils demeurent constants d'un lot à l'autre.

Par exemple, si un dispositif nécessite une stérilisation après chaque fabrication d'un lot d'antibiotiques et d'un lot de vaccins, une durée fixe de stérilisation de 30 minutes pourrait être nécessaire pour assurer l'élimination complète de toute trace de produit. Cette période est fixée à l'avance et n'est pas influencée par le genre de produit ou la quantité produite, procurant donc une certaine prévisibilité dans la planification.

1.3.4.2 Temps de stérilisation variables

Ils sont tributaires de divers éléments : Le genre de produit fabriqué : certains articles, du fait de leur composition chimique ou de leur concentration en agents actifs, peuvent exiger des processus de nettoyage plus étendus ou plus compliqués.

Le volume de déchets à traiter : des lots plus imposants ou plus compliqués peuvent produire une plus grande quantité de déchets, prolongeant par conséquent la durée de stérilisation requise.

Les paramètres de production spécifiques : des éléments comme l'humidité, la température et le type d'équipement peuvent avoir un impact sur l'efficacité du processus de nettoyage et donc, le temps requis pour la stérilisation. Dans

cette situation, les durées de stérilisation sont moins prévisibles et nécessitent des ajustements basés sur les attributs de chaque lot, ce qui rend la planification plus complexe.

Par exemple, si un équipement est employé pour fabriquer un médicament puissant, le temps de nettoyage peut être prolongé afin d'assurer l'élimination totale des résidus, contrairement à un produit classique.

1.3.5 Impact sur l'ordonnancement

Les durées de stérilisation, qu'elles soient fixes ou variables, influencent directement la planification de la production. Elles prolongent le temps de production total (makespan), diminuent l'efficacité des ressources et imposent des restrictions supplémentaires sur l'ordre des tâches. Il est donc essentiel d'intégrer ces périodes de stérilisation dans les modèles de planification lors de la programmation. On peut intégrer des variables spécifiques dans le modèle d'optimisation pour réduire leur incidence, tel que l'emploi de groupes de compatibilité pour rassembler les lots qui requièrent des durées de stérilisation comparables.

1.3.6 Revue des approches prenant en compte la stérilisation et les incompatibilités de produits

Ces dernières années, plusieurs travaux de recherche ont mis en évidence la complexité de la planification en environnement pharmaceutique, notamment lorsqu'il s'agit de prendre en compte des contraintes de stérilisation et d'incompatibilité entre produits. La thèse de Boukef Ben Othman (2009) aborde ce sujet en combinant des environnements flowshop et jobshop flexibles, tout en intégrant des contraintes de changement liées à la nature des produits traités. D'autres études, comme celle de Rossi et Lanzetta (2013), se sont intéressées à la gestion de la stérilisation dans un cadre hospitalier, en modélisant le processus comme un flowshop à deux étapes (lavage puis stérilisation), avec des heuristiques spécifiques pour améliorer l'efficacité de l'ordonnancement. Plus récemment, Hertrich et Weiss (2020) ont étudié un problème similaire dans un contexte industriel biopharmaceutique, en tenant compte des incompatibilités entre lots et des contraintes de regroupement (batching), montrant l'intérêt d'algorithmes adaptés pour optimiser le temps de production.

1.4 Algorithmes d'optimisation

Dans l'industrie pharmaceutique, l'ordonnancement de la production, notamment en tenant compte de contraintes spécifiques telles que la contamination croisée, les périodes de stérilisation et les ensembles de compatibilité, requiert le recours à des techniques d'optimisation sophistiquées.

Ces techniques facilitent la recherche de solutions optimales ou presque optimales pour des problèmes complexes, qui sont souvent ardues à déchiffrer avec des méthodes précises du fait de l'ampleur considérable des instances et des diverses contraintes.

1.4.1 Méthodes exactes

Les techniques exactes assurent la découverte d'une solution optimale, cependant elles s'avèrent souvent inapplicables pour des problèmes de grande envergure à cause de leur coût élevé en temps de calcul. Les techniques les plus fréquemment employées dans l'établissement du planning de production parmi ces méthodes sont :

1.4.1.1 Programmation linéaire

la programmation linéaire en nombres entiers mixtes (MILP) est largement utilisée pour optimiser des processus complexes tels que la planification de la production, la gestion des stocks, la distribution des médicaments et l'ordonnancement des opérations en salle blanche. Elle permet de modéliser avec précision les contraintes spécifiques du secteur, comme les temps de stérilisation, les incompatibilités entre produits, les capacités limitées des machines, ou encore les délais de livraison stricts.

Grâce à MILP, les décideurs peuvent élaborer des plannings efficaces qui respectent les normes de qualité et de sécurité tout en minimisant les coûts opérationnels. Cette approche mathématique joue un rôle clé dans l'amélioration de la performance et de la traçabilité des chaînes de production pharmaceutiques.

1.4.1.2 Branch & Bound

Branch & Bound est une méthode de recherche organisée qui permet d'obtenir la solution optimale en parcourant toutes les solutions envisageables, tout en supprimant efficacement celles qui ne peuvent pas conduire à une meilleure option que celle déjà identifiée comme étant la plus performante. Cette technique est particulièrement efficace pour traiter des problèmes combinatoires sophistiqués, tels que ceux liés à la planification avec diverses contraintes (durée de stérilisation, incompatibilités, etc.).

Bien que cette méthode soit précise, elle exige également beaucoup de puissance de calcul, notamment pour des problèmes volumineux, ce qui la rend fréquemment non viable dans des contextes industriels pratiques.

1.4.2 Heuristiques

Les heuristiques sont des techniques d'optimisation visant à trouver des solutions assez satisfaisantes dans un délai raisonnable, sans toutefois assurer leur caractère optimal.

Elles conviennent particulièrement aux problèmes de grande envergure, où les méthodes précises s'avèrent trop lentes ou onéreuses en termes de calcul.

1.4.2.1 NEH(Nawaz-Enscore-Ham)

L'algorithme NEH (Nawaz et al. (1983)) est une heuristique largement utilisée pour les problèmes d'ordonnancement, en particulier dans le contexte des problèmes de Flow Shop. Cet algorithme élabore progressivement une solution en intégrant une tâche dans une séquence partiellement ordonnée afin de réduire le makespan. Il opère en trois phases :

1. Classer les tâches selon leur durée totale d'exécution.
2. Intégrer chaque tâche, individuellement, à l'endroit le plus approprié dans la séquence existante.
3. Continuez cette insertion jusqu'à ce que toutes les tâches soient disposées.

L'algorithme NEH est performant et fréquemment employé dans les secteurs requérant des réponses rapides, cependant il n'assure pas la solution idéale.

1.4.2.2 L'algorithme de CDS

Une autre heuristique largement utilisée pour les problèmes d'ordonnancement, notamment dans le cadre du Flow Shop, est l'algorithme CDS (Campbell et al. (1970)). Il se base sur l'élaboration de solutions en deux étapes :

- La première étape vise à **classer les tâches** selon divers critères (comme le temps de traitement),
 - Alors que la deuxième étape cherche à déterminer la séquence idéale en permutant les tâches dans l'ordre suggéré.
- Cet algorithme est aussi fréquemment employé dans les contextes industriels, du fait de sa capacité à générer rapidement des solutions de qualité, même pour des instances de grande envergure.

1.4.2.3 L'algorithme de Palmer

Une autre heuristique employée pour traiter les enjeux du Flow Shop est l'algorithme Palmer (Palmer (1965)). Il est spécialement conçu pour les enjeux où les tâches doivent transiter par plusieurs machines dans une séquence préétablie. L'algorithme détermine un indice de priorité pour chaque tâche, puis les organise en fonction de cet indice.

Les tâches ayant une priorité supérieure sont gérées en premier lieu, ce qui contribue à créer une chaîne de production efficace.

Cet algorithme est performant, néanmoins, à l'instar des autres heuristiques, il n'assure pas une solution optimale.

1.4.3 Métaheuristiques

Les métaheuristiques sont des techniques d'optimisation globale qui parcourent un vaste domaine de solutions afin d'identifier des résolutions à peu près optimales, notamment pour les problèmes complexes.

Ces techniques sont aptes à gérer des problématiques avec différentes restrictions, telles que les durées de stérilisation, les incompatibilités et les exigences de nettoyage.

1.4.3.1 Recuit simulé(simulated annealing)

Le recuit simulé est une technique inspirée du refroidissement des métaux, où l'on ajuste progressivement les solutions afin de dénicher un minimum global. Cette méthode s'avère particulièrement efficace pour les problèmes combinatoires complexes, tels que l'ordonnancement, car elle facilite une exploration performante de l'espace des solutions.

L'utilisation du recuit simulé dans la planification pharmaceutique permet de gérer des contraintes complexes, comme les périodes de nettoyage et de stérilisation, tout en prévenant les minima locaux.

1.4.3.2 Recherche tabou(Tabu search)

L'optimisation par recherche tabou est une approche qui fait usage d'une mémoire pour empêcher la redécouverte de solutions déjà analysées. Cette méthode favorise l'évasion des minima locaux et l'exploration de nouvelles solutions.

Elle est employée pour traiter des problèmes de planification où plusieurs restrictions doivent être considérées. La recherche tabou est performante pour les problèmes d'envergure et peut être ajustée pour relever les défis spécifiques associés aux restrictions de production dans le secteur pharmaceutique.

1.4.3.3 Algorithme génétique(AG)

Les techniques d'optimisation basées sur les mécanismes de sélection naturelle sont connues sous le nom d'algorithmes génétiques. On les utilise pour résoudre des problèmes dans lesquels l'espace des solutions est à la fois vaste et complexe. En reproduisant la progression de populations de solutions, ces algorithmes ont la capacité de produire des solutions à proximité de l'optimal pour des problématiques d'envergure, comme l'ordonnancement intégrant les contraintes de stérilisation, de compatibilité et de nettoyage.

Les algorithmes génétiques sont spécialement conçus pour les problématiques d'ordonnancement pharmaceutique, car ils sont capables de gérer efficacement des contraintes complexes tout en fournissant rapidement des solutions de haut niveau.

1.4.3.4 Optimisation par essaim de particule(PSO)

L'optimisation par essaim de particules (PSO, pour Particle Swarm Optimization) est une technique d'optimisation tirée du comportement des colonies d'animaux, tels que les oiseaux en vol. Elle exploite un ensemble de solutions envisageables (particules) se déplaçant dans l'espace solution pour dénicher la meilleure option.

Cette approche est pertinente pour les problèmes d'organisation complexes, y compris ceux nécessitant des restrictions supplémentaires, comme c'est le cas dans le secteur pharmaceutique, et peut fournir de solides résultats dans des contextes de production évolutifs.

1.5 Motivation

Dans l'industrie pharmaceutique, les lignes de production suivent souvent une organisation de type flow shop, où plusieurs lots de produits sont traités de manière séquentielle sur les mêmes équipements. Lorsque ces lots appartiennent à des familles incompatibles — par exemple, des antibiotiques et des vaccins — le risque de contamination croisée devient un enjeu critique. Ce risque impose l'obligation de stériliser l'ensemble des machines entre deux lots incompatibles, ce qui entraîne des arrêts supplémentaires et allonge la durée de production.

La contrainte de stérilisation constitue donc un aspect central dans l'ordonnancement des lots. Elle est à la fois une exigence de qualité et de sécurité, un impératif réglementaire imposé par les bonnes pratiques de fabrication (GMP), et un facteur ayant un impact opérationnel direct sur le makespan. Ignorer cette contrainte peut conduire à des plannings irréalistes, voire non conformes.

Pour répondre à cette problématique spécifique, une double approche est proposée dans ce travail : d'une part, une modélisation mathématique rigoureuse du problème sous forme d'un programme linéaire en variables mixtes (MILP) intégrant explicitement la contrainte de stérilisation ; d'autre part, une heuristique dédiée, conçue spécifiquement pour ce contexte. Cette heuristique s'appuie sur la structure des incompatibilités entre lots pour guider la construction de séquences plus efficaces, avec pour objectif de réduire les temps de stérilisation tout en maintenant un bon niveau de performance globale.

Cette démarche permet de générer des solutions à la fois réalistes, conformes aux exigences du secteur, et adaptées aux contraintes industrielles.

1.6 Conclusion

Ce chapitre a permis de poser le cadre général du problème d’ordonnancement en environnement Flow Shop, en mettant l’accent sur les particularités du domaine pharmaceutique. Les contraintes liées à la stérilisation, au nettoyage et à l’incompatibilité entre produits rendent la planification plus complexe qu’en environnement industriel classique. Plusieurs approches ont été explorées dans la littérature, mais elles montrent parfois des limites lorsqu’il s’agit d’intégrer ces contraintes spécifiques.

C’est ce constat qui justifie l’intérêt d’adapter ou de développer des méthodes plus ciblées. Le chapitre suivant s’inscrit dans cette logique : il présente une modélisation mathématique du problème, ainsi que deux heuristiques de résolution, dont une variante personnalisée de NEH pensée pour mieux répondre aux exigences du secteur pharmaceutique.

2.1 Introduction

La modélisation rigoureuse d'un problème d'ordonnancement constitue une étape essentielle pour développer des méthodes de résolution efficaces et adaptées. Dans le cadre de l'industrie pharmaceutique, cette tâche se complexifie en raison de la multiplicité des contraintes à intégrer, notamment celles liées à la stérilisation et à l'incompatibilité des lots.

Ce chapitre a pour objectif de formuler mathématiquement le problème d'ordonnancement dans un atelier Flow Shop en intégrant les spécificités du contexte pharmaceutique. Une formulation MILP est tout d'abord proposée afin de formaliser l'objectif de minimisation du makespan tout en respectant les contraintes de séquençement, d'incompatibilité et de stérilisation.

Par la suite, une adaptation de l'algorithme heuristique NEH est présentée, prenant en compte les contraintes de compatibilité et les temps de stérilisation, avant d'introduire une nouvelle heuristique appelée G-NEH-S, construite sur une approche par groupes de lots compatibles.

Ce chapitre détaille les méthodes de résolution développées, ainsi que leur mise en œuvre sur une instance de test représentative.

2.2 Une instance de problème

Pour mieux comprendre le problème étudié, nous proposons une instance illustrative appliquée à un environnement de production pharmaceutique. Il s'agit d'un problème d'ordonnancement de type Flow Shop de permutation, dans lequel plusieurs lots doivent être traités successivement sur une série de machines, tout en respectant des contraintes spécifiques liées à l'incompatibilité des produits et aux exigences de stérilisation.

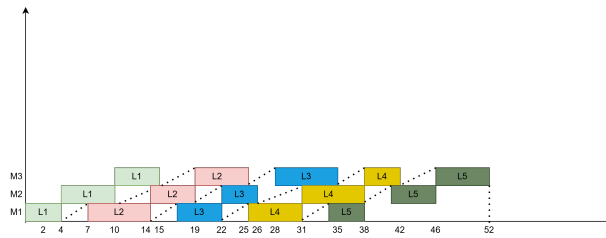


Figure 2.1: Diagramme de Gantt d'une séquence aléatoire

Dans cet exemple, cinq lots de production L_1 à L_5 doivent être planifiés sur trois machines M_1 , M_2 et M_3 . Chaque lot appartient à une famille de produits (par exemple, antibiotique ou vaccin), et certaines paires de lots sont considérées comme incompatibles. Lorsqu'un lot est suivi d'un autre jugé incompatible, une opération de stérilisation d'une durée fixe est requise avant de continuer le traitement, afin d'éviter toute contamination croisée.

Cette instance met en évidence la complexité du problème : il ne s'agit pas seulement de minimiser le makespan, mais aussi de tenir compte des interruptions dues à la stérilisation et des contraintes de compatibilité, qui influencent fortement l'ordonnancement optimal.

Pour illustrer cette situation, nous présentons les données de l'instance sous forme de tableaux :

Tableau des temps de traitement (Tab. 2.1) indiquant, en unité de temps, la durée nécessaire pour traiter chaque lot L_i sur chaque machine M_j .

Lot	M1	M2	M3
L1 (A)	4	6	5
L2 (B)	7	5	6
L3 (A)	5	4	7
L4 (B)	6	7	4
L5 (A)	4	5	6

Table 2.1: Temps de traitement des lots sur chaque machine

Tableau d'incompatibilité (Tab. 2.2): 1 signifie que les lots L_i et L_j sont incompatibles (et nécessitent une stérilisation entre eux) ; sinon zéro signifie pas de stérilisation.

	L1	L2	L3	L4	L5
L1	0	1	0	1	0
L2	1	0	1	0	1
L3	0	1	0	1	0
L4	1	0	1	0	1
L5	0	1	0	1	0

Table 2.2: Matrice d'incompatibilité entre les lots

Temps de stérilisation fixe $T=3$ est imposé entre deux lots incompatibles dans cette instance.

On considère une séquence aléatoire comme L_1, L_2, L_3, L_4, L_5 , les transitions entre lots incompatibles peuvent être identifiées à l'aide du tableau Tab.2. 1, et la durée totale de production (makespan) varie selon la position des stérilisations. Un diagramme de Gantt permettrait ici de visualiser clairement la répartition des traitements et des temps de stérilisation.

2.3 Modélisation mathématique

Cette partie introduit un modèle de programmation linéaire en nombres entiers mixtes (MILP) pour aborder le problème d'ordonnancement Flow Shop tenant compte des temps de stérilisation liés à l'incompatibilité entre lots.

2.3.1 Paramètres et indices

- n : nombre de lots.
- m : nombre de machines.
- i : indice de lot, où $i \in \{1, \dots, n\}$.
- j : indice de machine, où $j \in \{1, \dots, m\}$.
- k : indice de position dans la séquence, où $k \in \{1, \dots, n\}$.
- T : temps de stérilisation.
- $P_{i,j}$: temps de traitement du lot i sur la machine j .
- $y_{i,i'} \in \{0, 1\}$: indicateur de compatibilité entre les lots i et i' (0 = incompatibles, 1 = compatibles).

2.3.2 Variables

- $X_{i,k}$: variable binaire, égale à 1 si le lot i se trouve à la position k dans la séquence, 0 sinon.
- $W_{i,i',k}$: égale à 1 si le lot i est à la position k et le lot i' est à la position $k + 1$, 0 sinon.
- Z_k : variable binaire, égale à 1 si une stérilisation est effectuée entre les positions k et $k + 1$, 0 sinon.
- $S_{k,j}$: date de début du lot en position k sur la machine j .

2.3.3 Modèle

En utilisant les paramètres et les variables décrits précédemment, le problème peut être modélisé de la façon suivante :

$$(1) \text{ Min } (S_{n,m} + \sum_{i=1}^n X_{i,n} \cdot P_{i,m})$$

Sous les contraintes :

$$\begin{aligned}
(2) \quad & \sum_{k \in K} X_{ik} = 1, & \forall i \in I \\
(3) \quad & \sum_{i \in I} X_{ik} = 1, & \forall k \in K \\
(4) \quad & W_{ii'k} \geq X_{ik} + X_{i'(k+1)} - 1, & \forall i, i' \in I, k = 1, \dots, n-1 \\
(5) \quad & Z_k \geq \sum_{i \in I} \sum_{i' \in I} y_{ii'} \cdot W_{ii'k}, & \forall k = 1, \dots, n-1 \\
(6) \quad & S_{11} = 0 \\
(7) \quad & S_{(k+1)1} \geq S_{k1} + \sum_{i \in I} X_{ik} \cdot P_{i1} + T \cdot Z_k, & \forall k = 1, \dots, n-1 \\
(8) \quad & S_{kj} \geq S_{k(j-1)} + \sum_{i \in I} X_{ik} \cdot P_{i(j-1)}, & \forall k \in K, j = 2, \dots, m \\
(9) \quad & S_{(k+1)j} \geq S_{kj} + \sum_{i \in I} X_{ik} \cdot P_{ij} + T \cdot Z_k, & \forall k = 1, \dots, n-1, j = 2, \dots, m
\end{aligned}$$

2.3.4 Signification des équations

Nous donnons dans ce paragraphe la signification de chacune des contraintes de notre modèle :

- **Fonction 1** : représente la fonction objectif du problème. Elle consiste à minimiser le temps d'exécution maximum, appelé *makespan* ou $C_{\max}$, qui correspond à la date de début du lot en dernière position sur la dernière machine, plus son temps d'exécution.
- **Contrainte 2** : chaque position k de la séquence est occupée par un seul lot .
- **Contrainte 3** : chaque lot i est assigné à une seule position dans la séquence .
- **Contrainte 4** : cette contrainte construit les variables $W_{i,i',k}$ qui valent 1 si le lot i est placé en position k et le lot i' en position $k+1$.Elle permet d'identifier les couples de lots consécutifs, étape essentielle pour détecter les cas d'incompatibilité.
- **Contrainte 5** : déduit la variable Z_k , qui devient 1 si deux lots incompatibles sont placés successivement (en positions k et $k+1$) Cela modélise la nécessité d'une stérilisation entre les deux lots , conformément à la matrice d'incompatibilité $y_{ii'}$.
- **Contrainte 6** : cette contrainte fixe le temps de début du premier lot sur la première machine à zéro posant ainsi l'origine temporelle de l'ordonnancement.
- **Contrainte 7** : gère la propagation des temps sur la première machine : chaque lot ne peut commencer qu'après la fin du précédent , avec l'ajout du temps de stérilisation si une incompatibilité est détectée.
- **Contrainte 8** : impose la continuité verticale sur les autres machines ($j \geq 2$) : un lot placé à la même position k ne peut débuter sur la machine j qu'après avoir terminé sur la machine $j-1$.
- **Contrainte 9** : traite la continuité horizontale pour les machines $j \geq 2$: un lot placé en position $k+1$ ne peut commencer sur la machine j qu'après la fin du lot précédent en position k , avec ajout du temps de stérilisation si nécessaire .

2.4 Adaptation de NEH classique

2.4.1 Présentation de l'algorithme NEH

L'algorithme NEH est reconnu comme l'une des heuristiques les plus performantes pour aborder les problématiques d'ordonnancement Flow Shop à permutation, en particulier pour réduire le makespan (temps global de production). Sa performance dépend d'une élaboration progressive et méthodique de la séquence idéale. Les étapes de l'algorithme standard NEH :

1. Estimation du poids de chaque lot : addition de ses durées de traitement sur l'ensemble des machines.
2. Classement décroissant des lots en fonction de leur poids.
3. Élaboration progressive de la séquence :
 - (a) Positionner le second lot de façon optimale par rapport au premier.
 - (b) À chaque itération, placez le lot suivant dans toutes les positions accessibles de la séquence actuelle.
 - (c) Déterminez le makespan pour chaque position et conservez l'option la plus favorable.
4. Continuer jusqu'à obtenir la séquence complète.

2.4.2 Adaptation pour le cas avec stérilisation

Dans le domaine pharmaceutique, des incompatibilités existent entre certains produits, nécessitant un délai de stérilisation fixe entre deux lots de différentes catégories (par exemple, passant du vaccin à l'antibiotique). Ainsi, l'algorithme NEH est ajusté pour inclure cette contrainte :

À chaque ajout de lot dans la séquence, on contrôle les types du lot précédent et du lot nouvellement ajouté.

Si ces types sont différents, on introduit un temps de stérilisation.

2.4.3 Exemple explicatif

Pour illustrer concrètement le fonctionnement de l'algorithme NEH, nous appliquons ses différentes étapes sur une instance simplifiée composée de cinq lots à ordonnancer sur trois machines. Cette instance, déjà présentée dans la section 2.2, intègre une contrainte de stérilisation fixe qui s'applique lorsqu'un lot est suivi d'un autre issu d'une famille incompatible. L'objectif est de montrer, étape par étape, comment l'algorithme construit une séquence de production efficace tout en tenant compte de cette contrainte supplémentaire.

2.4.3.1 calcul du poids de chaque lot (somme des temps de traitement sur M1,M2,M3)

- $L_1 = 15$
- $L_2 = 18$
- $L_3 = 16$
- $L_4 = 15$
- $L_5 = 15$

2.4.3.2 Tri décroissant des lots selon le poids

Ordre des lots après tri : L_2, L_4, L_3, L_1, L_5

2.4.3.3 Construction progressive de la séquence, en partant de L_2

Insérention de L_4 , donc soit $[L_2, L_4]$ ou $[L_4, L_2]$ Pour chaque séquence, on calcule le makespan en ajoutant un temps fixe de stérilisation $T=3$ lorsque nécessaire entre les lots incompatibles.

La séquence qui donne le makespan le plus faible est retenue comme séquence globale. Pour $[L_2, L_4]$ le makespan trouver est de : 24 Pour $[L_4, L_2]$ le makespan trouver est de : 24 Egalité; on choisit $[L_4, L_2]$.

2.4.3.4 Insertion de L_3 dans $[L_4, L_2]$

Tester les positions suivantes:

$[L_3, L_4, L_2]$

$[L_4, L_3, L_2]$

$[L_4, L_2, L_3]$ Pour chaque séquence, on calcule le makespan en ajoutant un temps fixe de stérilisation $T=3$ lorsque nécessaire entre les lots incompatibles. Pour $[L_3, L_4, L_2]$, le makespan trouver est de : 31

Pour $[L_4, L_3, L_2]$, le makespan trouver est de : 36

Pour $[L_4, L_2, L_3]$, le makespan trouver est de : 33

Donc la séquence retenue est $[L_3, L_4, L_2]$

2.4.3.5 Insertion de L_1 dans $[L_3, L_4, L_2]$

Tester les positions suivantes:

$[L_1, L_3, L_4, L_2]$

$[L_3, L_1, L_4, L_2]$

$[L_3, L_4, L_1, L_2]$

$[L_3, L_4, L_2, L_1]$ Pour chaque séquence, on calcule le makespan en ajoutant un temps fixe de stérilisation $T=3$ lorsque nécessaire entre les lots incompatibles.

Pour $[L_1, L_3, L_4, L_2]$, le makespan trouver est de : 35

Pour $[L_3, L_1, L_4, L_2]$, le makespan trouver est de : 35

Pour $[L_3, L_4, L_1, L_2]$, le makespan trouver est de : 44

Pour $[L_3, L_4, L_2, L_1]$, le makespan trouver est de : 40 On remarque, une égalité entre les deux premières séquences, on choisit $[L_1, L_3, L_4, L_2]$ ou $[L_3, L_1, L_4, L_2]$.

2.4.3.6 Insertion de L_5 dans $[L_3, L_1, L_4, L_2]$

Tester les positions suivantes:

$$[L_5, L_3, L_1, L_4, L_2]$$

$$[L_3, L_5, L_1, L_4, L_2]$$

$$[L_3, L_1, L_5, L_4, L_2]$$

$$[L_3, L_1, L_4, L_5, L_2]$$

$$[L_3, L_1, L_4, L_2, L_5]$$

Lorsqu'on a calculé le makespan de chaque séquence, on a trouvé que la séquence retenue est $[L_5, L_3, L_1, L_4, L_2]$, le makespan trouver est de : 40.

Cette séquence est donc la meilleure trouvée par l'algorithme NEH , en tenant compte à la fois des temps de traitement et des stérilisations dues aux incompatibilités. La figure. 2.2 schématise le digramme de Gantt de la meilleure séquence obtenue avec NEH classique.

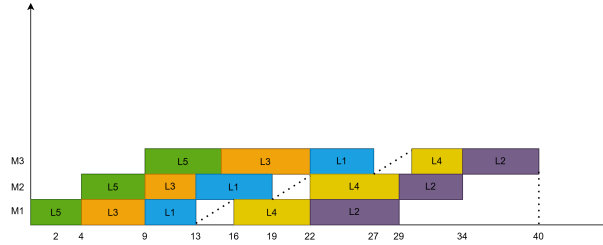


Figure 2.2: Diagramme de Gantt de la séquence finale

2.5 Heuristique proposée G-NEH-S

Les problèmes d'ordonnancement en flow shop deviennent particulièrement complexes lorsqu'ils intègrent des contraintes industrielles spécifiques. L'une des plus critiques dans les secteurs comme la pharmaceutique est la contrainte de stérilisation entre lots incompatibles.

Lorsqu'un lot succède à un autre avec lequel il n'est pas compatible (du point de vue de la sécurité ou de la qualité), une opération de stérilisation s'impose, entraînant des pertes de temps non négligeables.

Classiquement, les heuristiques d'ordonnancement comme NEH ne tiennent pas compte de ce type de contrainte. C'est dans cette optique que nous proposons une heuristique originale, nommée G-NEH-S (pour Grouped NEH with Stérilisation), qui adapte la logique de NEH à un contexte industriel enrichi par des contraintes de compatibilité.

L'idée centrale repose sur une stratégie en deux niveaux : regrouper intelligemment les lots compatibles, puis ordonnancer ces groupes de façon à minimiser à la fois le makespan et les opérations de stérilisation.

L'heuristique proposée comporte les étapes suivantes:

2.5.1 Modélisation des compatibilités sous forme de graphe

La première étape consiste à exploiter la matrice de compatibilité $Y \in \{0, 1\}$ de taille $n \times n$, où $Y_{ii}'=1$ signifie qu'une stérilisation est obligatoire entre les lots i et i' , et $Y_{ij}=0$ signifie qu'ils sont compatibles.

Pour exploiter cette donnée efficacement, on construit un graphe non orienté de compatibilité $G=(V,E)$, dans lequel :

- chaque nœud représente un lot,
- une arête relie deux lots si ceux-ci sont compatibles (i.e., $Y_{ii}'=0$).

On procède ensuite à une exploration de ce graphe par l'application de l'algorithme de recherche en profondeur (DFS) pour identifier les composantes connexes du graphe, chacune correspondant à un groupe de lots compatibles entre eux.

Ce regroupement permet d'éviter toute stérilisation à l'intérieur d'un même groupe, ce qui constitue un gain significatif.

2.5.2 Évaluation et tri des groupes

Une fois les groupes de lots compatibles identifiés, l'objectif est de déterminer un ordre d'insertion pertinent dans la séquence globale.

Pour cela, on évalue le poids de chaque groupe, défini comme la somme des temps de traitement des lots qui le composent, toutes machines confondues. Formellement, pour un groupe G_k contenant les lots $L_1, L_2, \dots, L_n$. On calcule :

$$w_{G_k} = \sum_{L_i \in G_k} \sum_{j=1}^m P_{ij}$$

Les groupes sont ensuite triés par ordre décroissant de poids. Cette stratégie de tri priorise les groupes les plus "lourds", conformément à la logique de l'heuristique NEH classique, en cherchant à positionner en tête les groupes les plus contraignants pour optimiser globalement le makespan.

2.5.3 Ordonnancement inter-groupe, séquençement groupe par groupe

Dans cette étape, l'objectif est de construire une séquence globale des groupes de lots incompatibles. Une fois les groupes de lots compatibles $G_1, G_2, \dots, G_K$ constitués à l'étape précédente sont triés en fonction de leur poids w_{G_k} , la séquence finale est construite de manière itérative selon une logique d'insertion incrémentale.

La construction de la séquence s'effectue de manière itérative. On commence par insérer le groupe ayant le poids maximal, G_1 , en première position. Puis, à chaque étape k , le groupe G_k est inséré dans la séquence courante à la position qui minimise le makespan total. Cependant, contrairement à l'algorithme NEH classique, l'originalité de notre approche repose sur l'introduction systématique d'une opération de stérilisation fixe entre deux groupes successifs, car les groupes ont été construits en partant du principe qu'ils sont mutuellement incompatibles.

Ainsi, pour chaque insertion, un temps fixe de stérilisation T est ajouté avant le début du traitement du groupe G_k . Ce surcoût est systématique et ne dépend pas d'une vérification de compatibilité.

Le groupe G_k est inséré à toutes les positions possibles dans la séquence, chaque fois précédé d'un temps de stérilisation, et on retient la position qui donne le plus petit makespan. Ce processus se répète jusqu'à ce que tous les groupes soient ordonnés. Le résultat final est une séquence $S=(G_{\Pi_1}, G_{\Pi_2}, \dots, G_{\Pi_k})$, dans laquelle chaque transition entre deux groupes inclut un temps de stérilisation, garantissant ainsi le respect des contraintes d'incompatibilité entre lots.

La séquence obtenue à ce stade constitue une base hiérarchique cohérente, dans laquelle chaque groupe est positionné en respectant les exigences de compatibilité et les contraintes de stérilisation. La phase suivante s'intéresse désormais à l'ordonnancement interne des lots au sein de chaque groupe, dans le but de raffiner encore davantage le séquençement à une échelle plus fine.

2.5.4 Ordonnancement intra-groupe : séquençement des lots compatibles

Une fois la séquence inter-groupe établie, chaque groupe G_k , composé exclusivement de lots compatibles entre eux, peut être traité de manière indépendante. L'objectif de cette étape est d'optimiser la séquence interne de chaque groupe, c'est-à-dire l'ordre d'exécution des lots $L_i \in G_k$, de manière à minimiser la contribution de chaque groupe au makespan global.

Contrairement à l'étape précédente, aucune opération de stérilisation n'est requise à l'intérieur d'un groupe. Cette propriété permet de modéliser chaque groupe comme une instance standard du problème flow shop de permutation, sur lequel des heuristiques classiques peuvent être appliquées efficacement. Plusieurs stratégies peuvent être envisagées à ce niveau, mais nous adoptons une approche heuristique basée sur l'adaptation locale de NEH.

On calcule d'abord un poids individuel pour chaque lot $L_i \in G_k$, défini par la somme de ses temps de traitement sur l'ensemble des machines : $w_i = \sum p_{ij}$

Les lots du groupe sont ensuite triés par ordre décroissant de w_i , puis insérés un à un dans la position de la séquence partielle qui minimise le makespan du groupe. Ce processus est répété pour chaque groupe G_k indépendamment, ce qui permet de raffiner le plan de production en intégrant une optimisation locale au sein de chaque bloc compatible.

En construisant les séquences groupe par groupe, on affine progressivement la solution globale tout en respectant les contraintes de compatibilité. Cette approche modulaire simplifie la complexité du problème initial, en le divisant en sous-problèmes plus faciles à résoudre, ce qui permet d'améliorer efficacement la qualité de l'ordonnancement final.

2.5.5 Optimisation globale par insertion

L'optimisation intra-groupe, une première séquence globale est obtenue en concaténant les groupes de lots selon l'ordre établi à l'étape précédente. Cette solution, bien que respectueuse des contraintes de compatibilité, n'est pas nécessairement optimale vis-à-vis du critère de makespan. Afin d'améliorer la qualité de cet ordonnancement, une phase d'ajustement global par insertion est introduite.

Cette étape consiste à déplacer chaque lot individuellement dans toutes les positions possibles de la séquence courante. À chaque insertion testée, le makespan est recalculé en tenant compte des éventuelles opérations de stérilisation induites par des changements de type entre lots successifs. Si une nouvelle position permet de réduire le temps d'achèvement total, le déplacement est conservé. Ce processus est appliqué successivement à l'ensemble des lots.

Contrairement à la logique initiale de regroupement par compatibilité, cette phase permet de déplacer un lot hors de son groupe si cela améliore l'ordonnancement général. Cela peut entraîner davantage de stérilisations, mais le gain obtenu sur le temps total de production peut compenser ce coût. Ce processus d'ajustement permet d'explorer différentes possibilités autour de la solution précédemment construite et d'améliorer le résultat final en trouvant un meilleur équilibre entre les durées de traitement et les contraintes de stérilisation.

L'heuristique G-NEH-S proposé permet de traiter efficacement le problème d'ordonnancement flow shop avec contraintes de stérilisation entre lots incompatibles. En combinant une phase de regroupement des lots compatibles, un ordonnancement inter-groupe tenant compte des opérations de stérilisation, puis un ordonnancement intra-groupe, cette méthode offre une approche structurée et modulaire.

Cette décomposition progressive facilite la gestion des contraintes complexes tout en limitant la taille des sous-problèmes à résoudre, ce qui rend l'heuristique à la fois pragmatique et performante.

2.6 Mise en œuvre de l'heuristique G-NEH-S sur une instance du problème

Pour illustrer le fonctionnement de l'heuristique G-NEH-S, nous présentons ici son application à une instance représentative du problème d'ordonnancement flow shop avec contraintes de stérilisation. Cette instance comporte un nombre limité de lots et de machines afin de faciliter la compréhension de chaque étape de l'algorithme.

2.6.1 Présentation de l'instance

Dans cette partie, on reprend la même instance précédente définie dans la section 2.2.

2.6.2 Regroupement des lots compatibles

Pour respecter les contraintes de stérilisation, nous utilisons la matrice de compatibilité Y définie ainsi :

- $Y_{ii}' = 1$ signifie que les lots L_i et L_i' sont incompatibles et nécessitent une opération de stérilisation,
- $Y_{ii}' = 0$ signifie qu'ils sont compatibles et peuvent se succéder sans stérilisation.

Dans notre exemple, la règle est que les lots de types différents sont incompatibles, alors que ceux du même type sont compatibles.

La matrice Y s'écrit donc :

	L_1	L_2	L_3	L_4	L_5
L_1	0	1	0	1	0
L_2	1	0	1	0	1
L_3	0	1	0	1	0
L_4	1	0	1	0	1
L_5	0	1	0	1	0

Table 2.3: Matrice d'incompatibilité entre les lots

À partir de cette matrice, on construit un graphe où chaque lot est un nœud, et une arête relie deux lots s'ils sont compatibles



Figure 2.3: Graphe de compatibilité des lots

L'analyse du graphe permet d'identifier deux groupes de lots compatibles :

- Groupe $G_1 = L_1, L_3, L_5$, correspondant aux lots de type A.
- Groupe $G_2 = L_2, L_4$ correspondant aux lots de type B.

2.6.3 Évaluation et tri des groupes

Une fois les groupes compatibles identifiés, l'objectif est de déterminer un ordre pour les enchaîner dans la séquence globale, en tenant compte de leurs poids respectifs. On calcule donc :

- Pour $G_1 = L_1, L_3, L_5$
 $w_{G_1} = (4+6+5) + (5+4+7) + (4+5+6) = 15+16+15 = 46$
- Pour $G_2 = L_2, L_4$

$$w_{G_2} = (7+5+6) + (6+7+4) = 18+17 = 35$$

Ensuite, on trie les groupes par ordre décroissant de poids :

$$G_1(46) > G_2(35)$$

2.6.4 Ordonnancement inter-groupe

Avec seulement deux groupes incompatibles, G_1 et G_2 , l'objectif est de déterminer la meilleure séquence entre ces deux groupes.

On teste les deux ordres possibles : (G_1, G_2) ou (G_2, G_1) . Pour chaque ordre, on calcule le makespan en ajoutant un temps fixe de stérilisation $T=3$ entre les deux groupes, car ils sont incompatibles.

La séquence qui donne le makespan le plus faible est retenue comme séquence globale.

- Pour (G_1, G_2) le makespan trouver est de : 41

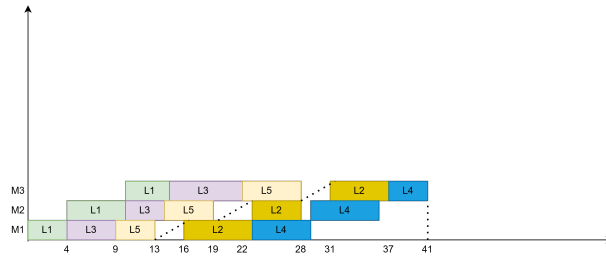


Figure 2.4: Diagramme de Gantt de la séquence G_1G_2

- Pour (G_2, G_1) le makespan trouver est de : 47

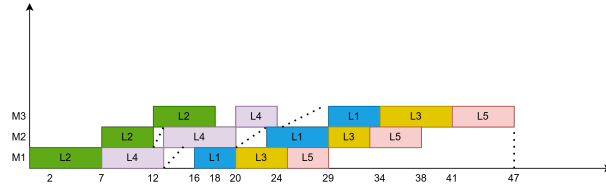


Figure 2.5: Diagramme de Gantt de la séquence G_2G_1

Donc la séquence retenue est : (G_1, G_2)

2.6.5 Ordonnancement intra-groupe

Pour optimiser l'ordre des lots dans chaque groupe, on applique l'heuristique NEH localement.

- Pour G_1 la meilleure sequence trouver avec NEH est $[L_5, L_3, L_1]$ avec un makespan de 27

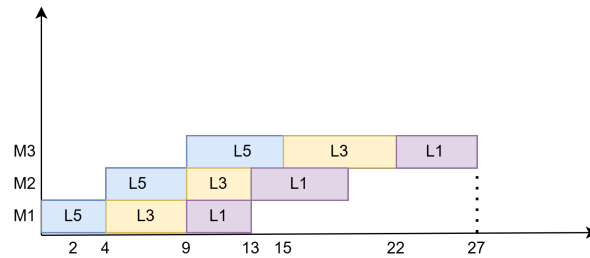


Figure 2.6: Diagramme de Gantt de la meilleure séquence du G_1

- Pour G_2 la meilleure sequence trouver avec NEH est $[L_4, L_2]$ avec un makespan de 24

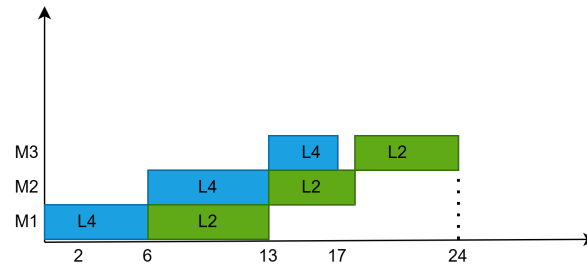


Figure 2.7: Diagramme de Gantt de la meilleure séquence du G_2

2.6.6 Construction de la séquence finale

Après avoir ordonné les lots au sein de chaque groupe compatible (intra-groupe), nous procédons à l'assemblage des groupes dans une séquence globale.

La séquence finale se construit en plaçant d'abord les lots optimisés de G_1 , suivis par une opération de stérilisation de durée $T=3$, puis par les lots optimisés de G_2 . Enfin, nous calculons le makespan total de cette séquence complète, qui intègre à la fois les temps de traitement des lots et les durées de stérilisation.

Le diagramme de Gantt ci-dessous illustre la séquence obtenue avec G-NEH-S avec une durée totale de production égale à 40.

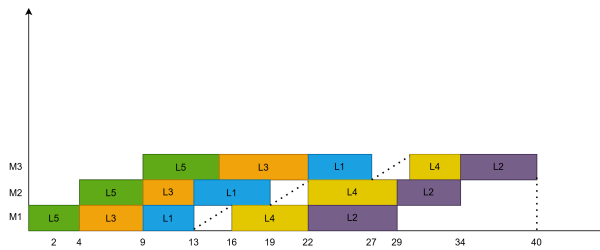


Figure 2.8: Diagramme de Gantt de la séquence finale avec G-NEH-S

2.6.7 Optimisation globale par insertion

Après avoir construit la séquence des lots de l'étape précédente $L_5-L_3-L_1-L_4-L_2$, on applique l'insertion globale. L'idée est simple : on prend chaque lot un par un, on le retire de sa position, puis on teste toutes les autres positions possibles dans la séquence. À chaque essai, on recalcule le temps total de production (makespan), en tenant compte des éventuelles stérilisations nécessaires si deux lots sont incompatibles. Si on trouve une meilleure position qui réduit le makespan, on la garde.

À l'issue de cette étape d'insertion globale, la séquence $L_5-L_3-L_1-L_4-L_2$ s'est révélée être la plus performante. En effet, après avoir testé toutes les positions possibles pour chaque lot, aucune autre configuration n'a permis d'obtenir un makespan inférieur. Cette séquence est donc conservée comme solution finale à ce stade de l'optimisation.

2.7 Conclusion

Dans ce chapitre, nous avons présenté une approche complète de modélisation et de résolution du problème d’ordonnancement d’un atelier Flow Shop en milieu pharmaceutique, en intégrant des contraintes complexes liées à la stérilisation et à la compatibilité des lots. Une première formulation mathématique du problème a été développée sous forme de programme linéaire en variables mixtes (MILP), ce qui a permis de modéliser rigoureusement les contraintes spécifiques à l’environnement pharmaceutique.

Compte tenu de la nature NP-difficile du problème, des approches heuristiques ont été mises en œuvre pour résoudre des instances de taille plus importante. L’adaptation de l’algorithme NEH au contexte pharmaceutique a permis d’intégrer les temps de stérilisation dans la séquence d’ordonnancement. Par la suite, une heuristique originale, appelée G-NEH-S, a été proposée afin de mieux gérer la compatibilité entre les lots, en les regroupant selon leurs caractéristiques.

L’ensemble de ces méthodes constitue une base solide pour l’analyse expérimentale qui sera menée dans le chapitre suivant, visant à comparer leur efficacité et leur applicabilité sur des données réelles ou simulées.

3.1 Introduction

Ce chapitre est consacré à l'analyse comparative des différentes approches développées pour résoudre le problème d'ordonnancement flow shop avec contrainte de stérilisation. Trois méthodes ont été considérées : MILP, NEH classique et G-NEH-S proposée dans ce travail.

Le modèle MILP fournit des solutions optimales pour les instances de petite taille et constitue une base de référence pour évaluer les performances des méthodes approchées.

L'heuristique NEH, largement utilisée dans les problèmes de type flow shop, ne prend pas en compte les contraintes spécifiques de stérilisation, ce qui limite son efficacité dans le contexte étudié.

Afin d'y remédier, l'heuristique G-NEH-S a été développée : elle intègre une phase de regroupement des lots compatibles et construit la séquence tout en respectant les contraintes de stérilisation entre groupes incompatibles.

L'objectif de ce chapitre est de comparer ces différentes approches selon deux indicateurs : qualité des solutions (via le makespan obtenu) et temps de calcul. L'analyse met également en lumière les forces et les limites de chaque méthode, selon la taille et la structure des instances considérées.

3.2 Paramètre expérimentaux

3.2.1 Benchmark

Afin d'évaluer les performances du modèle mathématique (MILP), de l'heuristique NEH classique, de l'heuristique proposée G-NEH-S, plusieurs instances de type flow shop avec contraintes de stérilisation ont été générées.

Les instances sont réparties selon leur taille afin de tester les approches sur des cas variés :
Petite taille : 5 à 15 lots, répartis sur 3 à 10 machines. Ces cas peuvent être résolus exactement avec le modèle MILP.
Taille moyenne à grandes: 20 à 50 lots sur 5 à 20 machines. Elles permettent de comparer les performances des différentes heuristiques. Ces cas sont utilisés pour évaluer la robustesse et la capacité des heuristiques à gérer des situations

plus complexes.

Les durées d'usinage sont générées aléatoirement, selon une loi uniforme entre 1 et 99. Chaque lot appartient à l'un des deux types de produits : antibiotique ou vaccin.

Pour générer la matrice d'incompatibilité entre lots, on a d'abord attribué à chaque lot un type aléatoire, soit A (antibiotique) soit B (vaccin). Ensuite, on a considéré que deux lots sont incompatibles s'ils ne sont pas du même type. Une matrice symétrique a été construite à partir de cette règle, en mettant un 1 quand deux lots sont incompatibles, et 0 sinon. Le taux global d'incompatibilité (le pourcentage de paires incompatibles) a été ajusté automatiquement pour qu'il soit proche d'une valeur cible : 20%, 50% ou 80%, selon les cas. Cette méthode permet de générer des instances variées avec plus ou moins de contraintes de stérilisation, ce qui aide à bien tester les performances des approches proposées.

3.2.2 Implémentation des méthodes

3.2.2.1 Modèle PLNE

Programmé avec CPLEX, il a été utilisé pour résoudre uniquement les petites instances, en raison de sa complexité computationnelle.

3.2.2.2 Heuristique NEH et G-NEH-S

Toutes deux codées sous MATLAB.

Les résultats obtenus seront présentés et analysés dans la section suivante.

3.3 Résultats expérimentaux

Afin d'évaluer l'efficacité de chaque approche, les résultats obtenus avec le modèle MILP (résolu par CPLEX), l'algorithme heuristique NEH classique, et l'heuristique G-NEH-S ont été comparés sur plusieurs instances. Les tableaux suivants présentent les valeurs de makespan (C_{max}) et les temps de calcul correspondants pour différents paramètres d'instance et taux d'incompatibilité.

		MILP				NEH				G-NEH-S	
	incomp %	Cmax	CPU		incomp %	Cmax	CPU		incomp %	Cmax	CPU
5*3	20	437	0.34	5*3	20	442	0.0039	5*3	20	437	0.0039
	50	427	0.14		50	427	0.00126		50	427	0.00126
	80	427	0.16		80	427	0.0108		80	427	0.0108
5*5	20	353	0.17	5*5	20	353	0.0022	5*5	20	353	0.0022
	50	353	0.16		50	353	0.0019		50	353	0.0019
	80	373	0.22		80	373	0.0017		80	373	0.0017
5*8	20	769	0.28	5*8	20	787	0.0023	5*8	20	769	0.0023
	50	779	0.16		50	787	0.0052		50	779	0.0052
	80	769	0.27		80	797	0.0023		80	769	0.0023
5*10	20	720	0.19	5*10	20	762	0.003	5*10	20	720	0.003
	50	738	0.11		50	742	0.0028		50	740	0.0028
	80	738	0.11		80	742	0.004		80	740	0.004
				10*3	20	633	0.0067	10*3	20	624	0.0067
					50	653	0.0058		50	633	0.0058
					80	653	0.0085		80	633	0.0085
				10*5	20	710	0.007	10*5	20	695	0.007
					50	736	0.0086		50	720	0.0086
					80	715	0.0074		80	709	0.0074
				10*8	20	1004	0.0106	10*8	20	1004	0.0106
					50	1024	0.0104		50	1024	0.0104
					80	1044	0.0117		80	1036	0.0117
				10*10	20	1158	0.0132	10*10	20	1150	0.0132
					50	1176	0.0129		50	1176	0.0129
					80	1178	0.0119		80	1149	0.0119
				15*3	20	942	0.0129	15*3	20	922	0.0129
					50	953	0.0145		50	953	0.0145
					80	942	0.0147		80	933	0.0147
				15*5	20	909	0.0184	15*5	20	874	0.0184
					50	906	0.0194		50	904	0.0194
					80	891	0.0266		80	878	0.0266
				20*5	20	1299	0.0458	20*5	20	1299	0.0458
					50	1339	0.0943		50	1313	0.0943
					80	1319	0.0386		80	1319	0.0386
				20*10	20	1735	0.0763	20*10	20	1715	0.0763
					50	1740	0.0757		50	1732	0.0757
					80	1763	0.0679		80	1757	0.0679
				20*20	20	2446	0.134	20*20	20	2349	0.134
					50	2398	0.1557		50	2384	0.1557
					80	2409	0.1515		80	2374	0.1515
				50*5	20	2796	0.507	50*5	20	2796	0.507
					50	2811	0.527		50	2793	0.527
					80	2846	0.5044		80	2841	0.5044
				50*10	20	3273	1.093	50*10	20	3161	1.093
					50	3272	0.9338		50	3201	0.9338
					80	3258	0.917		80	3191	0.917
				50*20	20	4076	1.831	50*20	20	4013	1.831
					50	4104	1.973		50	4057	1.973
					80	4026	1.833		80	3994	1.833

Table 3.1: Tableaux de résultats

3.4 Comparaison des performances

3.4.1 Comparaison sur les petites instances

Pour les petites tailles de problèmes (jusqu'à 5 lots et 5 machines), le modèle MILP résolu par CPLEX donne des résultats optimaux en un temps de calcul très court (souvent inférieur à une seconde). Dans ces cas, la solution exacte est obtenue rapidement, ce qui en fait une référence fiable pour évaluer la qualité des heuristiques. Par exemple, pour l'instance 5×5 avec un taux d'incompatibilité de 50 %, CPLEX retourne un makespan de 353 avec un temps de 0,16 seconde, tandis que G-NEH-S donne le même résultat avec un temps négligeable ($\approx 0,002$ s).

On remarque que l'heuristique G-NEH-S, bien qu'approximative, est capable de reproduire les résultats du MILP sur ces petites instances avec une précision remarquable et un temps de calcul beaucoup plus faible. L'algorithme NEH classique, quant à lui, fournit parfois des solutions équivalentes mais est plus sensible à la présence d'incompatibilités. Il peut proposer un makespan légèrement plus élevé dans certains cas.

3.4.2 Limites du solveur MILP (CPLEX)

Lorsque la taille des instances augmente (à partir de 10 lots avec 5 machines ou plus), les performances de CPLEX se dégradent significativement. Le temps de calcul devient alors un facteur bloquant : la résolution devient plus lente, voire impraticable pour les plus grandes tailles (20×10 , 20×20 , $50 \times 10 \dots$). Par exemple, pour l'instance 15×5 , le temps de résolution peut dépasser plusieurs dizaines de secondes, voire ne pas garantir l'optimalité dans un délai raisonnable.

Ce constat met en évidence la nécessité d'heuristiques rapides comme G-NEH-S ou NEH pour traiter efficacement des instances de taille industrielle, où le solveur exact devient inadaptable.

3.4.3 Comparaison G-NEH-S vs NEH sur les grandes instances

Pour les moyennes et grandes instances, l'heuristique G-NEH-S montre clairement sa supériorité sur NEH classique. Cette amélioration s'explique par la prise en compte explicite des contraintes d'incompatibilité et des stérilisations dans la construction de la solution. En effet, NEH n'intègre pas directement ces aspects, ce qui peut conduire à des séquences irréalistes ou coûteuses en temps (due aux stérilisations nombreuses non anticipées).

Par exemple, sur l'instance 20×10 avec un taux d'incompatibilité de 50 %, NEH retourne un makespan de 1740, alors que G-NEH-S propose 1732. De même, sur l'instance 10×10 à 80 % d'incompatibilité, G-NEH-S réussit à réduire le makespan de 1178 (NEH) à 1149, tout en gardant un temps de calcul inférieur à 0,01 s. Cela montre l'intérêt d'adapter l'algorithme de construction à la nature des contraintes industrielles.

3.4.4 Impact du taux d'incompatibilité

Un autre facteur influençant les performances est le taux d'incompatibilité entre les lots. Plus ce taux est élevé, plus les risques de stérilisation augmentent, ce qui a un impact direct sur le makespan. Dans les trois approches, on observe une hausse générale du makespan avec l'augmentation de ce taux. Toutefois, G-NEH-S gère mieux cette contrainte que NEH : grâce à son regroupement initial par familles compatibles, il limite le nombre de transitions nécessitant une stérilisation.

3.4.5 Comparaison des temps de calcul

Les résultats montrent une nette différence entre les méthodes en termes de temps de calcul. Le solveur CPLEX, basé sur une approche exacte, devient rapidement coûteux dès que la taille de l'instance augmente. Par exemple, pour des tailles modestes comme 5×3 ou 5×5 , CPLEX fournit des résultats en moins d'une seconde. Cependant, à partir d'instances de taille 10×10 ou 15×5 , le temps nécessaire croît sensiblement, ce qui peut rendre cette méthode impraticable dans un contexte industriel nécessitant des décisions rapides.

À l'inverse, les heuristiques NEH classique et G-NEH-S affichent des temps de calcul extrêmement faibles, même pour les grandes instances (par exemple, inférieurs à 0,01 seconde pour des problèmes allant jusqu'à 50 lots).

Par ailleurs, G-NEH-S, bien qu'un peu plus complexe que NEH, reste tout aussi rapide, tout en offrant de meilleures performances sur le critère du makespan, notamment en présence d'un taux élevé d'incompatibilité.

En résumé, le modèle MILP offre un étalon théorique, mais son usage reste limité à des cas simples. L'heuristique NEH, bien que rapide, montre ses limites dans les environnements complexes. À l'inverse, G-NEH-S représente un compromis idéal, combinant qualité de solution, rapidité d'exécution et adaptabilité aux contraintes du domaine pharmaceutique. Cette méthode se révèle particulièrement bien adaptée pour une utilisation concrète en milieu industriel.

3.5 Conclusion

Tout d'abord, la résolution par MILP, bien qu'efficace pour des instances de petite taille, devient rapidement inadaptée en termes de temps de calcul dès que la complexité du problème augmente. Cela confirme la nécessité de recourir à des approches alternatives pour les contextes industriels réels.

L'algorithme NEH classique, quant à lui, propose une solution rapide mais reste limité face aux contraintes complexes telles que les incompatibilités entre lots et les temps de stérilisation obligatoires. C'est dans ce contexte que l'heuristique G-NEH-S se distingue : en intégrant une logique de regroupement des lots compatibles et en adaptant dynamiquement la séquence de production, elle permet d'obtenir des résultats très satisfaisants en minimisant à la fois le makespan et les temps d'arrêt liés aux stérilisations.

Ainsi, les expérimentations réalisées confirment la validité et l'efficacité de l'approche heuristique développée. Elle constitue une solution réaliste et performante pour optimiser l'ordonnancement dans un environnement aussi contraignant que celui de l'industrie pharmaceutique.

CONCLUSION GÉNÉRALE

L'ordonnancement de la production en atelier Flow Shop, lorsqu'il est appliqué à un contexte aussi exigeant que celui de l'industrie pharmaceutique, nécessite des approches à la fois rigoureuses et flexibles. Ce mémoire a permis d'explorer et de mettre en œuvre plusieurs méthodes de résolution adaptées à cette problématique complexe.

La formulation mathématique développée a permis de modéliser précisément les contraintes industrielles spécifiques, telles que les temps de stérilisation fixes ainsi que les incompatibilités entre lots. L'utilisation de CPLEX a permis de résoudre des instances de petite taille de manière optimale, tandis que les heuristiques développées, notamment l'algorithme NEH adapté et l'heuristique personnalisée G-NEH-S, ont montré leur efficacité sur des cas plus complexes ou de plus grande taille.

Les résultats expérimentaux ont mis en évidence les avantages des approches combinant modélisation mathématique et heuristiques adaptées. Ce travail ouvre la voie à plusieurs perspectives, notamment l'intégration d'autres types de contraintes (ressources humaines, disponibilité machines), l'extension à d'autres environnements de production, ou encore l'implémentation en temps réel dans des systèmes de pilotage industriel.

Ainsi, ce mémoire apporte une contribution à la fois méthodologique et pratique à la problématique de l'ordonnancement pharmaceutique, en conciliant optimisation, faisabilité et respect des normes industrielles.

BIBLIOGRAPHIE

Boukef Ben Othman, H. (2009). Méthodes de résolution pour des problèmes d’ordonnancement avec contraintes de production dans un environnement pharmaceutique. Thèse de doctorat, Université Paris-Dauphine.

Campbell, H. G., Dudek, R. A., Smith, M. L. (1970). A heuristic algorithm for the n-job, m-machine sequencing problem. *Management Science*, 16(10), B630–B637.

Hertrich, T., Weiss, S. (2020). Lot-sizing and scheduling with product changeovers and incompatibilities in biopharmaceutical production. *Computers Chemical Engineering*, 141, 106980.

Nawaz, M., Ensore, E. E., Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1), 91–95.

Palmer, D. S. (1965). Sequencing jobs through a multi-stage process in the minimum total time – A quick method of obtaining a near optimum. *Operations Research Quarterly*, 16(1), 101–107.

Rossi, R., Lanzetta, M. (2013). A flow shop scheduling approach for the sterilization service in hospitals. In *Proceedings of the 23rd International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*.