

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة أبي بكر بلقايد - تلمسان
Université Aboubakr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE



THÈSE

Présentée pour l'obtention du grade de **DOCTORAT 3^{ème} Cycle**

En : Génie Industriel

Spécialité : Productique

Par : SEKKAL Djazia Nadjat

Sujet

**Métaheuristiques pour l'ordonnancement des
systèmes industriels**

Soutenue publiquement, le **11/06/2024**, devant le jury composé de :

M.	MEGNOUNIF Abdellatif	Professeur	Université de Tlemcen	Président
M.	BELKAID Fayçal	Professeur	Université de Tlemcen	Directeur de thèse
Mme.	KOULOUGHLI Sihem	Professeur	Université de Tlemcen	Examinatrice
Mme.	KERBOUA-ZIARI Yasmina	Professeur	USTHB- Alger	Examinatrice
M.	BRAHAMI Mustapha Anwar	MCA	ESSAT-Tlemcen	Examinateur
M.	KAHOUADJI Housseyn Amin	MCB	Université de Tlemcen	Invité

À mes parents Wassila et Djamel,
À mon mari Faudel,
À mon fils Rachad

Remerciements

«*Tout vient à point à qui sait attendre*». Ce proverbe résume parfaitement mon parcours universitaire, qui représente l'aboutissement de plusieurs années de travail, symbolisé par la rédaction de cette thèse. Tout cela n'aurait pas été possible sans le soutien et les encouragements de nombreuses personnes. Je suis consciente que mes mots ne seront pas suffisants pour exprimer toute la gratitude que je ressens envers tout ce qu'elles m'ont apporté.

Je tiens tout d'abord à exprimer ma profonde gratitude envers mon directeur de thèse, le professeur **Fayçal Belkaid**, pour ses directives si précieuses, sa disponibilité et son dévouement tout au long de cette aventure académique. Ses conseils éclairés, son expertise et son exigence ont été des atouts essentiels dans la réalisation de cette recherche, m'aidant à me surpasser et à donner le meilleur de moi-même. Je lui suis profondément reconnaissante pour son investissement personnel dans ce travail et pour sa contribution à mon développement professionnel. Un grand merci pour cet encadrement de qualité, tant sur le plan intellectuel qu'humain.

Mes remerciements les plus sincères aux membres du jury : Monsieur **Abdellatif Megnounif**, Professeur à l'université de Tlemcen, d'avoir pris le temps d'expertiser cette thèse et de me faire l'honneur d'en assurer la présidence. J'adresse également mes vifs remerciements à Madame **Sihem Kouloughli**, Professeur à l'université de Tlemcen, d'avoir participé à ma formation universitaire. Merci pour ses précieux conseils, sa gentillesse, son écoute et surtout de me faire le privilège d'évaluer et de juger ce manuscrit.

Je voudrais remercier Madame **Yasmina Kerboua-Ziari**, Professeur à l'université des sciences et de la technologie Houari-Boumediene, pour ses suggestions et conseils avisés, ainsi que d'avoir accepté de faire partie de mon jury de thèse et d'expertiser mon travail, malgré ses nombreuses charges.

Je remercie également Monsieur **Mustapha Anwar Brahami**, Maître de conférences à l'École supérieure en sciences appliquées de Tlemcen, pour ses conseils constructifs et

d'avoir eu l'amabilité de juger et d'évaluer ce travail de thèse.

Je remercie vivement Monsieur **Housseyn Amin Kahouadji**, Maître de conférences à l'université de Tlemcen, d'avoir accepté d'être parmi nous et de nous honorer de sa présence dans mon jury.

J'aimerais adresser un remerciement particulier à Madame **Zahera Dib** et à Madame **Lamia Sari-Triqui**, Maîtres de conférences à l'université de Tlemcen, pour leurs sympathies et leur soutien tout au long de ma formation universitaire.

Je souhaite exprimer ma gratitude envers tous les enseignants et chercheurs du laboratoire MELT. Le temps que j'ai passé au sein de ce laboratoire de productique a été une opportunité précieuse qui m'a permis de collaborer avec ceux qui étaient autrefois mes enseignants. Je tiens tout particulièrement à remercier notre directeur de laboratoire, Monsieur **Sidi Mohammed Meliani**, pour son soutien constant et ses conseils tout au long de mon parcours doctoral. Mes remerciements vont également à Monsieur **Mohammed Bennekrouf** pour son implication et ses conseils.

Mes remerciements les plus chaleureux sont adressés à Monsieur **Sidi Moh Ahmed Nait**, Professeur à l'université Jean Monnet-Saint Étienne, pour l'accueil chaleureux qu'il m'a réservé lors de mon séjour au laboratoire LASPI, ainsi que pour l'intérêt qu'il a manifesté à l'égard de mes travaux.

Je souhaite aussi saluer l'ensemble des ingénieurs et doctorants du laboratoire MELT et du laboratoire d'automatique de Tlemcen (LAT), sans qui ces quelques années auraient paru bien plus mornes, je pense spécialement à Mellouk Wafaa, Zeddou Besma, Boumezirene Thiziri, Sekkal Nor El Houda, Elouchdi Mouna, Bakdi Khadidja, Fandi Wahiba et Houar Esmaa. Un merci particulier à mon amie, ma collègue et ma compagne de route Tahraoui Nacera sans qui cette aventure aurait été bien moins lumineuse et passionnante.

Un grand merci à mes trois meilleures amies d'enfance, Sarah, Farah et Lilya, pour leur amitié, leur présence et leur soutien.

Je tiens à exprimer tout particulièrement et du fond du cœur ma gratitude envers toute ma famille, en pensant plus spécialement à mes parents pour leur soutien, leur confiance et leurs encouragements. Sans eux, je ne serais pas arrivé jusqu'ici. Ma reconnaissance va également à mes grands-parents, mes tantes, mes beaux-parents et mes cousins, qui ont été présents du début à la fin et ont su trouver les mots pour m'encourager et me

soutenir. Je souhaite également adresser mes remerciements à ma sœur Lilya pour son aide, sa patience et sa gentillesse, ainsi qu'à ma petite nièce adorée, Éline.

Je garde une place spéciale dans mes remerciements à mon mari, Faudel, mon pilier, mon complice, pour son soutien infailible, ses encouragements dans les moments difficiles et les périodes de doute. Tu as su trouver les mots et prodiguer les conseils nécessaires pour m'élever et me motiver à dépasser mes limites. Pour tout cela, je te serai infiniment reconnaissante. Merci aussi à celui qui m'a donné la force et le courage d'avancer au travers de ses beaux sourires, mon petit prince, mon trésor, mon fils Rachad.

Ce dur labeur et ce travail acharné n'auraient pas vu le jour sans l'aide de Dieu tout-puissant. Je lui suis infiniment reconnaissante pour la volonté de fer, la patience et la persévérance qu'il m'a octroyée et qui ont guidé tout mon parcours d'études.

Table des matières

Table des figures	i
Liste des tableaux	iii
Introduction Générale	1
1 Ordonnancement de la production et technologie de groupe	5
1.1 Introduction	5
1.2 Les systèmes de production	6
1.2.1 Présentation	6
1.2.2 Classification des systèmes de production	8
1.2.2.1 Classification selon le volume de production	9
1.2.2.2 Classification selon la gestion de la demande	10
1.2.2.3 Classification selon la disposition des flux physique	11
1.2.2.4 Classification selon la circulation du produit dans l'atelier	12
1.2.2.5 Classification selon la relation avec le client	12
1.3 Les différents niveaux de décisions	13
1.3.1 Les décisions stratégiques	13
1.3.2 Les décisions tactiques	14
1.3.3 Les décisions opérationnelles	14
1.4 Notions préliminaires sur les problèmes d'ordonnancement	15
1.4.1 Définition	16
1.4.2 Concepts de base	16
1.4.2.1 Les tâches	17
1.4.2.2 Les ressources	17
1.4.2.3 Les contraintes	17
1.4.2.4 Les critères d'évaluation	18
1.4.2.5 Les objectifs	19
1.4.3 Typologie des ordonnancements de la production	19
1.4.4 Les ateliers de production	21

1.4.4.1	Problème d’ordonnancement à une seule machine	21
1.4.4.2	Problème d’ordonnancement à machine parallèle	21
1.4.4.3	Problème d’ordonnancement à cheminement unique (flow shop)	22
1.4.4.4	Problème d’ordonnancement à cheminement multiple (job shop)	22
1.4.4.5	Problème d’ordonnancement à cheminement libre (open shop)	22
1.4.5	Représentation des problèmes d’ordonnancement	23
1.4.6	Notation des problèmes d’ordonnancement	24
1.5	Temps de traitement variables	27
1.5.1	Effet de détérioration	28
1.5.2	Effet d’apprentissage	29
1.5.3	Temps de traitement contrôlables	31
1.6	Temps de configuration dans les problèmes d’ordonnancement	32
1.6.1	Structure de base des temps de configuration	33
1.6.2	Classification des temps de configuration	34
1.6.2.1	Temps de configuration indépendants de la séquence (SI)	35
1.6.2.2	Temps de configuration dépendants de la séquence (SD)	35
1.6.2.3	Temps de configuration dépendants de la machine/des ressources (MD)	36
1.6.2.4	Temps de configuration dépendants du temps (TD)	37
1.7	Technologie de groupe	38
1.7.1	Définition	40
1.7.2	Évolution du concept de technologie de groupe	41
1.7.3	Applicabilité du concept de technologie de groupe	42
1.8	Conclusion	44
2	État de l’art	45
2.1	Introduction	45
2.2	Aperçu de la littérature sur les problèmes d’ordonnancement d’atelier flow shop	46
2.3	Aperçu de la littérature sur les problèmes de flow shop avec technologie de groupe	52
2.3.1	Temps de configuration indépendant de la séquence	54
2.3.2	Temps de configuration dépendant de la séquence	55
2.4	Aperçu sur les travaux d’ordonnancement avec effet d’apprentissage	60
2.4.1	Effet d’apprentissage basé sur la position	63

2.4.2	Effet d'apprentissage basé sur la somme des temps de traitement . . .	66
2.4.3	Combinaison des deux modèles	67
2.5	État de l'art sur les problèmes d'ordonnancement axé sur l'efficacité éner- gétique	67
2.5.1	Vers une optimisation énergétique des problèmes d'ordonnancement flow shop	67
2.5.2	Vers une optimisation énergétique des problèmes d'ordonnancement flow shop avec technologie de groupe	70
2.6	Positionnement de notre problématique	70
2.7	Conclusion	71
3	Problème flow shop avec technologie de groupe, temps de configuration dépendant de la séquence et effet d'apprentissage	73
3.1	Introduction	73
3.2	Présentation du problème	75
3.2.1	Modélisation des temps opératoires	76
3.2.2	Modèle global pour l'effet d'apprentissage	76
3.2.3	Extrapolation des modèles généraux	77
3.2.4	Description du modèle proposé	80
3.3	Flow shop avec effet d'apprentissage pour une optimisation mono-objectif .	81
3.3.1	Approche intégrative de l'effet d'apprentissage	81
3.3.2	Formulation du problème	83
3.4	Ordonnancement de groupes avec temps de configuration dépendant de la séquence	85
3.4.1	La technologie de groupe : Un concept innovant	85
3.4.2	Hypothèses et notations relatif à l'ordonnancement de groupe . . .	87
3.4.3	Le modèle mathématique et les contraintes	89
3.5	Minimisation du Cmax en tenant compte des contraintes énergétiques . . .	90
3.5.1	Formulation du problème	91
3.5.2	Évaluation de l'efficacité de notre approche	94
3.5.2.1	Paramètres et méthodologie d'analyse des instances de l'étude	94
3.5.2.2	Analyse des Résultats Expérimentaux	95
3.6	Conclusion	97
4	L'ordonnancement dans le FSDGS avec contrainte de transport	98
4.1	Introduction	98
4.2	Le FSDGS avec un système multi-transports à capacité limité	99
4.2.1	Description et formulation du problème	100

4.2.2	L'approche de résolution	102
4.2.3	Évaluation expérimentale de l'approche d'optimisation multi-objectifs du FSDGS avec transport	103
4.2.4	Analyse et discussion des résultats obtenus	104
4.3	Cas d'étude	109
4.3.1	Présentation du cas d'étude de bielles forgées	110
4.3.2	L'adaptation de la technologie de groupe pour le cas d'étude	110
4.3.3	Processus de fabrication de bielles forgées	112
4.3.3.1	Le chauffage	112
4.3.3.2	Le laminage	113
4.3.3.3	Le matriçage	114
4.3.3.4	Le poinçonnage	114
4.3.4	Données et caractéristiques relatives au cas d'étude	115
4.3.4.1	Les machines	116
4.3.4.2	Les produits	116
4.3.4.3	Le système de transport	117
4.4	Le FSDGS avec un système mono-transport à capacité limité	118
4.4.1	La consommation d'énergie du système global	118
4.4.1.1	La consommation d'énergie pendant la phase de production	119
4.4.1.2	La consommation d'énergie durant les phases de configura- tions des machines	119
4.4.1.3	La consommation d'énergie durant l'inactivité de la machine	120
4.4.1.4	La consommation d'énergie durant la phase de transport .	120
4.4.2	Définition mathématique du problème	121
4.4.3	Exemple numérique	127
4.5	La méthode de pondération	129
4.5.1	Exemple numérique	130
4.6	Développement des bornes inférieures pour le problème de FSDGS	134
4.6.1	La première borne inférieure <i>LB1</i>	134
4.6.2	Validation de la première borne inférieure <i>LB1</i>	136
4.6.3	La deuxième borne inférieure <i>LB2</i>	136
4.6.4	Validation de la deuxième borne inférieure <i>LB2</i>	139
4.7	Conclusion	140
5	Algorithme MOSA pour l'ordonnancement du problème FSDGS avec effet d'apprentissage des opérateurs	142
5.1	Introduction	142
5.2	Présentation de l'algorithme MOSA	144

5.3	le codage de la solution	145
5.4	Solution initiale et voisinage	146
5.4.1	Solution initiale	146
5.4.2	Génération du voisinage	146
5.5	Les paramètres de l’algorithme MOSA	148
5.5.1	Le processus de refroidissement	148
5.5.1.1	La température initiale	149
5.5.1.2	Le nombre d’itérations à chaque température	149
5.5.1.3	La fonction de refroidissement	149
5.5.1.4	Le critère d’arrêt	149
5.5.2	La probabilité d’acceptation	150
5.6	La procédure de l’algorithme MOSA	150
5.7	La recherche locale	152
5.8	Analyse de sensibilité de l’algorithme MOSA proposé	153
5.8.1	Étude de variation des paramètres de MOSA	154
5.8.2	Étude de variation des vitesses de traitement	155
5.9	Expérimentations et résultats	156
5.9.1	Génération des instances	156
5.9.1.1	Environnements de tests pour les machines	156
5.9.1.2	Environnement de tests pour les groupes	156
5.9.1.3	Environnement de tests pour les jobs	156
5.9.1.4	Environnement de tests pour les temps de configuration	157
5.9.1.5	Environnement de tests pour les caractéristiques de l’étude	157
5.9.2	Résultats numériques	158
5.9.2.1	Proportion d’instances résolues avant l’atteinte de la condition d’arrêt.	158
5.9.2.2	Temps de calcul et performances	159
5.10	Conclusion	179
	Conclusion Générale et Perspectives	181
	Références	187

Table des figures

1.1	le modèle conceptuel d'un système de production	7
1.2	La hiérarchie des niveaux de décisions	15
1.3	Typologie des ordonnancements	20
1.4	Représentation d'un ordonnancement par diagramme de Gantt	24
1.5	Corrélation entre les modèles de temps de configuration (ST). Le cercle représente un modèle d'ordonnement standard	37
1.6	Représentation du concept de technologie de groupe dans un atelier de fabrication automobile	39
2.1	Échéancier des problèmes d'ordonnement avec effet d'apprentissage . . .	63
3.1	Représentation de la courbe d'apprentissage avec différents taux d'appren- tissage.	82
4.1	Comparaison entre les trois expériences pour la minimisation du makespan à travers les trois scénarios	107
4.2	Comparaison entre les trois expériences pour la minimisation de la consom- mation d'énergie à travers les trois scénarios	107
4.3	L'impact des coefficients de pondération sur l'optimisation des deux objec- tifs pour le scénario 1	108
4.4	L'impact des coefficients de pondération sur l'optimisation des deux objec- tifs pour le scénario 2	108
4.5	L'impact des coefficients de pondération sur l'optimisation des deux objec- tifs pour le scénario 3	109
4.6	Illustration d'une bielle	111
4.7	Processus de fabrication d'une bielle forgée	115
4.8	Disposition des machines de fabrication de bielles forgées	116
4.9	Représentation du problème FSDGS pour la fabrication de bielles forgées .	122
4.10	Impact de l'effet d'apprentissage sur la fonction objective et l'ordonnance- ment des jobs et des groupes	128

4.11	Impact des coefficients de pondération et de l'effet d'apprentissage sur la minimisation des fonctions objectives et l'ordonnancement des tâches et des groupes	133
4.12	L'efficacité de la deuxième borne inférieure	140
5.1	La procédure de génération du voisinage (NGP2)	148
5.2	Le pourcentage d'erreur entre MOSA et LB2 pour les grandes instances . .	161
5.3	Impact de la recherche locale sur la valeur de la fonction pondérée en considérant différents taux d'apprentissage et différents coefficients de pondération pour les petites instances.	177
5.4	Impact de la recherche locale sur la valeur de la fonction pondérée en considérant différents taux d'apprentissage et différents coefficients de pondération pour les moyennes instances.	177
5.5	Impact de la recherche locale sur la valeur de la fonction pondérée en considérant différents taux d'apprentissage et différents coefficients de pondération pour les grandes instances.	177
5.6	Les trois principaux axes de la thèse	182

Liste des tableaux

1.1	Temps de traitement des jobs	23
1.2	Description des symboles d'utilisation dans la notation de Graham : partie 1	25
1.3	Description des symboles d'utilisation dans la notation de Graham : partie 2	27
1.4	Les modèles de temps de traitement en fonction de la position	31
1.5	Les modèles de temps de traitement en fonction du temps	31
2.1	Synthèse des articles qui ont considéré un ordonnancement à machines en série	52
2.2	Synthèse des problèmes d'ordonnancement flow shop avec groupe	61
3.1	Impact du taux d'apprentissage sur le traitement des tâches	83
3.2	Les données des deux scénarios	95
3.3	les résultats des deux scénarios	96
4.1	données communes des scénarios	104
4.2	caractéristiques des instances générées	104
4.3	les résultats de la première expérience	105
4.4	les résultats de la seconde expérience	106
4.5	les résultats de la troisième expérience	106
4.6	Temps de traitement des opérations	117
4.7	Temps de transport des produits sur la machine	118
4.8	Les paramètres et variables du problème d'ordonnancement	123
4.9	Les données du problème de l'exemple numérique	127
4.10	Les données de l'exemple numérique	131
4.11	Comparaison de la borne inférieure avec le modèle mathématique (<i>LB1</i> vs le modèle MILP)	137
5.1	L'écart moyen entre MOSA et l'optimum pour différents niveaux de facteurs contrôlables..	154
5.2	Protocole d'intégration de la vitesse et son impact sur la solution finale . .	155
5.3	Caractéristiques des instances générées	157

5.4	Le temps de configuration de chaque machine	158
5.5	Pourcentage des petites instances résolues avant 30 minutes	158
5.6	Pourcentage des moyennes instances résolues avant 30 minutes	159
5.7	Pourcentage des grandes instances résolues avant 30 minutes	159
5.8	Solutions détaillées pour les petites instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de 100%	163
5.9	Solutions détaillées pour les petites instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [70%.90%].	165
5.10	Solutions détaillées pour les petites instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [90%.70%].	167
5.11	Solutions détaillées pour les moyennes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de 100%	169
5.12	Solutions détaillées pour les moyennes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [70%.90%].	171
5.13	Solutions détaillées pour les moyennes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [90%.70%].	172
5.14	Solutions détaillées pour les grandes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de 100%.	173
5.15	Solutions détaillées pour les grandes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [70%, 90%].	174
5.16	Solutions détaillées pour les grandes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [90%, 70%].	175

Introduction Générale

La concurrence mondiale influence grandement la croissance économique. Dans la production industrielle, le mot d'ordre est la compétitivité, et le marché a évolué d'une économie d'échelle à une économie de gamme. Afin d'être compétitive, une entreprise manufacturière doit s'automatiser, s'intégrer et être flexible pour améliorer sa productivité, la qualité de ses produits ainsi que le temps et le coût de production. L'automatisation partielle ou complète des systèmes de production entraîne un investissement coûteux qui serait vain sans un système de gestion efficace et des méthodes d'ordonnancement judicieuses.

L'ordonnancement est une méthodologie qui organise l'utilisation des ressources humaines et technologiques dans de nombreuses industries manufacturières et de services afin de répondre directement aux exigences des clients. Cette fonction coordonne de manière optimale l'exécution simultanée de plusieurs tâches en utilisant des ressources disponibles en quantité limitée, tout en respectant un ensemble de contraintes pour optimiser un ou plusieurs objectifs. Le séquençage des tâches peut varier selon le domaine d'activité. Ainsi, la définition d'un planning des opérations dans un hôpital (système de production de services) fait partie de cette problématique, tout comme la planification des opérations ou des tâches à effectuer par les machines (système de production de biens).

Par ailleurs, la plupart des problèmes d'ordonnancement sont de nature déterministe. Dans cette perspective, les chercheurs supposent que toutes les données du problème sont parfaitement connues à l'avance et développent des méthodes pour générer un ordonnancement unique qui respecte les contraintes du système et optimise les performances. Cependant, cette approche est devenue trop contraignante dans des environnements où les données ne sont pas toujours connues à l'avance et où de nouvelles informations peuvent apparaître pendant l'exécution du programme prévisionnel, rendant ainsi sa mise en œuvre obsolète. C'est pourquoi de nombreux spécialistes se sont penchés sur la question de l'ordonnancement sous incertitude. Leur objectif est de permettre une application pratique des recherches théoriques en ordonnancement dans des conditions réelles et dynamiques.

Dans ce contexte, les temps de traitement des tâches peuvent varier en raison des conditions de production. Dans plusieurs industries, des facteurs tels que l'état de la machine ou l'expérience des opérateurs peuvent influencer les performances de la production. Ces variations sont habituellement modélisées dans la littérature par des effets de détérioration ou d'apprentissage.

Les problèmes d'ordonnancement avec effet d'apprentissage sont devenus fréquents dans la gestion des opérations. Cependant, une grande partie des études sur ces problèmes ignore les temps de configuration entre les tâches, qui sont pourtant essentiels et doivent être considérés indépendamment des temps de traitement. Ces temps de configuration, ou temps de réglage, comprennent des actions telles que le nettoyage des zones de fabrication, l'ajustement des paramètres de la machine ou le changement d'outils. De plus, s'ils ne sont pas correctement gérés, ces temps peuvent facilement consommer plus de 20% de la capacité disponible de la machine. (M. L. Pinedo, 2012).

Toujours dans une optique d'économie de temps et de coût pour accroître l'efficacité de la production, une approche se présente comme particulièrement judicieuse. Il s'agit de la technologie de groupe qui postule que les tâches nécessitant des réglages similaires de la machine peuvent être regroupées en familles ou en groupes, en fonction de divers critères conceptuels tels que la forme, la taille, la texture de la surface et le type de matériaux. Cette classification peut également s'effectuer en lien avec les critères de fabrication, englobant les opérations et les séquences, la taille des lots, le temps de traitement, ou encore la quantité de production. Cette approche, par sa nature novatrice, offre de nombreux avantages en assurant une parfaite adéquation avec le contexte industriel pratique.

Parallèlement, les tendances prédominantes de la recherche en théorie de l'ordonnancement au cours des dernières années ont visé à améliorer la modélisation des problèmes concrets et industriels. Ces investigations ont abouti au développement de modèles intégrant les aspects de l'ordonnancement et de la prise de décision logistique, y compris des considérations telles que la disponibilité des machines, en particulier celles dédiées au transport des produits inter-production. Ainsi, l'évolution de ces deux domaines offre des perspectives complémentaires pour une gestion plus efficace des processus de production.

La finalité de la résolution des problèmes d'ordonnancement est d'optimiser des critères de performance, généralement d'ordre économique, car ne l'oublions pas, le but premier d'une entreprise est de générer des bénéfices. Toutefois, face au changement climatique, il est devenu urgent pour les entreprises de réduire leur impact environnemental en réponse aux exigences de plus en plus strictes fixées par le gouvernement et le marché. L'intégration

de l'ensemble de ces contraintes confère au problème une visibilité significative, permettant son adaptation au système manufacturier concret, mais accroît également sa complexité, l'éloignant ainsi des standards et des problèmes d'ordonnancement classiques traités dans la littérature.

Face à ce paradigme, il est nécessaire de recourir à des méthodes de résolution dites approchées, telles que les heuristiques ou métaheuristiques, capables de fournir des solutions de bonne qualité en un temps assez réduit. Cela permet de concilier les objectifs économiques et environnementaux, tout en tenant compte de la complexité accrue des nouvelles contraintes.

Cet ensemble de considérations nous a motivés, dans le cadre de notre thèse, à explorer l'adaptation des métaheuristiques pour la résolution multiobjectif du problème d'ordonnancement à machine en série (flow shop), en prenant en considération l'ensemble des variantes décrites précédemment. Cette thèse est structurée autour de cinq chapitres.

Le premier chapitre offre un aperçu approfondi des défis liés à l'ordonnancement dans les systèmes de production. Nous entamons notre discussion en présentant les systèmes de production, suivi d'une exploration de la terminologie associée aux problèmes d'ordonnancement. Nous mettons en lumière l'importance de ces problèmes en exposant les concepts fondamentaux, les notations utilisées, ainsi que les divers ateliers de production. Dans le but de faciliter la compréhension de la problématique, nous détaillons également les divers effets exercés sur les temps de traitement. Une attention particulière est accordée à la compréhension du concept de technologie de groupe, accompagnée d'une explication détaillée de la notion de temps de configuration qui lui est associée. En conclusion, nous examinons la nature intrinsèque de ces problèmes, passons en revue les différentes méthodes de résolution disponibles, et abordons la complexité inhérente à leurs processus de résolution.

Le deuxième chapitre présente une revue de la littérature visant à offrir un panorama exhaustif de la prise en compte des diverses variantes exposées dans cette étude et adaptées dans les problèmes d'ordonnancement flow shop. Pour ce faire, nous commençons par donner un aperçu de la littérature sur les problèmes classiques d'ordonnancement à machine en série. Ensuite, nous approfondissons notre analyse avec un état de l'art sur les problèmes flow shop incluant la technologie de groupe et les temps de configurations. Nous présentons également les résultats des travaux antérieurs portant sur les problèmes d'ordonnancement avec effet d'apprentissage. Enfin, nous passons en revue les différentes approches de résolution du problème d'ordonnancement flow shop multiobjectif. L'objectif de cette section va au-delà de la simple synthèse de l'état de l'art des problèmes d'ordonnancement flow shop avec technologie de groupe et effet d'apprentissage. Il s'agit

également d'identifier les disparités entre les différentes études et de positionner judicieusement notre recherche par rapport à la littérature existante.

Le troisième chapitre est dédié à la présentation de notre approche globale pour résoudre le problème d'ordonnancement flow shop, en tenant compte de l'effet d'apprentissage. Dans un premier temps, nous mettons en lumière cette approche, puis nous intégrons la notion de technologie de groupe dans les problèmes de flow shop avec effet d'apprentissage. Nous détaillons les caractéristiques essentielles de ce problème, en particulier en ce qui concerne l'intégration des temps de configuration dépendant de la séquence. Deux modèles mathématiques linéaires en nombres entiers mixtes sont développés, le premier visant à résoudre le problème de flow shop de base avec effet d'apprentissage, et le second dédié à la résolution des problèmes de flow shop avec technologie de groupe.

Le quatrième chapitre est consacré à mettre en avant une autre variante du problème d'ordonnancement flow shop avec technologie de groupe et effet d'apprentissage, à savoir l'insertion des temps de transport. Cette contrainte a été abordée à travers deux modélisations distinctes : la première suppose des transports à capacité illimitée, tandis que la seconde met en avant l'hypothèse de transports à capacité limitée. Ces modélisations définissent l'ensemble des variables qui influent sur le modèle final, constituant ainsi le fondement principal du développement de notre approche de résolution. De plus, deux bornes inférieures ont été élaborées en se basant sur le principe de relaxation partielle de l'intégrité des variables du modèle et sur l'heuristique SPT.

Le cinquième et dernier volet de ce manuscrit se concentre sur l'adaptation de la métaheuristique multiobjectif basée sur le recuit simulé (Multi Objectif Simulated Annealing MOSA) pour résoudre le problème flow shop avec technologie de groupe, effet d'apprentissage, temps de configuration dépendant de la séquence et temps de transport, visant à minimiser simultanément le temps total d'exécution (le makespan) et la consommation d'énergie totale du système. Nous avons présenté un codage direct pour une solution complète, ainsi qu'une méthode de voisinage permettant de déterminer à la fois le séquençement des groupes et des jobs appartenant à ces groupes. En outre, une procédure de recherche locale a été intégrée pour renforcer les performances de l'algorithme. La seconde partie de ce chapitre est consacrée à l'exposition des instances générées du problème et des diverses expérimentations menées pour analyser les performances de MOSA, basées sur une étude de cas réelle.

Nous clôturons cette thèse par une conclusion ainsi que des perspectives de recherche et des orientations futures.

Chapitre 1

Ordonnancement de la production et technologie de groupe

1.1 Introduction

Tout au long de leur existence, les entreprises ont constamment cherché à optimiser leurs processus de production pour maintenir leur compétitivité. La gestion de la production est donc un concept aussi vieux que l'entreprise elle-même. Dans cette optique, l'ordonnancement joue un rôle essentiel, qu'il s'agisse d'industries ou d'entreprises de services. Il représente un élément fondamental de la gestion de la production, contribuant ainsi aux performances économiques et financières de l'entreprise. Les systèmes de production, souvent composés de machines polyvalentes, nécessitent des réglages fastidieux et parfois coûteux lors des transitions d'une production à une autre. Bien que certaines bonnes pratiques permettent de réduire ces ajustements, il est rare de pouvoir les éliminer totalement. La prise de décisions concernant les temps de configuration représente un défi complexe. D'un côté, il est nécessaire de répondre à la demande en effectuant fréquemment des modifications sur les machines, tout en maintenant des stocks réduits et des délais de livraison courts. D'un autre côté, ces modifications régulières peuvent diminuer la capacité de production et engendrer une perte de temps considérable. Donc, pour améliorer la productivité et répondre à ces défis, les entreprises explorent de nouvelles approches telles que la technologie de groupe GT, une philosophie de production novatrice qui réduit les temps de configuration et de production, tout en aidant à atteindre les objectifs fixés.

Un autre aspect essentiel abordé dans cette thèse concerne les effets d'apprentissage dans les problèmes d'ordonnancement où la composante humaine est significative, telle que la configuration, la maintenance, le fonctionnement et le contrôle des machines, ainsi que l'interprétation des données et les travaux manuels. Ces effets d'apprentissage de-

viennent particulièrement importants lorsque l'environnement de production évolue. Les changements, qu'ils soient liés à l'arrivée de nouveaux travailleurs, à l'investissement dans de nouvelles machines, à la modification du flux de travail ou à la création de nouveaux emplois, entraînent des effets d'apprentissage. Même de petites modifications, comme une mise à jour logicielle ou un changement de format de documents, ont un impact sur l'apprentissage. Chaque fois qu'un travailleur doit s'habituer à une nouvelle situation, une expérience d'apprentissage se produit. Étant donné la fréquence de ces activités humaines dans un environnement de production en constante évolution, il est important de tenir compte des effets d'apprentissage dans l'ordonnement des systèmes de production.

les éléments centraux de cette thèse s'articulent autour des problèmes d'ordonnement qui feront l'objet de ce chapitre. la deuxième section présente une vue d'ensemble des systèmes de production, englobant leur définition, leurs caractéristiques et les différents types de décisions associées. Ceci constitue un point d'ancrage pour les problèmes d'ordonnement de la production, exposé en détail dans la troisième section. La quatrième section présente les différents effets contribuant à la variation du temps de traitement. Le champ d'application de la thèse impliquant ses points clés relatif au temps de configuration et à la technologie de groupe est exposé dans la cinquième section. les deux dernières sections (section 6 et 7) donnent un aperçu sur la complexité des problèmes d'ordonnement et leurs méthodologies de résolution.

1.2 Les systèmes de production

1.2.1 Présentation

Les systèmes de production, au cœur de l'industrie et de nombreuses autres activités, ont connu une évolution significative au fil du temps. À l'origine, la société de consommation a favorisé une concurrence féroce entre les producteurs, donnant naissance à la fonction mercatique, axée sur la satisfaction des besoins et des demandes. Toutefois, les nouvelles exigences du marché, telles que l'amélioration continue de la qualité des produits, l'accélération des processus de conception, la gestion efficace des stocks et la réduction des délais de livraison, ont redonné à la fonction de production une place essentielle.

La production est définie comme étant le processus de création de biens et/ou de services par une combinaison de matériaux, de prestations et de ressources. Cette approche ne se limite pas au domaine industriel, elle englobe également d'autres secteurs d'activités, par exemple lorsqu'un formateur anime un stage ou qu'un professeur crée du matériel pédagogique, comme des prestations de cours ou des exercices, il y a aussi production.

Par ailleurs, il existe un lien évident entre la production de biens et de services. La consommation constitue la force motrice supérieure de toute production. Les biens produits doivent d'une manière ou d'une autre être distribués pour être consommés. La production de biens est donc souvent sans intérêt si elle n'est pas combinée à la production de services, par exemple dans le domaine de la logistique (Mattsson & Jonsson, 2003). Toutefois, le type spécifique de production auquel il est fait référence dans cette thèse est la production industrielle. Nous nous limitons à la production de biens, où la transformation de la matière première en produits est effectuée dans un système de production.

Les systèmes de production sont des ensembles d'unités, qu'elles soient matérielles ou humaines, organisées dans le but de produire des biens ou des services. Ils doivent répondre à une charge, correspondant à la demande du marché, et disposent d'une capacité limitée définissant la quantité de biens ou de services pouvant être produits sur une période donnée. Ces systèmes utilisent différentes ressources, telles que des machines, des opérateurs et des matières premières, dont la quantité et le type varient en fonction du système de production. Il se décompose généralement en trois sous-systèmes comme on peut le remarquer à travers la figure 1.1 (Javel & Le Bert, 1993) : le système physique de production, le système de décision et le système d'information.

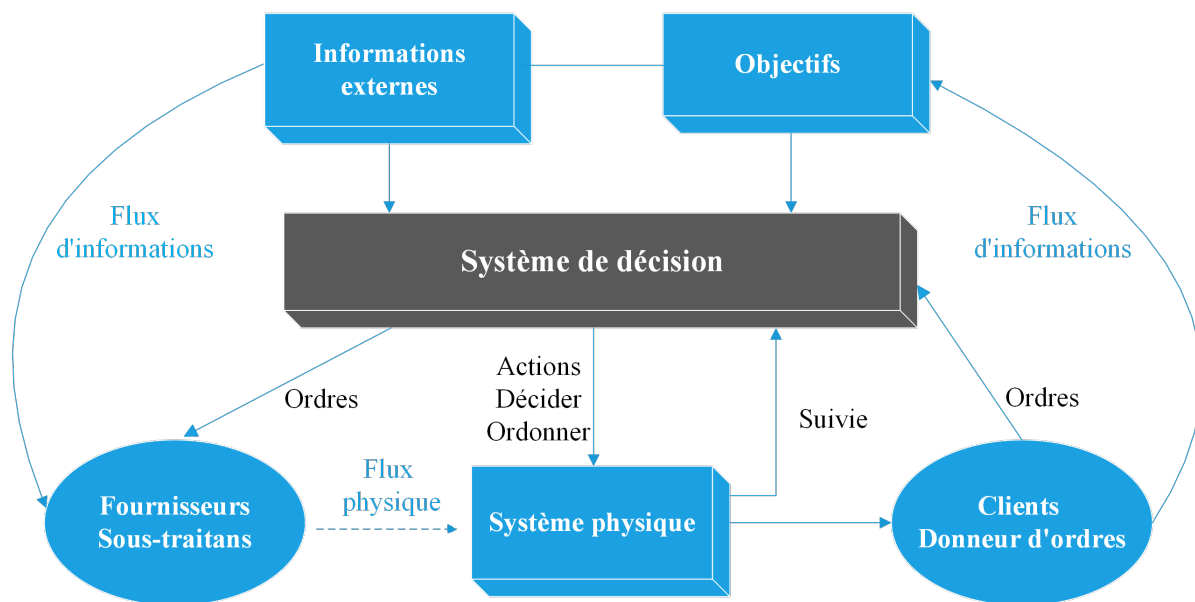


Figure 1.1: le modèle conceptuel d'un système de production

L'objectif de chaque système de production est de créer des produits, et chaque produit peut être fabriqué individuellement ou par lot. Dans le contexte de l'ordonnancement des opérations, chaque pièce ou lot de pièces à fabriquer est appelé "job". Chaque job suit une gamme opératoire, décrivant les opérations à réaliser, l'ordre de réalisation, les

machines affectées, la durée nécessaire à l'achèvement de l'opération, ainsi que les éventuelles contraintes, telles que la consommation de ressources additionnelles ou les relations de précedence entre les opérations.

Néanmoins, la gestion de ces systèmes est un domaine assez complexe en raison de leur nature et de la difficulté à générer, dans la majorité des cas, des solutions optimales manuellement. La taille des problèmes traités, ainsi que leur caractère déterministe ou stochastique, représentent un défi pour les praticiens. Les problématiques clés qui en découlent incluent la conception, la planification et le contrôle. La conception consiste à définir les caractéristiques du système, telles que la taille, les machines et les itinéraires de production. La planification implique l'allocation des ressources aux opérations et la définition des dates d'exécution. Enfin, le contrôle vise à gérer les perturbations, telles que les pannes ou les nouvelles commandes, afin de maintenir le fonctionnement efficace du système.

L'évolution récente des systèmes de production est marquée par un besoin accru de flexibilité et de réactivité. Les progrès technologiques, notamment dans le domaine de l'internet des objets (IoT), ont donné naissance à des systèmes de production cyber-physiques connectés en temps réel. Ces systèmes peuvent s'adapter rapidement pour produire des produits personnalisés, répondant ainsi à la demande changeante du marché tout en minimisant les coûts liés aux stocks.

De plus, les systèmes de production modernes doivent idéalement être flexibles en termes de volume, ce qui leur permet de suivre les fluctuations de la demande. Une usine idéale serait capable de s'ajuster à l'augmentation de la production lors du démarrage d'une nouvelle ligne, évitant ainsi des investissements inutiles en cas de sous-utilisation des installations.

1.2.2 Classification des systèmes de production

L'organisation et la classification des systèmes de production constituent une démarche essentielle, reposant sur la nécessité de définir des objectifs et de déterminer les ressources disponibles pour les atteindre. Cette approche vise à établir un équilibre approprié entre différentes contraintes telles que le facteur temps, la disponibilité des données et d'autres ressources cruciales. Toutefois, l'ampleur de la diversité des systèmes de production existants, caractérisée par une multitude de critères et de classes différentes, rend désormais difficile l'établissement d'une classification exhaustive qui englobe toutes ces caractéristiques. Dans ce contexte, nous allons passer en revue les classifications principalement répertoriées dans la littérature, dans le but de synthétiser les aspects que nous considé-

rons comme les plus pertinents.

1.2.2.1 Classification selon le volume de production

Cette classification est liée à l'un des critères clés, à savoir l'importance des séries et de la répétitivité. Cela implique de prendre en considération à la fois la quantité de produits fabriqués et la fréquence des séries produites tout au long du processus de production (Hitomi, 2017). Le volume de vente ou de production escompté, ainsi que la quantité de produit demandée sur une période donnée, que ce soit par mois, par saison ou par an, ont une grande influence sur les types de productions qui seront discutés ultérieurement.

- **Production unitaire** : Également connue sous le nom de "production sur mesure" ou "production à l'unité", se caractérise par la fabrication individuelle de chaque article, répondant ainsi aux exigences spécifiques de chaque client. Cette méthode de production est mise en œuvre lorsque les produits sont hautement personnalisés ou lorsque les quantités produites sont très limitées. Cette approche exige souvent une main-d'œuvre importante, entraînant des cycles de production relativement longs. Elle est réservée aux produits uniques, souvent complexes, nécessitant une coordination minutieuse de plusieurs ressources pour être livrés en temps voulu. La production unitaire est principalement adaptée aux cas où la taille du produit ou la demande impose une production en très faible quantité, mettant l'accent sur la réunion opportune et précise des ressources nécessaires pour chaque projet.
- **Production en petite et moyenne série** : Il s'agit de la production de quantités modérées de produits standardisés, également connue sous le terme de "production par lots". Ce mode de production se distingue par sa capacité à répondre aux besoins de personnalisation des clients tout en maintenant une efficacité opérationnelle. Elle repose sur des ateliers dans lesquels la variété des produits exige une flexibilité élevée des ressources, sans une automatisation poussée, favorisant ainsi la mobilité des produits au sein des installations. L'objectif est de garantir une coordination maîtrisée et une réponse agile aux demandes diversifiées du marché.
- **Production en grande série** : Qualifiée aussi par production de masse, constitue un procédé de fabrication uniforme et répétitif de biens en grandes quantités. Cette méthode est couramment employée pour la production de biens de consommation, tels que les automobiles, les appareils électroniques, les denrées alimentaires, et bien d'autres. Ses principes fondamentaux englobent la standardisation des produits pour assurer une uniformité maximale, ainsi que la mise en place d'une chaîne de produc-

tion composée de diverses machines conçues pour minimiser les temps de transfert et d'attente des pièces, tout en exigeant des performances et des débits similaires pour garantir une efficacité maximale et prévenir tout arrêt complet de la chaîne de production. De plus, cette approche se caractérise par une forte spécialisation des ressources de production, à la fois humaines et mécaniques, chacune étant dédiée à des tâches spécifiques, le tout accompagné d'un niveau élevé d'automatisation, optimisant ainsi le rendement et la qualité des produits fabriqués.

- **Production en continu** : Un concept, largement appliqué dans des secteurs tels que la chimie, les raffineries de pétrole et les centrales électriques, se distingue par sa capacité à maintenir une production constante, limitant au minimum les arrêts significatifs. Son objectif principal est de maintenir un flux de production régulier en utilisant en permanence les machines et les ressources pour produire des quantités importantes de produits de manière uniforme, à l'exception des pauses planifiées pour des changements de lot ou d'autres interruptions majeures. De plus, tous les produits suivent la même séquence d'opérations, passant par une série de postes de travail, dont la longueur peut varier. Bien que cette approche présente des similitudes avec la production en grande série, cette dernière peut impliquer des interruptions plus fréquentes en raison des ajustements nécessaires pour faire face à la variabilité de la production.

1.2.2.2 Classification selon la gestion de la demande

Cette seconde classification des systèmes de production repose sur la manière dont la production est orchestrée par rapport à la demande, ainsi que sur la gestion de la chaîne d'approvisionnement. En se plaçant au niveau opérationnel et en se basant sur des méthodologies distinctes, cette classification permet de distinguer deux catégories principales : les systèmes fonctionnant en flux tirés de ceux fonctionnant en flux poussés (Giard, 2003).

- **Production à flux tirés** : Dans ce cadre, la production est étroitement alignée sur la demande effective du marché ou du client. Les produits ne sont fabriqués ou livrés qu'en réponse à une demande réelle, ce qui les rend essentiellement "tirés" par cette demande. Cette coordination est généralement déclenchée par des signaux tels que les commandes des clients ou la constatation de faibles niveaux de stocks. Cette approche vise à éviter la surproduction, réduire les coûts de stockage et améliorer la réactivité à la demande.
- **Production à flux poussés** : La production est planifiée à l'avance en fonction

des prévisions de la demande ou des quotas de production, indépendamment de la demande réelle du marché. Dans ce système, le démarrage de chaque étape de la production est conditionné par la disponibilité des composants ou des produits provenant de l'étape précédente, assurant ainsi une progression étape par étape du processus de fabrication, connue sous le terme de "pilotage en amont". Cette approche peut se révéler efficace dans certaines industries nécessitant une production de masse, mais elle comporte le risque de gaspiller des ressources et d'engendrer des coûts de stockage élevés lorsque la demande réelle ne concorde pas avec les prévisions.

Cependant, en pratique, une classification purement théorique en flux tirés ou en flux poussés est rarement rencontrée. En réalité, les systèmes opérationnels intègrent souvent une approche hybride, combinant à la fois les deux modes de pilotages. Par exemple, dans un environnement de production, les produits simples sont généralement fabriqués en flux poussés, tandis que les produits complexes sont assemblés en flux tirés. Cette combinaison permet d'optimiser la gestion des ressources tout en répondant de manière plus précise à la demande du marché.

1.2.2.3 Classification selon la disposition des flux physique

Il s'agit d'une classification de la production dictée principalement par la nature du système physique et le volume des produits réalisés. Elle peut être répartie en trois catégories distinctes :

- **Système à flux continu** : Ce type de système est typiquement associé à "l'industrie de process", il se caractérise par un processus de transformation ininterrompu des matières premières entre les postes de travail, évitant tout stockage intermédiaire. Cette méthode de production exige des investissements substantiels, rentabilisés par une utilisation intensive des ressources et une automatisation poussée. Ces systèmes sont spécialisés dans la production de grandes quantités de produits, comme l'acier, le pétrole et l'électricité, offrant des avantages tels qu'une efficacité élevée, des coûts unitaires de production réduits et une utilisation optimale de l'équipement. Cependant, ils peuvent rencontrer des défis pour s'adapter rapidement aux fluctuations de la demande.
- **Système à flux discret** : Dans ce système, les éléments individuels sont traités séparément, avec des pauses ou des temps d'arrêt délibérés entre les cycles de production. Contrairement aux systèmes à flux continu, où la production est ininterrompue, les systèmes discrets fonctionnent par lots ou en séries. Ils sont couramment utilisés dans la production manufacturière, tels que la fabrication de voitures, d'ap-

pareils électroniques et d'équipements industriels.

- **Système à flux discontinu** : Ces systèmes présentent certaines similitudes avec les systèmes discrets en ce qui concerne les temps d'arrêt entre les cycles de production. Cependant, à la différence de ces derniers, où les périodes d'arrêt sont généralement planifiées, les systèmes à flux discontinu permettent des pauses variables en fonction des besoins de la production, des modifications dans le processus ou des demandes spécifiques. La production dans ce type de système se caractérise par de petites quantités de produits variés, fabriqués à partir d'équipements polyvalents, offrant ainsi une grande flexibilité.

L'avantage majeur de cette approche réside dans sa capacité d'adaptation en temps réel, permettant ainsi de répondre aux fluctuations de la demande et aux exigences particulières des clients. Cela en fait une option idéale pour les environnements caractérisés par une variation de production, tels que la fabrication de produits sur mesure.

1.2.2.4 Classification selon la circulation du produit dans l'atelier

Deux types de configurations de fabrication distinctes sont recensés dans cette classe, principalement en fonction de la séquence d'opérations et de la flexibilité du processus de production. Lorsque les produits suivent une séquence d'opérations prédéfinie, où chaque job passe par les mêmes étapes de production dans le même ordre avec éventuellement des temps opératoires variables, dans ce cas une production de type "flow shop" est mise en avant. En revanche, dans une production de type "job shop", l'ordre d'utilisation des postes de travail varie en fonction du processus d'élaboration de chaque produit, c'est-à-dire que chaque job possède sa propre gamme opératoire. Dans ce type d'atelier, les postes de charges sont souvent regroupés en section homogène ou en îlots de fabrication.

Cette distinction reflète la différence entre une production axée sur l'efficacité et la répétition (flow shop) et une production flexible adaptée à la diversité des produits et des processus (job shop).

1.2.2.5 Classification selon la relation avec le client

La nature du processus de fabrication d'un produit joue un rôle essentiel dans l'établissement de la relation entre l'entreprise et le client, déterminant ainsi la disponibilité et les spécifications des articles produits. Cette classification découle de la manière dont le processus de fabrication est initié, résultant en la création de trois systèmes distincts de gestion de la production.

- **Production à la commande** : La fabrication dans ce type de système est déclenchée en réponse à une commande précise du client, souvent accompagnée d'un travail de conception pouvant nécessiter la création de nouveaux composants. C'est le cas idéal pour une entreprise, car dans cette situation, elle ne produit que ce qu'elle vend.
- **Production sur stock** : Elle se déclenche en anticipant une demande solvable pour des produits dont les caractéristiques sont prédéfinies par le fabricant. Cela s'applique généralement lorsque la gamme de produits finis est limitée, et la demande pour chaque article doit être significative et prévisible.
- **Production mixte** : C'est une production hybride des deux approches précédentes, visant à réduire les délais de réalisation en produisant des sous-ensembles en stock, assemblés en personnalisant le produit final au moment de la commande. De plus en plus d'entreprises optent pour cette dernière option pour améliorer leurs performances vis-à-vis de leurs clients.

Dans cette section, une classification détaillée des systèmes de production a été exposée, en se basant sur divers critères tels que le mode de production, la nature et le volume des flux physiques, entre autres. L'objectif de cette typologie est de comprendre les processus de production et d'impacter la prise de décision en milieu industriel. Une organisation précise des systèmes de production joue un rôle majeur dans le processus décisionnel, donc la section suivante sera dédiée à exposer en détail les différents types de décisions.

1.3 Les différents niveaux de décisions

La gestion d'une organisation repose en grande partie sur la prise de décisions, un aspect crucial qui guide les actions de l'entreprise, influence ses résultats et détermine son succès ou son échec. Afin de mieux comprendre la complexité de ce domaine et d'accompagner les décideurs dans le processus décisionnel, différents types de décisions ont été élaborés, répartis à différents niveaux hiérarchiques. Dans la suite, nous décrirons les trois principales catégories de décisions illustrées dans la figure 1.2 : les décisions stratégiques, tactiques et opérationnelles, reflétant respectivement des horizons à long, moyen et court terme (Fisher, Jaikumar, & Van Wassenhove, 1986).

1.3.1 Les décisions stratégiques

Ce type de décision englobe l'ensemble des directives fondamentales prises au plus haut niveau de direction, définissant la trajectoire à long terme de l'entreprise. Avec un

horizon temporel étendu, elles influencent l'ensemble de l'organisation et sont alignées sur la vision et la mission de l'entreprise. Ces décisions comportent un niveau de risque élevé, car elles ont le potentiel d'avoir un impact significatif sur les résultats. De plus, une fois prises, elles sont généralement difficiles à modifier rapidement en raison de leur portée. La garantie de leur pertinence, ainsi que leur adéquation avec les tendances du marché, la concurrence et la réglementation en vigueur, n'est possible que par le biais d'une analyse minutieuse de l'environnement externe.

Par ailleurs, ces décisions affectent profondément la conception de l'entreprise, remettant en question des aspects tels que l'ouverture/fermeture de sites de production, la délocalisation, le développement de nouveaux produits et l'adaptation de nouveaux modes de production, etc.

1.3.2 Les décisions tactiques

Ces décisions sont prises à un niveau intermédiaire de gestion et visent à mettre en œuvre la stratégie de l'entreprise à moyen terme. Elles impliquent une coordination étroite entre les différents départements de l'organisation, couvrant divers domaines tels que la distribution, l'approvisionnement en matières premières, la taille de l'inventaire, la capacité d'utilisation des ressources, la planification de la production, etc.

Mesurables et adaptables, ces décisions ajustent les ressources internes pour atteindre les objectifs stratégiques, tout en s'adaptant aux évolutions du marché.

1.3.3 Les décisions opérationnelles

Les décisions opérationnelles revêtent une importance cruciale dans la gestion quotidienne des activités de l'entreprise. Elles s'inscrivent dans un horizon de temps limité, avec pour objectif principal d'assurer le bon pilotage et le contrôle du système de production à court terme. Leur rôle ne se limite pas à la gestion prévisible de la demande et des ressources, mais elles doivent également réagir de manière proactive aux imprévus et aux aléas, tout en respectant les décisions prises au niveau tactique. Ces décisions opérationnelles se manifestent à travers des activités telles que la gestion des stocks, garantissant ainsi la disponibilité des matières premières et des composants, et l'ordonnancement, qui consiste en une programmation détaillée des ressources mobilisées pour la production.

En somme, ces décisions ont un impact principalement interne à l'entreprise, assurant flexibilité et réactivité dans le processus opérationnel.

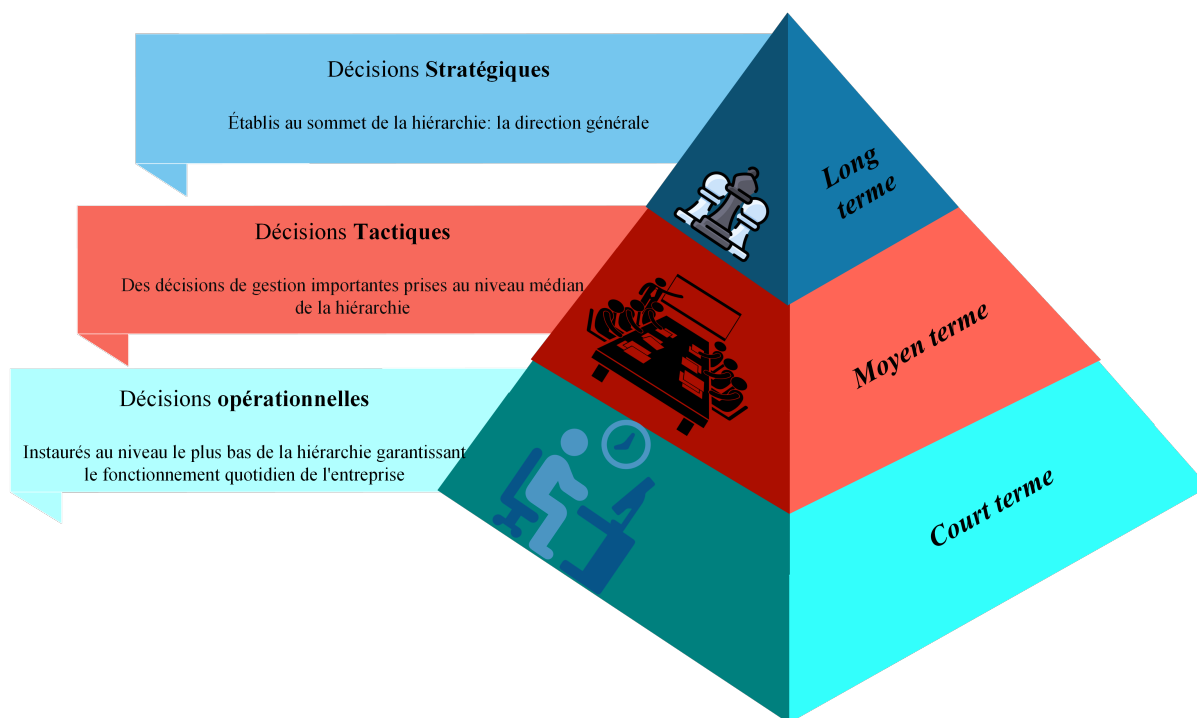


Figure 1.2: La hiérarchie des niveaux de décisions

Les décisions prises au niveau de la planification ne prennent généralement pas en compte les événements aléatoires susceptibles d'altérer le bon fonctionnement du système de production, comme c'est le cas lors de l'apparition de défauts de fabrication ou de retards éventuels dans la production. Ainsi, pour contrer ces événements indésirables et non programmables, il est crucial de définir des méthodes d'optimisation rapides qui permettent d'ajuster les décisions prises afin qu'elles restent applicables à l'environnement de production actuel en cas de changement. L'ordonnancement en particulier joue un rôle central dans ces décisions et constitue le principal sujet de notre étude. Ce concept sera décrit en détail dans la suite de ce chapitre.

1.4 Notions préliminaires sur les problèmes d'ordonnancement

L'ordonnancement, dans son sens le plus large, est aussi ancien que l'humanité. En gros, il consiste à définir les priorités ou à organiser les activités de manière à répondre à certaines exigences, contraintes ou objectifs. Ce concept peut être appliqué dans de nombreuses situations, de la planification de la production dans une usine à la gestion des horaires du personnel, en passant par l'optimisation des itinéraires de livraison pour une entreprise de transport. Ce domaine fait partie des sujets brûlants de la science moderne, par conséquent, plusieurs concepts, notations et méthodes d'optimisation relatifs à la

problématique de l'ordonnancement seront abordés.

1.4.1 Définition

“ Ordonnancer, c'est programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution. [...]. Enfin, il faut programmer les tâches de façon à optimiser un certain objectif...”

Carlier & Chrétienne, 1988, *Problèmes d'ordonnancement*

L'ordonnancement est le processus d'allocation d'un ensemble de ressources à un ensemble de tâches dans le but d'atteindre ou d'optimiser des objectifs spécifiques prédéfinis. Il implique la détermination de l'ordre dans lequel les tâches ou les opérations doivent être exécutées, en prenant en compte diverses contraintes, telles que les délais, les priorités et d'autres facteurs pertinents. L'objectif est d'optimiser un ou plusieurs critères qui peuvent prendre de nombreuses formes, notamment la maximisation de l'utilité du système, la minimisation de la durée d'exécution, la minimisation de la consommation d'énergie, etc. Plus précisément, l'ordonnancement est un sous-ensemble du produit cartésien de trois ensembles.

- L'ensemble des tâches (**What**) qui doivent être exécutées.
- L'ensemble de périodes de temps (**When**), les intervalles [début, fin], intuitivement, on peut associer à une tâche la période de temps pendant laquelle elle est exécutée.
- L'ensemble des ressources occupées par les tâches au cours de leurs exécutions. Nous désignons cet ensemble par le terme (**Where**), il comprend non seulement des ressources géographiquement localisées, mais aussi des matières premières et d'autres ressources.

Un ordonnancement est donc un sous-ensemble de **WhatxWhenxWhere**. En d'autres termes, il spécifie les ressources dont chaque tâche a besoin et le moment où elle en a besoin, afin de répondre à des objectifs préalablement définis. Les concepts de base, les notations, ainsi qu'une classification des problèmes d'ordonnancement seront examinés en détail dans les sous-sections à venir.

1.4.2 Concepts de base

De nombreux concepts et composantes cruciaux entrent en jeu lors de l'élaboration des modèles d'ordonnancement. Voici les éléments clés à prendre en considération dans ce contexte.

1.4.2.1 Les tâches

Les tâches représentent les unités de travail ou d'activités à planifier et à ordonnancer. Chacune d'entre elles est caractérisée par ses besoins en ressources, sa durée et l'heure à laquelle elle doit être achevée. Par ailleurs, la séquence d'exécution de ces activités est fortement influencée par l'ensemble des contraintes technologiques, telles que les relations de précédence. Dans le contexte de l'ordonnancement de la production, les tâches sont souvent appelées "jobs" et peuvent être des opérations dans un processus de production ou des exécutions de programmes informatiques. Parfois, les jobs peuvent être composés de plusieurs tâches élémentaires, appelées opérations.

De manière générale, il existe deux types de tâches ; les tâches morcelables, qui peuvent être interrompues en cours d'exécution et reprises ultérieurement, offrant ainsi une certaine flexibilité. En revanche, les tâches non morcelables doivent être exécutées en une seule fois, sans interruption. Dans ce cas, on parle de problèmes d'ordonnancement préemptifs et non préemptifs.

1.4.2.2 Les ressources

Les ressources sont des dispositifs techniques ou humains, dont la disponibilité peut être préalablement connue, qu'elle soit limitée ou non. Elles sont nécessaires à l'accomplissement des tâches et se divisent en deux catégories. On distingue les ressources renouvelables, qui demeurent disponibles dans la même quantité et état après chaque utilisation. En revanche, les ressources consommables, telles que la matière première, s'épuisent après avoir été utilisées.

De plus, parmi les ressources renouvelables, on peut distinguer les ressources disjonctives, qui ne peuvent être affectées qu'à une seule tâche à la fois, et les ressources cumulatives, qui peuvent être partagées entre plusieurs tâches simultanément, mais en quantité limitée.

1.4.2.3 Les contraintes

les contraintes définissent les règles et les restrictions qui guident la manière dont les tâches sont liées aux ressources et au temps. Généralement, elles se déclinent en deux catégories principales ; les contraintes temporelles et les contraintes de ressources.

- Les contraintes de ressources : Elles assurent une allocation efficace des ressources disponibles en quantités limitées, tout en garantissant que les tâches respectent les capacités et les exclusions requises. À leur tour, les contraintes de ressources se divisent en deux types, selon leur nature : disjonctive ou cumulative. Les contraintes de ressources disjonctives ne peuvent être allouées qu'à une seule tâche à la fois,

tandis que les contraintes de ressources cumulatives peuvent être attribuées à un nombre variable de tâches, mais celles-ci ne peuvent pas être réalisées simultanément.

- Les contraintes temporelles : Elles sont liées aux délais et à l'ordre dans lequel les tâches doivent être effectuées. Ce type de contrainte peut être répertorié de la manière suivante :
 - ◇ Les contraintes de temps alloué : Permettent d'englober les délais prescrits, souvent établis en fonction des exigences de gestion, liées aux échéances des tâches (comme les délais de livraison et la disponibilité des approvisionnements), ainsi qu'à la durée totale du projet. Elles servent à établir et maintenir la cohérence temporelle des tâches en déterminant, par exemple, le moment auquel une tâche doit commencer au plus tôt et la date limite pour son achèvement au plus tard.
 - ◇ Les contraintes de précedence : Plus généralement liées à la cohérence technologique, spécifient la disposition relative de certaines tâches par rapport à d'autres.
 - ◇ Les contraintes de calendrier : Associées à la conformité des horaires de travail.

D'autres catégories de contraintes peuvent être identifiées en plus de celles précédemment mentionnées. Parmi celles-ci, on peut notamment distinguer les contraintes liées au système de production. Dans ce contexte, ces contraintes se divisent en deux principales catégories. Les contraintes endogènes qui sont intrinsèquement liées au système de production et influencent directement ses performances. Elles englobent des éléments tels que les dates de disponibilité des machines, ainsi que leurs capacités et les gammes opératoires. En ce qui concerne les contraintes exogènes, elles sont imposées de manière externe au système de production, indépendamment de ses caractéristiques internes. Ces contraintes peuvent inclure des priorités spécifiques de commandes et de clients, ainsi que d'éventuels délais accordés pour la réalisation de certains produits.

1.4.2.4 Les critères d'évaluation

Les critères sont des mesures établies pour évaluer la qualité et la pertinence d'un ordonnancement de tâches, en fonction d'objectifs spécifiques tels que la minimisation du temps de réalisation, la réduction des coûts ou la maximisation de l'utilisation des ressources. Cela s'effectue en prenant en compte des contraintes spécifiques au système étudié. En milieu industriel, il est fréquent d'avoir plusieurs fonctions objectives à optimiser, transformant ainsi le problème d'ordonnancement en un problème multicritère. Face à cette complexité, des compromis doivent être faits entre ces critères, car l'optimisation d'un aspect peut avoir des répercussions négatives sur d'autres.

1.4.2.5 Les objectifs

Lors de la résolution des problèmes d'ordonnancement, deux stratégies principales se dessinent : une axée sur l'optimalité des solutions par rapport à des critères spécifiques, et une autre centrée sur la faisabilité en respectant les contraintes.

Ces objectifs sont cruciaux pour évaluer l'efficacité des méthodes de résolution dans les problèmes d'ordonnancement, ils offrent un cadre adéquat pour mesurer l'efficacité des résultats obtenus et guident la prise de décision dans la gestion des tâches et des ressources. La classification d'Esquirol et al. (s. d.) distingue ces objectifs de la manière suivante :

- Liés au temps : se rapporte à la minimisation du temps total d'exécution de l'ensemble des tâches, qui correspond au Makespan, l'objectif le plus étudié dans la littérature, aussi la réduction des temps de réglage, des temps de cycles, ou encore la minimisation des retards par rapport aux dates de livraison.
- Liés aux ressources : Ils englobent la maximisation du taux d'utilisation d'une ressource donnée ou la minimisation des ressources utilisées.
- Liés aux coûts : Ces critères visent à minimiser divers coûts, tels que les coûts de lancement, de stockage, de production, de transport et les coûts résultant de l'inoccupation des machines.

Par ailleurs, les objectifs liés à l'environnement ont connu une ascension fulgurante ces dernières années. Parmi ces objectifs, la consommation d'énergie ainsi que les émissions de dioxyde de carbone sont devenues des préoccupations primordiales pour le secteur industriel, en raison de leur impact sur l'environnement. Dans ce contexte, plusieurs chercheurs se sont penchés sur la résolution des problèmes d'ordonnancement éco-énergétique en proposant différentes approches de résolution (Fang, Uhan, Zhao, & Sutherland, 2013), (Mouzon, Yildirim, & Twomey, 2007), (D. N. Sekkal & Belkaid, 2020), (Lamy, 2017).

Pour plus de détails et de précisions concernant les problèmes d'ordonnancement, les lecteurs peuvent s'appuyer sur de nombreux ouvrages de référence couvrant une multitude de domaines d'application, en particulier le secteur industriel. Parmi ces ouvrages, citons (Baker, 1974), (Carlier et al., 1993), (Esquirol et al., s. d.), (Giard, 2003), (M. L. Pinedo, 2012) et (M. L. Pinedo & Pinedo, 2016).

1.4.3 Typologie des ordonnancements de la production

Plusieurs solutions satisfaisant les critères préalablement définis sont découvertes lors de la recherche d'un ordonnancement, comme l'ont mis en évidence les travaux de Caumond (2006), Shahzad (2011) et Zahmani (2018). Pour cette raison, il s'avère essentiel de seg-

menter l'ordonnancement en différentes catégories afin de restreindre la recherche de solutions à des sous-ensembles spécifiques. Cette approche a pour objectif d'éviter une exploration exhaustive de l'ensemble de l'espace de recherche, ce qui permet de réaliser un gain significatif en termes de temps nécessaire pour résoudre le problème. La figure 1.3 illustre la manière dont ces diverses catégories d'ordonnancement interagissent les unes avec les autres.

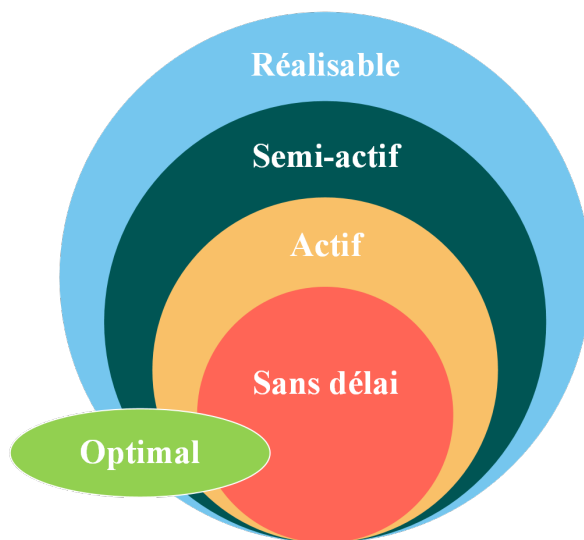


Figure 1.3: Typologie des ordonnancements

- **Ordonnancement réalisable** : il fait référence à un ordonnancement de tâches, d'opérations ou de ressources qui peut être effectivement mis en œuvre dans un système donné en respectant les contraintes et les limitations spécifiques.
- **Ordonnancement semi-actif** : dans ce type d'ordonnancement, aucune tâche ne peut être achevée plus tôt sans modifier l'ordre de traitement sur une machine dans une configuration non préemptif réalisable. Par conséquent, la seule façon d'améliorer les performances consiste à réorganiser la séquence des opérations sur les machines.
- **Ordonnancement actif** : un ordonnancement faisable est qualifié d'actif lorsqu'il devient impossible d'avancer le traitement d'une opération sans retarder le début d'une autre. De plus, aucune opération ne peut être insérée dans un emplacement vide plus tôt dans la séquence non préemptif sans que sa faisabilité ne soit compromise. Il est important de souligner que tout ordonnancement actif est invariablement de nature semi-actif.
- **Ordonnancement sans délai** : la politique appliquée dans cette classe stipule qu'aucune machine ne reste inactive pendant qu'une tâche est en attente de traitement. Par conséquent, un ordonnancement sans délai est inévitablement actif et donc semi-actif.

- **Ordonnancement optimal** : l'objectif à travers l'optimisation de tels problèmes est de générer les meilleures valeurs possibles adaptées à une mesure de performance spécifique. Dans un problème d'ordonnancement donné, il est parfois possible de générer plusieurs solutions qui conduisent à des résultats identiques. L'atteinte de l'optimalité peut varier significativement en fonction des objectifs, en particulier dans le contexte de l'optimisation multi-objectifs.

1.4.4 Les ateliers de production

L'ordonnancement est un domaine d'une grande ampleur qui a suscité l'intérêt de nombreux chercheurs au fil du temps. Il a été largement exploré, depuis la définition de ses composants et de ses caractéristiques jusqu'à la proposition de classifications exhaustives visant à évaluer son importance dans la planification de la production. Une classification particulièrement controversée concerne les différents types d'ateliers de production présents dans le secteur industriel, basée principalement sur la manière dont les tâches sont séquencées et les machines configurées. Fondamentalement, un atelier peut être caractérisé par son type et le nombre de machines qu'il comporte. Concernant son type, plusieurs classifications sont possibles.

1.4.4.1 Problème d'ordonnancement à une seule machine

Ce type d'atelier est composé d'une seule machine sur laquelle un ensemble de tâches va être exécuté suivant un séquençement précis, optimisant ainsi des critères et respectant des contraintes prédéfinies. Les tâches, quant à elles, sont composées d'une seule opération réalisée par la même machine.

En fin de compte, le cas d'une machine unique est le plus simple de tous les environnements de machines possibles et constitue un cas particulier pour tous les autres environnements de machines complexes. Il sert de fondement à la compréhension des principes essentiels de l'ordonnancement, tout en étant souvent utilisé comme point de départ pour aborder des problèmes plus complexes. Des exemples de modèles de machines uniques sont présentés dans (Biskup, 1999), (Ji, He, & Cheng, 2007), (Shrouf, Ordieres-Meré, García-Sánchez, & Ortega-Mier, 2014), (Aghelinejad, Ouazene, & Yalaoui, 2019).

1.4.4.2 Problème d'ordonnancement à machine parallèle

Ce type de problème représente une extension des problèmes d'ordonnancement à une seule machine. Dans cet atelier, les tâches, se composant d'une seule opération, peuvent être exécutées sur un ensemble de machines disposées en parallèle, bien que seule une machine soit nécessaire pour effectuer l'opération.

L'objectif est double dans ce type de problème, car il englobe non seulement l'affectation des tâches aux machines, mais aussi leur séquençage. Les machines exécutent les tâches de la même manière, mais avec des cadences différentes, d'où la nécessité de différencier entre les configurations suivantes :

- Machines parallèles identiques (P) : toutes les machines fonctionnent à la même vitesse, et par conséquent les durées d'exécution des opérations ne dépendent pas de la machine.
- Machines parallèles uniformes (Q) : chaque machine a sa propre vitesse d'exécution constante et distincte des autres.
- Machines parallèles non-uniformes (R) : la vitesse d'exécution varie non seulement entre les machines, mais également pour chaque opération.

1.4.4.3 Problème d'ordonnancement à cheminement unique (flow shop)

Ce problème se caractérise par des ensembles d'opérations pour chaque tâche, présentant une gamme opératoire commune qui implique une visite de toutes les machines disposées en série selon un ordre spécifique. Lorsque les tâches suivent un cheminement uniforme à travers toutes les machines, sous l'hypothèse d'une séquence identique de passage sur chacune, une vision simplifiée de ce problème est alors considérée, appelée flow shop de permutation. Ce type d'atelier est fréquemment observé dans la production de masse, où les schémas de fabrication sont répétitifs pour tous les produits. Cependant, malgré cette répétition, l'hétérogénéité des temps de traitement crée un défi en termes d'optimisation des délais. Ce problème constitue le point central de cette thèse et sera analysé en détail par la suite.

1.4.4.4 Problème d'ordonnancement à cheminement multiple (job shop)

Cet atelier se distingue par la variabilité des gammes opératoires, permettant à chaque tâche de suivre un cheminement unique à travers les machines. Généralement, les job shops sont considérés comme des problèmes d'ordonnancement généraux, englobant les problèmes de flow shop et d'open shop, qui sont eux-mêmes considérés comme des cas spéciaux de ce problème. Cela garantit une plus grande flexibilité dans la séquence des opérations, mais malheureusement, cela s'applique généralement à des volumes de production relativement faibles.

1.4.4.5 Problème d'ordonnancement à cheminement libre (open shop)

Dans un atelier de type open shop, la séquence des opérations n'est pas préétablie ; seules les machines sur lesquelles les opérations doivent être effectuées sont connues. Les

opérations de chaque tâche peuvent être exécutées dans n'importe quel ordre, ce qui rend ce type de problème complexe et de nature stochastique. Il est important de noter que ce type de problème est rarement rencontré dans l'industrie et peu exploré dans la littérature en raison de sa complexité et de sa rareté.

1.4.5 Représentation des problèmes d'ordonnancement

Les problèmes d'ordonnancement peuvent être représentés par le biais de plusieurs méthodes de modélisation. Deux approches courantes sont employées dans la littérature, à savoir la modélisation mathématique, qui constitue un outil puissant pour formaliser et quantifier des phénomènes complexes. Cette approche consiste à représenter les données, les contraintes et les critères d'évaluation sous forme d'équations, de variables et de concepts mathématiques.

Il existe également la représentation graphique, qui utilise des outils tels que la méthode Pert, les réseaux de Petri et les diagrammes de Gantt pour illustrer visuellement les tâches, les opérations et les ressources.

En général, en matière d'ordonnancement, le diagramme de Gantt est l'outil graphique le plus utilisé en raison de sa simplicité de lecture et d'interprétation. Il a été développé par Gantt en 1910 et trouve également des applications dans la planification de projets au sein des entreprises. Ce diagramme est composé de lignes horizontales représentant les machines utilisées, tandis que les tâches attribuées à chaque machine sont symbolisées par des rectangles dont la longueur est proportionnelle à leur durée, ainsi le diagramme de Gantt offre une vision d'ensemble des tâches réalisées sur toutes les machines au fil du temps. Un exemple d'atelier composé de trois machines disposées en série est illustré dans la figure 1.4, avec les durées opératoires de chaque tâche répertoriées dans le tableau 1.1.

Tableau 1.1: Temps de traitement des jobs

	J_1	J_2	J_3	J_4
P_{j1}	4	2.5	1.5	3
P_{j2}	3	3	1	5
P_{j3}	2.5	3	4.5	1

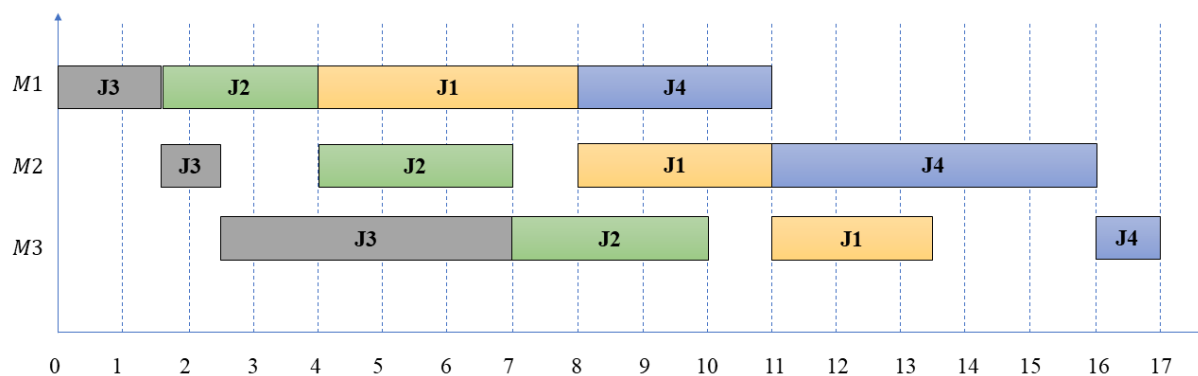


Figure 1.4: Représentation d'un ordonnancement par diagramme de Gantt

1.4.6 Notation des problèmes d'ordonnement

En dehors des méthodes de représentation graphique, il existe également un système de notation $(\alpha | \beta | \gamma)$ qui est largement utilisé pour décrire les problèmes d'ordonnement. Cette notation proposée par Graham et al. (1979) et actualisée par Brucker (2007) spécifie en détail les trois champs. Le champ α définit l'environnement des machines, qui est décomposé en deux sous-champs α_1 et α_2 où α_1 désigne l'organisation de la machine et α_2 le nombre de machines. Dans le cas où $\alpha_1 = \emptyset$ et $\alpha_2 = 1$, une organisation à une seule machine est alors considérée. Si $\alpha_1 \in \{\emptyset, P, Q, R\}$ alors chaque job est constitué d'une seule opération. Si $\alpha_1 \in \{F, J, O\}$ alors le job est composé d'un ensemble d'opérations. De plus, si α_2 n'est pas précisé, alors le nombre de machines est quelconque.

Le deuxième champ (β) indique les contraintes à respecter qui se rapportent à des conditions de traitement et des détails importants qui caractérisent le problème. Plusieurs entrées peuvent être énumérées. Ce champ peut également être vide s'il n'y a pas de conditions particulières. Les principaux symboles possibles et leurs définitions pour les champs α et β sont énumérés dans le tableau 1.2

Tableau 1.2: Description des symboles d'utilisation dans la notation de Graham : partie 1

α		β	
Symbole	Description	Symbole	Description
1	Une machine	<i>pmtn</i>	La préemption des jobs est autorisée
<i>Pm</i>	Machines parallèles identiques	<i>nwt</i>	Sans attente
<i>Qm</i>	Machines uniformes	<i>prec</i>	Contraintes de précédence
<i>Rm</i>	Machines indépendantes	<i>batch(b)</i>	Traitement par lots
<i>Fm</i>	Flow shop	<i>brkdw</i>	Possibilité de pannes ou d'arrêts
<i>Jm</i>	Job shop	<i>rcrc</i>	Recirculation
<i>FFc</i>	Flow shop flexible	r_j	Date de disponibilité
<i>FJc</i>	Job shop flexible	d_j	Date d'exécution au plus tard
<i>Om</i>	Open shop	M_{jk}	Restrictions d'éligibilité des machines

L'ensemble des contraintes et caractéristiques apparaissant dans le champ β peut être varié, nous n'expliquons en dessous que celle utilisée dans cette thèse.

- ***pmtn*** : la préemption est autorisée, Cela implique que la tâche peut être interrompue, ce qui permet de la démarrer sur une machine, de la suspendre, puis de la reprendre ultérieurement, même sur une machine différente. Lorsque les préemptions sont autorisées, le terme "*pmtn*" est inclus dans le champ β ; lorsque "*pmtn*" est exclu, les préemptions ne sont pas autorisées.
- ***prmu*** : Désigne que l'ordre ou la permutation des tâches sur les machines est le même. Cette contrainte est rencontrée dans l'environnement du flow shop.
- ***fmls*** : Cette contrainte stipule que les jobs sont répartis en famille (groupes). Les tâches d'une même famille peuvent avoir des temps de traitement différents, mais elles peuvent être exécutées séquentiellement sur une même machine sans nécessiter de configuration entre elles. Toutefois, lorsqu'une machine passe d'une famille à une autre, par exemple de la famille g à la famille h , une phase de préparation est requise. Si cette durée de préparation dépend à la fois des familles g et h ainsi que de la séquence, elle est notée s_{gh}
- **$S_{sd,gh}$** : Temps de préparation des familles dépendant de la séquence. Le temps de préparation d'une machine pour traiter un job appartenant à la famille g dépend du job précédent de la famille h traitée sur cette machine. Si aucun $S_{sd,gh}$ n'apparaît dans le champ β , tous les temps de préparation sont supposés être 0 ou indépendants de la séquence, auquel cas, ils sont simplement inclus dans les temps de traitement.

Toute autre entrée pouvant apparaître dans le champ β est explicite, par exemple $P_j = P$ implique que tous les temps de traitement sont égaux, et $d_j = d$ implique que toutes les

dates d'échéance sont égales.

Le troisième champ γ dans la notation de Graham et al. (1979) indique la fonction objective à optimiser. Il peut contenir une seule entrée ou une combinaison de plusieurs critères, comme c'est le cas pour l'optimisation multiobjectif. Étant donné qu'il existe de nombreuses fonctions objectives étudiées dans la littérature, il est essentiel de déterminer les variables les plus fréquemment utilisées dans la formulation de ces objectifs. Parmi ces variables, on trouve la date C_j de fin d'exécution de la tâche j et d_j la date due qui exprime la date d'exécution au plus tard de la tâche j . Les principales fonctions objectives considérées sont les suivantes :

- **$Cmax$** : Minimiser le makespan qui s'avère être la fonction objective la plus populaire en ordonnancement, il correspond à la minimisation de la date de fin de la dernière tâche exécutée (durée de l'ordonnancement), i.e $\min \left\{ \max_{j \in J} \{C_j\} \right\}$.
- $\sum C_j$: Minimiser la somme des dates de fin de chaque tâche. i.e $\min \sum_{j \in J} C_j$.
- $\sum T_j$: Minimiser la somme des retards des tâches. i.e $\min \sum_{j \in J} T_j$, avec $T_j = \max \{C_j - d_j, 0\}$
- **$Lmax$** : Minimiser le retard maximum des tâches, où L_j est communément appelé retard algébrique. i.e $\min \left\{ \max_{j \in J} \{L_j\} \right\}$, où $L_j = C_j - d_j$, représentant l'écart par rapport à la fin souhaitée.
- $\sum U_j$: Minimiser le nombre de tâches en retard. La fonction de pénalité unitaire U_j est définie telle que $U_j = 1$ si la tâche j est en retard ($C_j > d_j$) et 0 sinon. Par conséquent, l'objectif est de minimiser la somme $\min \sum_{j \in J} U_j$. Cependant, il se peut que dans la littérature la signification de U_j soit inversé et dans ce cas l'objectif est de maximiser la somme des U_j qui représente dans ce cas le débit.
- $\sum E_j$: Minimiser la somme de l'avance des tâches. i.e $\min \sum_{j \in J} \max(0, d_j - C_j)$.

Il est important de noter qu'en minimisant une somme, il est courant d'attribuer des poids aux différents jobs. Par exemple, au lieu de minimiser simplement $\sum_{j \in J} T_j$, il est souvent nécessaire de minimiser $\sum_{j \in J} w_j \cdot T_j$, où w_j représente le poids (ou coefficient de pondération) associé au job j . Cela permet de modéliser des situations où certains jobs sont plus importants que d'autres.

De plus, lorsque nous avons plusieurs objectifs, il est possible d'avoir des poids distincts pour les jobs et pour les fonctions. Par exemple, en cherchant à minimiser $\sum_{j \in J} w'_j \cdot E_j + w''_j \cdot T_j$, nous accordons plus de poids au retard qu'à l'avance (ou vice versa). Cette notion de pondération est essentielle pour refléter les priorités et les contraintes spécifiques

d'un problème donné. D'autres fonctions objectives faisant appels à différents symboles répertoriés dans le champ γ sont listés dans le tableau 1.3

Tableau 1.3: Description des symboles d'utilisation dans la notation de Graham : partie 2

γ	
Symbole	Description
$C_{\max} = \max\{C_1, \dots, C_n\}$	Makespan (Cmax)
$TCT = \sum_{j=1}^n C_j$	La somme des dates de fin
$TWCT = \sum_{j=1}^n w_j \cdot C_j$	La somme des dates de fin pondérées
$L_{\max} = \max\{L_1, \dots, L_n\}$	Retard algébrique maximal
$TT = \sum_{j=1}^n T_j$	Retard total
$TWT = \sum_{j=1}^n w_j \cdot T_j$	Retard total pondéré
$\sum_{j=1}^n U_j$	Nombre total de jobs en retard
$\sum_{j=1}^n w_j \cdot U_j$	Nombre total pondéré de jobs en retard

1.5 Temps de traitement variables

Au cours des deux dernières décennies, la littérature scientifique a accueilli de nouveaux concepts visant à mieux aligner les modèles d'ordonnancement sur les réalités des problèmes rencontrés dans divers domaines. Parallèlement, de nouvelles approches ont émergé pour appréhender la principale variable d'entrée, à savoir le temps de traitement. Traditionnellement considéré comme constant et prévisible, ce dernier s'est avéré être sujet à des fluctuations, une constatation éloignée des contraintes souvent rigides imposées par les modèles théoriques et peu représentatifs des environnements industriels réels.

De surcroît, l'exécution des programmes prévisionnels peut être confrontée à l'émergence de nouvelles informations susceptibles d'obsolescence. Cette réalité complexe, caractérisée par un environnement non déterministe et incertain, a incité de nombreux chercheurs à remettre en question la viabilité des approches traditionnelles en matière d'ordonnancement. Ils se sont ainsi penchés sur la question de l'ordonnancement sous incertitude afin de rendre les modèles plus adaptés à la pratique industrielle.

Dans ce contexte, il est crucial de reconnaître que les temps de traitement des tâches peuvent varier en fonction des conditions de production. Par exemple, dans de nombreux secteurs industriels, des facteurs tels que l'état de la machine ou l'expérience des opérateurs peuvent influencer les performances. Ces variations sont généralement représentées dans la littérature par des effets de détérioration ou d'apprentissage (Browne & Yechiali, 1990); (Biskup, 1999); (J.-B. Wang, 2006).

Peu importe l'effet appliqué à la machine (détérioration ou apprentissage), la littérature explore deux grandes classes de variations des tâches :

- Les effets dépendant de la position (Position-dependent effects), où le temps de traitement d’une tâche j varie en fonction de sa position dans l’ordonnancement, noté $p(j,r)$. Cette approche a été étudiée notamment dans les travaux de Bachman et Janiak (2004), Eren et Güner (2007).
- Les effets dépendants du temps (Time-dependent effects), où le temps de traitement d’une tâche j fluctue en fonction de sa date de début, noté $p(j,t)$. T. E. Cheng et al. (2004) synthétisent les recherches portant sur les problèmes d’ordonnancement impliquant des tâches dont la durée varie en fonction de leur date de début.

1.5.1 Effet de détérioration

Au fil des dernières décennies, les problèmes d’ordonnancement comportant un effet de détérioration ont été étudiés sous différentes configurations et avec diverses mesures de performance. Les travaux de Gupta et Gupta (1988) ainsi que de Browne et Yechiali (1990) sont souvent cités comme pionniers dans l’introduction de cet effet dans les problèmes d’ordonnancement.

Dans les cas où un effet de détérioration est présent, le temps nécessaire pour exécuter une tâche augmente proportionnellement à l’utilisation de la machine, ce phénomène étant attribué au vieillissement de l’équipement.

De nombreux exemples issus de l’industrie ou de la vie courante peuvent illustrer ce phénomène. Par exemple, dans la lutte contre les incendies, les efforts et le temps nécessaires pour les éteindre augmentent si le début de l’intervention est retardé. Cela ressemble à la situation où un traitement médical plus intensif peut être nécessaire lorsque l’état de santé d’un patient se détériore. Dans le secteur industriel, cela peut se manifester lorsque la température d’un lingot qui attend d’entrer dans une machine à laminier descend en dessous d’un certain niveau, obligeant ainsi à le réchauffer avant de le laminier. De même, lors de l’étape de perçage, l’outil utilisé peut se détériorer à force d’être utilisé, ce qui entraîne une augmentation du temps de traitement pour les produits suivants. Dans ce cas, les temps opératoires sont représentés par une fonction croissante dépendant des tâches précédemment ordonnancées.

L’effet de détérioration a été représenté par diverses méthodes analytiques. Comme mentionné précédemment, on peut distinguer :

- Effet de détérioration dépendant de la position :

$$P(j, 1) \leq P(j, 2) \leq \dots \leq P(j, n) \quad \forall j = 1, \dots, n \quad (1.1)$$

Avec $P(j, r)$ le temps de traitement de la tâche j dans la position r .

Dans ce cas, plus la tâche est exécutée ultérieurement, c'est-à-dire plus sa position est éloignée, plus son temps opératoire augmente.

- Effet de détérioration dépendant de la date de début :

$$P(j, t_1) \leq P(j, t_2) \quad \forall j = 1, \dots, n, \forall t_1 \leq t_2 \quad (1.2)$$

Où $P(j, t_r)$ représente le temps opératoire de la tâche exécutée dans la position r , avec t_r indiquant sa date de début.

En raison de leur pertinence pour diverses situations réelles, les problèmes d'ordonnancement avec effet de détérioration ont été largement étudiés dans la littérature, impliquant différentes hypothèses au sein de diverses configurations d'ateliers. Les travaux réalisés par Ji et Cheng (2007) ont considéré une fonction de temps de traitement linéaire par morceaux, où celui-ci n'augmente que si la date de début dépasse un seuil minimal donné "d" et cesse d'augmenter lorsque la date de début atteint un seuil maximal donné "D". De même, Rachaniotis et Pappis (2006) ont démontré que le défi consistant à allouer une seule ressource pour maîtriser plusieurs incendies dans une forêt peut être représenté par un problème d'ordonnancement à une machine avec un effet de détérioration.

Cet effet a été inclus dans plusieurs études, notamment dans les systèmes de production comprenant des périodes de maintenance. D.-L. Yang et al. (2012). ont abordé un problème d'ordonnancement à machine parallèle équipé d'outils de coupe. Ils ont constaté que des actions de maintenance programmables permettaient de réduire les effets de détérioration des machines dus à l'usure de l'outil de coupe, permettant ainsi à la machine de retrouver son état initial. M. Cheng et al. (2015) ont étudié un problème d'ordonnancement de type flow shop à deux machines avec effet de détérioration, où le temps de traitement des tâches dépend de leurs dates de début dans la séquence. Ils ont proposé un modèle de programmation en nombres entiers mixtes ainsi qu'un algorithme de Branch and Bound pour la résolution du problème, dans le but de minimiser la somme pondérée du makespan et la somme des dates de fin.

Plusieurs travaux portant sur l'effet de détérioration dans les problèmes d'ordonnancement ont émergé ces dernières années, notamment ceux de Sánchez-Herrera et al. (2019), N. Sekkal et Belkaid (2020), X. Sun et al. (2023). Les fonctions analytiques mentionnées (temporelles et positionnelles) sont représentées dans les tableaux 1.4 et 1.5.

1.5.2 Effet d'apprentissage

Les variations des temps de traitement, même dans des conditions quasi identiques, peuvent être attribuées à divers facteurs. Par exemple, lorsqu'un travailleur assemble une série de produits similaires, le temps nécessaire pour chaque produit peut varier en fonction de ses compétences, de son expérience et de l'organisation de l'environnement de

travail. À mesure que le travailleur répète les mêmes tâches, il acquiert de l'expérience et améliore son efficacité, ce qui se traduit par une réduction du temps de traitement. Ce phénomène est connu sous le nom d'effet d'apprentissage, initialement observé dans l'industrie aéronautique par Wright (1936) et introduit ultérieurement dans le domaine de l'ordonnancement par Biskup (1999), T. Cheng et Wang (2000).

La réduction du temps de traitement peut être modélisée par une fonction décroissante en fonction du nombre de produits déjà assemblés. Des études telles que celles menées par Biskup (1999), Ji et al. (2015), Fichera et al. (2015), et D. N. Sekkal et Belkaid (2020) ont exploré cette dynamique. Ainsi, la fonction dépendant de la position, comme mentionné précédemment, peut être formalisée pour rendre compte de ces variations comme suit :

$$P(j, 1) \geq P(j, 2) \geq \dots \geq P(j, n) \quad \forall j = 1, \dots, n \quad (1.3)$$

Par ailleurs, les concepts de temps de traitement dépendant du temps et d'effet de détérioration/apprentissage ont été introduits dès le début pour les problèmes d'ordonnancement à une seule machine, comme décrit dans l'article de J.-B. Wang et al. (2008). Quelques exceptions existent pour les machines parallèles Shokoufi et al. (2019) et les modèles flow shop à deux machines G. Li et al. (2013). L'objectif principal est généralement de minimiser les mesures de performance traditionnelles telles que le makespan, le temps d'écoulement, le retard maximal et le nombre de tâches en retard. La fonction utilisée pour modéliser le temps de traitement d'une tâche dépendant du temps est généralement une fonction linéaire ou linéaire par morceaux de la date de début de la tâche dans un ordonnancement, et elles sont souvent représentées comme suit :

$$P(j, t_1) \geq P(j, t_2) \quad \forall j = 1, \dots, n, \forall t_1 \leq t_2 \quad (1.4)$$

De nombreuses applications concrètes de l'effet d'apprentissage ont été examinées. Souvent, cet effet est intégré dans des problèmes qui représentent des systèmes de production manuels et semi-automatiques. Par exemple, K. I. Ho et al. (1993) ont appliqué un modèle d'ordonnancement incluant un effet d'apprentissage pour simuler un problème de détection des menaces aériennes. Dans ce cas, le temps requis pour la détection diminue à mesure que la menace se rapproche de la station radar. En s'appuyant sur le modèle d'apprentissage de Biskup (1999), une gamme variée de modèles basés sur la position de la tâche r a été étudiée dans la littérature. Des détails supplémentaires sur les modèles proposés sont fournis dans les tableaux 1.4 et 1.5.

Tableau 1.4: Les modèles de temps de traitement en fonction de la position

Modèle $p(j, r)$	l'effet de détérioration avec a, a_j, b_j, σ et σ_j sont des facteurs de détérioration	l'effet d'apprentissage avec a, a_j, b_j, σ et σ_j sont des facteurs d'apprentissage	Références
$P_j r^a$	$a > 0$	$a < 0$	Biskup (1999), N. Sekkal et Belkaid (2020), D. N. Sekkal et Belkaid (2020)
$P_j r^{a_j}$	$a_j > 0$	$a_j < 0$	Mosheiov et Sidney (2003); S.-J. Yang et Yang (2010); Senyigit et Atici (2018)
$P_j + b_j r$	$b_j > 0$	$b_j < 0$	Bachman et Janiak (2004); M. Cheng et al. (2018); Alfares et al. (2021)
$P_j \sigma^{(r-1)}$	$\sigma > 1$	$0 < \sigma < 1$	Dolgui et al. (2012).
$P_j \sigma_j^{(r-1)}$	$\sigma_j > 1$	$0 < \sigma_j < 1$	S.-J. Yang et Yang (2013)
$P_j g(r)$	$g(r)$ fonction croissante	$g(r)$ fonction décroissante	Rustogi et Strusevich (2012a, 2012b)
$P_j g_j(r)$	$g_j(r)$ fonction croissante	$g_j(r)$ fonction décroissante	Yin, Wu, et al. (2014)

Tableau 1.5: Les modèles de temps de traitement en fonction du temps

Fonction $p(j, t)$	Effets de détérioration où $a, a_j,$ b_j, σ et σ_j sont des facteurs de détérioration	Effet d'apprentissage où a, a_j, b_j, σ et σ_j sont des facteurs d'apprentissage	Références
P_{jt}	✓	-	Wu et al. (2008); Low et Lin (2013)
$P_j(a + bt)$	$b > 0$	$b < 0$	J.-B. Wang et al. (2008); Rustogi et Strusevich (2015)
$P_j + b_j t$	$b_j > 0$	$b_j < 0$	Bosio et Righini (2006); Luo (2015)
$P_j f(t)$	$g(r)$ fonction croissante	$g(r)$ fonction décroissante	Yin, Cheng, et al. (2014)

1.5.3 Temps de traitement contrôlables

Dans les problèmes d'ordonnancement avec des temps de traitement contrôlables, les durées des tâches ne sont pas prédéfinies comme des constantes comme dans l'ordonnance-

ment classique. Au contraire, elles peuvent être ajustées (c'est-à-dire réduites) en allouant des ressources supplémentaires telles que des fonds additionnels, des heures supplémentaires, de l'énergie, du carburant, des catalyseurs, des services sous-traités ou encore une main d'œuvre supplémentaire aux opérations des tâches. Ce concept a été étudié à partir des travaux pionniers de Vickson (1980a, 1980b). Au cours des trente dernières années, plusieurs recherches ont exploité cette catégorie de variables de décision pour améliorer les performances d'un système.

Une définition générale des problèmes d'ordonnancement avec des temps de traitement de tâches contrôlables peut être énoncée comme suit : Il s'agit de traiter n tâches indépendantes, notées $J = 1, 2, \dots, n$, sur m machines, représentées par $M \in M1, M2, \dots, Mm$, où O_{ij} désigne l'opération de la tâche j sur la machine i pour $i = 1, \dots, m$, et $j = 1, \dots, n$. Les machines sont disposées dans une configuration technologique spécifique, pouvant être une seule machine ($m = 1$), des machines en parallèle, des machines dans un atelier séquentiel, un atelier de fabrication, ou un atelier ouvert. Le temps de libération r_j de la tâche j correspond à son arrivée dans le système et à sa disponibilité pour le traitement. Le temps de traitement p_{ij} de la tâche j sur la machine i est une fonction décroissante de la quantité u_{ij} de la ressource allouée à l'opération O_{ij} . Dans le cas d'une seule machine, l'indice de la machine est omis, de sorte que p_j représente le temps de traitement de la tâche j sur cette unique machine. La ressource peut être utilisée soit de manière continue, soit de manière discrète. Dans le premier cas, le temps de traitement d'une tâche dépend de la quantité d'une ressource divisible (comme le gaz et l'électricité) qui lui est allouée, et peut donc varier de manière continue. En revanche, un type de ressource discrète est indivisible (par exemple, la main-d'œuvre et l'équipement de support), ce qui signifie que le temps de traitement d'une tâche ne peut prendre qu'un nombre fini de valeurs.

Plusieurs travaux ont intégré cette variante dans la résolution des problèmes d'ordonnancement, dont Giglio (2015), C.-H. Liu et Tsai (2016), S. Jiang et al. (2015), C. Lu, Li, et al. (2017), l'état de l'art le plus récent sur les problèmes d'ordonnancement avec des temps de traitement contrôlables a été présentée par Shabtay et Steiner (2007).

1.6 Temps de configuration dans les problèmes d'ordonnancement

Dans les systèmes de fabrication réels, il est courant de traiter un grand nombre de variantes de produits dans un même flux, ce qui nécessite des ajustements à chaque changement. Selon la définition de Shingo (2019), une opération de réglage est une préparation

ou un ajustement effectué avant et après le traitement de chaque lot. Le temps nécessaire pour passer d'un produit à un autre sur une machine donnée est identifié dans la théorie de l'ordonnancement comme un temps de préparation ou un coût de production supplémentaire. Les problèmes d'ordonnancement prenant en compte les temps de préparation sont considérés comme étant NP-difficile, comme l'a démontré M. L. Pinedo (2012) pour le cas d'une seule machine avec prise en compte des temps de réglage. Malgré cela, les planificateurs industriels et les chercheurs portent une grande attention à ce sujet en raison de son impact significatif sur l'efficacité de la production.

Le temps de préparation est un paramètre vital utilisé dans toute industrie manufacturière pour chaque machine ou poste de travail. Il est crucial d'aborder explicitement les temps de préparation dans la plupart des applications, car ils jouent un rôle essentiel dans la planification de la production dans de nombreux cas. Les activités de réglage peuvent varier de quelques millisecondes à plusieurs heures et peuvent facilement consommer plus de 20% de la capacité disponible de la machine si elles ne sont pas correctement gérées. Dans une production discrète, les activités typiques de réglage ou de mise en place d'une machine de traitement sont essentielles. Cette section examine la structure et les types d'activités de préparation pour une meilleure compréhension de ce concept.

1.6.1 Structure de base des temps de configuration

Le processus opérationnel d'une machine se décompose en trois phases distinctes : l'installation, le traitement et le retrait des jobs. Dans le cadre de la production discrète, les activités associées à la configuration et au retrait d'une machine comprennent principalement les ajustements de la machine pour le traitement du prochain job, le démontage après le traitement du job précédent, le réglage de la machine, l'inspection des outils et des accessoires, ainsi que le nettoyage des machines et des zones environnantes, et ainsi de suite. Étant donné qu'une machine ne devient libre que lorsque les réglages de tous les jobs ont été effectués, les temps de retrait sont soit nuls, soit positifs. Par conséquent, ils peuvent également être inclus dans le calcul du délai de fabrication. Les tâches de préparation peuvent varier en fonction du type de job, du type de machine, ou des deux. Dans tous les cas, le laps de temps entre la fin de la production d'une série et le début de la production de la série suivante, répondant à toutes les exigences de qualité, est souvent perçu comme une perte de temps ou un coût additionnel.

Pour réduire le temps d'arrêt total dû aux ajustements, deux approches sont envisageables :

- Diminuer la fréquence des ajustements ;
- Réduire la durée nécessaire à leur réalisation.

La réduction de la fréquence des ajustements est moins avantageuse que celle de leur du-

rée. Les principaux bénéfices associés à cette réduction incluent des délais plus courts, une productivité accrue, une capacité augmentée, une plus grande flexibilité et une diminution des défauts. Dans la littérature, la période entre la fin de la production précédente sur une machine et le début de la production suivante est dénommée temps de changement. Bien que les termes "préparation" et "changement" soient parfois utilisés de façon interchangeable, cette pratique est cependant incorrecte.

Un changement peut être divisé en trois "phases" :

1. La phase de nettoyage : revient à nettoyer l'espace, les produits, les matériaux ou les composants de la machine, de la ligne (poussière, débris, etc.) ;
2. La phase de Réglage : couvre le processus d'ajustement de l'équipement ;
3. La phase de démarrage/de lancement : comprend le temps passé pour la mise au point de l'équipement après son redémarrage. Il est généralement causé par la variabilité du nettoyage et du réglage, ainsi que par la diversité du produit ou de ses composants.

Au sens strict, un réglage n'est qu'une composante d'un processus de changement. Dans certains cas, les opérations de réglage peuvent affecter la configuration des lignes de production ou des cellules de fabrication dans l'usine. Pour plus de détails concernant ce point de vue, le lecteur peut se référer aux travaux réalisés par Andrés et al. (2005).

1.6.2 Classification des temps de configuration

La gestion des temps de préparation en ordonnancement est conditionnée par les hypothèses spécifiques du problème. En termes de traitement, les temps de configuration peuvent être considérés comme séparables ou non séparables. La plupart des études portant sur l'ordonnancement de la production supposent que les temps de réglage sont constants pour toutes les tâches et sont donc intégrés aux temps de traitement, ou bien qu'ils sont négligeables et donc omis. L'hypothèse de la non-séparabilité des temps de préparation simplifie l'analyse, permettant ainsi de formuler et résoudre ces problèmes sous forme d'instructions d'ordonnancement standard. Une opération est considérée comme séparable du traitement si elle ne fait pas partie intégrante de celui-ci. Lorsque les temps de préparation et de retrait séparables ne peuvent être négligés dans un problème d'ordonnancement, ils doivent être explicitement pris en compte. Dans certains cas, les opérations de préparation et de retrait non séparables, telles que le chargement et le déchargement, qui sont considérées comme faisant partie du traitement, doivent être explicitement prises en compte. Ces situations surviennent principalement dans les systèmes de production automatisée, impliquant des dispositifs de manutention intermédiaires comme les véhicules à guidage automatique (AGV) et les robots. La coordination étroite entre ces opérations

de préparation non séparables et les opérations de traitement est essentielle. Les temps de préparation séparables des tâches ou des lots, pris en compte dans les problèmes d'ordonnancement, peuvent être classés comme suit :

- Temps de configuration indépendants de la séquence (SI)
- Temps de configuration dépendants de la séquence (SD)
- Temps de configuration dépendants de la machine/des ressources (MD)
- Temps de configuration dépendants du temps (TD).

La dissociation de ces différents temps de configuration dépend du système de production considéré et de ses caractéristiques.

1.6.2.1 Temps de configuration indépendants de la séquence (SI)

Les temps de réglage séparables peuvent être soit indépendants, soit dépendants de la séquence des tâches sur la machine. Lorsque les activités de préparation d'une tâche sur n'importe quelle machine de l'atelier ne sont pas influencées par d'autres tâches dans la séquence à traiter, ce temps de préparation est défini comme étant indépendant de la séquence. Dans ce cas, la durée de préparation (coût) est uniquement fonction de la tâche ou du lot à traiter.

Les temps de préparation indépendants de la séquence, peuvent être classés en deux catégories : anticipés (détachés) ou non anticipés (attachés). La distinction réside dans la manière dont le temps d'inactivité de la machine est traité. Un réglage est considéré comme anticipé s'il peut être effectué avant que le job ou le lot correspondant ne soit disponible sur la machine. Par défaut, sauf indication contraire, les réglages sont supposés anticipés, ce qui signifie qu'un réglage sur une machine ne nécessite pas la présence d'un job (Potts & Kovalyov, 2000).

Les ouvrages qui abordent explicitement les temps de réglage indépendants de la séquence et leurs variations se concentrent principalement sur les environnements à une seule machine. Cependant, il existe des variations dans l'interprétation des temps de préparation. Nous citons notamment les travaux faits par Mor et Mosheiov (2014), Pei et al. (2016), Song et al. (2016), Kong et al. (2020), par ailleurs davantage de travaux abordant cette thématique seront cités dans la section 2.3.1 du chapitre suivant.

1.6.2.2 Temps de configuration dépendants de la séquence (SD)

Dans le cas où la durée (ou le coût) de la préparation dépend à la fois du job en cours et de ceux qui le précèdent immédiatement sur une machine spécifique, c'est-à-dire qu'elle est influencée par la séquence des jobs, cette configuration est qualifiée de dépendante de la séquence. Les temps de préparation dépendants de la séquence peuvent être illustrés

par l'exemple suivant : imaginons qu'une partie des ajustements de préparation nécessaires pour la réalisation d'un job k puisse également être utilisée pour traiter le job j suivant. Dans ce scénario, une fois le job k terminé, ces ajustements de préparation ne sont pas retirés, mais restent en place pour faciliter le traitement du job j . Ainsi, le temps requis pour retirer le job k dépendra des activités de préparation du job j , tandis que les préparations du job j dépendront du job k . Pour cette raison, les temps de préparation dépendants de la séquence sont de nature anticipative et indissociable. En d'autres termes, les informations et les détails de la préparation ne peuvent pas être inclus dans le temps de traitement et ne peuvent pas être déplacés avec le job.

Les temps de réglage dépendants de la séquence sont toujours de type anticipatif. Dans ce cas, le temps d'inactivité d'une machine peut être utilisé pour achever la préparation de la tâche. Les temps de préparation anticipés sont distincts des temps de traitement (détachés). Dans cette configuration, le temps de préparation et le temps de traitement sont associés à chaque tâche. Les informations et les détails relatifs à la préparation ne sont pas transférés avec le job si les temps de préparation sont anticipés.

En revanche, un réglage est considéré comme non anticipé lorsque les opérations de réglage ne peuvent commencer qu'à l'arrivée du job sur une machine, car le réglage est associé au job. Dans ce cas, le temps d'inactivité d'une machine ne peut pas être utilisé, et le temps de préparation est donc considéré comme faisant partie du temps de traitement. Par conséquent, le problème est formulé et résolu comme un problème d'ordonnancement standard sans tenir compte du temps de préparation. Sauf indication contraire explicite, les réglages sont considérés comme non anticipés.

Si le type de configuration n'est pas explicitement indiqué, il est considéré comme indépendant de la séquence. La figure 1.5 présente une représentation graphique des modèles de temps de préparation décrits.

La prise en compte explicite des temps de préparation dépendant de la séquence produit une large classe de problèmes de production réels. Une brève revue de la littérature de cette catégorie de problèmes est présentée dans la section 2.3.2 du chapitre 2.

1.6.2.3 Temps de configuration dépendants de la machine/des ressources (MD)

Les temps de préparation dans ce modèle dépendent de la ressource utilisée lors de cette phase. Cette thématique a été abordée initialement par T. E. Cheng et al. (2001) ainsi que par W.-J. Chen (2008). Les premiers auteurs ont examiné une variante du problème d'ordonnancement par lots sur une seule machine, où les temps de préparation et de traitement des tâches dépendaient d'un processus continuellement divisible. Cette variante du problème trouve son origine dans la fabrication de pièces sur un centre d'usinage polyvalent, où les pièces sont montées sur des palettes, traitées puis retirées de celles-ci.

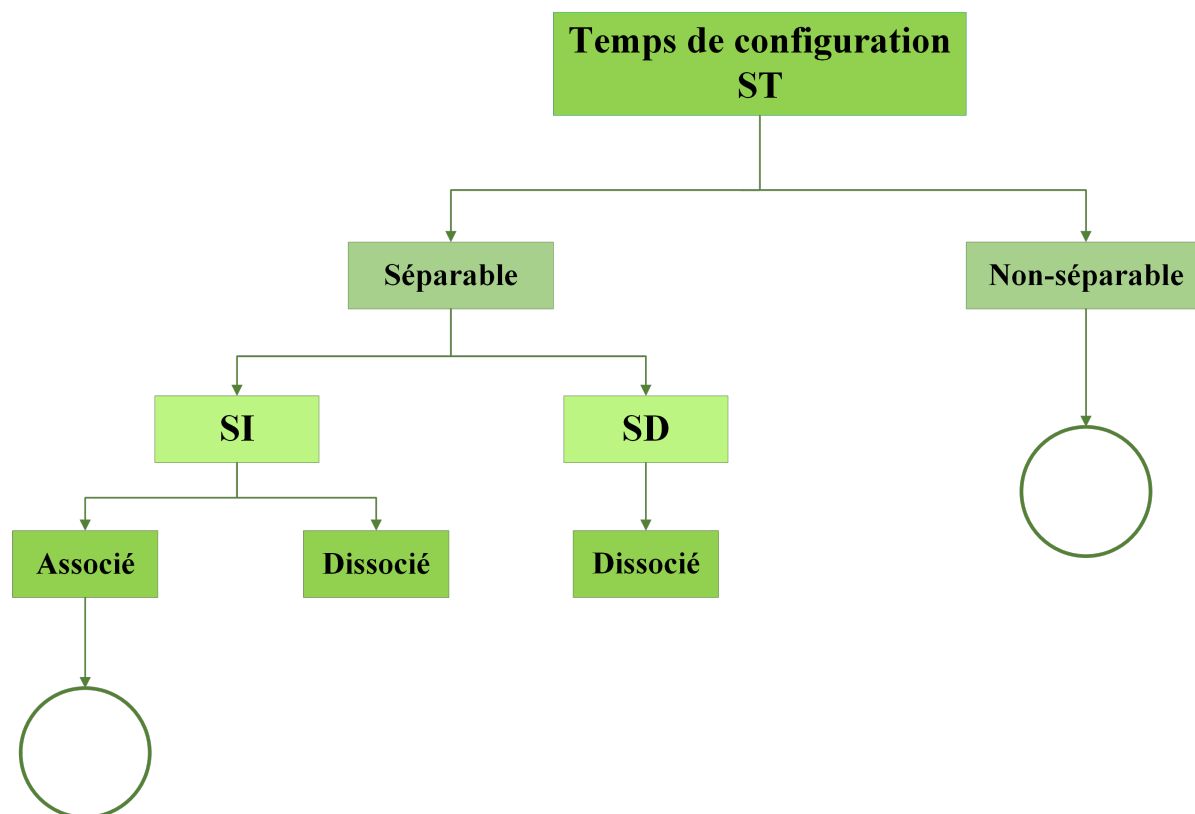


Figure 1.5: Corrélation entre les modèles de temps de configuration (ST). Le cercle représente un modèle d'ordonnement standard

Le retrait et la mise en place des palettes sont effectués par un robot, et le temps nécessaire pour ces opérations dépend de la productivité du robot et de la consommation d'énergie. W.-J. Chen (2008) a également pris en compte les événements perturbateurs, les définissant comme des "temps de repos des machines". Ces événements peuvent inclure les week-ends, pendant lesquels une machine exploitée par des travailleurs doit être mise hors service, impactant ainsi le temps de préparation. Pour résoudre le problème de la minimisation du retard maximal tout en respectant les contraintes de temps de préparation et de repos, un algorithme heuristique efficace a été développé. Cette variante a été étudiée par plusieurs chercheurs, notamment Chyu et Chang (2010), Koh et al. (2013), Avalos-Rosales et al. (2015), ainsi que Y.-K. Lin et Hsieh (2014).

1.6.2.4 Temps de configuration dépendants du temps (TD)

Le concept de temps de préparation dépendant du temps, également connu sous le nom de temps de préparation dépendant de la séquence passée (p-s-d), a été introduit presque simultanément par Koulamas et Kyparisis (2007) et Kuo et Yang (2006) dans l'étude des problèmes d'ordonnement. Cela implique que le temps de préparation est proportionnel à la longueur des jobs déjà planifiés. Ce type de configuration est générale-

ment présent dans les énoncés de problèmes avec des hypothèses spécifiques telles que les effets d'apprentissage/détérioration. Ces phénomènes peuvent influencer non seulement les temps de traitement des jobs, mais aussi les durées des processus de préparation.

Koulamas et Kyparisis (2007) ont motivé l'introduction de cette nouvelle forme de temps de réglage en soulignant certaines situations dans la fabrication de haute technologie, où un lot de jobs se compose d'un groupe de composants électroniques montés ensemble sur une carte de circuit imprimé (CI). Ces jobs doivent être traités individuellement par une machine alors qu'ils sont montés ensemble sur la carte. Le fonctionnement de la machine sur l'un de ces composants a un impact négatif sur la "préparation" de tous les autres composants qui n'ont pas encore été traités, en raison du flux de courant électrique à travers la carte de circuit imprimé pendant que la machine fonctionne.

Une fois qu'un composant est entièrement traité, son état n'est plus affecté par le fonctionnement ultérieur de la machine, même s'il reste monté sur la carte CI. Le degré d'indisponibilité d'un composant est proportionnel à la durée pendant laquelle il a été exposé au fonctionnement de la machine sur d'autres composants. Par conséquent, avant le traitement d'un composant, une opération de préparation est nécessaire pour le ramener à l'état de "disponibilité totale". Cette opération de réglage est proportionnelle au degré d'impréparation du composant concerné et n'a aucun effet sur l'état de préparation des autres composants non traités. Lorsque tous les composants de la carte CI sont gérés par la machine, le processus de fabrication global est terminé.

Ces auteurs ont également étendu ce concept à un environnement d'apprentissage, dans lequel les temps de préparation p-s-d ne sont plus des fonctions linéaires du temps de traitement déjà écoulé en raison des effets de l'apprentissage. Wu et al. (2008) ont introduit pour la première fois la détérioration du temps de préparation dans l'ordonnancement de groupe, où les temps de préparation des groupes sont généralement supposés connus et fixes. En réalité, le temps nécessaire à la mise en place ou à la préparation peut varier, notamment lorsque la qualité des éléments à préparer se détériore ou que l'état d'un patient s'aggrave. Dans ce contexte, si une tâche doit attendre d'être traitée, tant les temps de préparation que les temps de traitement de cette tâche sont prolongés.

1.7 Technologie de groupe

La technologie de groupe (GT) est fondée sur le concept de simplification et de standardisation, émergeant au début du 20^e siècle (Andrés et al., 2005). Ce processus consiste à regrouper des pièces ou à en séparer d'autres sur la base de critères prédéterminés. Deux catégories de critères sont identifiées : les critères de conception, englobant la forme, la taille, la texture de surface, le type de matériaux et la nature des matières premières ; et les critères de fabrication, incluant les opérations, les séquences, la taille des lots, le

temps de traitement et la quantité de production. La classification selon les critères de conception permet aux ingénieurs de normaliser la conception de nouvelles pièces et d'estimer les coûts, tandis que la classification selon les critères de fabrication vise à améliorer le contrôle du processus, à réduire les temps de préparation et à normaliser les plans de processus.

Donc, lors de l'observation d'une usine produisant une variété de produits, on constate souvent des similitudes entre certains d'entre eux, tant au niveau de leur forme, de leur construction, que de leur processus de production et des ressources nécessaires. Par exemple, une entreprise fabricant N produits différents peut identifier des groupes de produits similaires, tels que le groupe x ou le groupe y. Pour optimiser l'efficacité, toutes les ressources requises par chaque groupe sont regroupées dans une zone spécifique, appelée la cellule des machines (CM). Ainsi, la production d'une famille de pièces (PF), composée de produits similaires, est facilitée, car toutes les ressources nécessaires sont concentrées à proximité. La GT explore les similitudes entre les composants des produits et, grâce à des méthodes de simplification et de normalisation, organise les activités de fabrication afin d'améliorer l'efficacité de la production. Cette approche comporte plusieurs aspects et peut être représenté par la figure 1.6

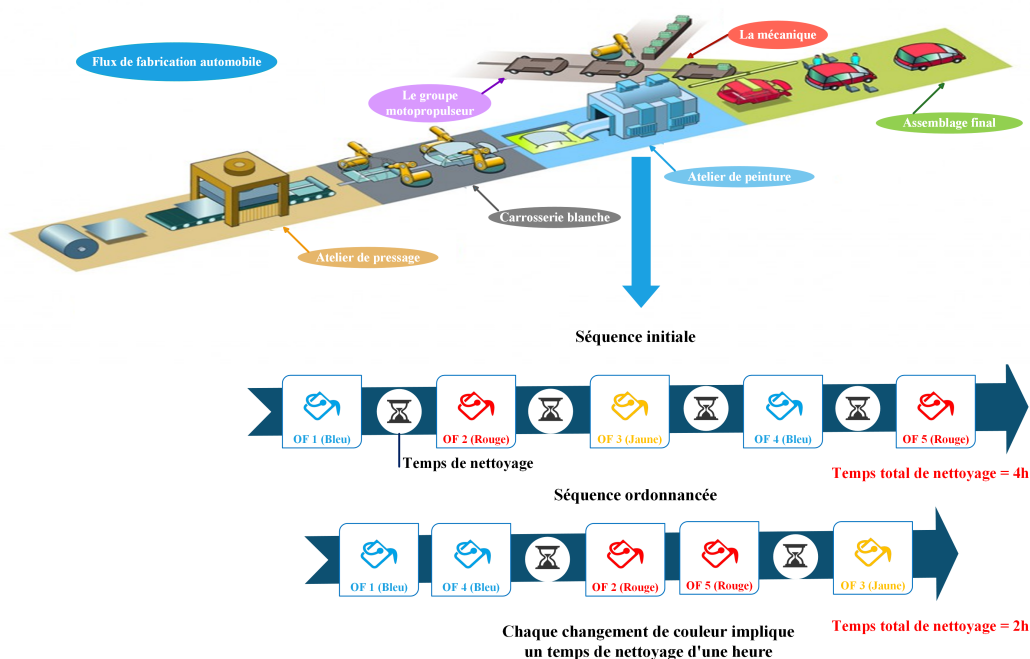


Figure 1.6: Représentation du concept de technologie de groupe dans un atelier de fabrication automobile

1.7.1 Définition

Dans les systèmes de production avancés, tels que ceux utilisés dans les industries des circuits intégrés, des composants électroniques et diverses chaînes d'assemblage, les produits fabriqués présentent souvent des caractéristiques de volume et d'assortiment élevés (HV/HM) de nomenclatures. Dans ces industries HV/HM, les composants du produit sont traités en groupes de plusieurs articles identiques, ce qui se traduit par la fabrication d'un grand nombre de composants de même taille et de même configuration.

L'équipement nécessaire à ce type de production exige un investissement en capital élevé et une grande flexibilité opérationnelle. Cependant, au cours des dernières années, les entreprises manufacturières ont orienté leurs démarches vers une production multi-produits et de petits lots afin de s'adapter aux fluctuations du marché. Ces mouvements sont caractérisés par des demandes diversifiées et spécialisées de la société ainsi que par des cycles de vie courts des produits.

La gestion d'une grande variété de produits à fabriquer avec des lots de petite taille entraîne d'énormes complexités dans le processus de production. Des problèmes fréquents peuvent survenir, tels que l'inefficacité de la fabrication, un mauvais flux de travail, un coût élevé des machines, un temps de préparation élevé, des stocks importants et des délais de livraison, entre autres.

Pour faire face à ces difficultés dans des environnements dynamiques, les méthodologies de gestion de la production classiques s'avèrent souvent insuffisantes pour maintenir une grande flexibilité de fabrication. Les approches GT (Group Technology) et Cellular Manufacturing (CM) utilisent des instruments internes pour améliorer le système de production, réduire les délais de livraison et cherchent à éliminer ou à minimiser la complexité de la planification tout en améliorant ou en maximisant la productivité.

Le terme "technologie de groupe" possède de nombreuses définitions. La définition la plus fréquemment citée est peut-être celle de l'un des premiers auteurs, le professeur Solaja et Ursoevic (1969), qui était le directeur de l'Institut des machines-outils de Belgrade (ex-Yougoslavie) :

“ La technologie de groupe consiste à reconnaître que de nombreux problèmes partagent des similitudes, et en les regroupant, il est possible de trouver une solution commune à un ensemble de ces problèmes, ce qui permet d'économiser du temps et des efforts. ”

Dans une première revue de la littérature consacrée à la CM, Greene et Sadowski (1984) ont défini la GT comme :

“ la réunion et l’organisation (regroupement) de concepts, de principes, de problèmes et de tâches (technologie) communs”, tandis que la CM est définie comme “la division physique des machines de fabrication de l’atelier fonctionnel en cellules de production, où chaque cellule est conçue pour produire une famille de pièces.”

L’un des auteurs les plus reconnus dans le domaine de la GT, Burbidge a proposé dans son dernier article publié en 1996 la définition suivante :

“ La technologie de groupe est une méthode d’organisation des usines dans laquelle les machines-outils, les autres installations de traitement et les personnes sont divisées en groupes. Chaque groupe complète toutes les pièces qu’il fabrique, au stade de transformation où il opère. Les machines de chaque groupe doivent être disposées ensemble en un seul endroit.”

Pendant de nombreuses années, la GT n’a pas reçu la reconnaissance officielle qu’elle méritait et n’a pas été rigoureusement pratiquée en tant qu’approche holistique de l’amélioration de la productivité. Pourtant, ses principes fondamentaux ont été appliqués à travers le monde depuis longtemps, bien que sous différentes appellations et formes, dans des publications antérieures, telles que les "bonnes pratiques d’ingénierie" et la "gestion scientifique". Dans ce qui suit, nous allons survoler brièvement l’évolution de ce concept à travers les différents travaux réalisés auparavant.

1.7.2 Évolution du concept de technologie de groupe

Le terme "technologie de groupe" a été introduit par (Mitrofanov, Harris, & Grayson, 1966), professeur à l’université d’État de Leningrad (URSS), dans le cadre de ses recherches sur les liens entre la configuration des composants et les méthodes de traitement. Cette nouvelle théorie a été formellement exposée dans son ouvrage intitulé "Les Principes Scientifiques de la Technologie de Groupe", d’abord publié en russe en 1946, puis traduit en anglais en 1966. Y. Zhang (2013) résume les contributions de Mitrofanov et al. comme suit :

- Il a découvert que des réductions considérables des temps de préparation, conduisant à une augmentation de la capacité, pouvaient être obtenues avec des tours. Dans ce cas, un groupe de pièces similaires est créé. Ensuite, ces pièces sont chargées sur la machine les unes après les autres avec le même montage.
- Il a également spécifié qu’une simplification de la conception et une normalisation du processus sont des prérequis impératifs pour un programme de Technologie de Groupe, et a suggéré un système de classification et de codage double : une partie pour la simplification de la conception et l’autre pour la normalisation du processus.

- Il a continuellement affirmé que la classification des opérations techniques basée sur la forme des composants, les surfaces et les caractéristiques offraient la meilleure solution à ce problème. Il a illustré le rôle de la classification comme le problème de base ou la solution sur laquelle repose la GT.

La GT, popularisée par Mitrofanov et al., a attiré une attention mondiale en tant que méthode visant à réduire les temps de préparation. Initialement conçue comme une solution pour les configurations à machine unique, elle a été étendue par Burbidge et a servi de base à la création du concept de CM. Cependant, l'ingénieur Gombinski a souligné le besoin urgent d'appliquer la GT dans le secteur industriel pour améliorer la productivité, car en l'appliquant, une augmentation substantielle de la productivité des industries mécaniques peut être obtenue, en particulier lorsque la production se fait sur une base unique ou par petits lots. Ces augmentations résultent d'une meilleure utilisation de la main-d'œuvre et des machines, avec une réduction correspondante des coûts. En accordant une telle importance à l'application de la GT, Gombinski a conclu que la résistance au changement pouvait être l'une des raisons de la lenteur de l'acceptation de la GT. Plus de 50 ans après la publication de cet article, on peut conclure que les principes et les méthodes de la GT sont encore insuffisamment mis en œuvre dans l'industrie moderne. Dans les années 1970, le concept de normalisation a été stabilisé dans la GT, tandis qu'une classification générale des attributs des composants a été établie, à savoir :

- Caractéristiques géométriques ou graphiques : taille, forme, etc.
- Fonctions des composants : manipulation, serrage, etc ;
- En fonction du type de fabrication : taille du groupe, déroulement du processus, etc ;
- Matière première de la pièce.

Malgré les progrès technologiques, certains processus de production présentent toujours des temps de préparation élevés atteignant jusqu'à 8 à 24 heures, dans certaines phases des processus de production, comme dans la fabrication de tuiles ou de semi-conducteurs (Delgado-Arana et al., 2017). Ceci justifie la poursuite du développement et de l'application de la GT en conjonction avec d'autres méthodologies telles que le Juste-à-Temps (JIT) pour réduire les délais d'exécution.

1.7.3 Applicabilité du concept de technologie de groupe

Les principes de la GT peuvent être appliqués à toute entité industrielle, depuis les pièces fabriquées et les biens d'équipement jusqu'aux processus de décision et aux caractéristiques humaines, dans le but de tirer parti des similitudes qui existent entre les éléments et d'accroître l'efficacité correspondante. Parallèlement, le problème du flow shop, qui reflète fidèlement la réalité industrielle, est l'un des défis les plus courants. Leur association

constitue donc un atout majeur pour l'organisation de la fabrication, représentant ainsi le point culminant de cette thèse.

L'exemple de la fabrication de cartes de circuits imprimés (PCB) sur des machines d'insertion automatisées HS-180 en est une illustration (Schaller, Gupta, & Vakharia, 2000). Afin d'optimiser le rendement, l'entreprise a regroupé les différents types de PCB en familles de sorte à réduire les temps morts lors du passage d'une famille à une autre. Cela signifie que les puces communes entre les différentes familles restent chargés sur la machine, tandis que celles spécifiques à chaque famille sont remplacés.

L'industrie de la fabrication de tuiles repose également sur le principe d'exploiter les similitudes entre les produits, inspiré de la philosophie de la technologie de groupe (Andrés et al., 2005). Cela vise à améliorer la flexibilité du système tout en réduisant les temps de fabrication. Dans ce secteur, la ligne de production comprend une presse où l'argile est déposée pour former les carreaux, suivie d'un convoyeur, composé de plusieurs machines, appelé ligne d'émaillage. Cette ligne ajoute d'autres éléments aux carreaux, tels que le smalt et différentes couleurs. Ces éléments déterminent l'aspect final des carreaux après cuisson dans le four. Une fois cuits, les produits sont triés et emballés à l'aide de machines automatisées.

Le changement de production d'un type de carreaux à un autre nécessite des ajustements sur les différentes machines. Par exemple, pour la presse, il est essentiel de changer la matrice en cas de modifications de taille ou de forme des carreaux. Sur la ligne d'émaillage, cela implique de modifier la largeur du convoyeur, de remplacer les matières premières nécessaires au nouveau modèle, voire d'installer de nouvelles machines ou de revoir leur agencement. En conclusion, une constitution intelligente des différentes familles de carreaux réduit considérablement les temps de préparation. De plus, un séquençage optimal de ces familles permet d'utiliser des machines à haut rendement qui n'étaient pas rentables auparavant en raison des longs temps de préparation.

La GT trouve également son application dans l'industrie automobile, notamment dans les ateliers de peinture. En effet, pour peindre un lot de voitures, des configurations préalables sont nécessaires, impliquant des activités de nettoyage de l'équipement et de la zone. Le temps de configuration dépend du lot précédemment traité et du lot en cours de traitement. Ainsi, il est plus judicieux de planifier la peinture des voitures vertes après celle des voitures blanches, plutôt que celle des voitures noires, car cela nécessite moins d'activités de nettoyage.

1.8 Conclusion

Dans ce chapitre, nous avons abordé les fondamentaux d'un problème d'ordonnement, en mettant en avant les notions essentielles liées à la technologie de groupe. Nous avons discuté des définitions de base et des termes spécifiques à ce domaine, tout en illustrant les différentes classifications et types de décisions qui s'y rapportent.

Par la suite, nous avons examiné les diverses structures des problèmes d'ordonnement. En parallèle, nous avons exposé les implications sur les ressources, telles que l'effet d'apprentissage et l'effet de détérioration. De plus, une analyse approfondie a été menée sur l'incorporation de temps de configuration de natures variées dans les problèmes d'ordonnement, ainsi que sur les différentes approches d'insertion de la technologie de groupe et leurs postulats dans le domaine de l'ordonnement.

Afin de positionner notre travail de manière plus contextuelle, le chapitre suivant se consacrera à un état de l'art sur les problèmes d'ordonnement flow shop impliquant la technologie de groupe, tout en tenant compte de l'effet d'apprentissage et des temps de configuration, en particulier ceux dépendant de la séquence.

Chapitre 2

État de l'art

2.1 Introduction

Les premières recherches sur l'ordonnancement ont débuté dans les années 50. Depuis lors, les problèmes sont devenus de plus en plus complexes en raison des nombreuses contraintes pratiques à prendre en compte. En effet, il existe de nombreuses variantes de problèmes d'ordonnancement parmi la vaste littérature de ce domaine. Comme mentionné dans (Graham et al., 1979) ; (Carlier & Chrétienne, 1988) ; (Gotha, 1993) ; (Lenstra, Shmoys, & Tardos, 1990) ; (Carlier et al., 1993) ; (Esquirol et al., s. d.) ; (M. Pinedo & Chao, 1999) ; (Allahverdi, Gupta, & Aldowaisan, 1999), (Baptiste, 2005) ; (Artiba, Dhaevers, Duvivier, & Elmaghraby, 2011) ; (Bassem, Patrick, & Jacques, 2013), (Sule, 2007) ; (M. L. Pinedo & Pinedo, 2016) ; (Allahverdi, 2015). Ces problèmes sont présents dans tous les secteurs de l'économie : industrie, hôpitaux, administrations, informatique, etc. L'ordonnancement est une fonction cruciale, car la qualité des plannings générés affecte directement la production (en termes d'efficacité, de compétitivité, de flexibilité, de respect des délais, de satisfaction client, etc).

Ce chapitre est consacré, dans un premier temps, à un aperçu de la littérature sur les problèmes classiques d'ordonnancement de flow shop. Ensuite, nous poursuivons avec un état de l'art détaillé sur les problèmes flow shop intégrant la technologie de groupe et des temps de configuration dépendant de la séquence (FSDGS). Nous présentons également les résultats des travaux antérieurs sur les problèmes d'ordonnancement avec effet d'apprentissage, suivis des travaux intégrant les contraintes de transport. Enfin, nous passons en revue les différentes approches de résolution du problème d'ordonnancement multiobjectif de l'atelier de production.

2.2 Aperçu de la littérature sur les problèmes d'ordonnement d'atelier flow shop

L'ordonnement concerne l'allocation de ressources sur des périodes de temps données dans le but de minimiser un ou plusieurs objectifs. En tant que processus de prise de décision, l'ordonnement joue un rôle crucial dans les systèmes de fabrication, de production, les environnements de transport, de distribution, et même certains types de services (M. Pinedo, 2005). Les chercheurs ont apporté une contribution significative à ce domaine, couvrant des problèmes classiques de la littérature jusqu'aux applications pratiques (Gupta & Stafford Jr, 2006), (Allahverdi, Ng, Cheng, & Kovalyov, 2008), (Allahverdi, 2015), (W. Qin, Zhuang, Liu, & Tang, 2019), (Fu, Wang, Wang, & Pu, 2020).

Il existe différentes variantes des problèmes de flow shop, telles que le problème de permutation, de non-permutation, sans temps d'inactivité, sans attente, avec processus réentrant, avec ou sans pannes, avec ou sans blocage. De plus, d'autres problèmes dérivés de celui-ci existent, comme c'est le cas du problème de flow shop hybride. Ce dernier représente une combinaison entre les problèmes d'ordonnement à machines parallèles et les problèmes de flow shop. Il est également abordé par une grande communauté de chercheurs, dont (Tosun, Marichelvam, & Tosun, 2020), (Neufeld, Schulz, & Buscher, 2023), (Y. Li, Li, Gao, & Meng, 2020), (Fan, Zhai, Li, & Wang, 2018), (Peña Tibaduiza et al., 2017).

Le problème d'ordonnement flow shop à permutation est la variante la plus importante de ce type de problème, attirant une attention particulière depuis sa première publication par Johnson (1954). Cet environnement se caractérise par le traitement de n tâches sur m machines. Toutes les tâches suivent le même ordre, débutant sur la première machine et se déplaçant jusqu'à la dernière machine. La capacité de stockage intermédiaire entre les machines est considérée comme infinie, et les machines sont toujours disponibles pour traiter les tâches (M. L. Pinedo & Pinedo, 2016).

Une des suppositions courantes dans les problèmes d'ordonnement flow shop stipule que les files d'attente des tâches doivent être disponibles à tout moment sur n'importe laquelle des machines de la séquence de traitement. Donc, cette hypothèse s'apparente au fait que la zone de stockage est infinie et que les tâches peuvent attendre au niveau des machines, le temps que cette dernière se libère (Allahverdi et al., 1999). Par ailleurs, cette supposition n'est pas toujours vraie, car il existe des cas particuliers de problème dans lesquelles les tâches ne sont pas autorisées à former des files d'attente. Les problèmes flow shop avec zéro buffer et sans attente en sont quelques exemples. Dans ces situations, un

job j qui vient de se terminer sur la machine i ne peut pas passer à la machine $i + 1$ si cette dernière est encore en train de traiter son prédécesseur dans la séquence des jobs. Ainsi, le job j doit demeurer sur la machine i , temporairement, privant cette dernière de la possibilité de traiter son successeur dans la séquence des jobs, jusqu'à ce que le job j puisse passer à la machine suivante.

Abadi et al. (2000) ont été parmi les premiers auteurs à prendre en considération cet aspect dans les problèmes de flow shop, définissant ainsi les contraintes de blocage liées à cette restriction. Depuis lors, de nombreux travaux se sont penchés sur cet axe de recherche, des plus anciens aux plus récents. Parmi ces travaux figurent ceux réalisés par S. Q. Liu et Kozan (2009), qui ont pris en compte quatre conditions variables relatives aux buffers entre les machines (c'est-à-dire sans attente, sans buffer, buffer limité et illimité). Ces quatre conditions sont combinées pour générer un problème d'ordonnement plus généralisé et complet, capable de couvrir les problèmes classiques d'ordonnement flow shop (FSS) avec buffer illimité, le FSS avec blocage (BFSS) sans buffer, le FSS sans attente (NWFSS) et le FSS avec buffer limité (LBFSS).

En analysant les propriétés structurelles de ces quatre conditions de stockage, les auteurs proposent un algorithme constructif novateur appelé l'algorithme de Liu-Kozan pour construire la solution réalisable d'ordonnement d'atelier de flow shop à buffer combiné (CBFSS). L'algorithme comprend principalement une procédure de détermination du temps et une procédure de mise au point avec huit ensembles de formules. Des mises en œuvre numériques détaillées ont été analysées pour indiquer que l'algorithme de Liu-Kozan est très générique, car il permet de construire des solutions réalisables pour les problèmes FSS, BFSS, NWFSS, LBFSS et CBFSS sans modification.

En ce qui concerne les contributions récentes, Aqil et Allali (2021) ont pris en compte des contraintes de blocage dans un problème de flow shop hybride (HFS) avec des temps de configuration dépendant de la séquence. L'objectif est la minimisation des retards et des avances. Les auteurs ont développé six métaheuristiques basées sur des phénomènes naturels inspirés des oiseaux migrateurs et des mouvements des vagues. Trois versions de chaque algorithme ont été développées, constituant des approches d'amélioration pour la phase d'exploration du système de voisinage. Ces approches reposent sur l'algorithme de recherche itérée, la procédure de recherche adaptative randomisée et les procédures de recherche locale.

Parallèlement, dans les applications pratiques (Grabowski & Pempera, 2000), des blocages surviennent à certaines étapes de la production de blocs de béton en raison de contraintes technologiques. Dans l'industrie chimique, des produits chimiques partielle-

ment traités doivent parfois rester dans les machines en raison d'un manque de stockage intermédiaire (S. Q. Liu & Kozan, 2009). H. Chen et al. (2014) ont étudié un atelier de fabrication électronique qui assemble et teste des cartes de circuits imprimés (PCB) pour des produits électroniques grand public. Après l'assemblage, un lot de PCB est soumis à une séquence de tests dans deux chambres de contrôle des conditions environnementales (ESS) : la première chambre est utilisée pour soumettre le lot à un test de vibration, et la deuxième chambre est utilisée pour le tester à des températures extrêmes. Les exigences technologiques ne permettent pas à un lot d'attendre entre deux chambres ESS. Par conséquent, un lot doit être bloqué à l'intérieur de la première chambre ESS si la deuxième chambre ESS est occupée.

D'autres systèmes de production, tels que l'industrie chimique, pharmaceutique (Hall & Sriskandarajah, 1996), les lignes de production juste-à-temps (Prasad et al., 2006), et les cellules robotiques (Ribas et al., 2015), peuvent être modélisés comme des flow shop avec blocage. Une autre cause possible de blocage réside dans la technologie de production elle-même, par exemple, la température ou d'autres caractéristiques des matériaux, exigeant que la tâche terminée reste sur la machine pour éviter sa détérioration ou des coûts supplémentaires.

Dans le traitement des déchets industriels, différents types de déchets sont apportés par des camions et déchargés dans des réservoirs. Chaque produit doit être traité dans un mélangeur. Le produit s'écoule lentement du réservoir vers le mélangeur. Le réservoir est prêt à traiter un nouveau job uniquement lorsque le job précédent est entièrement transféré dans le mélangeur. Dans le traitement des pièces métalliques, chaque pièce doit d'abord être chauffée à la bonne température dans l'un des fours (première étape), afin d'être préparée pour le traitement sur une presse unique (deuxième étape). Le four est bloqué pour traiter le lot suivant de pièces jusqu'à ce que la dernière pièce du lot précédent soit terminée sur la presse (Martinez et al., 2006).

Gong et al. (2010) ont examiné un flow shop à deux machines dans l'industrie du fer et de l'acier, dans les opérations de trempage et de laminage, représentées par une machine de traitement par lots soumise à une contrainte de blocage et la deuxième machine est une machine discrète avec des temps de configuration partagés. Une fois terminé, l'ensemble des tâches doit rester à l'intérieur de la machine de lot pour éviter la détérioration du matériau, bloquant ainsi de nouvelles opérations à traiter. Dans la production de cidre, il est impossible de mélanger des pommes de différents clients. Tout d'abord, les pommes sont versées dans un bain, puis pressées pour obtenir du jus de pomme. Le blocage se produit, car les nouvelles pommes du client ne peuvent pas être versées dans le bain avant que toutes les pommes du premier client aient été pressées (Trabelsi et al., 2012). Le problème du flow shop avec blocage, avec deux machines et minimisation du makespan équivaut au problème du flow shop sans attente, avec deux machines et minimisation du

makespan (M. Pinedo & Chao, 1999).

Par ailleurs, Les récentes recherches sur le problème de flow shop hybride avec blocage se concentrent sur l'optimisation d'un ou plusieurs critères d'ordonnancement. Dans l'étude menée par Mollaei, Mohammadi, et Naderi (2019), ils ont introduit un modèle basé sur la programmation linéaire en nombre entière mixte (MILP) pour résoudre ce type de problème, visant à optimiser simultanément le makespan et le coût d'utilisation des ressources. De manière similaire, Meng et al. (2020) a développé à la fois un modèle MILP et un algorithme génétique (GA) pour aborder le problème HFS avec blocage sous la contrainte de temps de configuration dépendant de la séquence (SDST), avec un objectif principal de minimiser le makespan.

Une autre approche a été adoptée par Moccellin et al. (2018), qui ont présenté un ensemble d'heuristiques visant à résoudre le même problème, prenant en compte diverses dispositions des temps de configuration. Récemment, C.-C. Lin et al. (2020) ont proposé une recherche hybride harmonique et un algorithme génétique (AG) pour résoudre le problème HFS réentrant avec une capacité des buffers et des stokers limités en minimisant le makespan et le temps de flux moyen. Pour plus de détails concernant le problème de flow shop avec blocage et sans attente, les lecteurs peuvent se référer à l'état de l'art proposé par Miyata et Nagano (2019).

Comme évoqué précédemment, le flow shop traditionnel capte l'essence du traitement séquentiel. Cependant, lors du traitement des jobs, des événements inattendus peuvent survenir, généralement liés aux pannes des machines. Prendre en considération cette variante est donc une composante essentielle de la résolution des problèmes d'ordonnancement flow shop, étroitement liés au contexte industriel réel des systèmes de production.

Parmi les travaux qui ont intégré cette perspective, citons ceux réalisés par Y. Han et al. (2016), qui ont abordé un problème flow shop avec blocage et pannes de machines pour une optimisation multiobjectif incluant des critères de robustesse et de stabilité. La résolution de ce problème se fait par le biais d'un algorithme évolutif multiobjectif d'ordonnancement robuste. Dans ce cadre, des solutions obtenues par une variante de l'heuristique mono-objectif NEH proposée par (Nawaz, Ensore Jr, & Ham, 1983) sont incorporées dans l'initialisation de la population. De plus, deux nouveaux opérateurs de croisement sont proposés pour tirer parti des solutions non dominées. Enfin, une stratégie de replanification basée sur la recherche locale est présentée pour réduire davantage l'influence négative résultant des pannes de machines.

Ce type de problème a également été récemment traité par Hnaien et Arbaoui (2023), qui ont examiné la panne de la première machine dans un problème d'ordonnancement flow shop à deux machines. Une caractéristique particulière de leur approche est que le temps

de démarrage de la panne est de nature stochastique, avec une distribution de probabilité connue. L'objectif principal est de trouver une solution minimisant le makespan tout en respectant un niveau de service défini. Pour résoudre ce problème, deux modèles linéaires en nombres entiers mixtes ont été développés, accompagnés de bornes inférieures, et une inégalité valide a été introduite pour renforcer les deux modèles, améliorant ainsi leurs performances. De plus, cette étude présente deux heuristiques exploitant la modélisation par intervalles.

Par ailleurs, l'indisponibilité des machines dans le contexte concret de l'ordonnancement peut être causée par différentes circonstances, notamment des actions de maintenance et des pannes imprévues. En relation avec ces variations, les modèles de traitement des tâches sont divisés en deux catégories principales : 'résumable' et 'non résumable'. Dans le cas du modèle résumable, lorsque la période de maintenance interrompt le traitement d'une tâche, les opérations peuvent se poursuivre sans pénalité jusqu'à ce que la maintenance soit terminée. Cependant, pour le cas non résumable, une pénalité est appliquée, et l'ensemble des opérations de traitement sur la machine indisponible doit être repris depuis le début (Lee, 1999).

Parmi les travaux qui ont intégré les actions de maintenance, on trouve ceux élaborés par Safari et Sadjadi (2011), qui ont intégré des actions de maintenance programmées en fonction de l'état de chaque machine, contrôlé à intervalles de temps relativement espacés. Si la variable indiquant l'état est supérieure à un seuil de maintenance préventive, une action de maintenance préventive instantanée est exécutée, et les variables d'état reprennent leur valeur initiale. Les auteurs proposent un algorithme hybride combinant l'algorithme génétique et le recuit simulé pour la minimisation du makespan dans un problème flow shop non résumable, où le traitement des tâches après une maintenance préventive est repris depuis le début. Les résultats préliminaires indiquent que la méthode proposée offre des résultats significativement meilleurs par rapport à d'autres algorithmes performants dans la littérature.

L'optimisation conjointe de l'ordonnancement de la production et de la planification de la maintenance exerce une influence significative sur la stabilité de la production et la fiabilité des machines. Dans cette perspective, des études ont été menées par Z. Zhang et al. (2021) pour l'optimisation bi-objectif du problème de flow shop de permutation, en prenant en compte des actions de maintenance préventive et corrective dans le but de minimiser le makespan et les coûts de maintenance. Deux lemmes sont déduits pour relaxer le nombre attendu de pannes et les coûts de maintenance, rendant ainsi le modèle linéaire. Un algorithme RIPG (Restarted Iterated Pareto Greedy) est appliqué pour résoudre le problème et plusieurs expérimentations ont été réalisées pour démontrer que le RIPG

proposée surpasse quatre métaheuristiques bien connues, notamment MOPSO, MOSA, NSGA-II et NSGA-III.

Les recherches portant sur le problème de flow shop peuvent être examinées sous différents angles, en tenant compte de diverses fonctions objectives et techniques d'optimisation. La majorité de ces recherches se focalise principalement sur le makespan, également appelé temps de flux maximal, défini comme le moment d'achèvement où toutes les tâches ont accompli leur traitement. Cette fonction objective a été optimisée par divers chercheurs dont (Shao, Pi, & Shao, 2017), (Khatami, Salehipour, & Hwang, 2019), (Umam, Mustafid, & Suryono, 2022); (Allali, Aqil, & Belabid, 2022), (D. N. Sekkal & Belkaid, 2022); (Fasihi, Tavakkoli-Moghaddam, & Jolai, 2023). Pour la minimisation du makespan, Figielska (2018) a proposé une heuristique basée sur la règle de priorité pour résoudre les problèmes de flow shop flexible à deux étages avec des contraintes de ressources partagées renouvelables. Les contraintes résultant de cette définition sont incluses dans le problème d'optimisation résolu par un algorithme de génération de colonnes (CG). L'algorithme CG génère un ordonnancement composé de séquençement partiel assignant les tâches aux machines pour un traitement simultané dans le premier et le deuxième stage durant un laps de temps donné, afin que les contraintes de ressources soient satisfaites à tout moment. Deux modèles mathématiques linéaires à nombres entiers mixtes ont été proposés par Y. Zhang, Lu, et Zhang (2018) pour le dimensionnement des lots dans les problèmes de flow shop avec reconfiguration des ressources qui sont partagées entre les différentes étapes, ce qui diffère des problèmes d'ordonnancement classiques dans lesquels les machines sont configurées à l'avance.

Étant donné que les problèmes d'ordonnancement flow shop à n tâches et m machines appartiennent à la classe des problèmes NP difficiles (M. Pinedo & Chao, 1999), (Framinan, Gupta, & Leisten, 2004), les exigences de calcul pour obtenir une solution optimale augmentent de façon exponentielle à mesure que la taille du problème augmente, c'est-à-dire que, compte tenu de la complexité combinatoire et des contraintes de temps, la plupart des grandes instances des problèmes ne peuvent être résolues que par des métaheuristiques et heuristiques, (Jung, Woo, & Kim, 2017); (Wu et al., 2018); (Kheirandish, Tavakkoli-Moghaddam, & Karimi-Nasab, 2015); (Nikzad, Rezaeian, Mahdavi, & Rastgar, 2015); (Umam et al., 2022); (Sharma, Sharma, & Sharma, 2022). Cependant, dans de nombreuses situations pratiques, les problèmes d'ordonnancement impliquent généralement de multiples objectifs (Panwalker, Dudek, & Smith, 1973), (Ulungu, Teghem, Fortemps, & Tuyttens, 1999). En raison de ces situations, les problèmes d'ordonnancement multicritères font l'objet d'une grande attention ces dernières années (Ding, Song, & Wu, 2016); (Neufeld et al., 2023), (W. Han, Deng, Gong, Zhang, & Luo, 2021), (P. Li, Xue, Zhang, Chen, & Zhou, 2023).

Le tableau 2.1 synthétise les études précédemment exposées sur les problèmes d'ordonnement flow shop. Chaque référence y est répertoriée avec les informations suivantes : le type d'atelier, les contraintes, le critère d'optimisation, et l'approche de résolution.

Tableau 2.1: Synthèse des articles qui ont considéré un ordonnancement à machines en série

La référence	Le problème	L'approche de résolution
(Abadi et al., 2000)	$Fm block CC$	HDP
(Grabowski & Pempera, 2000)	$Fm prmu, no - wait, block Cmax$	TS
(Prasad et al., 2006)	$Fm b_{j,j+1} \sum_i C_{icont} + \sum_i C_{iparts} + \sum_i \sigma_i$	GA
(Martinez et al., 2006)	$Fm block, RCb Cmax$	CSA
(S. Q. Liu & Kozan, 2009)	$Fm b_{j,j+1}, prmu, no - wait, block Cmax$	l'algorithme de Liu-Kozan
(Gong et al., 2010)	$F2 no - wait, block, batch, shared_{st} Cmax$	HDP
(Safari & Sadjadi, 2011)	$Fm brkdown Cmax$	HGA
(Trabelsi et al., 2012)	$Fm prmu, block, Rcb^*, Rcb Cmax$	GA
(H. Chen et al., 2014)	$F2 block, batch, r_{ij} Cmax$	HDDE
(Ribas et al., 2015)	$Fm block \sum_i C_i$	DABC
(Y. Han et al., 2016)	$Fm brkdown, lot - str, block E(Cmax) + E[var(Tmax)]$	MOEA
(Moccellin et al., 2018)	$HFS block, ST_{sd}, ST_{si} Cmax$	Heuristique constructive (règle SPT et LPT)
(Figielska, 2018)	$FH2 res...1, prem Cmax$	CG
(Y. Zhang et al., 2018)	$FH res...1, lot - str, ST_{si} TPC$	MILP
(Mollaie et al., 2019)	$HFS block, ST_{sd} Cmax, TMC$	MILP, RPP
(Meng et al., 2020)	$HFS ST_{sd}, block Cmax$	MILP, GA
(C.-C. Lin et al., 2020)	$HFS block, recrc w_1 Cmax + w_2 f$	HHSGA
(Aqil & Allali, 2021)	$HFS r_j, d_j, ST_{sd}, block \sum_j (E_j + T_j)$	MBO, WWO
(Z. Zhang et al., 2021)	$DPm \rightarrow Fm PmetCm Cmax, TMC$	RIPG, MILP
(Hnaïen & Arbaoui, 2023)	$F2 brkdown Cmax$	MILP, IJO, TPIJO

Références des abréviations : HDP - heuristique de programmation dynamique ; TS - la recherche tabou ; GA - algorithme génétique ; CSA - algorithme de recherche par regroupement ; HGA - algorithme génétique hybride ; HDDE - algorithme hybride d'évolution différentielle discrète ; DABC - algorithme de colonie d'abeilles artificielles ; MOEA - algorithme évolutionnaire multi-objectif ; CG - algorithme de génération de colonnes ; RPP - la programmation possibiliste robuste ; HHSGA - la recherche hybride d'harmonie et l'algorithme génétique ; MBO - algorithme d'optimisation pour les oiseaux migrateurs ; WWO - algorithme d'optimisation des vagues d'eau ; RIPG - algorithme glouton itéré de Pareto avec redémarrage ; IJO- heuristique d'ordonnement Johnson par intervalles ; TPIJO- ordonnancement Johnson par intervalles à deux partition.

2.3 Aperçu de la littérature sur les problèmes de flow shop avec technologie de groupe

L'assortiment des tâches en groupe présente des avantages significatifs dans la résolution des problèmes de flow shop. Cette approche conduit à des temps de traitement réduits, une minimisation des temps de configuration, une utilisation optimale des machines, des exigences d'outillage réduites, des stocks d'encours plus modestes, un flux de pièces simplifié, et une intégration des nouvelles technologies simplifiée, comme c'est le cas lors de la fabrication intégrée par ordinateur. L'efficacité de cette méthode peut être

maximisée en exploitant des caractéristiques de conception telles que la taille ou la forme, ainsi que des caractéristiques de production comme les temps de traitement, les opérations et les séquences.

L'ordonnancement de groupe s'est avéré pertinent dans divers domaines au-delà des systèmes de fabrication cellulaire. Cette pertinence est démontrée par ses multiples applications pratiques, telles que les ateliers de peinture et de carrosserie automobile (Salmasi et al., 2010), la production de meubles (Wilson et al., 2004), la fabrication d'étiquettes autocollantes (H.-T. Lin & Liao, 2003), l'industrie des semi-conducteurs (Celano et al., 2010), la fabrication de pales de moteurs d'avion (S. Li, 1997), les poinçonneuses pour pièces métalliques (D. N. Sekkal & Belkaid, 2023), la fabrication de pompes centrifuges (Van Der Zee et al., 2011), la fabrication de cartes de circuits imprimés (PCB) (W. Qin et al., 2019), la production de TFT-LCD (Bozorgirad & Logendran, 2016), la fabrication électronique en général (Geloggulari & Logendran, 2010), ainsi que dans la construction d'ouvrage d'art tel que les ponts (D.-L. Yang et al., 2008).

Cependant, la plupart des travaux dans la littérature négligent souvent les temps de configuration lors du processus de fabrication, créant un écart entre la réalité et la pratique. En réalité, le processus de configuration est nécessaire dans un environnement de production réel pour que l'équipement effectue différentes opérations en raison des caractéristiques spécifiques de chaque tâche. Comme mentionné dans le chapitre 1, différents types de temps de configuration existent, parmi lesquels les temps de changement dépendants de la séquence et indépendants de la séquence sont les plus couramment discutés. Pour plus de détails concernant les problèmes d'ordonnancement avec des temps de configuration, une revue approfondie de la littérature est proposé par Allahverdi (2015).

Dans notre étude, le temps de configuration dépend de la séquence, ce qui signifie qu'il est déterminé par les opérations successives sur la même machine. Par exemple, le changement de fabrication d'une bielle utilisée dans l'essieu arrière d'une voiture diffère de celui d'une bielle dans le rotor d'un hélicoptère. Ainsi, l'application de temps de configuration au niveau des machines devient nécessaire. Dans ce contexte, regrouper les tâches en familles (groupes) est une approche judicieuse, où les tâches d'un groupe nécessitent la même configuration sur les machines.

2.3.1 Temps de configuration indépendant de la séquence

Les premières études sur l'ordonnancement du flow shop avec groupes considéraient initialement que le temps de configuration était indépendant de la séquence. Parmi ces travaux, Hitomi et Ham (1976) ont été les pionniers dans la prise en compte de ce problème. Ils ont présenté une version modifiée de l'algorithme Petrov PT. Par la suite, Ham et al. (1985) ont publié un ouvrage exposant les principes fondamentaux de la technologie des groupes. Ils ont également introduit une nouvelle borne inférieure, plusieurs théorèmes et l'algorithme de branch and bound pour résoudre le problème, en prenant en considération un temps de traitement variable en fonction des conditions d'utilisation de la machine. Ces travaux ont posé les bases et offert un point de départ pour les recherches futures sur l'ordonnancement en groupes.

Allison (1990) a combiné deux algorithmes pour séquencer simultanément les tâches et les groupes. Ils ont proposé quatre combinaisons de l'algorithme Petrov et l'algorithme Campbell Dudek and Smith (CDS), concluant que l'arrangement Petrov (séquençage des tâches) / CDS (séquençage des groupes) est plus performant que les autres combinaisons. Sur la base des résultats d'Allison (1990), plusieurs études ont été réalisées, explorant la combinaison d'heuristiques pour le problème d'ordonnancement de groupes. Logendran et al. (1995) ont développé une autre heuristique LN qu'ils ont combinée avec l'heuristique Petrov pour résoudre le problème à deux niveaux. Ils ont démontré que l'adaptation de l'heuristique LN au premier niveau (ordre des tâches) et l'heuristique PT au deuxième niveau (ordre des groupes) constitue l'arrangement le plus significatif.

Par la suite, plusieurs métaheuristiques ont été développées pour résoudre le problème. La première métaheuristique adaptée au problème est le recuit simulé réalisé par Vakharia et Chang (1990). Ils ont montré que leur algorithme surpassait les algorithmes NEH et CDS. Sridhar et Rajendran (1994) ont explicitement pris en compte les opérations manquantes dans les environnements de flow shop et ont introduit une équation récursive modifiée pour calculer le makespan. Leur algorithme génétique, également utilisé pour minimiser le temps total d'écoulement, a conduit à des résultats supérieurs par rapport à l'algorithme du recuit simulé de Vakharia et Chang (1990). D'autres métaheuristiques ont été introduites pour résoudre le problème selon différentes hypothèses. Solimanpur et Elmi (2011) ont abordé le problème en considérant un nombre limité de buffers. Ils ont proposé un modèle de programmation mixte en nombres entiers et la métaheuristique de recherche tabou pour résoudre le problème d'ordonnancement de groupes avec la minimisation du makespan comme critère d'optimisation.

Par ailleurs, peu de recherche ont considéré les problèmes flow shop de groupe avec temps de configuration indépendant de la séquence en abordant d'autres critères. Zolfaghari et Liang (1999) ont résolu le problème en minimisant le retard pondéré, tout en variant les vitesses des machines. Un modèle de programmation non linéaire à variables mixtes ainsi qu'un algorithme hybride itératif combinant la recherche tabou et le recuit simulé ont été développés. Gupta et Schaller (2006) ont présenté l'algorithme branch and bound ainsi que plusieurs approches constructives et métaheuristiques pour la résolution du problème avec la minimisation de la somme de dates de fin des tâches. Une comparaison avec l'algorithme génétique de Sridhar et Rajendran (1994) démontre l'efficacité des nouveaux algorithmes proposés. Dès lors, peu de travaux ont considéré l'attribut du temps de configuration indépendant de la séquence pour les problèmes d'ordonnement flow shop avec technologie de groupe.

2.3.2 Temps de configuration dépendant de la séquence

Alors que l'hypothèse d'un temps de configuration indépendant de la séquence existe depuis les années 1980 dans les problèmes d'ordonnement de groupe, ce n'est qu'après les années 2000 que les temps de configuration dépendants de la séquence ont été pris en compte dans les problèmes FSDGS. Schaller et al. (2000) sont les pionniers à avoir étudié ce problème. Ils ont développé 12 heuristiques visant à minimiser le makespan. Leur recherche a été soutenue par une étude de cas pratique menée au sein d'une entreprise "Fortune 100" spécialisée dans la production de circuits imprimés. Cette entreprise a classé différentes puces PCB en familles pour réduire le temps de préparation. Ils ont également proposé une borne inférieure pour évaluer la qualité des algorithmes proposés.

Leurs résultats ont montré que l'heuristique CMD donne de meilleurs résultats que les 11 autres heuristiques. CMD est constitué de trois algorithmes heuristiques : l'algorithme C, basé sur la procédure de Campbell-Dudek-Smith, pour planifier les tâches au sein du groupe ; puis l'algorithme M, où la procédure NEH modifiée est utilisée pour séquencer les familles des tâches. Enfin, pour améliorer les résultats de l'algorithme CM, ils ont proposé l'algorithme D, ou heuristique de décente, basé sur une procédure de recherche local.

Deux autres algorithmes ont été présentés par França et al. (2005) pour le problème FSDGS : l'algorithme génétique (GA) et l'algorithme mémétique (MA), accompagnés d'une stratégie de démarrage multiple (MS). Leurs algorithmes surpassent ceux proposés par Schaller et al. (2000). Andrés et al. (2005) ont introduit un problème d'ordonnement flow shop flexible avec groupes, où les temps de configuration dépendent de la séquence. Ils ont développé et appliqué une heuristique basée sur une règle de répartition dans le cadre d'une étude de cas menée au sein d'une entreprise de fabrication de tuiles.

Leur attention s'est particulièrement portée sur la structuration des familles, basée sur des coefficients de similarité entre les tâches, dans le but de réduire les temps de configuration et d'augmenter la productivité.

Hendizadeh et al. (2008) ont proposé cinq métaheuristiques basées sur la recherche tabou (TS) pour minimiser le makespan dans un contexte où les temps de préparation dépendent de la séquence au sein d'un atelier flow shop. Ces algorithmes reposent sur des procédures sophistiquées d'intensification et de diversification, en intégrant le concept d'élitisme et en acceptant les mouvements moins favorables. Ils ont également été comparés aux heuristiques CMD et aux métaheuristiques MA. En moyenne, les performances de tous les algorithmes proposés surpassent celles de CMD, bien que le temps de calcul soit plus long. Notamment, l'algorithme TS/SA, basé sur le concept de recuit simulé, a démontré des performances équivalentes à celles de l'algorithme MA. Eddaly et al. (2009) ont présenté un algorithme d'estimation de distribution suivi d'une recherche locale itérative.

Un autre algorithme, basé sur le recuit simulé, a été développé par S.-W. Lin, Gupta, Ying, et Lee (2009), qui a utilisé la fonction de Cauchy à la place de la fonction de Boltzmann lors du processus de recuit, montrant une efficacité remarquable. Les auteurs ont ajusté cette approche pour générer des ordonnancements de non-permutation au sein de la séquence de familles, mais pas dans la séquence des tâches de chaque famille. Une étude approfondie des avantages des ordonnancements de non-permutation a été fournie par Ying, Gupta, Lin, et Lee (2010). Celano et al. (2010) ont étudié la limitation de la capacité des buffers opérationnels entre les machines d'inspection de plaquettes de semi-conducteurs d'une société produisant des appareils électriques, ils ont évalué son impact sur la productivité en développant un algorithme génétique testé avec leurs propres instances du problème, son efficacité a été prouvée par rapport à l'algorithme TS et à l'heuristique NEH. Outre le makespan, pour lequel seules des améliorations légères étaient possibles, d'autres critères d'optimisation ont également été pris en compte. En outre, S.-W. Lin, Ying, Lu, et Gupta (2011) ont proposé une stratégie combinée, utilisant une recherche ascendante multi-démarrage en conjonction avec une procédure de recuit simulé. Leur algorithme a démontré des performances comparables à huit métaheuristiques existantes.

Les premiers travaux sur le problème du FSDGS, avec la fonction objective de minimiser la somme des dates de fin des tâches, ont été abordés par Salmasi et al. (2010). Dans cette lignée de recherche, ils ont développé un modèle mathématique linéaire en nombres entiers mixtes. De plus, ils ont présenté deux métaheuristiques, la recherche tabou (TS) ainsi qu'un algorithme hybride de colonie de fourmis (HACO) pour résoudre le problème.

Ces approches ont été comparées, et selon les résultats obtenus, l'algorithme HACO a démontré des performances supérieures par rapport à l'algorithme TS.

Pour évaluer l'efficacité des méta-heuristiques proposées, plusieurs bornes inférieures ont été introduites par les auteurs. Par la suite, l'algorithme HACO a été comparé à l'algorithme hybride d'optimisation par essaims de particules (HPSO) développé Hajinejad, Salmasi, et Mokhtari (2011) pour minimiser la somme des dates de fin. À la suite de multiples tests, il a été conclu que leurs algorithmes étaient plus performants que celui proposé par Salmasi et al. (2010).

Cette même problématique a été abordée par Naderi et Salmasi (2012), qui ont obtenu des résultats remarquables à l'aide d'un algorithme hybride combinant l'algorithme génétique et le recuit simulé. Ils ont également proposé deux formulations MILP pour la minimisation de la somme des dates de fin du problème FSDGS. Le second modèle proposé surpasse largement ceux précédemment présentés dans la littérature, car il peut résoudre des problèmes de grandes instances allant jusqu'à 60 jobs de manière optimale. Keshavarz et Salmasi (2014) se penchent sur le même problème et proposent un algorithme génétique hybride pour sa résolution. Ils démontrent que leur métaheuristique surpasse celles présentées par Hajinejad et al. (2011)

Costa et al. (2014) examinent le problème M-machines FSDGS avec l'allocation des employés qualifiés, en introduisant un modèle de programmation linéaire en nombres entiers mixtes ainsi que trois métaheuristiques basées sur les algorithmes génétiques. Costa et al. (2017) proposent un algorithme génétique hybride pour le problème FSDGS, en se concentrant sur la minimisation du makespan. Leur algorithme est amélioré par un schéma de recherche par échantillonnage aléatoire biaisé, visant à améliorer les solutions obtenues. À la suite de nombreux tests, ils prouvent l'efficacité de leur algorithme par rapport à celui proposé par Naderi et Salmasi (2012)

Adressi et al. (2016) ont intégré des pannes aléatoires pour le problème FSDGS flexible à deux stages. l'algorithme génétique et l'algorithme du recuit simulé ont été proposés pour résoudre le problème avec pour objectif la minimisation du temps maximal d'achèvement des tâches. Les résultats des calculs montrent que les algorithmes GA et SA fonctionnent correctement, mais que le SA semble fournir de meilleurs résultats pour les instances générées. Toujours face aux mêmes variantes du problème, Costa et al. (2020) ont proposé un nouvel algorithme génétique novateur auto-adaptif (PSAGA), caractérisé par sa capacité à modifier les paramètres génétiques selon un schéma évolutif, ce qui permet d'atteindre un équilibre adéquat entre l'exploration et l'exploitation, évitant ainsi une analyse de calibrage fastidieuse. Les auteurs ont démontré qu'en plus de l'avantage en termes de qualité des solutions, une caractéristique importante du PSAGA développé réside dans le

fait qu'aucune phase de réglage préliminaire n'est nécessaire, garantissant ainsi une mise en œuvre rapide et efficace. De plus, plusieurs expérimentations ont été réalisées, appuyé par une analyse statistique spécifique démontrant l'efficacité de l'approche proposée par rapport aux métaheuristiques HGA proposé par Costa et al. (2017), et l'algorithme GSA proposé par Naderi et Salmasi (2012). Yuan, Li, et Wang (2020) ont exploré un problème d'ordonnement flow shop de groupes à deux machines, prenant en compte les temps de blocage et de transport associés aux tâches dans l'industrie sidérurgique, avec pour objectif de minimiser le makespan. Afin d'améliorer la convergence de l'algorithme génétique coévolutif (CGA) proposé pour résoudre ce problème, ils ont développé une stratégie de construction de chromosomes artificiels basée sur l'exploration de blocs.

Une nouvelle approche d'ordonnement pour les FSDGS flexibles avec maintenance préventive a été présentée par Jain et Jain (2022). Cette approche utilise des algorithmes génétiques pour minimiser la durée d'exécution. Les résultats de quatre études de cas menées dans deux environnements de fabrication démontrent l'efficacité de la méthodologie proposée pour résoudre les complexités du problème d'ordonnement de groupe avec SDST et maintenance préventive. S. Lu et al. (2021) ont proposé un lemme très utile dans le cas où les machines sont dominantes et les tâches se détériorent, et ont optimisé le makespan et le temps total d'acheminement en concevant des heuristiques.

Par ailleurs, d'autres auteurs ont considéré l'aspect de distribution dans les problèmes de FSDGS, en supposant plusieurs usines de fabrication dans l'ordonnement de la production, comme c'est le cas des études réalisées par Z.-Y. Wang, Yu, Pan, et Li (2021) concernant un dérivé du problème FSDGS, connue sous le nom d'atelier flow shop de permutation de groupes avec distribution (DFSDGS). Pour la résolution du problème, un algorithme simple de recherche itéré (IG) renforcée avec des procédures améliorées d'initialisation, de construction et de destruction a été proposé avec comme objectif la minimisation de la somme des retards. Pan, Gao, et Wang (2020) aborde également la notion de distribution au sein des problèmes FSDGS et proposent un modèle de programmation linéaire en nombres entiers mixtes, un paradoxe contre-intuitif consistant à introduire une nouvelle famille à une cellule entraînant ainsi la réduction du makespan, et deux suites d'accélération pour économiser les efforts de calcul. En raison de la complexité du problème, les auteurs ont envisagé une stratégie de décomposition incluant un algorithme co-évolutif coopératif (CCEA) avec un nouveau modèle de collaboration et un schéma de réinitialisation. Plus tard, Z.-Y. Wang et al. (2022) ont conçu un algorithme de recherche itéré à deux stages pour minimiser la somme des retards. En tenant compte des contraintes de blocage et de report des configurations du DFGSP, Niu et Li (2022) ont amélioré un algorithme évolutionnaire coopératif en combinant IG et VNS pour minimiser

le makespan et la consommation totale d'énergie.

En suivant l'évolution du temps, les chercheurs se sont efforcés d'améliorer les méthodes de résolution approchée pour les problèmes de FSDGS. Parmi les travaux remarquables, H. Qin et al. (2022) ont présenté un algorithme IG modifié, appelé algorithme NIG, utilisé pour résoudre le problème DFSDGS hybride avec des contraintes de blocage. Cette étude compare plusieurs algorithmes avancés, et les résultats de simulations démontrent que l'algorithme NIG surpasse les algorithmes existants en termes de performance.

De plus, dans la pratique, il arrive souvent que seule une séquence d'ordonnement optimale soit nécessaire, ce qui requiert une exploration approfondie de l'espace de solution et une optimisation maximale des objectives. Sur cette base, Y. Wang et al. (2023) ont abordé le problème de DFSDGS en développant un modèle de programmation linéaire en nombres entiers mixtes. En explorant les caractéristiques implicites du problème, deux méthodes d'évaluation rapide basées sur l'insertion de groupes et l'insertion de tâches ont été conçues pour accélérer l'évaluation des objectifs.

Ensuite, un algorithme efficace de recherche itérée à deux stages (tIGA) a été proposé pour résoudre les trois sous-problèmes couplés mentionnés précédemment, à savoir l'affectation des groupes aux usines, l'ordonnement des groupes et le séquençage des tâches au niveau de chaque groupe. Les résultats expérimentaux démontrent son efficacité par rapport aux algorithmes existants, comparé en termes de valeurs des objectives et de pourcentages d'écart relatifs. En synthétisant les différents travaux considérés, on peut affirmer que le problème FSDGS a été abordé en adaptant diverses métaheuristiques. Parmi ces approches, on retrouve l'utilisation d'algorithmes génétiques (Celano et al., 2010 ; Costa et al., 2017, 2020), du recuit simulé (Nikjo & Rezaeian, 2014), d'algorithmes de recherche itérés (H.-M. Cheng & Ying, 2011 ; H. Qin et al., 2022 ; Zhao, Pan, & Gao, 2023), ainsi que des métaheuristiques hybrides ((Feng et al., 2018 ; Liou & Hsieh, 2015 ; Naderi & Salmasi, 2012).

En général, la stratégie d'optimisation adoptée par la majorité de la communauté scientifique pour les problèmes FSDGS privilégie une résolution mono-objectif, comme observé dans l'analyse précédente. Cependant, la pratique opérationnelle exige souvent l'optimisation de plusieurs critères simultanément afin de s'aligner au mieux avec l'environnement industriel concret.

Dans cette perspective, certains travaux, tels que ceux réalisés par, Keshavarz et al. (2019) ont développé un algorithme hybride basé sur l'optimisation par essaim de particules (HPSO) pour minimiser l'avance et le retard pondérés totaux dans le problème FSDGS. Pour évaluer ses performances, ils ont introduit un algorithme de synchronisa-

tion et une borne inférieure. Les résultats expérimentaux indiquent l'efficacité de l'algorithme HPSO proposé pour résoudre le problème, en particulier dans les cas où les dates d'échéance ne sont pas fixées à l'avance.

Plus récemment, les travaux de D. N. Sekkal et Belkaid (2023) ont abordé le problème du FSDGS en introduisant un effet d'apprentissage, en adoptant une résolution multiobjectif visant à minimiser simultanément le makespan et la consommation d'énergie de l'ensemble du système. Un modèle mathématique linéaire en nombre entier mixte a été développé pour déterminer l'ordonnancement optimal des groupes, des jobs appartenant aux groupes, ainsi que les vitesses de production et de transport appropriées afin d'améliorer les performances globales du système.

En raison de la complexité du problème abordé, les auteurs proposent deux bornes inférieures et une métaheuristique multiobjectif basée sur le principe du recuit simulé (MOSA), qui a été programmée et testée à travers une multitude d'expérimentations où plusieurs paramètres ont été variés, incluant les coefficients de pondération, les différents taux d'apprentissage, le ratio des temps de configurations, et les cadences de production et de transport. Cela offre un champ décisionnel vaste pour les décideurs afin de les aider à trouver le bon équilibre entre tous les objectifs. Les résultats obtenus démontrent l'efficacité de l'algorithme dans l'amélioration des performances des systèmes de production en réduisant le temps de traitement et la consommation d'énergie.

Pour plus d'informations, nous invitons le lecteur à se référer aux revues de littérature sur le problème FSDGS publiées par Neufeld, Gupta, et Buscher (2016) et İnce, Deliktaş, et Hakan Selvi (2023).

Nous résumons dans le tableau 2.2 tous les résultats sur les problèmes d'ordonnancement flow shop avec groupe. Pour chaque référence, nous spécifions le type d'atelier, les contraintes à respecter, les objectifs à optimiser et l'approche de résolution employé.

2.4 Aperçu sur les travaux d'ordonnancement avec effet d'apprentissage

Dans les problèmes d'ordonnancement classiques, le temps de traitement était autrefois considéré comme constant dans le temps et connu d'avance. Cependant, en réalité, il est sujet à des variations en raison de l'effet d'apprentissage et/ou de l'effet de détérioration. L'expérience de l'opérateur s'accroît avec le temps, car en effectuant la même tâche de manière répétée, il acquiert un savoir-faire et une organisation optimale de son poste de travail. Les pionniers à avoir introduit ce concept dans l'ordonnancement sont (Biskup, 1999) et (T. Cheng & Wang, 2000), bien que son utilisation ait été largement répandue

Tableau 2.2: Synthèse des problèmes d'ordonnement flow shop avec groupe

Référence	Caractéristiques des problèmes				Approches de résolution	
	Atelier	Contraintes	objectives	Méthodes Exactes	Méthodes Approchées	
(Hirami & Hani, 1976)	Fm	$fmls$ ✓ LR ✓ ST_{sd} ✓ TS_{tp} ✓ $speed$ ✓	économique $Cmax$	✓	✓	
(Han et al., 1985)	Fm	✓	$Cmax$	✓	✓	
(Allison, 1990)	Fm	✓	$Cmax$	✓	✓	
(Vakharia & Chang, 1990)	Fm	✓	$Cmax$	✓	✓	
(Sridhar & Rajendran, 1994)	Fm	✓	$Cmax, \sum C_j$	✓	✓	
(Solimanpur & Elni, 2011)	Fm	✓	$Cmax$	✓	✓	
(Logendran et al., 1995)	Fm	✓	$Cmax$	✓	✓	
(Zolfaghari & Liang, 1999)	Fm	✓	$\sum W_j \cdot T_j$	✓	✓	
(Gupta & Schaller, 2006)	Fm	✓	$\sum C_j$	✓	✓	
(Schaller et al., 2000)	Fm	✓	$Cmax$	✓	✓	
(França et al., 2005)	Fm	✓	$Cmax$	✓	✓	
(Andrés et al., 2005)	$FF3$	✓	$Cmax$	✓	✓	
(Hendizadeh et al., 2008)	Fm	✓	$Cmax$	✓	✓	
(Eddaly et al., 2009)	Fm	✓	$Cmax$	✓	✓	
(Ying et al., 2010)	Fm	✓	$Cmax, Tmax, \sum W_j \cdot T_j, \sum C_j$	✓	✓	
(Celano et al., 2010)	Fm	✓	$Cmax$	✓	✓	
(S.-W. Lin et al., 2011)	Fm	✓	$Cmax$	✓	✓	
(Salmasi et al., 2010)	Fm	✓	$\sum C_j$	✓	✓	
(Hajinejad et al., 2011)	Fm	✓	$\sum C_j$	✓	✓	
(Naderi & Salmasi, 2012)	Fm	✓	$\sum C_j$	✓	✓	
(Keshavarz & Salmasi, 2014)	Fm	✓	$\sum C_j$	✓	✓	
(Costa et al., 2017)	Fm	✓	$Cmax$	✓	✓	
(Adressi et al., 2016)	$FF2$	✓	$Cmax$	✓	✓	
(Costa et al., 2020)	Fm	✓	$Cmax$	✓	✓	
(Yuan et al., 2020)	$F2$	✓	$Cmax$	✓	✓	
(Jain & Jain, 2022)	FFm	✓	$Cmax$	✓	✓	
(H. Ojm et al., 2022)	HFS	✓	$Cmax$	✓	✓	
(Y. Wang et al., 2023)	DFm	✓	$Cmax$	✓	✓	
(Keshavarz et al., 2019)	Fm	✓	$\sum (W_j \cdot E_j + W_j^a \cdot T_j)$	✓	✓	
(D. N. Sekkal & Bellaid, 2023)	Fm	✓	$Cmax$	TEC	✓	

Références des abréviations : $fmls$ - technologie de groupe; LR - effet d'apprentissage; ST_{sd} - temps de configuration indépendant de la séquence; ST_{st} - temps de configuration; ST_{sd} - temps de configuration dépendant de la séquence; TS_{tp} - temps de transport; $speed$ - vitesse variable; Fm - flow shop flexible; DFm - flow shop de distribution.

dans les sciences de gestion depuis sa découverte par Wright (1936).

En présence de l'effet d'apprentissage, le temps de traitement du job est plus court s'il est ordonnancé ultérieurement dans la séquence. Son application pour la première fois aux systèmes de production remonte à plus de 80 ans, grâce à Wright (1936) qui a démontré à travers ses formules qu'une opération donnée connaît une amélioration de productivité de 20% à chaque fois que la quantité de production double. Après le succès de la formule de Wright, de nombreux autres modèles d'apprentissage ont été proposés afin de le représenter de manière aussi réaliste que possible. Par ailleurs, ce n'est qu'à partir des années 2000 que ce concept a été introduit dans le domaine de l'ordonnancement, notamment par Biskup (1999) et T. Cheng et Wang (2000)

Lorsque les individus s'engagent dans des activités significatives au sein de contextes d'ordonnancement, tels que la mise à jour de logiciels, la configuration ou la désinstallation entre les tâches, les opérations sur les machines, la maintenance ou le remplacement d'équipements, les effets d'apprentissage jouent un rôle crucial. Selon Biskup (1999), le temps requis pour accomplir une tâche tend généralement à diminuer de manière exponentielle à mesure que le nombre de répétitions de la tâche augmente.

Étant donné que les environnements de production continueront de changer constamment, plusieurs modèles d'effet d'apprentissage ont été proposés afin de rendre la formulation du problème aussi réaliste que possible.

Les modèles existants dans la littérature sont classés en deux approches principales : la première est celle des effets d'apprentissage basés sur la position, où l'expérience dépend du nombre de travaux traités précédemment. En revanche, la deuxième approche, appelée effets d'apprentissage basés sur la somme des temps de traitement, implique que l'expérience dépende de la somme du temps de traitement de tous les travaux déjà effectués. Ce concept a été proposé pour la première fois par Kuo et Yang (2006). Par la suite, d'autres études ont tenté de combiner les deux modèles, cherchant à prendre en compte plus d'une observation pour modéliser les effets d'apprentissage de la manière la plus réaliste possible. Citons notamment les travaux réalisés par T. E. Cheng et al. (2008).

Ainsi, nous pouvons synthétiser la séquence chronologique des études menées sur les problèmes d'ordonnancement avec effet d'apprentissage par la figure 2.1.

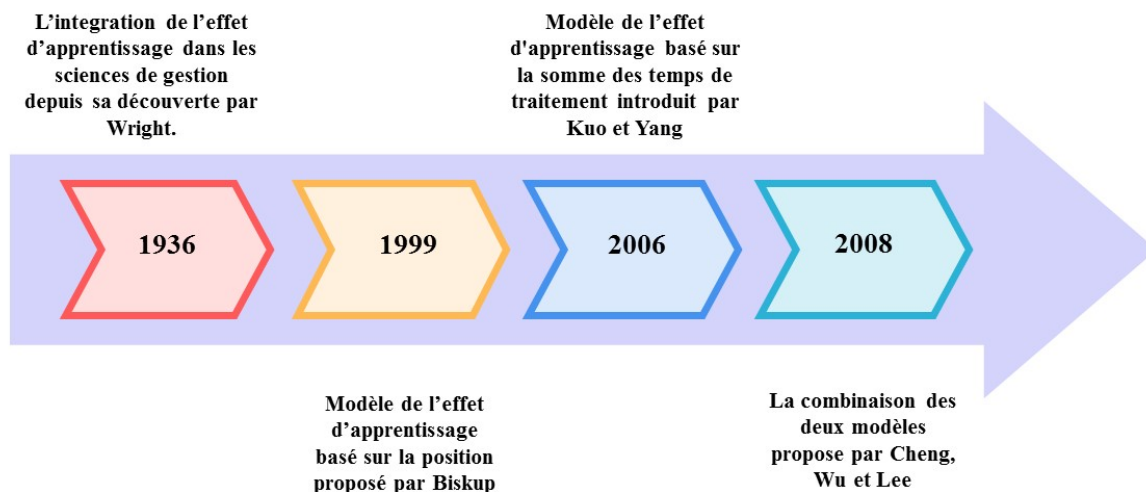


Figure 2.1: Échéancier des problèmes d'ordonnancement avec effet d'apprentissage

Dans le domaine de l'ordonnancement, l'influence de l'apprentissage se révèle à travers les temps de traitement des tâches, qui sont déterminés par des fonctions non croissantes dépendant de l'expérience du processeur. Comme énoncé précédemment, la littérature recense divers types de modèles d'ordonnancement intégrant des effets d'apprentissage, que nous exposons dans les sections suivantes.

2.4.1 Effet d'apprentissage basé sur la position

Biskup (1999) a présenté l'effet d'apprentissage basé sur la position, introduisant une diminution du temps de traitement d'un job en fonction de son ordre d'exécution. Ainsi, le temps de traitement du job j programmé en position r est exprimé par la formule suivante : $P_{j[r]} = P_j \cdot r^a$, où P_j représente le temps de traitement normal du job j , et $a < 0$ est l'indice d'apprentissage. La valeur de a est définie comme $a = \log_2 \cdot (LR) \leq 0$, qui est dépendant du taux d'apprentissage $LR \leq 1$. Dans ce contexte, le temps de traitement décroît à mesure que la tâche est répétée, ce modèle a été introduit par l'auteur dans le cadre d'un problème d'ordonnancement à une seule machine afin de minimiser l'écart par rapport à une date d'échéance et à des temps d'écoulement communs.

Sur la base de cette formule diverses études ont émergés dans lesquels les auteurs tentent de modifier ce modèle en fonction de leurs recherches, on retrouve notamment ceux proposés par Mosheiov et Sidney (2003) qui ont estimé que l'effet d'apprentissage

diffère d'un job à un autre, pour ce faire, les auteurs proposent un effet d'apprentissage dépendant de la tâche comme suite : $P_{j[r]} = P_j \cdot r_j^{a_j}$, où a_j est l'indice de l'apprentissage propre au job j . Sous ce modèle, Vahedi Nouri et al. (2013) proposent un algorithme hybride basé sur le principe du recuit simulé et du comportement des lucioles pour le problème de flow shop avec des activités de maintenance flexibles.

Une autre constatation provenant du secteur manufacturier réel suggère que lorsque la machine est complètement automatisée, il serait plus judicieux de supposer que l'apprentissage se produit principalement durant les phases de réglage de la machine ou bien durant les actions de maintenance. Pour ce faire, plusieurs auteurs abordent ce point de vue, Koulamas et Kyparisis (2007) assument que le temps de traitement P_j est composé du temps de réglage s_i et du temps de production v_i comme suit : $P_{j[r]} = s_{j[r]} + v_j$, avec $s_{j[r]} = s_j \cdot r^a$. Pour la même observation, Pargar et Zandieh (2012) abordent un problème flow shop hybride bi-objectif avec des temps de configuration. Ils proposent un modèle mathématique et une métaheuristique pour minimiser la somme pondérée du makespan et du retard total. D'autres auteurs ont abordé ce constat, on cite notamment les travaux faits par Costa et al. (2014), Fichera et al. (2015), Pargar et al. (2018), Mousavi et al. (2018) et Bektur (2022).

Eren et Güner (2007) ont étudié un problème d'ordonnancement à une seule machine qui a pour fonction objective la minimisation du retard total tout en tenant compte de l'effet d'apprentissage, ils ont proposé un modèle de programmation en nombres entiers et adapté trois métaheuristicques (la recherche tabou, le recuit simulé et la procédure de recherche locale) pour résoudre le problème lorsque les instances sont assez élevées allant jusqu'à 1000 tâches. Parallèlement, İşler, Toklu, et Çelik (2012) ont étudié l'effet de l'apprentissage basé sur la position dans un problème d'ordonnancement en flux tendu (sans avance ni retard). Hosseini et Tavakkoli-Moghaddam (2013) ont examiné un problème d'ordonnancement multi-objectif à deux machines, prenant en compte l'effet de l'apprentissage basé sur la position et l'heure d'arrivée des tâches, lequel est dynamique, c'est-à-dire que chaque tâche arrive selon une fenêtre temporelle. Comme le modèle est NP-difficile, ils ont proposé deux métaheuristicques, MOGA et MOSA, pour résoudre le problème. Après plusieurs expériences, il s'avère que MOGA est plus performant que MOSA en termes de fonction objective. De manière similaire, Vahedi-Nouri, Fattahi, et Ramezani (2013) ont abordé un problème d'ordonnancement de type flow shop sans permutation, dans lequel le temps de traitement est une fonction de l'effet d'apprentissage variant d'une machine à l'autre. L'indisponibilité des machines a également été prise en compte, leurs objectifs étant de trouver le séquençage optimal des tâches et la planification optimale des activités de maintenance simultanément.

Diverses études explorent une autre réalité où tout retard dans le traitement d'une tâche peut entraîner une augmentation de l'effort requis pour la compléter. Autrement

dit, le temps effectif de traitement d'une tâche est représenté comme étant proportionnel à son temps de commencement en raison des effets de détérioration. J.-B. Wang (2006) a considéré la détérioration et l'effet d'apprentissage simultanément dans un problème à une seule machine. l'auteur propose deux modèles mathématiques considérant différentes fonctions objectifs (le makespan, la somme pondérée des dates de fin et du retard maximum) et démontre que malgré la prise en compte de la détérioration et de l'effet d'apprentissage, plusieurs problèmes d'ordonnancement à une seule machine et flow shop demeurent solvables de façon polynomiale. Fu, Ding, Wang, et Wang (2018) traitent un problème d'ordonnancement multi-objectif dans un environnement flow shop sous contraintes d'apprentissage et de détérioration, en tenant compte de l'utilisation de machines intelligentes dans les systèmes de fabrication basés sur l'industrie 4.0.

Toutes les recherches précédentes ont contribué à l'essor et au développement de l'effet d'apprentissage dans le domaine de l'ordonnancement, ce qui a conduit à des études récentes telles que celle menée par Pargar et al. (2018). Ils ont proposé une méthode d'ordonnancement pour un environnement hybride de flow shop en mettant l'accent sur l'apprentissage des travailleurs. Pour minimiser le makespan et le retard total, ils ont développé un modèle de programmation mixte en nombres entiers bi-objectif. Deux méthodes de solution basées sur l'algorithme de flux d'eau hybride et les concepts de tri et de classement non dominés sont proposés pour optimiser l'ordonnancement et réduire les temps de configuration. Dans une autre étude, Xin, Jiang, Li, Li, et Chen (2023) ont introduit un effet d'apprentissage basé sur la position pour minimiser le délai d'exécution et la consommation d'énergie dans un problème d'ordonnancement flow shop avec permutation et économe en énergie. Un modèle de programmation bi-objectif est développé, et un algorithme multiobjectif itéré et gourmand (MOIG) est conçu pour atteindre le front de Pareto, améliorant ainsi l'efficacité écologique et maîtrisant les coûts d'exploitation de manière effective. Par ailleurs, d'autres auteurs se sont appuyés sur des méthodes de résolution exacte dans le processus d'optimisation des problèmes d'ordonnancement, comme c'est le cas pour Bai et al. (2021) qui ont considéré un problème d'ordonnancement flow shop avec effet d'apprentissage basé sur la position avec comme objectif la minimisation du retard maximal. Pour la résolution exacte des petites instances du problème, les auteurs proposent l'algorithme Branch and Bound équipé par des règles de branchement et une borne inférieure conçues dans le but d'améliorer sa capacité de recherche. Un algorithme basé sur le comportement des colonies d'abeilles artificielles (DABC) équipé d'un mécanisme de recherche de voisinage hybride est introduit pour fournir des solutions de haute qualité pour les moyennes et grandes instances. La performance de l'algorithme est également amélioré par l'heuristique EDDA (date d'échéance la plus proche).

2.4.2 Effet d'apprentissage basé sur la somme des temps de traitement

Kuo et Yang (2006) ont avancé une hypothèse alternative concernant l'effet d'apprentissage, suggérant que le temps de traitement est influencé par la durée consacrée aux tâches précédentes. En d'autres termes, plus on investit de temps dans le travail, plus on développe des compétences et plus, on améliore ses performances. Par conséquent, l'effet d'apprentissage est déterminé par la somme des temps de traitement des tâches déjà planifiées, et sa formulation est la suivante : $P_{ir} = (1 + \sum_{k=1}^{r-1} P_r)^a \cdot P_i$. Kuo, Hsu, et Yang (2012) examinent un effet d'apprentissage dépendant du temps dans le problème d'ordonnancement flow shop, où l'efficacité des tâches s'améliore en fonction des temps de traitement des tâches précédentes. Plusieurs fonctions objectives ont été étudiées par les auteurs : le makespan, le temps total d'écoulement, la somme des dates de fin pondérées, la somme du k-ième puissances des dates de fin, et le retard maximal. Un algorithme polynomial est proposé pour les cas spéciaux avec des temps de traitement identiques sur chaque machine et une série croissante de machines dominantes, respectivement. De récentes contributions ont été apportées par les auteurs concernant ce type d'effet d'apprentissage, notamment ceux faits par (Y.-Y. Lu, 2016), (W.-C. Lin, Wu, Yu, Zhuang, & Liu, 2017), (L. Liu, Shi, & Shi, 2018), (L. Liu, He, & Shi, 2019), (S.-C. Liu, 2020), (Paredes-Astudillo, Botta-Genoulaz, & Montoya-Torres, 2022), (Y.-C. Wang & Wang, 2023).

Une autre vision concernant l'effet d'apprentissage a été abordé par T. E. Cheng, Cheng, Wu, Hsu, et Wu (2011) dans laquelle les auteurs supposent que l'effet d'apprentissage doit être limité pour éviter que le temps de traitement n'atteigne zéro en raison des répétitions de tâches ou de la somme des temps de traitement des tâches précédemment séquencées, ce qui est communément appelé l'effet d'apprentissage tronqué. Pour résoudre ce problème, ils proposent un modèle dans lequel le temps de traitement est une fonction de la somme des temps de traitement des tâches déjà séquencées et d'un paramètre de contrôle. Ce modèle permet de limiter le phénomène d'apprentissage des travailleurs. (Lai, Hsu, Ting, & Wu, 2014) intègrent ce type d'effet d'apprentissage dans un problème d'ordonnancement flow shop à deux machines pour la minimisation du temps total d'achèvement. Les auteurs proposent l'algorithme branch-and-bound et trois algorithmes de recuit simulé et à travers les résultats expérimentaux, ils ont pu montrer que l'algorithme branch-and-bound résout efficacement des instances allant jusqu'à 18 tâches et que les algorithmes de recuit simulé offrent de bonnes performances en termes de temps de calcul CPU et de pourcentage d'erreur. D'autres auteurs ont abordé cette variante comme c'est le cas pour (Azizi & Hu, 2020), (Wu et al., 2021), (M.-H. Li, Lv, Zhang, & Wang, 2024).

2.4.3 Combinaison des deux modèles

Dans de nombreux systèmes de fabrication réels, les effets d'apprentissage peuvent découler simultanément de diverses observations. C'est pourquoi les recherches les plus récentes intègrent plusieurs observations pour modéliser de manière réaliste les effets d'apprentissage. (T. E. Cheng et al., 2008) ont avancé l'hypothèse que les effets d'apprentissage peuvent découler simultanément des apprentissages humains et machine. Pour ce faire, ils ont introduit un modèle général des effets d'apprentissage, combinant l'effet d'apprentissage basé sur la position pour modéliser les effets d'apprentissage machine et la somme du temps de traitement réel pour modéliser les effets d'apprentissage humain. Le temps de traitement réel de ce modèle s'inspire donc de Koulamas et Kyparisis (2007) pour représenter la somme du temps de traitement réel, et de Biskup (1999) pour modéliser les effets d'apprentissage basés sur la position. Ainsi, le temps de traitement réel de la tâche j , lorsqu'elle est programmée à la r ème position dans une séquence, est le suivant : $P_{j[r]} = P_j \cdot (1 - \frac{\sum_{l=1}^{r-1} P_l}{\sum_{l=1}^{r-1} P_l})^{a_1} \cdot r^{a_2}$, avec P_j le temps de traitement de base du job j , $P_{[l]}$ représente le temps de traitement de base du job programmé à la l ème position dans la séquence et a_1 et a_2 représentent les deux indices d'apprentissage avec $a_1 \geq 1$ et $a_2 < 0$. Récemment, les travaux de L. Sun, Ning, et Huo (2020) ont intégré les deux modèles pour la résolution du problème d'ordonnancement avec technologie de groupe pour la minimisation du makespan, du temps total d'achèvement et du temps total d'achèvement pondéré.

2.5 État de l'art sur les problèmes d'ordonnancement axé sur l'efficacité énergétique

Traditionnellement axé sur les objectifs d'optimisation liés au temps et aux coûts, le domaine de l'ordonnancement s'intéresse de plus en plus aux préoccupations environnementales. Des recherches récentes ont émergé, marquant un tournant dans cette perspective.

2.5.1 Vers une optimisation énergétique des problèmes d'ordonnancement flow shop

Parmi les travaux pionniers à l'origine de ce changement se trouvent ceux réalisés par Mouzon et al. (2007), qui ont servi de point de départ pour des études ultérieures. Leur focalisation sur la minimisation des dates de fin et de la consommation d'énergie dans un problème d'ordonnancement à une machine a révélé que les machines qui ne sont pas des goulots d'étranglement engendraient des temps d'inactivité élevés, entraînant une perte

significative d'énergie. Pour remédier à ce problème, plusieurs règles de répartition ont été proposées. La stratégie consistant à éteindre et allumer la machine pendant les temps d'inactivité semble représenter une économie significative et une réduction de la consommation d'énergie.

L'application de cette stratégie permet de réduire la consommation d'énergie pendant la phase d'inactivité de la machine, représentant environ 40% de la consommation totale d'énergie dans les secteurs industriels. Cependant, la majeure partie de la consommation d'énergie se produit pendant la phase de production. Fang, Uhan, Zhao, et Sutherland (2011) ont considéré la vitesse de fonctionnement de la machine comme une variable indépendante pouvant être modifiée pour influencer la charge de puissance maximale et l'empreinte carbone dans un problème d'ordonnancement flow shop à deux machines. Ils ont développé un modèle de programmation linéaire en nombre entier mixte pour résoudre le problème et ont étayé leur recherche avec une étude de cas sur la fabrication de plaques en fonte.

Fang et al. (2013) ont élaboré deux modèles de programmation linéaire en nombre entier mixte, intégrant des contraintes de consommation maximale de puissance. Ils ont varié les vitesses de production pour illustrer leurs impacts sur l'optimisation du temps d'exécution. À cette fin, ils ont pris en compte plusieurs hypothèses, telles que des vitesses de production continues ou discrètes, avec des espaces de stockage intermédiaires limités ou illimités. Cette approche a également été adoptée par Dai, Tang, Giret, Salido, et Li (2013) pour un problème d'ordonnancement flow shop flexible. Ils ont développé un algorithme génétique associé à la procédure du recuit simulé, en testant différents scénarios dans lesquels les poids des deux fonctions objectives (makespan vs consommation d'énergie) sont variés. Les résultats obtenus ont révélé l'aspect conflictuel entre les deux objectifs, rendant complexe la recherche d'une solution optimale satisfaisant les deux critères.

Par ailleurs, les travaux de (Fang et al., 2011, 2013) ainsi que ceux de Dai et al. (2013) négligent les temps de configuration entre les tâches et la consommation d'énergie correspondante. Afin de combler cette lacune, Mansouri, Aktas, et Besikci (2016) ont examiné un problème d'ordonnancement flow shop à deux machines, tenant compte des temps de configuration dépendants de la séquence. Ils ont développé des heuristiques et un algorithme génétique hybride pour trouver un compromis entre le makespan et la consommation d'énergie.

G.-S. Liu, Zhou, et Yang (2017) ont opté pour l'utilisation d'un algorithme génétique

pour résoudre un problème d'ordonnancement flow shop flexible (FFS). Leur étude a été motivée par le processus de fabrication de pneus en caoutchouc, où émergent deux caractéristiques essentielles : la première concerne le temps de préparation, dépendant non seulement de la séquence, mais également de l'état antérieur de la machine en raison d'un processus thermique. La deuxième caractéristique prend en compte des temps de traitement flous, résultant de l'expérience des travailleurs lors de la manipulation des dispositifs thermiques.

C. Lu, Gao, Li, Pan, et Wang (2017) ainsi que E.-d. Jiang et Wang (2019) ont abordé un problème d'ordonnancement flow shop, proposant un algorithme hybride multi-objectif visant à minimiser le makespan et la consommation d'énergie. Ils ont pris en compte des temps de préparation dépendants de la séquence et un temps de transport contrôlable. De plus, ils ont intégré la stratégie d'éteindre/allumer pendant les temps d'inactivité de la machine, tout en tenant compte de sa durée de vie.

Dans une perspective similaire, J.-q. Li, Sang, Han, Wang, et Gao (2018) ont examiné un flow shop flexible (FFS) avec une consommation d'énergie dépendante de la séquence. Ils ont développé un algorithme d'optimisation multi-objectifs axé sur les économies énergétiques pour résoudre le problème, utilisant deux opérateurs de croisement et quatre approches de décodage afin de minimiser le makespan et la consommation d'énergie.

M. H. Ho, Hnaien, et Dugardin (2021) ont étudié un problème d'ordonnancement flow shop à deux machines dans le but de minimiser les coûts énergétiques tout en tenant compte du makespan. Plusieurs solutions optimales ont été obtenues avec des coûts d'électricité différents pour un même makespan. Ces résultats aideront considérablement les gestionnaires à ajuster leur planification, réduisant ainsi les coûts énergétiques sans compromettre la productivité. En ce qui concerne les méthodes de résolution basées sur le recuit simulé, on trouve l'étude réalisée par N. Sekkal et Belkaid (2020), dans lequel un problème de machines parallèles a été considéré en tenant compte de l'effet de détérioration et de la consommation de ressources pour minimiser simultanément le makespan et le coût énergétique.

Récemment, plusieurs travaux en ordonnancement ont pris en considération l'impact disproportionné du changement climatique et ont intégré cette préoccupation dans le processus d'optimisation. Dans cette perspective, les travaux menés par Ghorbanzadeh et Ranjbar (2023) se sont penchés sur un problème d'ordonnancement flow shop énergétiquement conscient, soumis à des contraintes liées à l'énergie renouvelable. Ils ont proposé un algorithme heuristique basé sur la décomposition pour résoudre ce problème.

2.5.2 Vers une optimisation énergétique des problèmes d'ordonnement flow shop avec technologie de groupe

En présence d'un problème d'ordonnement d'atelier avec technologie de groupe, nous observons les travaux réalisés par S.-W. Lin et al. (2009, 2011) ; S.-W. Lin et Ying (2012), au cours desquels la métaheuristique basée sur le recuit simulé (SA) a été utilisée pour résoudre le problème FSDGS, comme détaillé par S.-W. Lin et al. (2009). Par la suite, ils ont développé l'heuristique MSA, qui combine les avantages du recuit simulé (SA) et des mécanismes multi-niveaux pour résoudre le même problème, comme décrit par S.-W. Lin et al. (2011). Cette même technique de résolution a été adaptée par S.-W. Lin et Ying (2012) pour résoudre un problème bi-objectif visant à minimiser simultanément le makespan et le temps total d'écoulement. Un autre aspect étroitement lié à l'ordonnement de la production dans les problèmes FSDGS a été traité par Khamseh, Jolai, et Babaei (2015), où ils ont pris en compte la maintenance préventive qui influence considérablement le makespan. Pour résoudre ce problème, ils ont proposé deux algorithmes, SA et GA, démontrant la supériorité de l'algorithme génétique (GA) par rapport à l'algorithme de recuit simulé (SA).

2.6 Positionnement de notre problématique

Dans cette thèse, nous nous intéressons aux problèmes d'ordonnement simultané de tâches et des groupes dans un atelier flow shop tout en intégrant l'effet d'apprentissage et les temps de configuration. L'étude bibliographique réalisée dans la section précédente aide à tracer les objectifs et les motivations sur lesquels reposent ces travaux de recherche. Nous constatons que les problèmes FSDGS ont été largement étudiés dans la littérature avec différentes fonctions qui modélisent la variation de tâches et des groupes. Également, plusieurs critères d'optimisation ont été évalués tels que : C_{max} , C_j , T_{max} , T_j , etc.

Malgré l'attention croissante portée à la problématique du FSDGS dans la recherche, seules quelques études ont pris en compte l'effet d'apprentissage, les temps de transport intermédiaires et les vitesses variables de production et de transport, et aucune n'a abordé ces contraintes simultanément. De plus, à notre connaissance, la minimisation simultanée du makespan et de la consommation d'énergie dans les problèmes FSDGS n'a pas encore été étudiée. Pourtant, une telle approche permettrait aux praticiens de l'ordonnement dans le secteur manufacturier de prendre en compte à la fois le niveau de service et les facteurs environnementaux dans leurs décisions. L'un des aspects attrayants de ces idées est leur focalisation sur les processus, évitant ainsi la nécessité d'investissements importants dans la reconception des machines ou des produits, ce qui pourrait poser problème

aux petites et moyennes entreprises. De plus, tous les algorithmes actuels pour le FSDGS traitent le problème de l'ordonnement des groupes et celui des tâches séparément au sein de chaque groupe, entraînant ainsi un manque d'efficacité et une certaine imprévisibilité. Notre objectif est de développer une solution qui adresse ces problématiques et surmonte ces limitations.

Pour conclure, les principales originalités de ces travaux de recherche sont :

- Proposer un modèle de programmation linéaire en nombres entiers mixtes bi-objectif pour déterminer le séquençage optimal des groupes et des tâches au sein des groupes, tout en tenant compte des facteurs tels que l'effet d'apprentissage, le temps de configuration dépendant de la séquence (SDST), et les vitesses des machines et du transport.
- Démontrer l'importance de la technologie de groupe et l'impact de l'effet d'apprentissage dans un problème FSDGS à travers une étude de cas réel d'une industrie de bielles forgées.
- Fournir un plan d'ordonnement efficace sur le plan énergétique, intégrant la consommation d'énergie pendant les phases de production, de transport, de réglage et les phases d'inactivité des machines.
- Développer la métaheuristique MOSA pour une optimisation multi-objectif du problème.
- Entreprendre une évaluation approfondie des performances de l'algorithme en utilisant une large combinaison des paramètres de l'étude et en comparant ses résultats avec deux bornes inférieures proposées.

2.7 Conclusion

Cette thèse se concentre sur les problèmes d'ordonnement simultané de groupes et des tâches les composant, en intégrant l'effet d'apprentissage. L'ordonnement est un domaine crucial dans la gestion des opérations, où l'objectif est d'organiser les tâches de manière optimale pour améliorer l'efficacité et la productivité. En particulier, l'ordonnement des groupes et des tâches nécessite une attention particulière en raison de la complexité accrue lorsqu'il s'agit de coordonner plusieurs niveaux d'activités. L'effet d'apprentissage, qui se traduit par une réduction progressive des temps de traitement au fur et à mesure que les opérateurs acquièrent de l'expérience, est un facteur souvent négligé dans les études d'ordonnement traditionnelles. Cette thèse vise à combler cette lacune en examinant comment cet effet peut être intégré pour optimiser les processus d'ordonnement. Une revue de la littérature sur le sujet est donc réalisée dans ce chapitre, afin

de situer notre recherche dans le contexte des études existantes et de mettre en lumière les domaines nécessitant davantage d'investigation.

"En examinant les différentes études, nous avons identifié les objectifs et les motivations de cette thèse, notamment le manque de recherches sur l'intégration de l'effet d'apprentissage dans les problèmes d'ordonnancement, en particulier les problèmes FSDGS, tout en incluant l'aspect énergétique dans le processus d'optimisation.

Cette analyse nous conduit à étudier ce problème plus en détail dans le chapitre suivant, consacré à la modélisation du problème. Nous y intégrerons progressivement ses différentes variantes, et plusieurs expérimentations seront réalisées pour démontrer l'efficacité de l'approche de résolution.

Chapitre 3

Problème flow shop avec technologie de groupe, temps de configuration dépendant de la séquence et effet d'apprentissage

3.1 Introduction

Avant d'entamer tout projet digne de ce nom, il est essentiel de mener des investigations préliminaires, de se familiariser en profondeur avec le domaine concerné, d'identifier les intervenants essentiels à sa réalisation, et de garantir son originalité. Ainsi, le terrain est propice à sa création. La structure de cette thèse s'harmonise parfaitement avec cette progression d'idées. Bien que le chapitre 1 soit parfois considéré comme superflu, il revêt au contraire une importance cruciale dans la définition du domaine de recherche. Le chapitre 2 met en lumière les avancées scientifiques élaborées par divers chercheurs au fil du temps, se focalisant sur le domaine de l'ordonnancement en général et des problèmes de flow shop en particulier. L'élaboration de ces deux chapitres a permis de tracer les grandes lignes de ce travail, donnant ainsi naissance au troisième chapitre.

Dans ce chapitre, nous examinons un problème d'ordonnancement flow shop dans lequel les jobs sont regroupés en familles afin de réduire les temps de configuration, appelé FSDGS. Comme souligné dans la revue de la littérature présentée dans le chapitre précédent, les problèmes d'ordonnancement flow shop ont été abordés avec des hypothèses simplificatrices, s'éloignant ainsi de la réalité industrielle. Dans les problèmes de FSDGS, on suppose généralement que le temps de traitement est constant et connu à l'avance. Cette présomption, jugée trop restrictive, ne correspond pas à la dynamique des envi-

ronnements manufacturiers, car le temps de traitement varie souvent en raison des effets d'apprentissage et/ou de détérioration. Cette variation s'inscrit dans le contexte de la réduction des temps de traitement avec l'effet d'apprentissage et de leur augmentation en présence de l'effet de détérioration.

Dans les problèmes d'ordonnancement avec effet d'apprentissage, le temps de traitement d'une tâche devient plus court à condition qu'elle soit traitée ultérieurement dans la séquence. Cela se manifeste par une amélioration de l'efficacité en réponse à l'accroissement des performances ou des compétences d'une personne ou d'un système à mesure qu'il acquiert de l'expérience ou de la pratique. Dans le contexte spécifique des machines ou des systèmes automatisés, l'effet d'apprentissage peut être intégré par le biais d'algorithmes d'apprentissage automatique. Ces algorithmes permettent à la machine d'ajuster ses performances en fonction des données qu'elle reçoit au fil du temps. Par ailleurs, il peut également être attribué à l'opérateur, démontrant ainsi sa capacité à progresser, à s'adapter et à s'améliorer progressivement grâce à l'expérience et à l'acquisition de nouvelles connaissances.

L'effet d'apprentissage examiné dans cette recherche est directement lié à l'opérateur intervenant lors du traitement de la tâche. Comme indiqué précédemment, ce concept a été initialement introduit par Biskup (1999), et malgré de nombreuses études menées dans ce domaine, l'effet d'apprentissage a été rarement pris en compte et étudié dans les travaux portant sur le problème de FSDGS, et ce, principalement au niveau des temps de configuration. Cependant, l'intervention humaine est inévitable dans les industries de fabrication, se produisant surtout au niveau des temps de production. À cet égard, un effet d'apprentissage dépendant de la position est considéré dans le problème de FSDGS.

L'objectif principal de cette étude est d'évaluer l'impact des différents taux d'apprentissage sur l'ordonnancement simultané des tâches et des groupes, d'une part, et sur l'optimisation des critères, d'autre part. Cette analyse est étayée par une étude de cas réelle sur la fabrication de bielles forgées, car dans ce secteur, l'intervention d'opérateurs expérimentés joue un rôle crucial.

Pendant de nombreuses années, l'objectif principal des industries manufacturières a été de produire à moindre coût et dans des délais très courts, souvent au détriment des préoccupations environnementales. Toutefois, avec le changement climatique, la prise en compte des enjeux environnementaux est devenue une urgence pour un développement durable. Le secteur manufacturier est actuellement responsable de près de 50% de la consommation annuelle mondiale d'énergie, et cette consommation devrait augmenter de plus de 50% d'ici à 2040 (Giret, Trentesaux, & Prabhu, 2015). Des études préliminaires sur l'impact environnemental des machines-outils ont révélé que plus de 90% de cet impact,

y compris la consommation d'énergie, survient pendant la phase d'utilisation (Tigane, Dahane, & Boudhar, 2019). Ainsi, la réduction des temps de traitement par l'intégration de l'apprentissage des travailleurs devient cruciale pour atténuer la consommation d'énergie.

De plus, étant donné que les prix de l'énergie devraient connaître une hausse significative à long terme, la plupart des industries manufacturières ont commencé à intégrer les coûts énergétiques dans leurs objectifs (Y. Liu et al., 2014), (M. H. Ho et al., 2021). Sur la base de ces constats, il est possible d'aborder le problème des FSDGS avec effet d'apprentissage en adoptant une perspective attentive à la gestion énergétique.

Par ailleurs, il a été observé que la plupart des travaux antérieurs portant sur les problèmes FSDGS optent pour la minimisation d'un seul objectif (Ghorbanzadeh & Ranjbar, 2023), alors que l'optimisation multi-objectifs prend en compte divers critères et contraintes dans le processus décisionnel. Ainsi, cette étude vise à minimiser d'une part le makespan, en tant que mesure dédiée au calcul de la durée totale d'un ordonnancement en tant qu'objectif économique, et d'autre part, à réduire la consommation d'énergie du système de production en tant qu'objectif écologique.

Dans ce chapitre, nous proposons un nouveau modèle et des outils d'aide à la décision pour l'optimisation mono-objectifs de l'ordonnancement du problème FSDGS en tenant compte de l'effet d'apprentissage et des caractéristiques du système. Ce modèle intègre simultanément les décisions opérationnelles relatives aux machines et au séquençage approprié des groupes et des tâches, afin d'aider les décideurs à trouver le bon équilibre entre l'objectif à minimiser en intégrant les contraintes énergétiques liées aux émissions carbone.

Nous entamons ce chapitre par une présentation détaillée du problème étudié, exposée dans la section suivante (section 3.2). La section 3.3 se consacre à une exploration approfondie du problème en intégrant l'effet d'apprentissage sous l'optimisation mono-objectif. Nous élargissons ensuite notre modèle dans la section 3.4 en adoptant une configuration dépendante du groupe, tout en intégrant des temps de changement dépendant de la séquence. Dans ce cadre, le regroupement des jobs réduit considérablement les temps de configurations. Plus spécifiquement, dans la section 3.5, nous nous penchons sur l'étude du problème en mettant l'accent sur la minimisation du C_{max} en tenant compte des contraintes énergétiques. ce chapitre se termine par une conclusion synthétisant les résultats obtenus.

3.2 Présentation du problème

Avant d'entamer la description du modèle proposé, il est essentiel d'exposer les modèles généraux employés dans la modélisation du problème étudié.

appliqués.

- Le premier type, proposé par Biskup (1999), stipule que le temps de traitement d'une tâche diminue à mesure du nombre de répétitions, communément appelé effet d'apprentissage basé sur la position, également connu sous le nom de "position-based learning effect". Il est représenté comme suit :

$$P_{jr} = P_j \cdot r^a \quad \forall j, r = 1, \dots, n \quad (3.3)$$

où $P(j, r)$ est le temps de traitement de la tâche j si elle est exécutée en position r , P_j est le temps de traitement normal ou basic de la tâche j , $a < 0$ est l'indice d'apprentissage.

- Le second type est basé sur la somme des temps de traitement des tâches ordonnancées ultérieurement dans la séquence, appelé "sum-of-processing-time-based learning effect". Il repose sur le principe que plus on s'entraîne, meilleures sont les performances. À mesure que vous vous entraînez davantage, vos performances s'améliorent, évaluées en fonction de la somme des temps de traitement réels. Ce concept est représenté par la formule suivante :

$$P_{jr} = P_j \cdot \left(1 + \sum_{k=1}^{r-1} P_{[k]}\right)^a \quad \forall j, r = 1, \dots, n \quad (3.4)$$

Le chiffre 1 est le terme modificateur qui affirme qu'il s'agit d'un effet d'apprentissage, c'est-à-dire que $0 < \left(1 + \sum_{k=1}^{r-1} P_{[k]}\right)^a \leq 1$, comme démontré par Kuo et Yang (2006). Cette expression garantit au moins que le temps de traitement du premier job ordonnancé est égale à son temps de traitement de base.

3.2.3 Extrapolation des modèles généraux

Un modèle global pour un système de production en présence d'effet d'apprentissage, adapté à un atelier composé d'une seule machine, a été présenté dans la section précédente. Par ailleurs, les ateliers de type flow shop suscitent un intérêt accru et se rapprochent davantage des défis réels rencontrés dans la pratique. Notre modèle peut être aisément étendu aux ateliers à machines en série (flow shop). Pour ce faire, un ensemble de n tâches doit s'exécuter sur un ensemble de m machines disposées en série. Une tâche T_j ordonnancée à la position r sur une machine i est définie par un temps opératoire P_{jri} .

Le temps opératoire va augmenté en présence de l'effet de détérioration de cette ma-

nière :

$$P(j, 1, i) \leq P(j, 2, i) \leq \dots \leq P(j, n, i) \quad \forall j = 1, \dots, n \quad \forall i = 1, \dots, m \quad (3.5)$$

En présence de l'effet d'apprentissage, c'est l'effet contraire qui va se produire :

$$P(j, 1, i) \geq P(j, 2, i) \geq \dots \geq P(j, n, i) \quad \forall j = 1, \dots, n \quad \forall i = 1, \dots, m \quad (3.6)$$

Dans un problème d'ordonnancement, deux types d'effet d'apprentissage peuvent être appliqué comme expliqué dans la section 3.2. Par conséquent ces modèles peuvent s'appliquer au problème flow shop de la manière suivante :

- Effet d'apprentissage basé sur la position :

$$P_{jri} = P_{ji} \cdot r^{a_i} \quad \forall j, \forall i, \forall r \quad (3.7)$$

- Effet d'apprentissage basé sur la somme des temps de traitement :

$$P_{jri} = P_{ji} \cdot \left(1 + \sum_{k=1}^{r-1} P_{i[k]}^{a_i}\right) \quad \forall j, \forall i, \forall r \quad (3.8)$$

Dans de nombreuses méthodes de production, l'amélioration de l'efficacité peut découler du regroupement d'éléments ou de produits partageant des caractéristiques similaires. Ce concept, connu sous le terme de "technologie de groupe" dans la littérature et expliqué en détail dans le chapitre 1, a été examiné à travers diverses applications, chacune présentant des avantages distincts.

Les modèles d'effet d'apprentissage peuvent être étendu aux problèmes d'ordonnancement par groupe dans les ateliers flow shop comme suite :

$$P_{g j r i} = P_{g j i} \cdot r^{a_i} \quad \forall j, \forall i, \forall g, \forall r \quad (3.9)$$

Pour l'effet d'apprentissage basé sur la position de la tâche.

$$P_{g j r i} = P_{g j i} \cdot \left(1 + \sum_{k=1}^{r-1} P_{g i[k]}^{a_i}\right) \quad \forall j, \forall i, \forall g, \forall r \quad (3.10)$$

Pour l'effet d'apprentissage basé sur la somme des temps de traitement.

La technologie de groupe permet d'améliorer l'efficacité des systèmes de production, se traduisant par une réduction drastique des temps de configuration, générant ainsi une

économie significative de temps, de coûts et d'énergie pour les industries.

Comme précisé dans la section 2.2 du chapitre précédent, deux catégories de temps de configuration existe :

- Les temps de configurations indépendant de la séquence :

Dans un problème d'ordonnancement et en présence de la technologie de groupe, on suppose que lorsque deux jobs se suivent de la même famille, il n'y a pas de temps de préparation entre eux. En revanche, dans le cas contraire, un temps de configuration S_g est nécessaire avant que le job j de la famille suivante g ne soit traité. Ce temps S_g est indépendant de la séquence. En d'autres termes le temps de préparation de la famille ne dépend que d'elle et d'aucun autre facteurs.

La date de fin de la tâche j appartenant au groupe g exécuté sur la machine i à la position r est alors égale à :

$$C_{gri} = P_{gjr_i} + S_{gi} \quad \forall j, \forall i, \forall g, \forall r \quad (3.11)$$

Dans ce cas, on intègre au temps de traitement, le temps de préparation qui est indépendant de la séquence.

- Les temps de configurations dépendant de la séquence :

Les temps de configuration sont dits dépendants de la séquence lorsque le temps de préparation d'un job donné dépend du job qui a été préparé sur la ressource avant l'exécution de ce job. Dans le cas où les jobs sont répartis en familles, le temps de configuration pour exécuter la famille g dépend de la famille précédemment traitée h sur la même machine. On le désigne par S_{hgi} . Dans ce contexte, la date de fin de la tâche j appartenant au groupe g exécuté sur la machine i dans la position r est égale à :

$$C_{gri} = P_{gjr_i} + S_{hgi} \quad \forall j, \forall i, \forall g, \forall r \quad (3.12)$$

Le temps de configuration mentionné peut être illustré à travers divers exemples. Dans le domaine de la production de vêtements, prenons le cas de la teinture du fil. Avant de teindre un nouveau groupe de fil, il est impératif de nettoyer la cuve de teinture, la machine responsable du traitement du fil. Le temps de préparation dédié au nettoyage, en vue de la teinture d'une future série, varie en fonction des couleurs du fil à venir et de celles du groupe précédemment teint Salmasi et al. (2010).

Si le groupe précédent était de couleur noire et que le suivant est prévu en blanc, un

nettoyage complet de la cuve de teinture s'impose. En revanche, si le groupe précédent était blanc et que le suivant sera noir, un nettoyage sommaire de la cuve de teinture est suffisant. Par conséquent, le temps de nettoyage est plus court lors du passage de la couleur blanche à la couleur noire que lors du passage inverse.

D'autres exemples similaires se retrouvent dans l'industrie de l'imprimerie, où les temps de nettoyage (réglage) des presses dépendent de la taille du papier et de la couleur de l'encre utilisée dans le groupe précédent. Cela s'applique également à l'industrie automobile au niveau de l'atelier de peinture des voitures, où le temps nécessaire pour peindre un groupe de voitures dépend du groupe précédemment peint.

3.2.4 Description du modèle proposé

Plongeant au cœur de cette étude, nous nous focalisons sur l'ordonnancement d'un ensemble de n tâches réparties en G groupes, opérant sur m machines en série. Nous examinons les subtilités de l'effet d'apprentissage, de la technologie de groupe, et des temps de configuration dépendant de la séquence. Les caractéristiques distinctives de notre système sous investigation comprennent :

- L'atelier contient plusieurs machines disposées en série.
- Les n tâches sont toutes disponibles pour être traitées simultanément dès le début de l'horizon de l'ordonnancement.
- Chaque machine ne peut traiter qu'une seule tâche à la fois.
- Aucune préemption n'est autorisée, c'est-à-dire que le temps de traitement d'une tâche sur une machine ne peut être interrompu.
- Toutes les machines sont disponibles en permanence, c'est-à-dire que les pannes et les actions de maintenance ne sont pas prises en compte.
- Le temps de traitement de la tâche j n'est pas constant, il varie en fonction de l'effet d'apprentissage.
- Un problème de flow shop de permutation "*prmu*" est considéré, c'est-à-dire que les tâches et les groupes sont traités dans la même séquence sur l'ensemble des machines.
- L'objectif est de déterminer non seulement la séquence des tâches mais aussi la séquence des groupes, afin de minimiser simultanément le makespan et la consommation d'énergie du système.

3.3 Flow shop avec effet d'apprentissage pour une optimisation mono-objectif

Le problème de flow shop a fait l'objet d'études approfondies par divers chercheurs, chacun apportant des hypothèses variées et proposant différentes méthodes de résolution. L'intérêt croissant pour ce problème s'explique par sa présence répandue dans la plupart des systèmes de production. L'introduction de l'effet d'apprentissage insufflé une nouvelle perspective à ce type de problème, offrant ainsi la possibilité de le rapprocher au mieux du contexte industriel réel.

Ainsi, nos travaux ont initialement porté sur la résolution d'un problème de flow shop avec effet d'apprentissage, visant à minimiser le makespan de manière mono-objectif, représentant la date de fin de la dernière tâche exécutée sur la dernière machine. Avant d'aborder la formulation du modèle de base que nous proposons, il est essentiel de clarifier l'effet d'apprentissage et les différents taux d'apprentissage pris en compte dans cette étude.

Comme exposé dans la section précédente (section 3.2), notre étude se concentre sur un effet d'apprentissage basé sur la position, appliqué spécifiquement à un problème de flow shop.

3.3.1 Approche intégrative de l'effet d'apprentissage

La théorie de l'apprentissage suppose un environnement de production de masse où des produits identiques sont traités consécutivement. Par conséquent, les temps de traitement des opérations individuelles seraient identiques s'ils n'étaient pas influencés par l'apprentissage (Biskup, 2008). L'apprentissage est souvent défini à l'aide de la formule de puissance :

$$P_{[k]} = P_{[1]} \cdot k^a \quad (3.13)$$

Avec $P_{[k]}$ symbolisant le temps de traitement requis pour la kème unité de la quantité de production cumulée k ; $P_{[1]}$ est le temps de traitement de la première unité et $a = \log_2.LR \leq 0$ est l'indice d'apprentissage dépendant du taux d'apprentissage LR . Ce contexte peut être clarifié comme suit : Le taux d'apprentissage est défini pour décrire l'effet d'apprentissage à chaque redoublement de la production :

$$P_{[2k]} = P_{[k]} \cdot LR \quad (3.14)$$

Plus le taux d'apprentissage est faible, plus les effets d'apprentissage sont importants. La substitution des temps de traitement par l'équation (3.13) conduit à la formule suivante :

$$LR = \frac{P_{[2k]}}{P_{[k]}} = \frac{P_{[1]} \cdot 2k^a}{P_{[1]} \cdot k^a} = 2^a$$

$$a = \frac{\log LR}{\log 2} = \log_2.LR$$
(3.15)

Ainsi, lorsque LR=90%, $a = \log_2.0,9 = -0,152$ s'applique. Supposons que le temps de traitement du premier produit est de 100, alors $P_{[k]} = 100.k^{-0.152}$ décrit la courbe d'apprentissage en fonction du nombre de produits cumulés. Comme la production cumulée est généralement mesurée par des nombres entiers, la courbe d'apprentissage consisterait en des points simples qui sont reliés entre eux dans la figure 4.5. Le taux d'apprentissage de 100% (pas d'effet d'apprentissage), et le taux d'apprentissage de 70%, de 80% et de 90% sont aussi représentés.

Donc, en observant la figure 4.5, il est notable qu'un taux d'apprentissage de 70% entraîne une réduction plus significative du temps de traitement par rapport aux taux de 80% et 90%. Ainsi, des taux moins élevés sont associés à une réduction plus importante du temps de traitement.

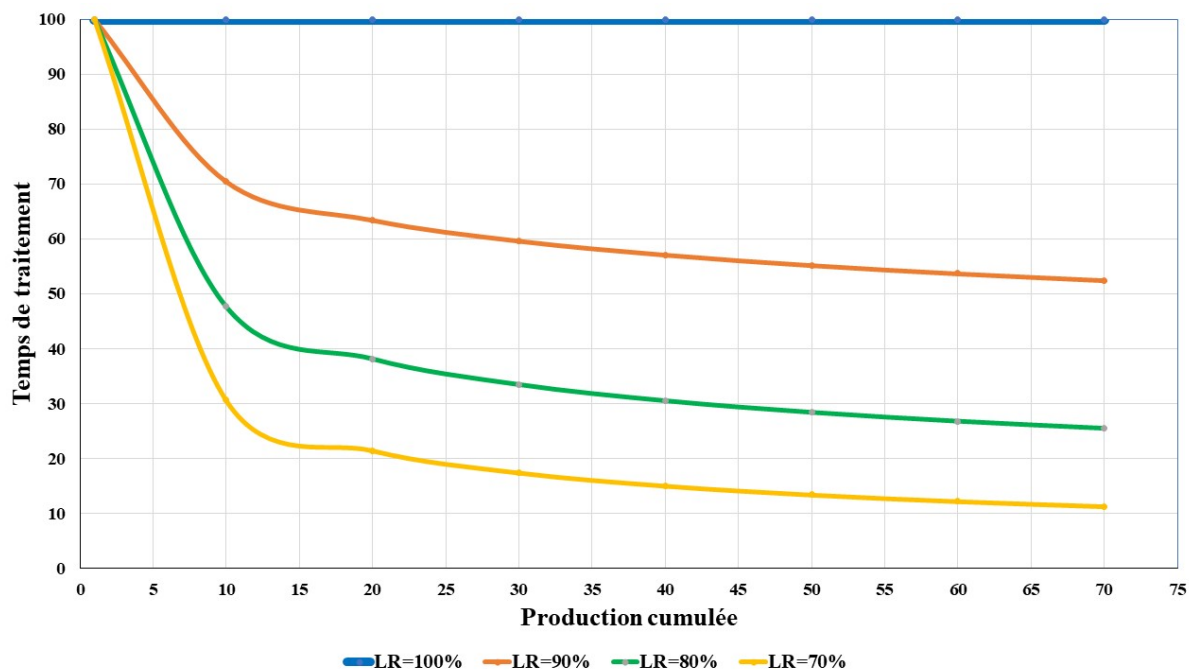


Figure 3.1: Représentation de la courbe d'apprentissage avec différents taux d'apprentissage.

L'impact des divers taux d'apprentissage sur l'exécution des tâches à différentes positions sur une machine peut également être illustré à l'aide du tableau 3.1.

Tableau 3.1: Impact du taux d'apprentissage sur le traitement des tâches

Taux d'apprentissage : LR	l'indice d'apprentissage : a	Effet d'apprentissage pour différentes positions sur la machine			
		$l = 1$	$l = 2$	$l = 3$	$l = 4$
70%	$a = \log_2 0.70 = -0.514$	$l^a = 1^{-0.514} = 1$	$l^a = 2^{-0.514} = 0.7$	0.56	0.49
90%	$a = \log_2 0.90 = -0.152$	$l^a = 1^{-0.152} = 1$	$l^a = 2^{-0.152} = 0.9$	0.84	0.81
100%	$a = \log_2 0.1 = 0$	$l^a = 1^0 = 1$	$l^a = 2^0 = 1$	1	1

En examinant le tableau 3.1, il devient évident qu'opter pour un taux d'apprentissage plus bas permet d'anticiper une réduction plus significative des temps de traitement, conduisant ainsi à une amélioration du rendement. On peut également déduire que l'effet d'apprentissage est négligeable lorsque le taux d'apprentissage est égal à 100%, ce qui peut être assimilé dans l'industrie à un changement fréquent de l'employé.

À partir de l'équation (3.13), on peut conclure qu'il n'y aurait pas d'effet d'apprentissage pour la première position ($l = 1$) sur les machines. Il est également remarquable que plus la tâche est exécutée ultérieurement dans la séquence, plus son temps opératoire diminue, comme illustré par les positions $l = 3$ et $l = 4$, pour lesquelles on obtient un effet d'apprentissage de 0.56 et 0.49 respectivement, avec un taux d'apprentissage de 70%.

3.3.2 Formulation du problème

En se basant sur les modèles généraux présentés dans la section 3.2, une première formulation du problème est proposée. Afin de déterminer une solution optimale pour le problème de flow shop avec effet d'apprentissage, nous proposons un modèle mathématique linéaire en nombre entier mixte. Ce modèle repose sur des variables de position, telles que précisées dans la section 3.2, et vise à minimiser le makespan. Cette modélisation s'inspire du modèle proposé par Quang Chieu TA (2015), en maintenant les mêmes hypothèses décrites dans la section de la problématique.

Pour présenter ce modèle mathématique du problème susmentionné, nous introduisons différentes notations, détaillées comme suit :

Les paramètres :

- m : le nombre de machines
- n : le nombre de tâches
- i : l'indice des machines, où $i = 1, \dots, m$
- j : l'indice de la tâche, où $j = 1, \dots, n$
- k : l'indice de la position, où $k = 1, \dots, n$
- P_{ij} : le temps de traitement sans effet d'apprentissage de la tâche j sur la machine i
- $a^i = \log_2.LR \leq 0$: l'indice d'apprentissage de la machine i en fonction du taux d'apprentissage LR

Les variables de décision :

- C_{ik} : la date de fin de la tâche se trouvant dans la position k sur la machine i .
- X_{jk} : la variable binaire qui va prendre la valeur 1 si la tâche j est exécuté dans la position k , 0 sinon
- P_{ijk} : le temps de traitement avec effet d'apprentissage de la tâche j exécuté dans la position k sur la machine i

Le modèle mathématique :

En utilisant les notations fournies ci-dessus, la fonction objectif ainsi que les contraintes du modèle mathématique proposé pour le problème de flow shop avec effet d'apprentissage sont présentées ci-dessous :

- **Fonction objectif :**

$$\text{Minimiser } C_{mn} \quad (3.16)$$

- **Contraintes :**

$$\sum_{k=1}^n X_{jk} = 1, \quad \forall j = 1, \dots, n \quad (3.17)$$

$$\sum_{j=1}^n X_{jk} = 1, \quad \forall k = 1, \dots, n \quad (3.18)$$

$$P_{ijk} = P_{ij}k^{a_i}, \quad \forall j = 1, \dots, n \quad \forall i = 1, \dots, m \quad \forall k = 1, \dots, n \quad (3.19)$$

$$C_{ik} \geq C_{i-1k} + \sum_{j=1}^n X_{jk}P_{ijk}, \quad \forall i = 2, \dots, m \quad \forall k = 1, \dots, n \quad (3.20)$$

$$C_{ik} \geq C_{ik-1} + \sum_{j=1}^n X_{jk}P_{ijk}, \quad \forall i = 1, \dots, m \quad \forall k = 2, \dots, n \quad (3.21)$$

$$C_{11} = \sum_{j=1}^n X_{j1}P_{1j1} \quad (3.22)$$

$$C_{i1} \geq C_{i-11} + \sum_{j=1}^n X_{j1}P_{ij1}, \quad \forall i = 2, \dots, m \quad (3.23)$$

$$C_{1k} \geq C_{1k-1} + \sum_{j=1}^n X_{jk}P_{1jk}, \quad \forall k = 2, \dots, n \quad (3.24)$$

$$C_{ik}, P_{ijk} \geq 0, \quad \forall i = 1, \dots, m \quad \forall k = 1, \dots, n, \quad \forall j = 1, \dots, n \quad (3.25)$$

$$X_{jk} \in \{0, 1\} \quad (3.26)$$

La fonction objectif (3.16) minimise le makespan, représenté par la date de fin de la tâche exécuté dans la dernière position sur la dernière machine. Les contraintes (3.17) et (3.18)

assurent que chaque tâche doit occuper une seule position et dans chaque position, il peut y avoir qu'une seule tâche. La contrainte (3.19) détermine le temps de traitement de la tâche en fonction de l'effet d'apprentissage. La contrainte (3.20) stipule que le traitement d'une tâche à la position k sur la machine i ne peut débuter qu'après la fin de son exécution sur la machine précédente. La contrainte (3.21) garantit que l'exécution d'une tâche ne peut avoir lieu qu'après la fin de celle qui la précède sur la même machine. La contrainte (3.22) est introduite pour déterminer la date de fin de la première tâche ordonnancée sur la première machine. La contrainte (3.23) calcule la date de fin de la tâche placée en première position sur la machine i par rapport à la machine précédente $i - 1$ et à la même position. La contrainte (3.24) permet d'obtenir les dates de fin des tâches sur la première machine. La contraintes (3.25) assure que la date de fin et le temps de traitement ne peuvent être négatifs. La contrainte (3.26) représente une variable binaire qui prend la valeur 1 si la tâche est ordonnancée à la position k et 0 sinon.

3.4 Ordonnancement de groupes avec temps de configuration dépendant de la séquence

Cette section vise à étendre le modèle principale proposé. Dans ce modèle, une instance du problème comprend un ensemble de tâches $j = \{1, \dots, N_g\}$ constituant un ensemble de groupes $g = \{1, \dots, G\}$ exécutés sur des machines $i = \{1, \dots, m\}$, la preemption n'est pas autorisée et chaque machine ne peut traité qu'une seule tâche à la fois. L'ordre des tâches est fixe et le problème implique la prise en compte simultanée de l'effet d'apprentissage et du temps de configuration dépendant de la séquence.

Le temps de traitement de la tâche j appartenant au groupe g exécuté sur la machine i , P_{gji} est en fonction de l'effet d'apprentissage a^i , qui a son tour varie d'une machine i à une autre, représentant ainsi l'affectation de différents employés possédant différentes capacité sur plusieurs machines.

3.4.1 La technologie de groupe : Un concept innovant

La technologie de groupe, dans le contexte de l'ordonnancement, représente une approche innovante visant à optimiser les opérations en regroupant les tâches similaires ou liées. Elle s'appuie sur l'idée de traiter certains éléments en tant qu'unités cohérentes plutôt qu'individuellement, ce qui peut conduire à des améliorations significatives en termes d'efficacité, de réduction des temps de configuration, et de meilleure utilisation des ressources. Ce concept offre une perspective stratégique pour l'optimisation des processus de production, notamment dans des environnements où des groupes spécifiques de tâches partagent des caractéristiques similaires.

Cette méthodologie présente des avantages significatifs. Cependant, pour qu'elle soit efficace au sein d'un système de production, il est nécessaire de procéder à une analyse approfondie du système étudié et à une détermination précise des temps de configuration au préalable.

- Description des temps de configuration :

La première phase de la méthodologie consiste en l'estimation du temps de préparation. Ainsi, une analyse des facteurs influant sur le temps de configuration s'avère nécessaire. Chaque modèle est caractérisé par une configuration spécifique du système productif. Lors d'un changement de modèle dans une installation, que ce soit dans le secteur automobile, la production de pièces métalliques ou encore la fabrication de carreaux en céramique, des opérations de réglage doivent être effectuées pour modifier la configuration des systèmes. Généralement, le temps de configuration peut être défini comme :

- ◇ Le réglage des machines, comme c'est le cas pour une presse, qui nécessite la mise en place de nouvelles matrices lors de modifications dans la taille et la forme des produits.
- ◇ La modification de la disposition des machines, voire l'ajout de nouvelles machines, surtout lorsqu'elles sont disposées sur un convoyeur.
- ◇ Le remplacement des produits bruts nécessaires au nouveau modèle (couleurs, fixateurs, teintes), ainsi que leur ajustement lorsque cela est requis.
- ◇ Le changement d'outils, une opération fréquente dans les ateliers de production, pouvant impliquer le remplacement de forets, de moules ou d'autres équipements spécifiques.

Les opérations de configuration présentent parfois des caractéristiques liées à des relations de précedence entre et au sein de chaque cellule de fabrication. Cependant, dans le cadre de cette étude, les relations de précedence demeurent identiques pour chaque produit, seule leur durée ainsi que la présence ou l'absence d'opérations varient.

- La formation des familles de produits :

Les familles de pièces, définies par des attributs de fabrication et de conception, ne sont pas nécessairement connectées. De nouvelles approches, proposées en remplacement de l'approche traditionnelle basée sur les attributs des pièces (Andrés et al., 2005), reposent sur l'utilisation d'un coefficient de similarité entre les pièces. La définition de ce coefficient s'appuie sur trois affirmations clés. La première considère la facilité d'ajustement

des machines et la modification de la disposition d'une pièce A à une pièce B (et vice versa) comme indicateur d'appartenance à la même famille. Ensuite, la deuxième affirmation repose sur le fait que lorsque le temps de configuration à partir de n'importe quelle pièce vers A ou B est similaire, A et B peuvent être regroupées dans la même famille. La troisième déclaration est une variante inversée de la deuxième, indiquant que des pièces A et B peuvent également appartenir à la même famille lorsque le temps de configuration depuis la production de A ou B vers n'importe quelle autre pièce est similaire.

La technique de formation de groupes ou de familles elle-même implique une étude approfondie et une analyse détaillée du système. Dans cette thèse, cet aspect a été pris en compte en adoptant une constitution de familles basée sur les attributs des tâches. Pour illustrer l'efficacité de cette technique, nous examinons un exemple lié à un problème de flow shop. Nous comparons une situation sans adaptation de la technologie de groupe à une situation où cette dernière est prise en considération.

3.4.2 Hypothèses et notations relatif à l'ordonnancement de groupe

En complément des hypothèses formulées lors de l'élaboration du modèle précédent, d'autres suppositions spécifiques à la notion de groupe s'ajoutent, détaillées comme suit :

- Le temps de configuration du groupe g dépend du groupe précédemment traité (i.e, le groupe h) exécuté sur la même machine (s_{hgi}).
- le temps de préparation entre les jobs d'un même groupe est négligeable et est inclus dans le temps de traitement.
- L'hypothèse de la technologie de groupe est appliquée, ce qui signifie qu'une fois que l'exécution du groupe a commencé, elle ne peut pas être interrompue par un autre groupe.
- Les ajustements sont prévus à l'avance, ce qui indique que la configuration de la machine pour un groupe peut être initiée avant même qu'un job du groupe ne soit disponible.
- Un groupe de référence ou groupe fictif est inclus dans le modèle pour calculer le temps de préparation requis pour le premier groupe sur chaque machine.

Avant d'approfondir l'explication et la modélisation du modèle proposé pour le problème de flowshop avec groupe et temps de changement dépendant de la séquence 'FSDGS', commençons par introduire les ensembles, données et variables de décision nécessaires à la formulation du problème en tant que MILP. Nous réutiliserons certaines notations fournies dans la section (3.3).

Les indices :

- h, g : l'indice des groupes, où $g = 1, \dots, G$
- j : l'indice des jobs, où $j = 1, \dots, N_g$
- l : l'indice des positions, où $l = 1, \dots, N_g$
- i : l'indice des machines, où $i = 1, \dots, m$

Les paramètres :

- N_g : le nombre total de jobs dans le groupe g
- G : le nombre total de groupes
- m : le nombre total de machines
- P_{gji} : le temps de traitement basique du job j appartenant au groupe g sur la machine i
- s_{hgi} : le temps de configuration pour traiter le groupe g sur la machine i , si le groupe h est le groupe précédent
- LR : le taux d'apprentissage
- a^i : l'indice d'apprentissage ($a = \log_2.LR \leq 0$)
- l^{a_i} : l'effet d'apprentissage dans l'ème position sur la machine i
- M : un grand nombre positif

Les variables de décisions :

- C_{gli} : la date de fin du job exécuté en position l dans le groupe g sur la machine i
- F_{gi} : la date de fin du groupe g sur la machine i
- S_{gi} : la date de début du groupe g sur la machine i
- P_{gjl} : le temps de traitement actuel du job j exécuté en position l dans le groupe g sur la machine i
- X_{gjl} : la variable de décision binaire, qui prend la valeur 1 si le job j occupe la position l dans le groupe g , 0 sinon
- U_{hg} : la variable de décision binaire, qui prend la valeur 1 si le groupe g est traité immédiatement après le groupe h , 0 sinon
- $Cmax$: le makespan

3.4.3 Le modèle mathématique et les contraintes

Cette partie est dédiée à la présentation de la fonction objective à minimiser et des contraintes à satisfaire, lesquelles peuvent être formulées comme suit :

- **Fonction objectif :**

$$\text{Minimiser } Cmax \quad (3.27)$$

- **Contraintes :**

$$\sum_{l=1}^{N_g} X_{gjl} = 1, \quad \forall j = 1, \dots, N_g \quad \forall g = 1, \dots, G \quad (3.28)$$

$$\sum_{j=1}^{N_g} X_{gjl} \leq 1, \quad \forall l = 1, \dots, N_g \quad \forall g = 1, \dots, G \quad (3.29)$$

$$P_{gjl} = P_{gji} l^{\alpha_i}, \quad \forall j, l = 1, \dots, N_g \quad \forall i = 1, \dots, m \quad \forall g = 1, \dots, G \quad (3.30)$$

$$C_{gli} \geq C_{gl-1i} + \sum_{j=1}^{N_g} X_{gjl} P_{gjl}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall l = 2, \dots, N_g \quad (3.31)$$

$$C_{gli} \geq C_{gli-1} + \sum_{j=1}^{N_g} X_{gjl} P_{gjl}, \quad \forall g = 1, \dots, G \quad \forall i = 2, \dots, m \quad \forall l = 1, \dots, N_g \quad (3.32)$$

$$C_{g1i} \geq S_{gi} + \sum_{j=1}^{N_g} X_{gjl} P_{gji}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (3.33)$$

$$F_{gi} \geq C_{gN_g i}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (3.34)$$

$$S_{gi} \geq F_{hi} + s_{hgi} - M(1 - U_{hg}), \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall h = 0, \dots, G \quad (3.35)$$

$$\sum_{h=0, h \neq g}^G U_{hg} = 1, \quad \forall g = 1, \dots, G \quad (3.36)$$

$$\sum_{g=1, h \neq g}^G U_{hg} \leq 1, \quad \forall h = 1, \dots, G \quad (3.37)$$

$$\sum_{g=1}^G U_{0g} = 1 \quad (3.38)$$

$$U_{hg} + U_{gh} \leq 1 \quad (3.39)$$

$$Cmax \geq C_{gN_g m}, \quad \forall g = 1, \dots, G \quad (3.40)$$

$$C_{gli}, F_{gi}, S_{gi}, P_{gji} \geq 0 \quad (3.41)$$

$$X_{gjl}, U_{hg} \in \{0, 1\} \quad (3.42)$$

avec $C_{g0} = F_{0i} = 0$

La fonction objectif vise à minimiser le makespan, comme indiqué dans l'équation (3.27). Les contraintes (3.28) et (3.29) précisent que chaque job occupe précisément une position dans le groupe auquel il appartient, et qu'à chaque position d'un groupe correspond un seul job de ce groupe. La contrainte (3.30) est nécessaire pour calculer le temps de traitement des jobs en considérant l'effet d'apprentissage basé sur la position de cette dernière. L'exécution d'un job sur une machine ne peut commencer que si l'exécution d'un autre job à la position précédente est terminée. C'est la raison pour laquelle la contrainte (3.31) est incorporée au modèle. La contrainte (3.32) est également intégrée pour assurer que l'exécution du job sur la machine i débute après son exécution sur la machine précédente. La contrainte (3.33) prend en charge le fait que le temps de traitement du job dans la première position de chaque groupe commence après le point de départ du groupe. Le temps d'achèvement d'un groupe est supérieur au temps d'achèvement du job dans la dernière position de ce même groupe, la contrainte (3.34) corrobore ce fait. La contrainte (3.35) stipule que la date de début du groupe g sur la machine i doit être supérieure à la date de fin du groupe précédemment traité, le groupe h , ajoutée au temps de configuration entre les deux groupes successifs, s_{hgi} . Chaque groupe doit avoir exactement un groupe qui le précède, la contrainte (3.36) garantit cette affirmation. La contrainte (3.37) est incorporée au modèle afin d'assurer que chaque groupe a au plus un groupe successeur. Le groupe fictif est pris en compte comme le premier groupe traité sur toutes les machines, et la contrainte (3.38) est incorporée dans le modèle pour étayer cette observation. Chaque groupe peut être traité avant ou après un autre groupe. Ceci est exprimé dans la contrainte (3.39). La contrainte (3.40) définit le makespan, qui est supérieur à la date de fin du job traité en dernière position dans le groupe sur la dernière machine. Les contraintes (3.41) et (3.42) définissent les variables de décision.

3.5 Minimisation du Cmax en tenant compte des contraintes énergétiques

L'atténuation du changement climatique s'impose comme l'un des plus grands défis du début de ce millénaire, suscitant des débats intenses sur ses origines, ses sources, ses impacts potentiels et les stratégies d'adaptation nécessaires. Malgré les efforts déployés et les résolutions déployées par les gouvernements, axée sur la lutte contre le changement climatique, les données indiquent une augmentation continue de l'utilisation mondiale de combustibles fossiles, tels que le charbon, le pétrole et le gaz, depuis environ 1850 jusqu'à nos jours.

Cette prédominance persistante des combustibles fossiles dans la consommation mon-

diale d'énergie demeure inébranlable, même après l'accord de Paris. Les analyses révèlent une croissance constante des émissions de CO₂, établissant un nouveau record en 2022. Entre 2015 et 2022, les émissions mondiales ont grimpé de 35,6 à 38,2 gigatonnes de CO₂, soit une augmentation de plus de 7,2%. Alarmante, cette hausse est principalement attribuée à la combustion d'énergie fossile, représentant 88% des émissions. Ces constatations cruciales ont été soulevées lors de la Conférence de 2023 à Dubaï.

Parallèlement, les entreprises manufacturières et d'autres secteurs industriels ont traditionnellement axé leurs objectifs sur une production rentable, souvent au détriment de considérations environnementales. Cependant, la conjonction de la pression croissante du changement climatique et de l'augmentation des coûts énergétiques souligne l'impératif d'intégrer davantage les paramètres environnementaux dans les pratiques industrielles.

Le constat de l'épuisement des énergies fossiles incite particulièrement les industries manufacturières à réduire leur consommation d'énergie, compte tenu de leur responsabilité dans près de 50% de la consommation énergétique mondiale annuelle. La conférence de Dubaï a mis en lumière l'importance cruciale de réduire les émissions de carbone, soulevant ainsi des enjeux et des défis majeurs. Face à cette réalité, la communauté internationale doit redoubler d'efforts et de résolutions pour contrer ce fléau et favoriser un développement plus durable.

Dans cette section, notre attention se porte sur la minimisation du makespan C_{max} , qui représente le temps d'achèvement de la dernière tâche sur la dernière machine m dans le contexte du problème de FSDGS. Nous prenons en compte l'effet d'apprentissage au niveau des machines nécessitant l'intervention d'un opérateur, tandis que pour les machines automatisées, nous ajustons les vitesses de traitement. L'objectif principal est de définir le séquençement des tâches au sein des groupes, tout en considérant également le séquençement des groupes, et ce, en tenant compte des émissions de CO₂ qui ne doivent pas dépasser un seuil Q_{max} prédéterminé.

Ainsi, pour atteindre cet objectif, un modèle de programmation linéaire mixte en nombres entiers (MILP) a été développé pour le problème en question. Par ailleurs, nous fournissons un exemple numérique illustrant l'impact de l'effet d'apprentissage et des variations de vitesses sur l'ordonnancement des tâches.

3.5.1 Formulation du problème

En se basant sur les modèles proposés dans la section précédente, cette partie aborde le problème d'atelier de type flow shop avec des contraintes d'émission de CO₂. Comme mentionné précédemment, une instance du problème FSDGS se compose d'un ensemble de groupes $g = \{1, \dots, G\}$ constitués d'un ensemble de tâches $j = \{1, \dots, N_g\}$ traitées sur un ensemble de machines $i = \{1, \dots, m\}$. Chaque tâche j sur la machine i a un temps

de traitement p_{ij} variant en fonction du type de machine.

Au niveau des machines entièrement automatisées, le temps de traitement varie en fonction des vitesses sélectionnées pendant le traitement. En revanche, pour les machines nécessitant l'intervention d'opérateurs, le temps de traitement dépend de l'effet d'apprentissage. Un ensemble de vitesses V est défini, où une tâche $j \in N_g$ traitée sur la machine $i \in m$ à la vitesse $v \in V$ a un temps de traitement p_{ijv} et une émission de CO2 q_{jiv} associée. De plus, un seuil Q_{max} sur les émissions totales de CO2 est imposé à tout moment de l'ordonnancement.

À cet effet, nous proposons un modèle mathématique linéaire en nombre entier mixte, intégrant les paramètres et notations suivants, en plus de ceux présentés dans la section précédente, tout en maintenant les hypothèses qui y sont décrites :

Les indices :

- v : l'indice des vitesses, où $v = 1, \dots, V$

Les paramètres :

- V : le nombre total de vitesses
- $P_{g j i v}$: le temps de traitement basic du job j appartenant au groupe g sur la machine i à la vitesse v
- $q_{i v}$: la puissance consommé par unité de temps durant la phase de production sur la machine i à la vitesse v
- Sep_i : la puissance de configuration par unité de temps sur la machine i
- $I_{g i}$: le temps d'attente du groupe g sur la machine i
- $I p_i$: la puissance consommé durant l'inactivité de la machine i
- Fe : facteur d'émission de CO2 par unité d'énergie qu'il consomme (g/kj)
- Cp : les émissions carbone durant la phase de production
- Cs : les émissions carbone durant la phase de configuration
- $Cidl$: les émissions carbone durant la phase d'inactivité de la machine

Les variables de décisions :

- $P_{g j i l v}$: le temps de traitement actuel du job j exécuté en position l dans le groupe g sur la machine i à la vitesse v
- $X_{g j l v}$: la variable de décision binaire, qui prend la valeur 1 si le job j est exécuté dans la position l à la vitesse v , 0 sinon

Le modèle mathématique :

• **Fonction objectif :**

$$\text{Minimiser } C_{max} \quad (3.43)$$

• **S.C :**

Les contraintes (3.35), (3.36), (3.37), (3.38) et (3.39)

$$\sum_{l=1}^{N_g} \sum_{v=1}^V X_{gjl v} = 1, \quad \forall j = 1, \dots, N_g \quad \forall g = 1, \dots, G \quad (3.44)$$

$$\sum_{j=1}^{N_g} \sum_{v=1}^V X_{gjl v} \leq 1, \quad \forall l = 1, \dots, N_g \quad \forall g = 1, \dots, G \quad (3.45)$$

$$P_{gjl v} = P_{gji v} l^{a_i}, \quad \forall j, l = 1, \dots, N_g \quad \forall i = 1, \dots, m \quad \forall g = 1, \dots, G \quad \forall v = 1, \dots, V \quad (3.46)$$

$$C_{gli} \geq C_{gl-1i} + \sum_{j=1}^{N_g} \sum_{v=1}^V X_{gjl v} P_{gjl v}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall l = 2, \dots, N_g \quad (3.47)$$

$$C_{gli} \geq C_{gli-1} + \sum_{j=1}^{N_g} \sum_{v=1}^V X_{gjl v} P_{gjl v}, \quad \forall g = 1, \dots, G \quad \forall i = 2, \dots, m \quad \forall l = 1, \dots, N_g \quad (3.48)$$

$$C_{g1i} \geq S_{gi} + \sum_{j=1}^{N_g} \sum_{v=1}^V X_{gjl v} P_{gjl v}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (3.49)$$

$$F_{gi} \geq C_{gN_g i}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (3.50)$$

$$C_p = \sum_{v=1}^V \sum_{g=1}^G \sum_{i=1}^m \sum_{j=1}^{N_g} \sum_{l=1}^{N_g} q_{iv} P_{gjl v} X_{gjl v} F_e \quad (3.51)$$

$$C_s = \sum_{h=0}^G \sum_{g=1}^G \sum_{i=1}^m s_{hgi} U_{hg} sep_i F_e \quad (3.52)$$

$$C_{idl} = \sum_{g=1}^G \sum_{i=1}^m I_{gi} I_{pi} F_e \quad (3.53)$$

$$C_p + C_s + C_{idl} \leq Q_{max} \quad (3.54)$$

$$C_{max} \geq F_{gm}, \quad \forall g = 1, \dots, G \quad (3.55)$$

$$C_{gli}, F_{gi}, S_{gi}, P_{gjl v} \geq 0 \quad (3.56)$$

$$X_{gjl v}, U_{hg} \in \{0, 1\} \quad (3.57)$$

la fonction objectif est de minimiser le C_{max} présenté dans l'équation (3.43). Les contraintes

(3.44) et (3.45) spécifient que chaque tâche soit exécutée à une seule vitesse et occupe exactement un emplacement dans les groupes auxquels elle appartient et que chaque emplacement d'un groupe est attribué à l'une des tâches de ce groupe. La contrainte (3.46) est nécessaire pour calculer le temps de traitement des tâches en considérant l'effet d'apprentissage basé sur la position de cette dernière. L'exécution d'une tâche sur une machine peut démarrer si l'exécution d'une tâche dans l'emplacement précédent est terminée, la contrainte (3.47) est incorporée au modèle pour cette raison. La contrainte (3.48) est intégrée au modèle pour garantir que l'exécution de la tâche sur la machine commence après son exécution sur la machine précédente. La contrainte (3.49) calcule la date de fin dans le premier emplacement de chaque groupe au niveau de chaque machine. Le temps d'achèvement d'un groupe est supérieur au temps d'achèvement de la tâche dans le dernier emplacement. La contrainte (3.50) est incorporée au modèle pour prendre en charge ce fait. Les contraintes (3.51), (3.52) et (3.53) expriment les émissions de CO₂ au niveau de la production, des temps de configuration et des temps d'inactivité de la machine respectivement. Les émissions de CO₂ ne doivent pas dépasser la valeur Q_{max} , la contrainte (3.54) stipule ce fait. La contrainte (3.55) assure que le C_{max} est égale à la date de fin du dernier groupe sur la dernière machine. Les contraintes (3.56) et (3.57) définissent les variables de décision.

3.5.2 Évaluation de l'efficacité de notre approche

Nous avons testé notre approche selon deux scénarios pour en démontrer son efficacité. Dans ces scénarios, nous avons supposé que la première et la dernière machine peuvent exécuter les tâches en variant les vitesses. Dans ce cas particulier, nous avons pris en compte trois vitesses (lente, rapide et moyenne). Lorsqu'une tâche est traitée à une vitesse élevée, son temps de traitement diminue, tandis que les émissions de CO₂ augmentent. Autrement dit, à mesure que la vitesse v augmente, P_{gjjv} diminue et C_p augmente.

Pour les deuxième et troisième machines, des opérateurs manipulent la pièce semi-finie en fonction de leur dextérité. Ainsi, la prise en compte de l'effet d'apprentissage s'impose. Selon les travaux de Pargar et al. (2018), nous avons considéré un taux d'apprentissage de 70% et de 90%.

3.5.2.1 Paramètres et méthodologie d'analyse des instances de l'étude

Les instances se composent de 2 à 4 groupes, avec un nombre de tâches variant entre 2 et 10 dans chaque groupe. Les temps de changement varient d'une machine à l'autre, c'est-à-dire que le temps de configuration sur la première machine peut être supérieur, inférieur ou égal à celui sur la seconde machine. Le temps de réglage évolue uniformément dans l'intervalle $[1, 40]$, tandis que le temps de traitement suit une distribution uniforme

d'entiers dans l'intervalle $[1, 60]$. Le facteur d'émissions de CO2 est fixé à 0,1524 g CO2/kJ (Fang et al., 2013).

Tableau 3.2: Les données des deux scénarios

			Scénario 1							
Machines			M1	M2	M3	M4				
No Groupes	No Tâches									
		vitesse	$v1$	$v2$	$v3$	$\forall v$	$\forall v$	$v1$	$v2$	$v3$
2	(2,3)	q_{iv}	8	2	1	1.5	3	7	4	2
		Sep_i		10		5	4		8	
	(5,8)	Ip_i		0.5		0.5	1		1.5	
4	(4,7,3,9)	q_{iv}	9	6	2	3	2.5	8	5	1.5
		Sep_i		9		8	7		10	
	(6,10,8,3)	Ip_i		1.5		0.5	1.5		3	
Effet d'apprentissage a^i			0			-0.514	-0.152		0	

			Scénario 2							
Machines			M1	M2	M3	M4				
No Groupes	No Tâches									
		vitesse	$v1$	$v2$	$v3$	$\forall v$	$\forall v$	$v1$	$v2$	$v3$
2	(2,3)	q_{iv}	8	2	1	1.5	3	7	4	2
		Sep_i		10		5	4		8	
	(5,8)	Ip_i		0.5		0.5	1		1.5	
4	(4,7,3,9)	q_{iv}	9	6	2	3	2.5	8	5	1.5
		Sep_i		9		8	7		10	
	(6,10,8,3)	Ip_i		1.5		0.5	1.5		3	
Effet d'apprentissage a^i			0			-0.152	-0.514		0	

3.5.2.2 Analyse des Résultats Expérimentaux

L'objectif de cette étude est de démontrer l'importance de l'effet d'apprentissage sur le séquençage des tâches et dans la minimisation de la fonction objectif. Dans ce cas, il est pris en compte uniquement sur les machines 2 et 3. De plus, l'étude vise à mettre en évidence l'impact des émissions de CO2 sur la fonction objectif, le séquençage des tâches, et le choix des différentes vitesses de traitement. Ces spécifications sont utilisées pour générer deux scénarios, comme représenté dans le tableau 2. Ensuite, chaque instance est résolue à l'aide du solveur IBM ILOG CPLEX (version 12.8).

Dans cette section, nous présentons les résultats expérimentaux des deux scénarios générés précédemment, comme indiqué dans le tableau 3. Pour chaque scénario, nous

fournissons les valeurs de chaque fonction objectif et de la variable Q_{\max} , qui représente la valeur maximale des émissions de CO2 à ne pas dépasser. Nous avons varié la valeur de Q_{\max} pour évaluer son impact sur la fonction objectif, ainsi que sur les vitesses des machines 1 et 4. Dans le cas des machines 2 et 3, des effets d'apprentissage différents ont été pris en compte.

Dans le premier scénario avec les instances composés de deux groupes, lorsque Q_{\max} est à sa valeur maximale, le makespan est faible, indiquant l'utilisation des vitesses maximales sans considération des émissions de CO2. Cependant, à mesure que la valeur de Q_{\max} diminue, le makespan augmente jusqu'à ce que Q_{\max} atteigne environ 50% de sa valeur maximale. À ce point, le solveur ne fournit plus de résultats, car il n'y a pas de compromis satisfaisant entre la fonction objectif et le Q_{\max} . L'augmentation du makespan se traduit par le choix de vitesses moyennes ou lentes. Au niveau des machines 2 et 3, les tâches qui ont un temps de traitement élevé sont séquencées en dernière position pour une diminution significative.

Dans le scénario 2 ($a_3 > a_2$), les tâches ayant un temps de traitement élevé au niveau de M3 sont favorisées et exécutées en dernier. Lorsque Q_{\max} diminue de 50%, le makespan obtenu pour le scénario 2 est moins important que dans le scénario 1, montrant que la considération d'un effet d'apprentissage élevé au niveau de M3 donne de meilleurs résultats.

En augmentant la taille des instances, les émissions de CO2 deviennent plus élevées, mais en intégrant Q_{\max} , cette valeur diminue, ce qui entraîne également une diminution du makespan. Cependant, il est crucial de trouver un compromis entre la minimisation du makespan et des émissions de CO2, car privilégier la réduction maximale des émissions conduirait à l'utilisation de la vitesse la plus faible, doublant ainsi le makespan, ce qui n'est pas souhaitable pour respecter les délais de livraison aux clients.

Tableau 3.3: les résultats des deux scénarios

Instances	Scénario 1							
	C_{\max}	Q_{\max}	C_{\max}	Q_{\max}	C_{\max}	Q_{\max}	C_{\max}	Q_{\max}
(2,3)	186.91	≥ 400	246.91	300	329.29	250	/	≤ 200
(5,8)	323.01	≥ 763	423.51	570	707.91	460	/	≤ 450
(4,7,3,9)	583.13	≥ 1645.2	691	1234	883.12	1100	/	≤ 1040
(6,10,8,3)	599.59	≥ 1900.8	736.75	1425	1050.1	1200	/	≤ 1150

Scénario 2								
Instances	Cmax	Qmax	Cmax	Qmax	Cmax	Qmax	Cmax	Qmax
(2,3)	189.69	≥ 390	249.69	300	309.26	250	/	≤ 200
(5,8)	323.14	≥ 769	663.81	570	733.3	450	/	≤ 450
(4,7,3,9)	535.52	≥ 1656	716.06	1234	946.39	1100	/	≤ 1050
(6,10,8,3)	590	≥ 1878.7	787.48	1400	1185.7	1200	/	≤ 1180

Pour conclure ce travail de publication, nous avons développé un modèle mathématique linéaire en nombres entiers mixtes pour résoudre un problème d'ordonnancement. L'objectif principal était de minimiser le makespan tout en prenant en compte les émissions de CO₂. Les expériences réalisées ont souligné l'importance de l'effet d'apprentissage, démontrant son impact significatif sur la réduction du temps de traitement des tâches et la structuration des groupes de tâches similaires.

L'analyse de la complexité du modèle, en prenant en considération l'effet d'apprentissage et les cadences des machines, a été approfondie à travers plusieurs scénarios. Les résultats ont indiqué que des taux d'apprentissage plus faibles entraînaient un effet d'apprentissage plus prononcé, améliorant ainsi le rendement global. De plus, le facteur d'émission de CO₂ a été identifié comme un paramètre influençant le choix des vitesses de traitement.

3.6 Conclusion

Dans ce chapitre, nous introduisons un modèle d'ordonnancement global qui intègre l'effet d'apprentissage au sein d'un atelier de production à machine en série. Cette représentation s'apparente aux modèles antérieurs décrits dans la littérature. Après analyse de ces modèles, notre sélection s'est orientée vers l'adaptation du modèle présenté par Biskup, centré sur les variables de position.

Par la suite, nous avons étendu ce modèle en y incorporant la technologie de groupe, ce qui implique la prise en considération des temps de configuration. Au sein de la communauté scientifique, deux types de temps de configuration ont été discutés, mais notre étude se focalise spécifiquement sur les temps de configuration dépendants de la séquence. Cette décision découle de leur meilleure adéquation avec les scénarios manufacturiers concrets.

Les deux prochains chapitres seront consacrés à la résolution de ce modèle en utilisant une optimisation multi-objectif. Des extensions et des cas particuliers seront également examinés dans le cadre d'une analyse approfondie.

Chapitre 4

L'ordonnancement dans le FSDGS avec contrainte de transport

4.1 Introduction

Les orientations actuelles de la recherche dans le domaine de la théorie de l'ordonnancement visent, depuis plusieurs années, à améliorer la modélisation des problèmes rencontrés dans des contextes pratiques et industriels. Ces efforts ont abouti à la création de modèles intégrant à la fois l'aspect ordonnancement et la prise de décision logistique, tenant compte de paramètres tels que la disponibilité des machines et le transport.

Au cours des soixante dernières années, les problèmes d'ordonnancement ont fait l'objet d'une étude approfondie. Dans la plupart des cas, les temps de transport sont rarement pris en considération, et les tâches sont généralement envisagées comme commençant immédiatement sur la machine. Cette hypothèse n'est pas toujours correcte, voire erronée. En réalité, il est impossible de commencer l'opération O_{ij} immédiatement après l'achèvement de l'opération O_{i-1j} , car le produit doit d'abord être transporté de la machine $i-1$ à la machine i . En intégrant cet aspect dans notre étude, nous conférons au problème abordé une dimension réaliste, le rendant ainsi plus proche des situations rencontrées dans l'industrie.

L'analyse de la contrainte de transport dans l'ordonnancement peut être abordée sous différents angles, tels que le plus court chemin, l'optimisation de l'utilisation des ressources de transport, la planification de trajectoire, et les temps de transport entre les ressources. Dans le cadre de notre thèse, nous nous concentrons particulièrement sur les temps de transport d'un job entre deux ressources. En raison de la complexité introduite par l'ajout de cette contrainte, les travaux réalisés dans ce domaine sont relativement rares.

L'intégration des contraintes de transport dans les problèmes d'ordonnancement a été abordée en tenant compte de diverses hypothèses, principalement liées au nombre de

véhicules de transport et à leurs capacités. Ainsi, la détermination du nombre de véhicules de transport joue un rôle crucial dans la gestion fluide du flux de pièces au sein du système de production.

En cas de sous-estimation du nombre de véhicules par rapport aux niveaux de production des machines, cela pourrait entraîner une saturation des zones de stockage en sortie de poste de travail, ainsi que des temps d'attente importants pour obtenir un chariot de transport. À l'inverse, un nombre excessif de véhicules par rapport aux taux de production des machines pourrait conduire à une attente importante des chariots, provoquant une congestion des parkings et des voies de circulation, pouvant éventuellement entraîner des blocages du système. Il est essentiel de souligner que les chariots, en plus de leur impact opérationnel, représentent un coût significatif, d'où la nécessité cruciale de déterminer leur nombre avec précision.

Cette détermination dépend directement de la configuration du système flexible de production, des gammes de production, des temps d'usinage, des vitesses de déplacement à vide et à charge, du nombre de pièces qu'ils peuvent transporter, ainsi que de leurs modes de fonctionnement (monodirectionnel ou bidirectionnel) et de leurs règles de gestion. Dans le cadre de notre étude en cours, notre contribution réside dans l'ajout de deux types de contraintes :

- la prise en compte de plusieurs moyens de transports avec capacité limitée.
- la prise en compte d'un nombre finie de transports avec capacité limitée.

4.2 Le FSDGS avec un système multi-transports à capacité limité

Ces travaux visent principalement à combler le fossé souvent observé entre les études théoriques et la pratique dans le domaine de l'ordonnancement, en se concentrant particulièrement sur les problèmes de type flow shop. Ainsi, nous avons cherché à explorer des aspects sous-investis, en mettant l'accent sur des éléments tels que l'effet d'apprentissage, l'intégration de la technologie de groupe, les temps de configuration dépendant de la séquence et leur impact sur le problème. De manière complémentaire, nous avons également intégré des considérations écologiques dans l'optimisation du FSDGS.

Dans cette section, nous élargissons notre contribution en incluant le transport des jobs entre les différentes machines dans le problème de FSDGS. Cette extension est peu fréquemment abordée dans la littérature en raison de sa complexité. Dans la suite de cette partie, nous cherchons à étendre les modèles proposés pour le FSDGS en introduisant une nouvelle modélisation mathématique qui intègre les contraintes de transfert des tâches entre les machines à l'aide de robots de transport autoguidé appelés Automated Guided

Vehicles (AGV). Nous proposons également un schéma de résolution basé sur l'adaptation de la méthode LP-metric pour la résolution bi-objectif du problème FSDGS. Cette section se conclut par une évaluation numérique de notre approche.

4.2.1 Description et formulation du problème

Les contraintes liées au transport sont généralement associées au nombre de moyens de transport utilisés et à leurs capacités. Dans cette section, nous avons intégré plusieurs AGV dotés d'une capacité limitée. Leur mission initiale est de transporter les jobs du stock de départ à la première machine, puis de la première machine vers les suivantes. Une fois le traitement terminé, ils sont à nouveau dirigés vers le stock final.

Une autre caractéristique cruciale des véhicules de transport est leur fonctionnement à des vitesses variables, ce qui signifie que le temps de transport dépend de la vitesse choisie, influant ainsi sur la consommation d'énergie. Le modèle formulé dans ce chapitre englobe les mêmes éléments que ceux définis précédemment, y compris les hypothèses et caractéristiques préalablement définies.

Par ailleurs, tandis que les modèles proposés dans le chapitre précédent sont tous orientés vers une optimisation mono-objectif, le modèle du FSDGS avec effet d'apprentissage et temps de transport présenté dans ce chapitre est un modèle bi-objectif visant à minimiser simultanément le makespan et la consommation d'énergie du système. En plus des paramètres et variables définis précédemment, s'ajoutent ceux-ci :

Les indices :

- s : l'indice de la vitesse du transporteur, où $s = 1, \dots, L$

Les paramètres :

- L : le nombre total de vitesses du transporteur
- T_{0s} : le temps de transport du job de l'entrepôt à la première machine à la vitesse s
- T_{is} : le temps de transport du job de la machine i à la machine $i + 1$ à la vitesse s
- Sep_i : la puissance de configuration par unité de temps sur la machine i
- Tp_s : la puissance de transport du véhicule à la vitesse s par unité de temps

Les variables de décisions :

- Y_s : Une variable de décision binaire qui prend la valeur 1 si le transport fonctionne à la vitesse s , 0 sinon.

Le premier objectif (F1) est de minimiser le temps de réalisation maximal (makespan), cet objectif vise à réduire les temps de production et de transport quitte à utiliser des vitesses élevées qui consomment beaucoup d'énergie comme le montre l'équation (4.1), et le second objectif (F2) est de minimiser la consommation totale d'énergie durant les quatre phases de traitement, qui est représenté par l'équation (4.2), en respectant les contraintes de l'étude.

Le modèle mathématique :

• **Fonction objectif :**

$$\text{Minimiser } F1 = Cmax \quad (4.1)$$

$$\text{Minimiser } F2 = TEC = Ec_p + Ec_s + Ec_i + Ec_t \quad (4.2)$$

• **S.C :**

Les contraintes (3.35), (3.36), (3.37), (3.38), (3.39), (3.44), (3.45) et (3.46)

$$\sum_{s=0}^L Y_s = 1 \quad (4.3)$$

$$C_{g11} \geq \sum_{s=0}^L T_{0s} Y_s + \sum_{j=1}^{N_g} \sum_{v=1}^V X_{gjl1v} P_{gjl1v}, \quad \forall g = 1, \dots, G \quad (4.4)$$

$$C_{gli} \geq C_{gl-1i} + \sum_{j=1}^{N_g} \sum_{v=1}^V X_{gjl1v} P_{gjl1v}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall l = 2, \dots, N_g \quad (4.5)$$

$$C_{gli} \geq C_{gli-1} + \sum_{j=1}^{N_g} \sum_{v=1}^V X_{gjl1v} P_{gjl1v} + \sum_{s=0}^L T_{i-1s} Y_s, \quad \forall g = 1, \dots, G \quad \forall i = 2, \dots, m \quad \forall l = 1, \dots, N_g \quad (4.6)$$

$$C_{g1i} \geq S_{gi} + \sum_{j=1}^{N_g} \sum_{v=1}^V X_{gjl1v} P_{gjl1v}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (4.7)$$

$$F_{gi} \geq C_{gN_g i}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (4.8)$$

$$Ec_p = \sum_{v=1}^V \sum_{g=1}^G \sum_{i=1}^m \sum_{j=1}^{N_g} \sum_{l=1}^{N_g} q_{iv} P_{gjl1v} X_{gjl1v} \quad (4.9)$$

$$Ec_s = \sum_{h=0}^G \sum_{g=1}^G \sum_{i=1}^m s_{hgi} U_{hg} sep_i \quad (4.10)$$

$$Ec_i = \sum_{g=1}^G \sum_{i=1}^m I_{gi} Ip_i \quad (4.11)$$

$$Ec_t = \sum_{s=0}^L CmaxTp_s Y_s + \Delta E \quad (4.12)$$

$$Cmax \geq F_{gm}, \quad \forall g = 1, \dots, G \quad (4.13)$$

$$C_{gli}, F_{gi}, S_{gi}, P_{gjliv} \geq 0 \quad (4.14)$$

$$X_{gjliv}, U_{hg}, Y_s \in \{0, 1\} \quad (4.15)$$

la contrainte (4.3) permet de limiter la vitesse du véhicule de transport. Les dates de fin des jobs de chaque groupe sont déterminés dans les contraintes (4.4) à (4.7). La contrainte (4.8) détermine la date de fin du groupe. La consommation d'énergie des phases de production, de préparation, d'inactivité et de transport est représentée par les contraintes (4.9), (4.10), (4.11) et (4.12). La contrainte (4.13) définit la durée de l'ordonnancement (makespan). Les variables de décision sont déterminées par les contraintes (4.14) et (4.15).

4.2.2 L'approche de résolution

Parmi les méthodes d'optimisation multi-objectifs abordées dans la littérature, la méthode LP-métrique se distingue par son efficacité remarquable, permettant d'obtenir des solutions acceptables même pour des problèmes complexes dans divers domaines. Cette méthode a fait l'objet d'études approfondies par plusieurs chercheurs, dont Arabzad, Ghorbani, et Zolfani (2015), qui l'ont appliquée à un problème de localisation visant à minimiser deux objectifs : le coût total de la chaîne d'approvisionnement et le taux de détérioration lors du transport. De même, Zeddami, Belkaid, et Bennekrouf (2020) ont développé une approche bi-objectif pour le problème de routage de véhicules de la société algérienne de distribution de carburant NAFTAL, axée sur la minimisation des émissions de CO₂ et la maximisation du gain. Une autre contribution, celle de Tahraoui, Sari-Triqui, et Bennkrouf (2022), repose sur une modélisation mathématique basée sur la méthode LP-métrique pour résoudre un problème de planification de la production. Leur objectif était de trouver la meilleure solution combinant deux objectifs contradictoires : la minimisation de la consommation d'énergie et la maximisation de la qualité du poulet produit.

Au cours des deux dernières décennies, de nombreux travaux se sont concentrés sur l'optimisation simultanée de plusieurs objectifs. La résolution d'un problème d'optimisation multi-objectifs consiste à déterminer la solution qui correspond le mieux aux objectifs définis parmi les solutions de compromis. Dans notre cas, nous cherchons à optimiser simultanément le makespan et la consommation d'énergie. Ces deux objectifs sont contradictoires, avec des unités de mesure différentes, et ne peuvent pas être directement combinés dans une seule fonction. Nous avons donc adopté la méthode LP-métrique pour

fusionner les deux objectifs en une seule fonction :

$$Min \quad w_1 \frac{F_1 - F_{MILP_1}^*}{F_{MILP_1}^*} + w_2 \frac{F_2 - F_{MILP_2}^*}{F_{MILP_2}^*} + \dots + w_n \frac{F_n - F_{MILP_n}^*}{F_{MILP_n}^*} \quad (4.16)$$

Cette équation représente la fonction LP-métrique, où F_1, F_2, \dots, F_n sont les fonctions objectif à optimiser simultanément. Les valeurs optimales des objectifs, $F_{MILP_1}^*, F_{MILP_2}^*, \dots, F_{MILP_n}^*$, sont obtenues à partir du modèle de programmation linéaire en nombres entiers mixtes. Les poids relatifs à chaque fonction objectif, w_1, w_2, \dots, w_n , sont déterminés par le décideur, où $0 \leq w_n \leq 1$ et $\sum w_n = 1$. Dans notre cas, la formulation du problème consiste à intégrer les paramètres et les variables des deux objectifs avec : $F_1 = Cmax$ et $F_2 = TEC$, ainsi l'équation devient :

$$Min \quad w_1 \frac{F_1 - F_{MILP_1}^*}{F_{MILP_1}^*} + w_2 \frac{F_2 - F_{MILP_2}^*}{F_{MILP_2}^*} \quad (4.17)$$

4.2.3 Évaluation expérimentale de l'approche d'optimisation multi-objectifs du FSDGS avec transport

Afin de mettre en avant l'efficacité de notre approche, plusieurs expérimentations ont été réalisées, au cours desquelles les poids des deux fonctions objectif ont été modifiés ainsi que le taux d'apprentissage, générant ainsi différents scénarios. Les instances sont générées en fonction de l'étude de cas considérée, composée essentiellement de quatre machines. La première et la dernière machine exécutent la tâche à trois vitesses différentes (rapide, moyenne, lente), tandis qu'au niveau de la deuxième et de la troisième machine, l'intervention de l'opérateur est essentielle. Ainsi, la prise en compte de l'effet d'apprentissage est inévitable. Le nombre de groupes varie de 2 à 10 groupes, répartis en deux catégories : les petites instances (2 à 5 groupes) et les moyenne instances (6 à 10 groupes). Le nombre de tâches dans les groupes varie de 2 à 10 tâches dans chaque groupe. Les temps de traitement suivent une distribution uniforme entière $U[1, 50]$. Les paramètres utilisés et les caractéristiques supplémentaires du problème sont répertoriés dans les tableaux 4.1 et 4.2. Chaque instance est résolue par le solveur IBM ILOG CPLEX (version 12.8). Trois vecteurs de poids w_1 et w_2 et trois taux d'apprentissage différents sont utilisés pour les instances générées.

Tableau 4.1: données communes des scénarios

Machines	M1			M2	M3	M4		
Sep_i	3.5			2.5	3	2.5		
Ip_i	2			2.6	3	2.13		
vitesse	$v1$	$v2$	$v3$	$\forall v$	$\forall v$	$v1$	$v2$	$v3$
q_{iv}	25	20	15	10	16	20	15	10
	stock vers M1		M1 à M2	M2 à M3	M3 à M4	M4 vers stock		
T_{is}	s_1	10		18	12	16	22	
	s_2	5		9	6	8	11	
Tp_s	s_1				1			
	s_2				4			

Tableau 4.2: caractéristiques des instances générées

Instances	taille des groupes	nombre de groupes	taille des jobs	nombre de jobs
1	petite	3	petite	(2,3,4)
2			moyenne	(5,7,9)
3	moyenne	7	petite	(4,4,2,3,2,3,4)
4			moyenne	(3,5,3,9,2,3,4)

4.2.4 Analyse et discussion des résultats obtenus

Pour étudier l'efficacité du modèle MILP pour le problème d'ordonnancement de FSDGS avec effet d'apprentissage et temps de transport à capacité limité, plusieurs tests ont été réalisés en variant les poids des deux fonctions objectif et le taux d'apprentissage. Cette étude sera divisée en trois expériences :

- La première expérience vise à minimiser le Cmax sans tenir compte de la consommation d'énergie.
- La deuxième expérience se concentre uniquement sur la minimisation de la consommation d'énergie.
- La troisième expérience combine les deux objectifs en un seul en utilisant la méthode LP-métrique.

Chaque expérience a été réalisée selon trois scénarios distincts. Dans le premier scénario, l'effet d'apprentissage est négligé, avec un taux d'apprentissage de 100% pour les machines M2 et M3. Le deuxième scénario privilégie un effet d'apprentissage plus élevé sur la machine M2 par rapport à M3. Enfin, le troisième scénario attribue un effet d'apprentissage supérieur à la machine M3 par rapport à la machine M2. Les résultats de ces expériences

sont exposés dans les tableaux 4.3, 4.4 et 4.5, tandis que leur comparaison est illustrée dans les figures 4.1 et 4.2.

- **1ère expérience : minimisation du makespan**

Les résultats de la première expérimentation, présentés dans le tableau 3, indiquent une augmentation du makespan proportionnelle à la taille croissante des problèmes générés. Au cours de ces essais, le C_{max} atteint sa valeur minimale, tandis que la consommation d'énergie atteint son niveau maximal en raison de l'utilisation des cadences de production et de transport les plus élevées.

Dans les trois différents scénarios, l'introduction de l'effet d'apprentissage au niveau des machines M2 et M3 se traduit par une réduction du makespan, attribuable à la diminution du temps de traitement. Cette réduction est plus significative lorsque l'effet d'apprentissage est élevé au niveau de la machine M3 par rapport à M2, en raison des temps de traitement intrinsèquement plus longs sur cette dernière.

Tableau 4.3: les résultats de la première expérience

Scénarios	Scénario 1		Scénario 2		Scénario 3	
Problèmes	C_{max}	TEC	C_{max}	TEC	C_{max}	TEC
1	535.25	5081	532.83	4733	532.83	4608
2	865.95	12303	819	11270	803	10864
3	1216.3	13328	1208.3	12679	1207.3	12574
4	1379.8	15822	1209.8	14898	1169.8	14644

- **2ème expérience : minimisation de la consommation d'énergie**

Dans la deuxième expérience, la priorité est donnée à la minimisation de la consommation d'énergie, au détriment du makespan. L'optimisation de cet objectif s'effectue en utilisant les vitesses de traitement et de transport les plus lentes, même si cela entraîne une augmentation du temps de traitement. Par ailleurs, il est observé que, dans les trois scénarios, la consommation d'énergie est significativement élevée lorsque le taux d'apprentissage est de 100% (pas d'apprentissage). Elle diminue considérablement lorsque le taux d'apprentissage est réduit à 90% et 70%. Cependant, une nette diminution est remarquée lorsque le taux d'apprentissage est fixé à 70% au niveau de la machine M3, car cette dernière consomme plus d'énergie que la machine M2.

Tableau 4.4: les résultats de la seconde expérience

Scénarios	Scénario 1		Scénario 2		Scénario 3	
Problèmes	Cmax	TEC	Cmax	TEC	Cmax	TEC
1	660.25	4473.5	660.28	4225.4	645.25	4050.5
2	1106.3	11304	1113.8	10267	1102.4	9989.3
3	1520	11719	1523.1	10917	1504.6	10664
4	1799.3	13728	1816.3	12694	1803.8	12351

- **3ème expérience : la méthode LP-métrique**

Les deux expériences précédentes ont clairement mis en évidence l'aspect conflictuel entre les deux fonctions objectives. Il est donc impératif de parvenir à un compromis satisfaisant les deux critères d'optimisation. À cette fin, nous avons utilisé la méthode LP-métrique. Il est important de noter que toutes les données demeurent inchangées dans l'ensemble des expériences et des scénarios.

Les résultats obtenus, présentés dans le tableau 5 grâce à la méthode LP-métrique, permettent de minimiser simultanément les deux fonctions objectifs, avec une prédominance marquée pour la minimisation du makespan dans le cas des deux premiers problèmes (petites instances). En revanche, lorsque la taille des problèmes augmente, la méthode LP-métrique s'efforce de trouver une solution qui satisfait les deux critères d'optimisation.

Tableau 4.5: les résultats de la troisième expérience

Scénarios	Scénario 1		Scénario 2		Scénario 3	
Problèmes	Cmax	TEC	Cmax	TEC	Cmax	TEC
1	535.25	4920.2	536.83	4599.6	536.83	4431
2	869.95	12068	823.4	11030	803.17	10707
3	1225.4	12803	1222.8	11871	1215.6	11710
4	1388.4	15058	1383.8	13937	1361.4	13579

Les deux fonctions objectives connaissent une réduction significative lorsque l'effet d'apprentissage est intégré à la méthode LP-métrique (scénarios 2 et 3) par rapport au scénario 1. Cette diminution résulte d'une réduction des temps de traitement, de l'énergie de production et d'attente. La baisse est plus marquée pour le deuxième objectif, en particulier dans le scénario 3, en raison de la consommation d'énergie plus élevée de la machine 3 par rapport à la machine 2.

- Comparaison entre les trois expériences

Une comparaison entre les différentes expériences et scénarios est présentée dans les figures 4.1 et 4.2, mettant en évidence l'utilité de la méthode LP-métrique dans le contexte de notre étude. Les résultats obtenus et les figures illustrent clairement que la méthode employée cherche à trouver une solution satisfaisant les deux fonctions objectives, indépendamment du scénario appliqué au problème.

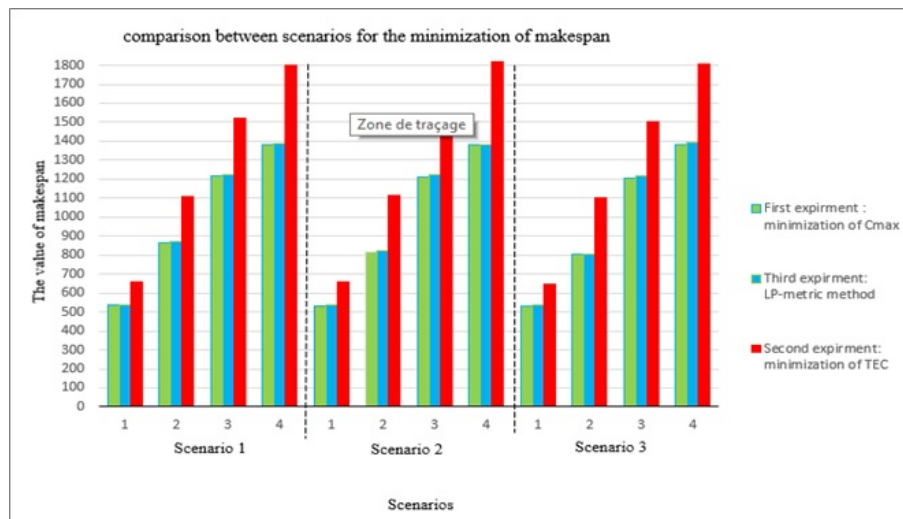


Figure 4.1: Comparaison entre les trois expériences pour la minimisation du makespan à travers les trois scénarios

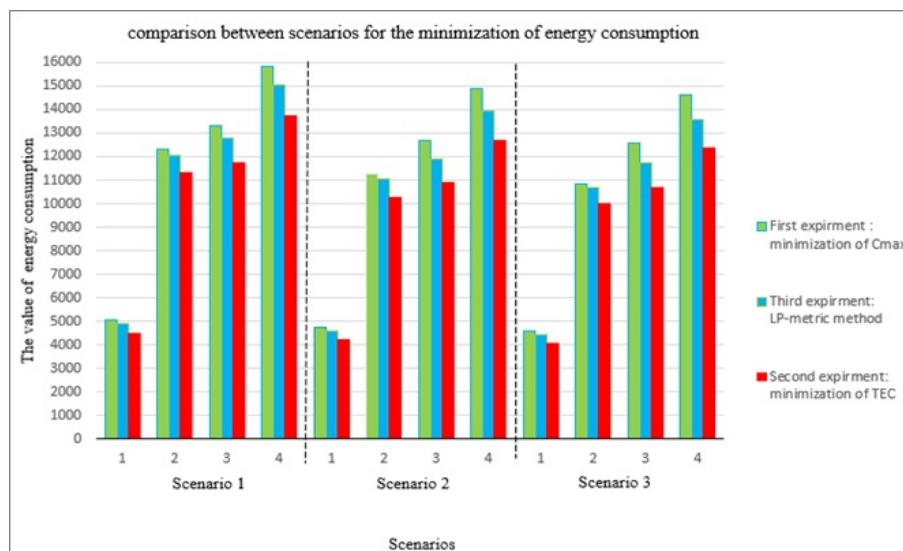


Figure 4.2: Comparaison entre les trois expériences pour la minimisation de la consommation d'énergie à travers les trois scénarios

- Impact des coefficients de pondération sur l'optimisation du problème étudié

En variant les poids des fonctions objectives dans les trois scénarios, comme illustré dans les figures 4.3, 4.4 et 4.5, nous constatons, pour le scénario 1, que les poids des deux fonctions objectives exercent une influence significative sur l'optimisation du problème considéré. Ainsi, avec la paire $[0.7, 0.3]$, la méthode favorise le premier objectif (minimisation du makespan), tandis qu'avec la paire $[0.3, 0.7]$, le deuxième objectif (minimisation de la consommation d'énergie) prend le dessus. Par ailleurs, la paire $[0.5, 0.5]$ attribue une solution intermédiaire en optant pour des vitesses moyennes de production et de transport.

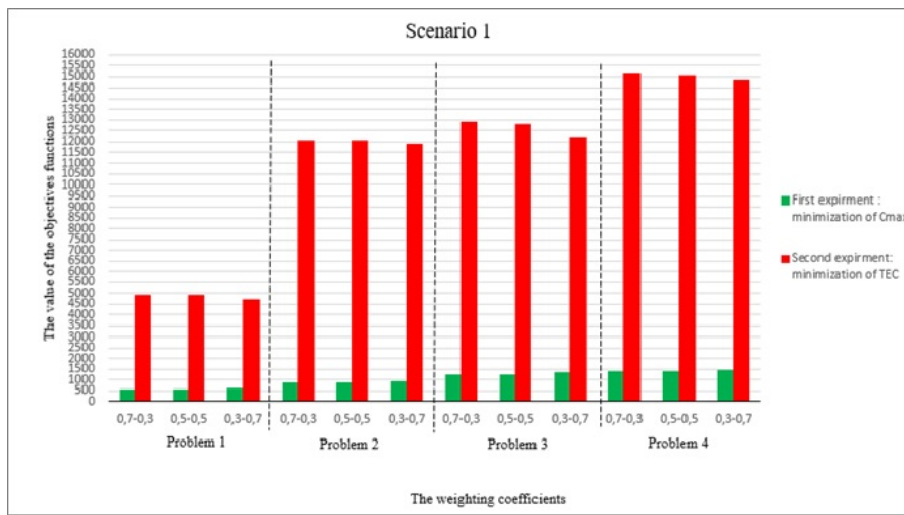


Figure 4.3: L'impact des coefficients de pondération sur l'optimisation des deux objectifs pour le scénario 1

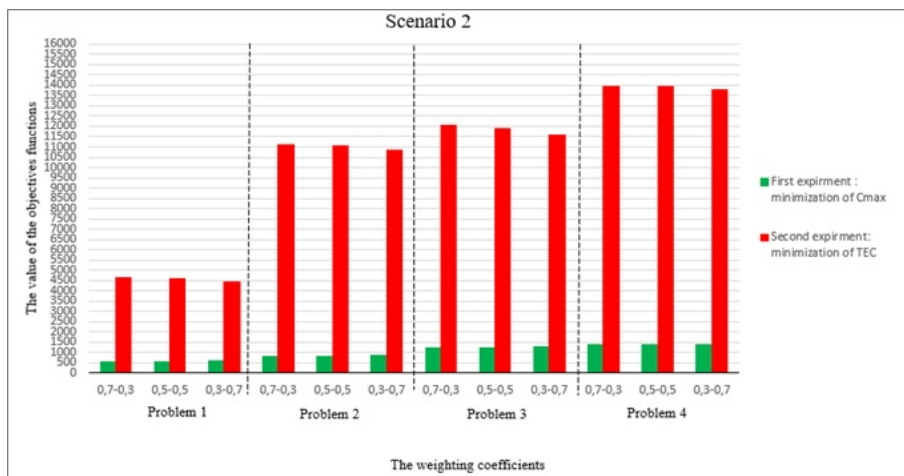


Figure 4.4: L'impact des coefficients de pondération sur l'optimisation des deux objectifs pour le scénario 2

Cette observation n'est pas immédiatement évidente dans le cas des deux premiers problèmes, mais elle devient de plus en plus apparente à mesure que la taille des problèmes

augmente. Dans les deux scénarios suivants, les deux fonctions objectives sont réduites, en mettant particulièrement l'accent sur la consommation d'énergie, étant donné son niveau initial élevé. Nous avons également noté que, pour la paire $[0.5, 0.5]$, la tendance est davantage orientée vers l'optimisation du makespan. En revanche, dans le scénario 3, un équilibre plus marqué est atteint, et la consommation d'énergie est réduite de manière plus significative.

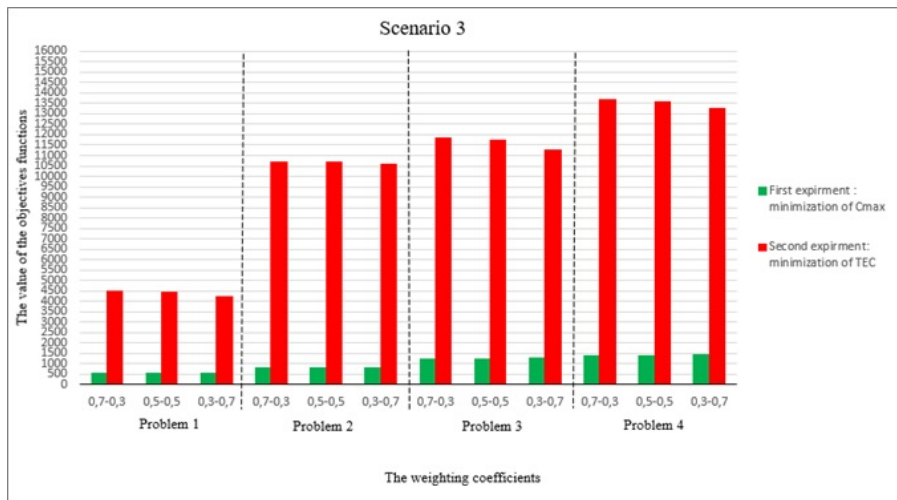


Figure 4.5: L'impact des coefficients de pondération sur l'optimisation des deux objectifs pour le scénario 3

Les résultats obtenus sont satisfaisants, car ils permettent au décideur de mieux gérer ses ateliers de production, d'attribuer efficacement les employés à leurs postes respectifs, et de réduire la consommation d'énergie des machines. Par ailleurs, nous avons conclu que les deux objectifs étaient contradictoires et que la minimisation de l'un d'entre eux négligeait l'autre. De plus, il est important de noter que le critère du makespan dépend considérablement de l'effet d'apprentissage, tandis que le critère de la consommation d'énergie dépend davantage de la vitesse des machines et du transport.

4.3 Cas d'étude

Afin de valider les différentes approches avancées dans le cadre de notre thèse, il a été impératif de procéder à des tests de performance en utilisant comme étude de cas le processus de fabrication des bielles forgées au sein d'une entreprise métallurgique. La cellule de production associée à ce processus a été exposée, modélisée et formalisée en référence aux travaux établis par D. N. Sekkal et Belkaid (2023). Cette section a pour objectif de présenter en détail le cas d'étude en question, tout en mettant en lumière le modèle mathématique qui a été employé pour conceptualiser, évaluer et confirmer la

validité des approches suggérées pour résoudre le FSDGS avec prise en compte de l'effet d'apprentissage et des temps de transport.

4.3.1 Présentation du cas d'étude de bielles forgées

Selon Neufeld et al. (2016), la plupart des articles de recherche se concentrent sur des problèmes simples de FSDGS, et comme mentionné dans les chapitres précédents, les travaux tenant compte à la fois des temps de transport et de l'effet d'apprentissage sont rares. En parcourant les recherches sur le FSDGS, on constate que la majorité s'attache principalement à l'aspect théorique du problème, négligeant ainsi les contraintes réelles présentes dans un système de production manufacturier, telles que les temps de transport, la maintenance, les temps de traitement flous/incertains, les effets d'apprentissage et/ou de détérioration, etc.

En raison du manque de recherche approfondie sur les contraintes réelles du FSDGS, il est rare de trouver des jeux de données qui respectent toutes ces contraintes dans ce domaine, à l'exception de certains benchmarks standard tels que ceux de Taillard (1993) ou Salmasi et al. (2010). Cependant, ces benchmarks ne reposent pas sur des cas d'usines concrets. Pour prendre en compte les contraintes réelles du Flow Shop, nous avons opté pour les instances de benchmark proposées par D. N. Sekkal et Belkaid (2023), qui sont inspirées d'une véritable cellule de fabrication de bielles forgées.

Cette étude de cas est définie comme un flow shop, ce qui donne lieu à la formulation d'un problème d'ordonnancement dans un flow shop avec technologie de groupe (FSDGS). Le benchmark offre également divers scénarios permettant d'évaluer la robustesse et l'efficacité d'une approche de résolution. Au sein de cette section, nous exposons le processus de fabrication des bielles forgées, ainsi que les informations concernant les ressources, les produits et le système de transport.

4.3.2 L'adaptation de la technologie de groupe pour le cas d'étude

Notre intérêt pour ce problème découle de son application industrielle spécifique. Il s'agit d'une entreprise spécialisée dans la fabrication de bielles forgées, adaptées à diverses exigences des clients et représenté dans la figure 4.6. Ces pièces jouent un rôle crucial dans le moteur en transmettant des charges de traction et de compression axiales, tout en supportant des contraintes de flexion résultant de la poussée et de la traction sur le piston, ainsi que de la force centrifuge générée par le vilebrequin en rotation.

La bielle, élément mécanique essentiel, est connectée à une extrémité du piston par un axe appelé axe de piston, tandis que l'autre extrémité est reliée au cou de vilebrequin. Elle assure la transmission de la force entre deux arbres en mouvement. Composée principalement de trois parties –la grande extrémité (circulaire/modifiée) est reliée au vilebrequin

et doit être aussi peu déformable que possible. La petite extrémité (circulaire) est reliée à l'axe de piston et doit résister à des charges de compression élevées dues à la combustion et à des charges de traction élevées dues à l'inertie à l'extrémité du piston. Entre ses deux alésages, il y a une partie qui relie la grande extrémité à la petite extrémité de la bielle appelée section du maneton. Elle offre une rigidité maximale avec un poids minimal.

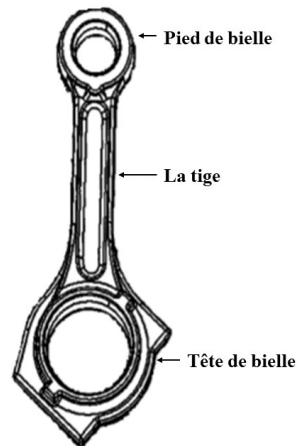


Figure 4.6: Illustration d'une bielle

Les bielles peuvent être classifiées selon deux critères principaux :

- le matériau de forgeage qui les constitue, soit en acier trempé revenu, soit en acier non trempé revenu.

Une bielle en acier trempé revenu désigne une bielle qui satisfait aux exigences de performance après un traitement thermique de trempe et de revenu à haute température suite au forgeage. Les bielles en acier non trempé revenu sont celles qui répondent aux exigences de performance en contrôlant la vitesse de refroidissement après le forgeage.

- On distingue deux types de structure de la bielle, : la bielle fendue et la bielle monobloc.

La bielle fendue se caractérise par la séparation du corps principal et du couvercle pendant le forgeage. Ces bielles permettent l'utilisation d'un vilebrequin monobloc, facilitant les interventions sur les pièces en mouvement. En revanche, la bielle monobloc fait référence à une forge complète de la bielle, du corps principal et du couvercle, suivie de la séparation par un traitement ultérieur. Ces bielles nécessitent une application sur un vilebrequin amovible.

Pour résister aux contraintes en cours d'utilisation, les bielles sont fabriquées à partir de matériaux extrêmement résistants. Notre étude se focalise sur la production de bielles

monobloc par forgeage, avec diverses longueurs et usages spécifiques. La longueur dépend de la rotation nécessaire au bon fonctionnement du moteur : une bielle plus grande prendra plus de temps pour effectuer une rotation. Ainsi, la composition, la longueur et la forme dépendent des contraintes auxquelles les bielles seront soumises, que ce soit dans le contrôle du pas collectif d'un rotor d'hélicoptère, la direction automobile ou la récupération du couple de freinage sur l'essieu arrière d'une voiture.

Compte tenu de ces différentes caractéristiques et utilisations multiples, il serait plus approprié de regrouper les différents types de bielles en familles pendant la fabrication afin d'optimiser les temps de configuration.

4.3.3 Processus de fabrication de bielles forgées

Avec le développement de l'industrie automobile et l'augmentation de la puissance des moteurs, les exigences en matière de qualité des bielles sont de plus en plus élevées. Le processus de fabrication englobe plusieurs types de bielles, chacune suivant des procédés presque similaires. La complexité réside dans le choix du matériau approprié et dans la garantie de l'équilibre du poids. La bielle doit être aussi légère que possible tout en restant suffisamment robuste pour supporter des charges maximales sans défaillance. Par conséquent, il est nécessaire de faire des choix judicieux concernant les paramètres d'usage tels que la vitesse, l'avance, la profondeur de coupe et les actions de nettoyage, lors de la transition de la fabrication d'un groupe de bielles à un autre.

Le processus de fabrication représenté dans la figure 4.7 comprend quatre étapes : le chauffage par induction de la billette, le laminage par étirement, le matriçage, l'élimination des bavures et le poinçonnage des grands et petits trous d'extrémité (Kar, 2019).

4.3.3.1 Le chauffage

La première étape consiste à chauffer une billette cylindrique à une température comprise entre 1050 °C et 1350 °C. Pour optimiser l'environnement de travail, améliorer la qualité du chauffage des billettes, et faciliter l'automatisation de la production, le chauffage par induction à moyenne fréquence est généralement privilégié. Cette méthode présente une faible consommation, généralement inférieure à 0.5%, et produit une bavure de forgeage très fine, facile à éliminer.

Étant donné le rythme de production, il est essentiel de contrôler la température de chauffage de la billette aux deux extrémités. Ainsi, la billette ne doit pas dépasser l'ouverture de la cavité avant d'être déchargée, un paramètre habituellement ajusté via le réglage de l'interrupteur de course. La température de chauffage impacte directement la qualité externe et interne de la bielle forgée. Une température basse favorise la résistance de la bielle et facilite le traitement thermique ultérieur. Cependant, une température trop

basse peut entraîner un mauvais écoulement du métal et une usure rapide de la matrice, ainsi que des fissures en cas de force d'impact excessive. À l'inverse, une température de chauffage trop élevée rend la bielle trop plastique, engendrant des défauts tels qu'un écoulement excessif du métal, un remplissage partiel insatisfaisant et une oxydation grave de la surface. Ainsi, une température de chauffage plus basse est généralement synonyme d'une meilleure qualité de la bielle forgée (Garde, 2015).

4.3.3.2 Le laminage

La deuxième étape du processus de fabrication est le laminage par étirement, une opération visant à déformer de manière contrôlée une billette cylindrique pour obtenir une pièce axisymétrique de diamètre réduit. Cette opération s'effectue à l'aide d'une machine à laminier, où la billette est positionnée entre deux surfaces d'outillage identiques. Chaque surface d'outillage est équipée d'un ou de plusieurs coins de façonnage, le nombre dépendant du degré de réduction de diamètre requis pour la pièce laminée. La distance entre les surfaces d'outillage est ajustée pour atteindre les dimensions nominales souhaitées de la pièce. Cette opération requiert l'intervention d'un opérateur qualifié pour guider la billette à travers le laminoir.

La qualité du laminage influe directement sur celle de la bielle forgée. Par conséquent, lors du laminage, il est crucial de prêter attention aux points suivants :

- a. La qualité de la surface des matrices de forgeage, qui, malgré leur longue durée de vie, peut devenir moins lisse au fil du temps en raison de l'écoulement du métal, entraînant le pliage des billettes de forgeage.
- b. L'inégalité du point d'entrée de chaque galet de forgeage, conduisant à un arc de cercle de section de transition réduit.
- c. L'inégalité de la surface des billettes avec des rayures.
- d. L'usure des matrices de forgeage pendant le laminage, entraînant une distribution inégale des billettes, avec davantage au centre et moins sur les côtés.

La présence de ces problèmes peut entraîner des défauts dans la bielle, tels que des courbures, des lignes de pliage et un remplissage incomplet. Pour remédier à ces problèmes, il est essentiel de prendre en considération les points suivants :

- a. Lors de la conception des matrices de forgeage, il est essentiel de déterminer de manière rationnelle la forme et la taille de chaque rouleau.
- b. Établissez une répartition rationnelle pour chaque coefficient d'extension, tout comme le choix du type de fente approprié.
- c. Effectuez un calcul précis de la taille de la section et de la fente de la billette de laminage.

4.3.3.3 Le matricage

Dans la troisième phase du processus, la préforme atteint sa forme finale à l'aide d'une matrice composée de deux moitiés, l'une fixée sur la plateforme et l'autre sur le poinçon. La billette est positionnée et transférée séquentiellement sur une série de matrices, incluant la matrice d'aplatissement, la matrice de blocage et la matrice de finition. En résultat, une ébauche avec une bavure sur sa périphérie, comprenant une tête, un corps et un pied, est obtenue. Une fois de plus, un opérateur est nécessaire pour déplacer l'ébauche entre les différentes presses.

Par ailleurs, plusieurs machines de forgeage des bielles peuvent être utilisées, chacune possédant des caractéristiques distinctes telles que des presses hydrauliques, presses mécaniques, presses à vis, et pilons ou marteaux. Le choix de la presse utilisée dépend du type de pièces forgées et du volume de production. Dans la cellule de fabrication considérée, la presse hydraulique est employée pour le matricage des bielles en raison de sa grande adaptabilité à différentes formes, son application à tous les matériaux et sa vitesse de forgeage ajustable.

Après le pré-forgeage, la bielle forgée est déplacée vers la matrice de forgeage finale, mais l'effet de l'élimination de la couche d'oxyde est évident lors du déplacement. Des rouleaux forgent la partie plate de la bielle pour faciliter son repérage, décollent la couche d'oxyde, et la soufflent en même temps. Le rythme de production est aussi régulier que possible, garantissant ainsi la température d'ébarbage et favorisant le contrôle de la qualité des processus ultérieurs, contribuant à l'amélioration de la qualité des bielles forgées.

4.3.3.4 Le poinçonnage

Les deux dernières étapes du processus sont regroupées en une seule, étant donné qu'elles sont effectuées sur la même machine, à savoir la presse. Au cours de cette phase, les bavures entourant l'ébauche sont éliminées, et les grands et petits trous d'extrémité sont percés.

Le découpage et le poinçonnage de la bielle sont généralement réalisés à haute température. Après le forgeage, la chaleur résiduelle est exploitée, et la contrainte de coupe est réduite. La matrice, comprenant à la fois le découpage et l'emboutissage, est utilisée dans la presse, permettant d'effectuer ces opérations en une seule fois. Cependant, lors du processus d'ébavurage, plusieurs problèmes peuvent survenir :

- a. Bavure résiduelle irrégulière : Cette irrégularité peut être causée par le décalage ou le jeu excessif de la matrice, lequel est généralement de l'ordre de 0,5 à 0,8 mm.
- b. Déformation de la découpe : La matrice d'ébavurage peut adopter une forme ondulée pour éviter les déformations. Outre les raisons susmentionnées, l'inadéquation entre la matrice de forgeage et la matrice d'ébavurage peut également en être la cause.

Afin de prévenir les déformations résultant de ce processus, l'utilisation d'une presse d'ébavurage à double pointe de hautes précisions ou d'un dispositif de guidage peut être envisagée dans l'outil de poinçonnage.

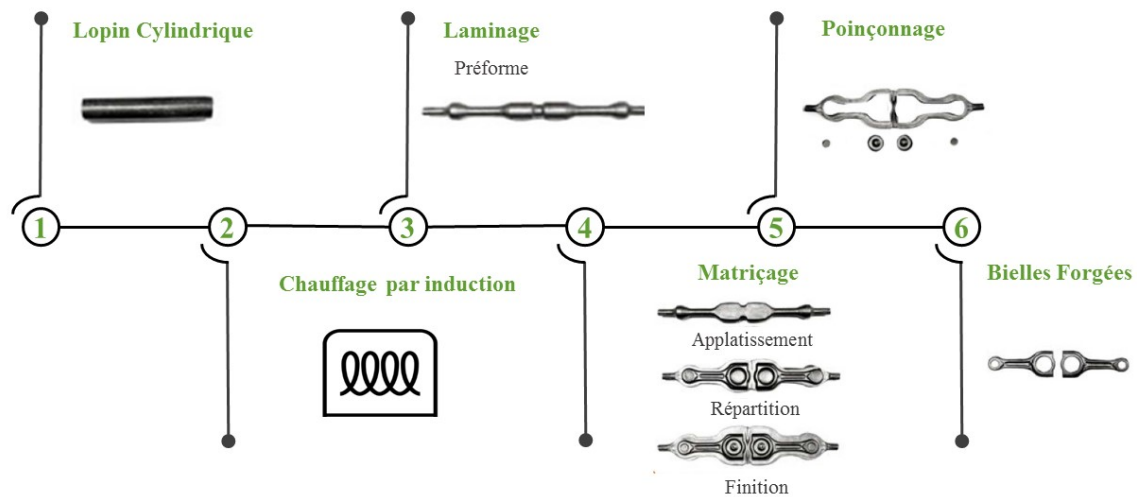


Figure 4.7: Processus de fabrication d'une bielle forgée

Par conséquent, le processus de production est fortement influencé par les effets d'apprentissage qui surviennent pendant les étapes semi-automatiques, nécessitant une intervention de l'opérateur. Un opérateur inexpérimenté peut entraîner d'importantes pertes de temps, de coûts et d'énergie. Pour améliorer l'efficacité, un système de technologie de groupe a été mis en œuvre sur la chaîne de production. Les bielles forgées sont classées en 13 groupes en fonction de leur forme et de leur taille (étiquetés de 1 à 13). Cependant, le traitement et le transport de ces bielles doivent se dérouler de manière séquentielle ; elles ne peuvent pas être traitées sur la machine à rouler si aucun transporteur n'est disponible, et un seul produit peut être transporté à la fois. L'AGV peut fonctionner à différentes vitesses, la vitesse de transport minimale étant considérée comme la vitesse normale, ces précisions seront détaillées dans la section suivante.

4.3.4 Données et caractéristiques relatives au cas d'étude

L'application industrielle du problème FSDGS en rapport avec la fabrication de bielles forgées est composé d'un ensemble de composants industriels, comprenant des AGV, des machines, des produits semi-finis, des opérateurs, ainsi qu'un système de manutention interconnecté par un système de transport. Cette configuration vise à traiter diverses tâches préalablement définies.

4.3.4.1 Les machines

Notre cas d'étude comporte quatre machines (Figure 4.8). Chaque machine est dédiée à la réalisation d'une seule opération, pouvant traiter plusieurs pièces, mais une seule à la fois et sans interruption (non préemptives).

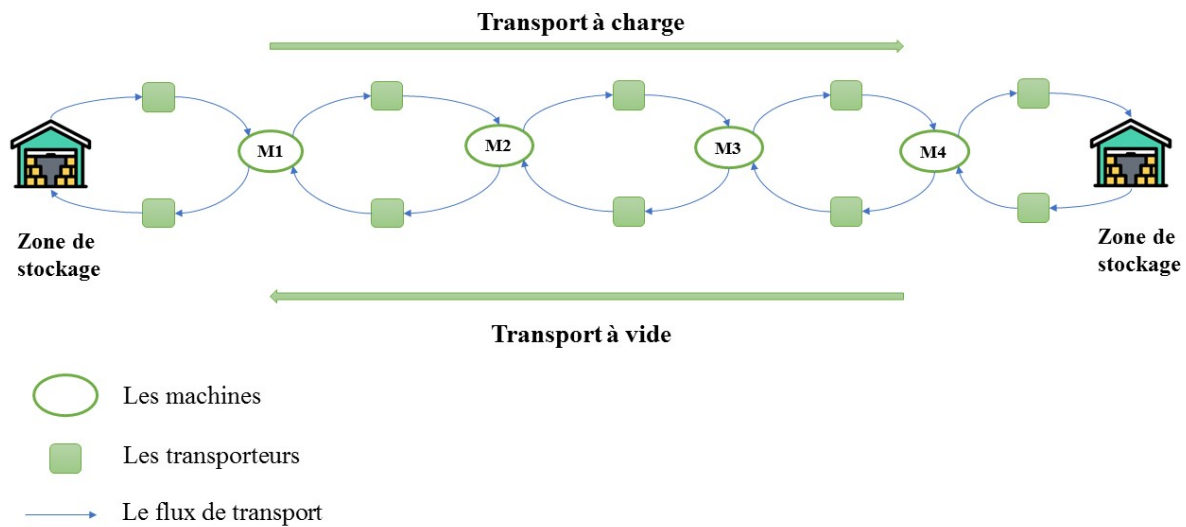


Figure 4.8: Disposition des machines de fabrication de bielles forgées

Les machines 2 et 3 nécessitent l'intervention des opérateurs qui sont disponibles à tout instant, mais disposant chacun de compétences différentes. Par contre, les machines 1 et 4 sont complètement automatisés et fonctionnant à différents régimes. Les opérations essentielles à la réalisation d'une bielle forgée sont de caractère successif, c'est-à-dire que l'étape de poinçonnage ne peut être réalisée avant l'étape de matriçage, et le laminage ne peut être effectué sans chauffage au préalable des lopins. En se basant sur cette classification, la cellule peut être vue comme un problème d'ordonnancement flow shop.

4.3.4.2 Les produits

Le tableau 4.6 présente les attributions des opérations aux ressources, ainsi que les durées opératoires respectives. Les valeurs spécifiées correspondent au temps requis (en secondes) pour que la machine/ressource effectue chaque opération.

Tableau 4.6: Temps de traitement des opérations

Id	Opérations	M1	M2	M3	M4
1	Chauffage des lopins	50			
2	Étirement du lopin préchauffé		40		
3	Estampage de la préforme			40	
4	Ébavurage de l'ébauche				20

Dans le cas d'étude considéré, la séquence d'opération des jobs identiques, elle commence par un transport des lopins sur le véhicule de transport auto-guidé (AGV), suivie des autres opérations, et se termine par une opération de transport vers le stockage finale des produits finis.

4.3.4.3 Le système de transport

Les ressources sont connectées entre elles grâce à un système de transport composé de véhicules guidés automatiquement. Ce dernier est équipé d'un dispositif d'aiguillage rotatif situé avant chaque point de divergence ou de convergence, il est utilisé pour diriger ou changer la trajectoire de ces véhicules dans un environnement automatisé.

Un système d'aiguillage pour les AGV peut prendre la forme de capteurs, de balises magnétiques, de bandes magnétiques, ou d'autres dispositifs qui permettent aux AGV de suivre un itinéraire prédéfini.

Pour le problème de FSDGS, deux opérations successives du même job ne peuvent être réalisées sur la même machine, il est nécessaire d'effectuer une opération de transport via des AGV. Le produit sera alors acheminé vers la machine suivante effectuant une autre opération.

Le temps de transport dépend de la disposition des machines, et une opération de transport englobe la récupération, le déplacement et le dépôt du job (produit) au niveau de la machine appropriée, ainsi que les temps de retour entre deux machines successives. Un autre détail important pris en charge dans cette étude concerne le fait que les AGV (Automated Guided Vehicles) utilisés fonctionnent à différentes vitesses. Par conséquent, les temps de transport dépendent également des cadences sélectionnées lors du transport. Dans notre étude de cas, une matrice de distance est fournie pour représenter les temps théoriques nécessaires aux opérations de transport qui sont représentés dans le tableau 4.7.

Tableau 4.7: Temps de transport des produits sur la machine

temps de transport (sec)					
vitesse(m/s)	stock vers M1	M1 vers M2	M2 vers M3	M3 vers M4	M4 vers stock
0.5	8	9	10	9.5	8
1	5.33	6	6.67	6.33	5.33
1.5	4	4.5	5	4.75	4
temps de retour (sec)					
/	6.5	7.5	8.2	7.33	9.5

4.4 Le FSDGS avec un système mono-transport à capacité limité

Dans un premier temps, nous avons considéré le problème de FSDGS avec un système multi-transports, que nous avons présenté en détail dans la section précédente. Cette supposition est très répandue dans la littérature, mais malheureusement, elle s'éloigne de la réalité industrielle, d'où la nécessité de prendre en compte des restrictions sur le nombre de véhicules utilisés pour l'acheminement des différents produits, limité à un seul produit par voyage. Dans cette partie, nous allons définir mathématiquement le problème en question, en exposant en détail les contraintes relatives à cette nouvelle restriction.

Le modèle mathématique linéaire en nombre entier mixte présenté dans cette section englobe l'ensemble des caractéristiques de notre problème et servira de base d'étude pour la globalité des expérimentations réalisées, qui seront exposées en détail dans le chapitre suivant. De plus, dans ces travaux, nous visons à minimiser simultanément le makespan en tant qu'indicateur de productivité et de niveau de service, ainsi que la consommation d'énergie en tant qu'indicateur environnemental et écologique. Pour ce faire, nous exposerons en détail la consommation d'énergie de l'ensemble du système en concordance avec le cas d'étude des bielles forgées.

4.4.1 La consommation d'énergie du système global

Dans cette section, nous aborderons de manière approfondie la question de la consommation d'énergie au sein de l'usine de bielles forgées. L'industrie de la métallurgie, tout comme les cimenteries, se distingue en tant qu'activité industrielle la plus énergivore, générant parallèlement des niveaux significatifs de certains gaz à effet de serre. Cette branche industrielle utilise des processus complexes pour l'élaboration et la transformation des métaux et alliages. Elle repose sur des méthodes physiques et un savoir-faire infallible

afin de prévenir tout incident et de réduire les quantités de déchets métalliques.

En analysant les caractéristiques de la consommation d'énergie, ainsi que les différentes machines impliquées, cette étude la divise en quatre catégories : la consommation liée à la phase de production, la phase de configuration, durant les temps d'inactivité des machines et enfin, celle liée à la phase de transport. En examinant chacune de ces zones, nous pouvons identifier des opportunités de réduction de la consommation d'énergie et d'amélioration de la durabilité de l'usine de bielles forgées.

4.4.1.1 La consommation d'énergie pendant la phase de production

Pendant la phase de production, diverses machines sont utilisées, telles qu'un laminoir, une presse hydraulique, et d'autres équipements. Malgré la production de produits identiques, ces machines présentent des variations de consommation d'énergie. Cette disparité peut être attribuée à plusieurs facteurs. Tout d'abord, les différences entre les types de machines, les fournisseurs d'énergie, et l'usure peuvent tous influencer la consommation d'énergie. Deuxièmement, la vitesse à laquelle chaque machine est programmée joue un rôle crucial dans cette consommation. Par exemple, faire fonctionner une machine à sa vitesse maximale peut réduire le temps de traitement, mais augmenter la consommation d'énergie.

Dans le processus de production des bielles forgées, l'opération de laminage et de matriçage nécessite l'intervention d'un opérateur sur les deux machines. En conséquence, le temps de traitement pour cette opération peut varier en fonction de l'expérience et du niveau de compétence de l'opérateur, plutôt que de la vitesse de la machine. Dans l'opération de laminage, par exemple, un opérateur expérimenté peut effectuer le travail plus rapidement qu'un opérateur inexpérimenté. Ainsi, le temps de traitement dans ce cas est fortement influencé par le taux d'apprentissage de l'opérateur, et à son tour impacte sur la consommation d'énergie en phase de production.

4.4.1.2 La consommation d'énergie durant les phases de configurations des machines

L'entreprise produit une variété de bielles de tailles différentes destinées à diverses applications. Pour s'adapter à ces variations, des ajustements sont nécessaires entre la production de chaque groupe de bielles. Ces ajustements incluent le nettoyage et la modification des paramètres, et leur durée peut varier de quelques minutes à une heure, en fonction des spécifications de chaque groupe. L'énergie consommée pendant ces ajustements est souvent gaspillée.

Dans l'opération de laminage, des ajustements sont effectués sur la température du laminoir ou l'écartement des outils. Il est important de noter que l'ajustement des paramètres des outils nécessite la construction d'un nouvel outil, ce qui prend du temps. Dans le processus de forgeage, la vitesse de la presse doit être ajustée, et les matrices doivent être chauffées à une température de 250 °C. Lors de l'opération de poinçonnage, la pression doit être régulée à l'aide d'une vanne de régulation de pression. En ce qui concerne le chauffage par induction, la température doit être régulée préalablement, et la puissance impliquée peut dépasser 100 kW.

La durée des ajustements est déterminée par les paramètres de réglage entre les deux groupes consécutifs de bielles. Par conséquent, la consommation d'énergie est influencée par la puissance de chaque machine et le temps nécessaire pour effectuer les modifications de production requises

4.4.1.3 La consommation d'énergie durant l'inactivité de la machine

La quantité d'énergie consommée par les machines pendant les périodes d'inactivité est directement liée à la durée pendant laquelle les machines attendent en raison de retards dans l'arrivée des tâches. Dans l'usine de bielles forgées, le temps d'arrêt est le plus fréquent pendant l'opération de forgeage, et la presse continue de fonctionner lorsque les intervalles entre les tâches sont courts, entraînant une consommation d'énergie supplémentaire. Selon Mouzon et al. (2007), les machines peuvent consommer jusqu'à 13% de l'énergie totale lorsqu'elles sont inactives. Bien que des chercheurs tels que C. Lu, Gao, et al. (2017) et E.-d. Jiang et Wang (2019) suggèrent d'éteindre les machines pendant les périodes d'inactivité pour minimiser la consommation d'énergie, le faire peut entraîner des défis significatifs. Dans de nombreux cas, l'énergie nécessaire pour redémarrer la machine est supérieure à l'énergie gaspillée pendant le mode d'inactivité. De plus, éteindre la machine peut entraîner des bugs dans le système, provoquant une perte de temps considérable et une diminution de l'efficacité de la production. Par conséquent, réduire la consommation d'énergie pendant les périodes d'inactivité nécessite une prise en compte minutieuse des caractéristiques uniques de chaque machine et du processus de production dans son ensemble.

4.4.1.4 La consommation d'énergie durant la phase de transport

La matière première est acheminée depuis l'entrepôt d'exportation jusqu'à la première machine. Une fois que la première machine a achevé son processus, le produit est dirigé vers la deuxième machine, et ainsi de suite jusqu'à son retour à l'entrepôt d'entrée. Un seul véhicule de transport est utilisé à la fois, et la distance entre les machines ainsi que

la vitesse de transport influent sur le temps de transport. Le véhicule de transport peut fonctionner à différentes vitesses, ce qui impacte à la fois le temps de transport et la consommation d'énergie. Si la vitesse de transmission est élevée, le temps de traitement diminuera, mais la consommation d'énergie augmentera.

4.4.2 Définition mathématique du problème

Le problème bi-objectif FSDGS présenté dans cette thèse est formulé sous la forme d'un modèle de programmation en nombres entiers mixtes. Une instance du problème comprend un ensemble de tâches $j = \{1, 2, \dots, N_g\}$, constituant un ensemble de groupes $g = \{1, 2, \dots, G\}$ exécutés sur un ensemble de machines $i = \{1, 2, \dots, m\}$. La préemption n'est pas autorisée, et chaque machine ne peut traiter qu'une tâche à la fois. L'ordre des tâches est fixé, et le problème implique la prise en compte simultanée de l'effet d'apprentissage et du temps de configuration dépendant de la séquence, ainsi que des temps de transport. Le temps de traitement de base de la tâche j sur la machine i dans le groupe g , noté P_{gji} , varie en fonction de l'effet d'apprentissage sur les machines M2 et M3. Au niveau des machines M1 et M4, le temps de traitement varie en fonction de la vitesse de la machine. Si la vitesse de la machine i est fixée à v_k ($k = 1, 2, 3$) pour la tâche j , alors le temps de traitement réel de la tâche j est P_{gji}/v_k noté P_{gjik} . Lorsqu'une tâche est effectuée à une vitesse élevée sur la première ou la dernière machine, son temps de traitement diminue, mais la consommation d'énergie augmente simultanément. Au niveau des machines M2 et M3, le temps de traitement diminuera selon le taux d'apprentissage, qui dépend de la position l de la tâche et du niveau d'apprentissage l^a .

Le temps de transport d'une tâche entre deux machines dépend également de la vitesse du véhicule de transport, qui peut être réglée à différentes vitesses. Le temps de transport diminue lorsque la vitesse du véhicule est élevée, mais la consommation d'énergie augmente simultanément. De plus, un seul véhicule de transport assure le déplacement entre les machines, et une seule tâche peut être transportée à la fois. La configuration du problème est illustrée dans la figure 4.9.

Les notations et les hypothèses utilisées pour la formulation du problème sont énumérées comme suit :

- Le temps de configuration du groupe h dépend de celui du groupe précédemment traité (c'est-à-dire le groupe t) exécuté sur la même machine (s_{hgi}).
- Le temps de configuration entre les tâches du même groupe est négligeable et est inclus dans le temps de traitement.
- L'hypothèse de la technologie de groupe est appliquée, ce qui signifie que, une fois l'exécution du groupe commencée, elle ne peut pas être interrompue par un autre

groupe.

- Les réglages sont censés être anticipés. Cela signifie que la configuration de la machine pour un groupe peut être commencée avant qu'aucune tâche appartenant au groupe ne soit disponible.
- Un groupe de référence nommé zéro ou groupe fictif est ajouté au modèle pour calculer le temps de configuration nécessaire pour le premier groupe sur chaque machine.

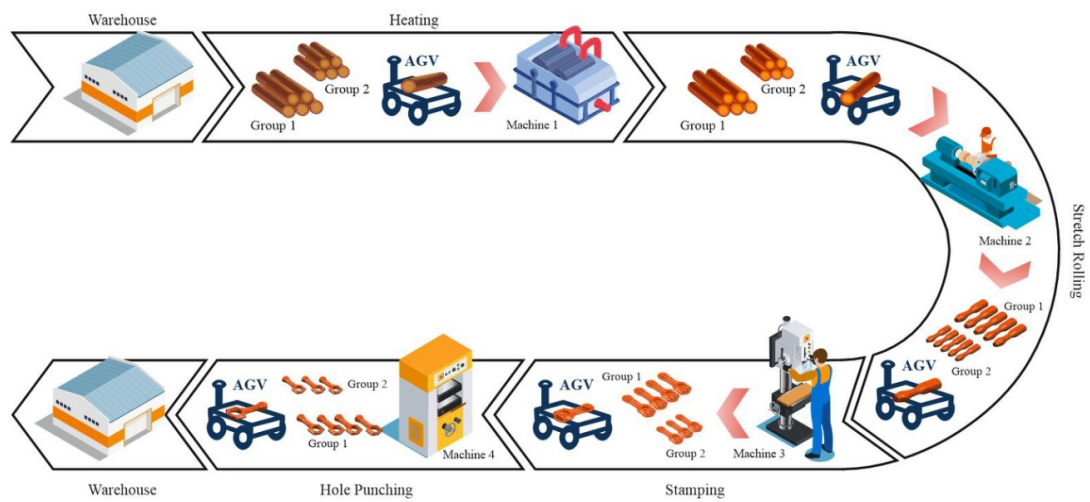


Figure 4.9: Représentation du problème FSDGS pour la fabrication de bielles forgées

Tableau 4.8: Les paramètres et variables du problème d'ordonnement

Les indices :

h, g l'indice des groupes, où $g = 1, \dots, G$
 j l'indice des jobs, où $j = 1, \dots, N_g$
 l l'indice des positions, où $l = 1, \dots, N_g$
 i l'indice des machines, où $i = 1, \dots, m$
 v l'indice des vitesses, où $v = 1, \dots, V$
 s l'indice de la vitesse du transporteur, où $s = 1, \dots, L$

Les paramètres

N_g le nombre total de jobs dans le groupe g
 G le nombre total de groupes
 m le nombre total de machines
 P_{gji} le temps de traitement basique du job j appartenant au groupe g sur la machine i
 s_{hgi} le temps de configuration pour traiter le groupe g sur la machine i , si le groupe h est le groupe précédent
 v_k facteur de vitesse de traitement, $k = 1, 2, 3$ pour les vitesses rapides, normales et lentes respectivement
 LR le taux d'apprentissage
 a^i l'indice d'apprentissage ($a = \log_2.LR \leq 0$)
 l^{a_i} l'effet d'apprentissage dans lème position sur la machine i
 T_{0s} Le temps de transport du job depuis l'entrepôt jusqu'à la première machine à la vitesse s
 T_{is} Le temps de transport du job de la machine i à la machine $i + 1$ à la vitesse s
 R_i Le temps nécessaire au véhicule de transport pour revenir de la machine $i + 1$ à la machine i
 I_{gi} Le temps d'inactivité du groupe g sur la machine i
 Ip_i La puissance au repos de la machine i par unité de temps
 Sp_i La puissance de réglage de la machine i par unité de temps
 Tp_s La puissance de transport du véhicule à la vitesse s par unité de temps
 Mp_{ik} La puissance de traitement de la machine i à la vitesse v_k par unité de temps
 M_j La masse du job j
 M un grand nombre positif

Les variables de décision

C_{gli} la date de fin du job exécuté en position l dans le groupe g sur la machine i
 TC_{gli} la date de fin du job exécuté en position l dans le groupe g , transporté de la machine i à la machine $i + 1$
 F_{gi} la date de fin du groupe g sur la machine i
 S_{gi} la date de début du groupe g sur la machine i
 D_{gi} La durée du groupe g dans la machine i
 $P_{gjl k}$ le temps de traitement actuel du job j exécuté en position l dans le groupe g sur la machine i à la vitesse v_k
 $X_{gjl k}$ la variable de décision binaire, qui prend la valeur 1 si le job j occupe la position l dans le groupe g à la vitesse v_k , 0 sinon
 U_{hg} la variable de décision binaire, qui prend la valeur 1 si le groupe g est traité immédiatement après le groupe h , 0 sinon
 Y_s la variable de décision binaire qui prend la valeur 1 si le véhicule de transport fonctionne à la vitesse s , 0 sinon
 $Cmax$ le makespan
 TEC la consommation d'énergie totale

Le modèle mathématique :

• **Fonctions objectif :**

$$\text{Min } Cmax \quad (4.18)$$

$$\text{Min } TEC \quad (4.19)$$

• **S.C :**

$$\sum_{l=1}^{N_g} \sum_{k=1}^3 X_{gjl k} = 1, \quad \forall j = 1, \dots, N_g \quad \forall g = 1, \dots, G \quad (4.20)$$

$$\sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} \leq 1, \quad \forall l = 1, \dots, N_g \quad \forall g = 1, \dots, G \quad (4.21)$$

$$\sum_{s=0}^L Y_s = 1 \quad (4.22)$$

$$P_{gjl k} = P_{gjk l}^{a_i}, \quad \forall j, l = 1, \dots, N_g \quad \forall i = 1, \dots, m \quad \forall g = 1, \dots, G \quad k = 1, 2, 3 \quad (4.23)$$

$$C_{g11} \geq \sum_{s=0}^L T_{0s} Y_s + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gj1 k} P_{gj11 k}, \quad \forall g = 1, \dots, G \quad (4.24)$$

$$C_{gli} = \begin{cases} C_{gl-1i} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} P_{gjl k} & Tr = 0, \quad l \geq 2 \\ \max\{C_{gl-1i}, TC_{gli-1}\} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} P_{gjl k} & Tr = 1, \quad l \geq 2 \end{cases} \quad (4.25)$$

$$TC_{gli} = \begin{cases} C_{gl-1i} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} P_{gjl k} + \sum_{s=0}^L T_{is} Y_s & Tr = 0, \quad l \geq 2 \\ \max\{C_{gl-1i} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} P_{gjl k}, TC_{gl-1i} + R_i\} + \sum_{s=0}^L T_{is} Y_s & Tr = 1, \quad l \geq 2 \end{cases} \quad (4.26)$$

$$C_{gli} \geq \max\{C_{gl-1i}, TC_{gli-1}\} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} P_{gjl k}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall l = 2, \dots, N_g \quad (4.27)$$

$$C_{gli} \geq C_{gli-1} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} P_{gjl k} + \sum_{s=0}^L T_{i-1s} Y_s, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall l = 1, \dots, N_g \quad (4.28)$$

$$C_{gli} \geq S_{gi} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gj1 k} P_{gj1 k}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (4.29)$$

$$TC_{g11} \geq C_{g11} + \sum_{s=0}^L T_{1s} Y_s \quad \forall g = 1, \dots, G \quad (4.30)$$

$$TC_{gli} \geq \max\{C_{gl-1i} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} P_{gjl k}, TC_{gl-1i} + R_i\} \quad (4.31)$$

$$+ \sum_{s=0}^L T_{is} Y_s, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall l = 2, \dots, N_g$$

$$TC_{gli} \geq TC_{gli-1} + \sum_{j=1}^{N_g} \sum_{k=1}^3 X_{gjl k} P_{gjl k} + \sum_{s=0}^L T_{is} Y_s, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall l = 1, \dots, N_g \quad (4.32)$$

$$TC_{gli} \geq C_{gli} \geq 0, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall l = 1, \dots, N_g \quad (4.33)$$

$$F_{gi} \geq C_{gN_g i}, \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (4.34)$$

$$S_{gi} \geq F_{hi} + s_{hgi} - M(1 - U_{hg}), \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad \forall h = 0, \dots, G, h \neq g \quad (4.35)$$

$$\sum_{h=0, h \neq g}^G U_{hg} = 1, \quad \forall g = 1, \dots, G \quad (4.36)$$

$$\sum_{g=1, h \neq g}^G U_{hg} \leq 1, \quad \forall h = 1, \dots, G \quad (4.37)$$

$$\sum_{g=1}^G U_{0g} = 1 \quad (4.38)$$

$$U_{hg} + U_{gh} \leq 1 \quad (4.39)$$

$$Cmax \geq TC_{gN_g m}, \quad \forall g = 1, \dots, G \quad (4.40)$$

$$Ec_p = \sum_{i=1}^m \sum_{g=1}^G \sum_{k=1}^3 D_{gi} M p_{ik} \quad (4.41)$$

$$Ec_s = \sum_{h=0}^G \sum_{g=1}^G \sum_{i=1}^m s_{hgi} U_{hg} S p_i \quad (4.42)$$

$$Ec_i = \sum_{g=1}^G \sum_{i=1}^m I_{gi} I p_i \quad (4.43)$$

$$Ec_t = \sum_{s=0}^L Cmax T p_s Y_s + \Delta E \quad (4.44)$$

$$\Delta E = Ec + Q = 2.8 \cdot 10^{-7} \cdot \sum_{i=1}^m M_i v^2 \quad (4.45)$$

$$TEC = Ec_p + Ec_s + Ec_i + Ec_t \quad (4.46)$$

$$C_{gli}, TC_{gli}, F_{gi}, S_{gi}, P_{gjl ik} \geq 0 \quad (4.47)$$

$$X_{gjl k}, U_{hg}, Y_s \in \{0, 1\} \quad (4.48)$$

La minimisation du makespan et de la consommation d'énergie du système est représentée respectivement par les équations (4.18) et (4.19). Les contraintes (4.20) et (4.21) indiquent que chaque job occupe une seule position dans le groupe auquel il appartient et est traité à une vitesse unique, et que chaque position du groupe est attribuée à l'un des jobs de ce groupe.

La contrainte (4.22) garantit que le véhicule de transport peut fonctionner à une seule vitesse à la fois. L'effet d'apprentissage du travailleur est modélisé par l'équation (4.23). La contrainte (4.24) détermine la date de fin du premier job, composé de son temps de traitement et du temps de transport depuis le stock jusqu'à la première machine. Le transport des jobs est effectué par un seul véhicule qui ne peut transporter qu'un produit à la fois. Les contraintes (4.25) et (4.26) sont intégrées dans le modèle pour déterminer les dates de fin des jobs avant et après le transport, où $Tr = 1$ représente un véhicule de transport occupé et 0 sinon. Le traitement du job sur une machine commence une fois que le job précédent est terminé, et également lorsque le job termine son transport de la machine i à la machine $i + 1$. La contrainte (4.27) est insérée dans le modèle pour justifier ce fait. La contrainte (4.28) est intégrée dans le modèle pour garantir que le traitement de chaque job commence après l'achèvement de son exécution sur la machine précédente, en tenant compte du transport du job de la machine $i - 1$ à la machine i . La contrainte (4.29) confirme que le traitement du job en première position de chaque groupe commence après le point de départ du groupe. La contrainte (4.30) définit le temps de réalisation du premier job une fois qu'il a été transporté de la première machine aux autres machines. La contrainte (4.31) indique qu'une machine ne peut pas exécuter deux jobs simultanément, y compris le temps de transport entre les machines et le temps de retour. La contrainte (4.32) assure le temps de réalisation du job une fois qu'il a été transporté de la machine $i - 1$ à la machine i . La date de fin du job incluant son temps de transport entre les machines est supérieure au temps de réalisation correspondant, qui doivent être supérieurs à zéro. La contrainte (4.33) maintient ces faits. L'ensemble de contraintes (4.34) et (4.35) est intégré dans le modèle pour le calcul du début et de la fin de chaque groupe sur chaque machine. Les contraintes de succession des groupes sont représentées par (4.36) et (4.37). Le groupe fictif est traité comme le premier groupe sur toutes les machines. La contrainte (4.38) confirme ce fait. La contrainte (4.39) exprime le fait que chaque groupe peut être traité soit avant, soit après un autre groupe. La contrainte (4.40) définit le makespan. L'ensemble d'équations (4.41)-(4.45) déterminent la consommation d'énergie pendant les phases de production, de configuration, d'inactivité et de transport de la machine, avec une énergie supplémentaire ΔE qui comprend l'énergie cinétique Ec et l'énergie thermique Q , ajoutée pour maintenir le véhicule de transport à une vitesse constante. La consommation d'énergie totale du système est calculée à l'aide de l'équation (4.46). Les ensembles de contraintes (4.47) et (4.48) définissent les variables de décision.

Le modèle proposé représente une approche novatrice visant à optimiser le processus de production en minimisant simultanément la consommation d'énergie de l'ensemble du système, tout en prenant en compte différentes vitesses de production et de transport. Cette approche revêt une importance particulière pour les industries axées sur la dura-

bilité et l'efficacité. En outre, le modèle intègre les contraintes réelles liées à l'utilisation d'un seul véhicule de transport entre les machines successives, tout en considérant l'effet d'apprentissage des opérateurs. Cela le rend mieux adapté aux environnements de fabrication réels, offrant ainsi des solutions plus précises et pratiques au problème d'optimisation multi-objectif.

4.4.3 Exemple numérique

Plusieurs expérimentations ont été conduites pour illustrer l'impact de l'effet d'apprentissage sur l'ordonnancement des jobs au sein des groupes et l'ordonnancement des groupes dans un problème de FSDGS. À cette fin, nous avons pris en considération l'exemple suivant représenté dans le tableau 4.9 et composé de deux machines, deux groupes et deux jobs dans chaque groupe. Les temps de traitement de chaque job au sein du groupe g sur les machines 1 et 2 sont indiqués par P_{gj1} et P_{gj2} respectivement. De plus, les temps de configuration sur chaque machine sont représentés par s_{hgi} .

Tableau 4.9: Les données du problème de l'exemple numérique

Groupes	Groupe 1		Groupe 2	
	Job 1	Job 2	Job 1	Job 2
P_{gj1}	100	60	120	60
P_{gj2}	50	40	60	50
s_{hg1}	s_{hg2}	s_{hg1}	s_{hg2}	
From/to				
G_0	6	10	8	16
G_1	/	/	15	80
G_2	12	60	/	/

Cinq scénarios ont été analysés pour illustrer l'impact de l'effet d'apprentissage sur l'ordonnancement des jobs et le séquençement des groupes. Dans ces scénarios, l'objectif est de minimiser exclusivement le makespan en faisant varier l'effet d'apprentissage entre les deux machines. Nous supposons que le temps de configuration entre les deux groupes est identique ($s_{hgi} = s_{ghi}$) dans les quatre premiers scénarios, et il varie uniquement dans le dernier scénario. La figure 4.10 présente l'ordonnancement optimal de l'exemple proposé.

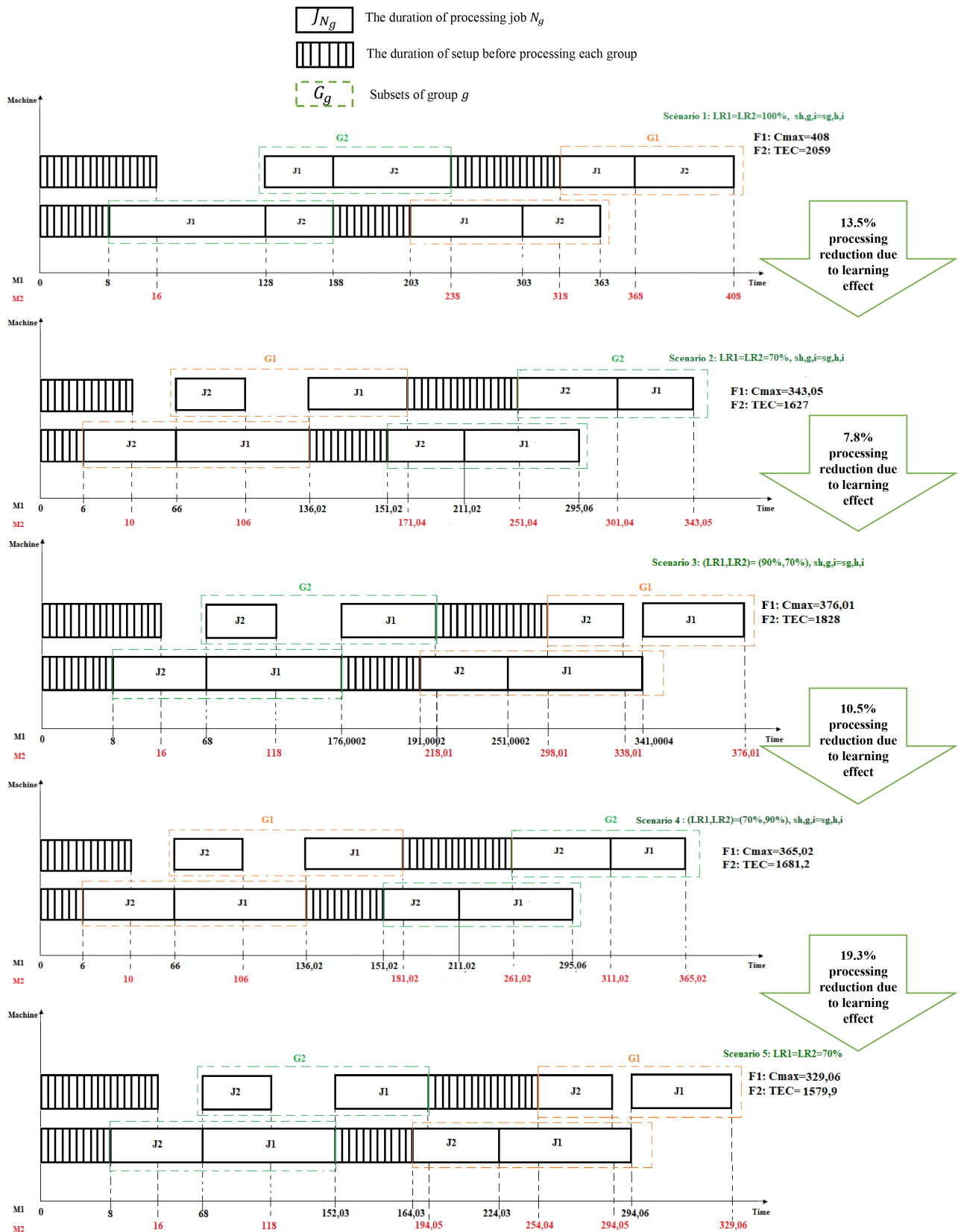


Figure 4.10: Impact de l'effet d'apprentissage sur la fonction objective et l'ordonnancement des jobs et des groupes

Comme illustré par la figure 3, dans le scénario 2, l'utilisation d'un taux d'apprentissage plus bas (70%) entraîne une réduction significative tant du makespan que de la consommation totale d'énergie. Cela modifie également l'ordonnancement optimale des groupes et des jobs. Pour atteindre cet objectif, le groupe comportant des jobs à temps de traitement élevé est positionné en dernière position, et les jobs avec des temps de traitement élevés dans les deux machines sont ordonnancés en dernier dans le groupe. Cette stratégie conduit à une réduction significative de leurs temps de traitement.

Dans les scénarios 3 et 4, le taux d'apprentissage diffère entre les deux machines. En attribuant le taux d'apprentissage le plus bas à la première machine, correspondant à un opérateur qualifié dans la réalité, nous pouvons minimiser davantage le makespan, étant donné que les temps de traitement des jobs sont plus élevés dans la première machine que dans la seconde. Lorsqu'un taux d'apprentissage de 70% est appliqué à la deuxième machine (scénario 3), l'ordonnancement des groupes reste inchangé dans le premier scénario, mais la séquence des jobs change. En revanche, lorsqu'un taux d'apprentissage de 70% est appliqué à M1 (scénario 4), la planification des groupes et des jobs change simultanément. Cette variation est due au fait que les temps de traitement des jobs du groupe G2 sur la première machine sont plus élevés que ceux du groupe G1, entraînant l'ordonnancement de G2 en dernière position. Ainsi, l'effet d'apprentissage influence à la fois l'ordonnancement des groupes et des jobs au sein de ces groupes.

Dans le scénario 5, la séquence des groupes évolue, même avec l'application d'un taux d'apprentissage de 70%. Cette modification découle de la dépendance du temps de configuration entre les groupes vis-à-vis de la séquence dans ce cas.

4.5 La méthode de pondération

Au cours des deux dernières décennies, de nombreuses recherches ont été entreprises pour optimiser simultanément plusieurs objectifs. La résolution d'un problème d'optimisation multi-objectif vise à déterminer la solution qui correspond au meilleur compromis entre les objectifs fixés. Par conséquent, l'aspect le plus délicat réside dans l'identification de la solution optimale de Pareto ou d'une approximation de celle-ci pour des problèmes complexes. Il existe différentes méthodes d'optimisation multi-objectif, notamment :

- Les méthodes agrégées, qui consistent à transformer les problèmes multi-objectifs en problèmes mono-objectifs. Parmi ces méthodes, on trouve les méthodes d'agrégation, la méthode LP-métrique, les méthodes de programmation par objectifs, et les méthodes e-contraintes.

- Les méthodes basées sur le principe de Pareto, telles que l'algorithme génétique avec objectifs multiples MOGA, NSGA II, etc.
- Les méthodes non agrégées et non fondées sur le principe de Pareto, comme l'algorithme VEGA.

Dans notre cas, nous cherchons à optimiser simultanément le makespan et la consommation d'énergie, en les combinant en un seul objectif à l'aide de la méthode d'agrégation. Cette dernière est l'une des approches de scalarisation les plus couramment utilisées, combinant de manière linéaire les fonctions objectives f_i en une seule fonction f .

$$f(x) = \sum_{i=1}^n w_i \cdot f_i(x) \quad (4.49)$$

Cette équation représente la fonction d'agrégation (de pondération), $f_i(x)$ sont les fonctions objectives à optimiser simultanément, et w_i sont les poids relatifs à chaque fonction objectif qui sont déterminés par le décideur où $0 \leq w_i \leq 1$ and $\sum w_i = 1$. Alors dans notre cas la formulation du problème en intégrant les paramètres et variables des deux objectifs avec : $f_1 = Cmax$ and $f_2 = TEC$, l'équation devient donc égale à :

$$Fo = w_1 \cdot f_1 + w_2 \cdot f_2 \quad (4.50)$$

Les deux fonctions objectives considérées dans cette étude ne sont pas à la même échelle, il faut donc procéder à la normalisation des objectifs (Talbi, 2009). La fonction pondérée précédente est transformée sous la forme suivante :

$$F(x) = \sum_{i=1}^n w_i \frac{f_i(x) - f_i^{min}}{f_i^{max} - f_i^{min}} \quad (4.51)$$

Où $f_i(x)$ représente les valeurs des fonctions objectives, f_i^{max} et f_i^{min} représentent les bornes supérieures et inférieures des différents objectifs, f_i et w_i est le poids du i ème objectif tel que défini précédemment (Doğan & Özgüner, 2005)

4.5.1 Exemple numérique

Cet exemple se compose de deux groupes, chacun comprenant quatre jobs, exécutés sur deux machines. L'objectif est d'optimiser deux critères : la durée d'exécution et la consommation totale d'énergie. Ces objectifs sont pondérés en fonction de leur importance relative, et quatre scénarios sont envisagés, chacun avec des vecteurs de pondération et des taux d'apprentissage différents pour la minimisation des deux fonctions objectives.

On suppose que la première machine fonctionne à des vitesses variables, et que la seconde machine est soumise à l'effet d'apprentissage. Les poids pour les objectifs makespan

et consommation totale d'énergie sont notés w_1 et w_2 , respectivement. Enfin, l'approche d'optimisation est mise en œuvre à l'aide d'IBM ILOG CPLEX Optimization Studio 12.8 pour résoudre les exemples numériques élaborés dans cette thèse.

Dans cet exemple, nous avons pris en compte les vitesses de fonctionnement variables dans la première machine. Le temps de traitement dépend du coefficient de vitesse v_k ($k = 1, 2, 3$) pour les vitesses rapides, normale et lente, respectivement. Si $v_k > v_{k'}$, alors $P_{gjik} < P_{gjik'}$, inversement $P_{gjik} \cdot Mp_{ik} > P_{gjik'} \cdot Mp_{ik'}$ (Mansouri et al., 2016). Cela indique que l'augmentation de la vitesse de la machine réduit le temps de traitement, mais augmente la consommation d'énergie. Les valeurs de cet exemple sont résumées dans le tableau 4.10.

Tableau 4.10: Les données de l'exemple numérique

Groups	Group1				Group2			
Jobs	job1	job2	job3	job4	job1	job2	job3	job4
P_{gj1}	15	20	30	35	70	40	55	75
P_{gj2}	55	45	35	40	55	25	35	60
k	1			2			3	
v_k	1.2			1			0.8	
Mp_{1k}	30			20			10	
Mp_{2k}	6			6			6	
	$shg1$		$shg2$		$shg1$		$shg2$	
From/to								
G_0	18		26		20		32	
G_1	/		/		90		20	
G_2	75		30		/		/	
Machines		M1				M2		
Sp_i		3.5				2.5		
Ip_i		3				2		

Dans cet exemple numérique, le modèle est analysé et les résultats sont résumés dans la figure 4.11, où le makespan et la consommation d'énergie sont minimisés.

Dans le scénario 1, nous avons accordé plus d'importance au premier objectif (80% vs. 20%). En fixant un taux d'apprentissage de 100% pour M2 et en utilisant des vitesses différentes pour M1, nous avons observé que les jobs du groupe 1 sont exécutés à une vitesse élevée pour réduire leur temps de traitement, tandis que pour le groupe 2, les jobs sont exécutés à différentes vitesses pour minimiser la consommation d'énergie de production. De plus, le groupe 1 est ordonnancé en premier, car le temps de préparation entre G_1 et G_2 est plus court que dans le cas où l'ordre est inversé.

En ajustant le taux d'apprentissage à 70% (scénario 2), nous avons constaté une diminution du makespan et des changements dans l'ordonnancement des jobs. Dans ce

cas, les jobs avec des temps de traitement élevés sont programmés en dernière position sur la machine M2. Il est important de noter que la plupart des jobs sont exécutés à grande vitesse sur la machine M1, entraînant une augmentation de la consommation d'énergie, même si les temps de traitement sont réduits sur la deuxième machine en raison de l'effet d'apprentissage.

Dans le scénario 3, le vecteur de poids est inversé (20% vs 80%). Nous avons observé que les jobs sont exécutés à faible vitesse pour réduire la consommation d'énergie de production, représentant 90% de la consommation d'énergie totale. En réduisant le taux d'apprentissage (scénario 4), nous avons réduit les critères d'optimisation et l'ordonnement des jobs a de nouveau changé. La séquence des groupes est restée inchangée, car le temps de préparation est resté constant dans les différents scénarios.

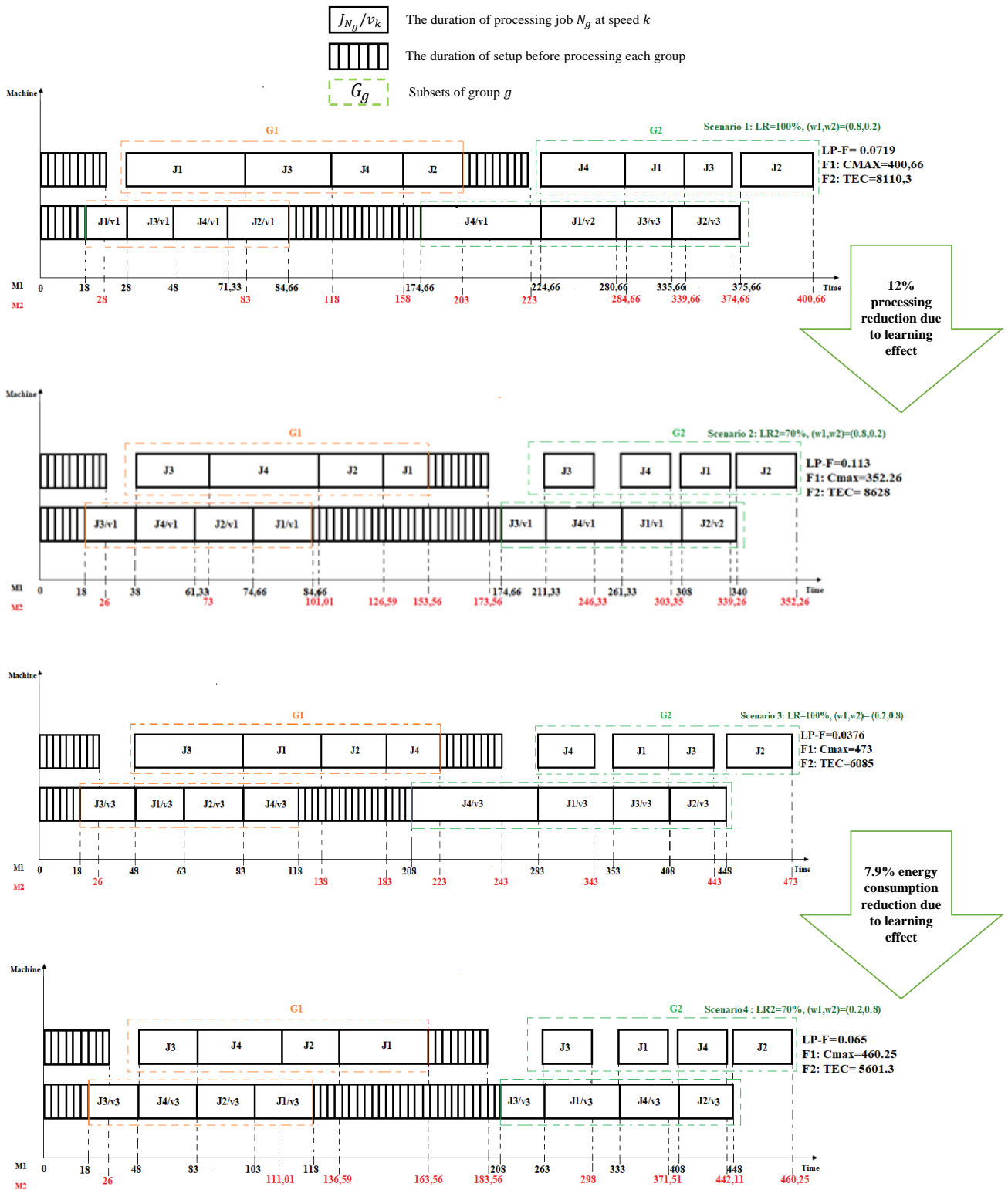


Figure 4.11: Impact des coefficients de pondération et de l'effet d'apprentissage sur la minimisation des fonctions objectives et l'ordonnancement des tâches et des groupes

4.6 Développement des bornes inférieures pour le problème de FSDGS

Un moyen courant d'évaluer l'efficacité des méthodes approximatives pour résoudre un problème NP-difficile consiste à développer des bornes inférieures. Ainsi, dans la suite de cette étude, nous nous efforçons de proposer des bornes inférieures pour le problème $Fm|prmu, fmls, s_{jk}, LR, t_s|Cmax, TEC$

4.6.1 La première borne inférieure $LB1$

Pour déterminer la première borne inférieure de la solution optimale du problème de FSDGS, nous avons introduit un modèle de programmation linéaire en nombres entiers mixtes relaxé. Ce modèle repose sur la technique de borne inférieure proposée par Keshavarz et Salmasi (2013). L'utilisation de ces bornes inférieures nous permet d'évaluer plus précisément les méthodes de résolutions proposées sur une large gamme de tailles de problèmes.

La première borne inférieure proposée repose sur les principes suivants :

- Chaque groupe est traité comme un seul job avec la sommation des temps de traitement des jobs du groupe.
- Cette approche permet de transformer les grandes instances du problème considéré en de petites instances d'un autre problème NP-difficile.
- Le modèle de la borne inférieure n'inclut que des variables de décision liées à la séquence des groupes et est capable de résoudre des problèmes plus vastes que le modèle original.
- En regroupant les tâches en groupes et en assouplissant les variables de décision liées à leur séquence et à leurs dates de fin, les éventuelles périodes d'inactivité entre les tâches sont ignorées.

Le modèle que nous avons élaboré offre simultanément une borne inférieure pour le makespan et la consommation d'énergie du système, démontrant ainsi son originalité et son efficacité. Une fois que les bornes inférieures des deux fonctions objectives ont été identifiées, elles sont pondérées en fonction de leur importance respective. Comme précédemment expliqué, le temps de traitement sur les première et quatrième machines dépend de la vitesse de production, tandis que sur les deuxième et troisième machines, il est influencé par l'effet d'apprentissage. Par conséquent, le modèle de borne inférieure proposé ordonnancera les jobs sur M2 et M3 selon la règle SPT (Shortest Processing Time) afin d'obtenir une meilleure réduction et de générer des solutions de borne inférieure de haute qualité.

En complément des paramètres exposés dans la section précédente, nous avons introduit les paramètres suivants afin d'enrichir le modèle de borne inférieure.

PT_{gji}	Temps de traitement du job j appartenant au groupe g dans la machine i avec l'effet d'apprentissage et la vitesse de traitement
AP_{gi}	La somme des temps de traitement des jobs du groupe g dans la machine i , i.e, $\sum_{j=1}^{N_g} PT_{gji}$
$MinP_{gi}$	Le temps de traitement minimum des jobs du groupe g dans la machine i , i.e, $\min_j \{PT_{gji}\}$.
Z_{gjk}	Variable de décision binaire qui prend la valeur 1 si le job j est traité à la vitesse v_k , sinon 0

Le modèle de la borne inférieure est le suivant :

• **Fonctions objectives :**

$$Min \quad Cmax \quad (4.52)$$

$$Min \quad TEC \quad (4.53)$$

• **S.C :**

Les contraintes (4.22), (4.36)-(4.46),

$$\sum_{k=1}^3 Z_{gjk} = 1, \quad \forall g = 1, \dots, G \quad \forall j = 1, \dots, N_g \quad (4.54)$$

$$PT_{gji} = \sum_{k=1}^3 \frac{P_{gji}}{v_k} \cdot l^{a_i} \cdot Z_{gjk} \quad \forall g = 1, \dots, G \quad \forall j, l = 1, \dots, N_g, \quad \forall i = 1, \dots, N_g \quad (4.55)$$

$$FT_{gi} = ST_{gi} + AP_{gi} \quad \forall g = 1, \dots, G \quad \forall i = 1, \dots, m \quad (4.56)$$

$$ST_{gi} \geq ST_{gi-1} + MinP_{gi-1} + \sum_{s=0}^L T_{(i-1)s} \cdot Y_s \quad \forall g = 1, \dots, G \quad \forall i = 2, \dots, m \quad MinP_{gi-1} \neq 0 \quad (4.57)$$

$$ST_{gi}, FT_{gi}, Cmax, TEC \geq 0 \quad (4.58)$$

$$U_{hg}, Z_{gjk} \in \{0, 1\} \quad (4.59)$$

La contrainte (4.54) stipule qu'un job doit être traité à une seule vitesse. Ensuite, la contrainte (4.55) est incorporée dans le modèle pour déterminer le temps de traitement d'un job en fonction de l'effet d'apprentissage et des vitesses de production. La contrainte (4.56) est chargée de calculer la date de fin de chaque groupe sur chaque machine, laquelle

est égale à la date de début additionnée à la somme des temps de traitement des jobs dans ce groupe. La date de début au plus tôt d'un groupe est déterminée en ajoutant au temps de transport entre les machines $i - 1$ et i le temps de traitement minimum du job appartenant au groupe, qui a été exécuté sur la machine précédente. Cette condition est garantie par la contrainte (4.57). Les contraintes antérieures demeurent inchangées, et leurs descriptions sont cohérentes avec celles du modèle original."

4.6.2 Validation de la première borne inférieure $LB1$

Afin de valider la première borne inférieure proposée, elle a été testée avec le modèle mathématique global développé pour de petites instances, en prenant en considération la variation des taux d'apprentissage, des différents axes, du temps de préparation et des coefficients de pondération. Les résultats de cette comparaison sont présentés dans le tableau 4.11. Au cours de ces tests, l'écart en pourcentage entre la borne inférieure et la solution optimale a été calculé à l'aide de la formule suivante :

$$100 \cdot \frac{\text{Lower bounding solution} - \text{Optimal solution}}{\text{Optimal solution}} \quad (4.60)$$

Le modèle mathématique relaxé de la première borne inférieure proposée $LB1$, a été comparé à la solution optimale pour les petites instances, comme illustré dans le tableau 4.11. L'écart en pourcentage a atteint une valeur maximale de 4,8% pour l'instance I1 lors de l'exécution avec différentes pondérations. À mesure que le nombre de jobs dans les groupes augmente (instances : I2 et I3), l'écart augmente également. Il a atteint une valeur maximale de 8,8% avec un taux d'apprentissage de 100%, de 9,7% avec un taux d'apprentissage de 70% sur la machine 2 et de 90% sur la machine 3, et de 9,6% lorsque les taux d'apprentissage étaient inversés entre les deux machines.

Les résultats du modèle de borne inférieure pour le problème FSDGS avec effet d'apprentissage, temps de transport, et vitesses de production et de transport variables sont prometteurs et raisonnables. Cependant, générer une solution optimale à l'aide du modèle de borne inférieure prend un temps considérable, surtout pour les grandes instances, d'où la nécessité de développer une seconde borne inférieure, qui sera détaillée dans la suite.

4.6.3 La deuxième borne inférieure $LB2$

Une deuxième borne inférieure a été développée pour évaluer la qualité des solutions liées au makespan et à la consommation totale d'énergie, générées par la méthode de résolution approchée basée sur le principe du recuit simulé pour une optimisation multi-objective. Cette seconde borne inférieure repose sur le même principe que la première, avec quelques détails distincts.

Tableau 4.11: Comparaison de la borne inférieure avec le modèle mathématique (LB1 vs le modèle MILP)

(w_1, w_2)	Instance	$N \times G$	Axes	LR1 = LR2 = 100%			LR1 = 70%, LR2 = 90%			LR1 = 90%, LR2 = 70%		
Code				Ave %GapCmax	Ave %GapTEC	PF%	Ave %GapCmax	Ave %GapTEC	PF%	Ave %GapCmax	Ave %GapTEC	PF%
SIC_11	(0.75,0.25)	I1	4 × 3	1	2.516	1.289	2.209	2.044	2.197	2.248	1.962	2.177
SIC_12				2	3.029	5.869	3.739	6.073	3.116	6.258	5.215	5.997
SIC_13				3	3.070	4.176	3.347	5.321	5.377	5.308	5.973	5.474
SIC_21		I2	5 × 3	1	1.051	0.669	0.956	1.899	0.635	0.503	0.935	0.611
SIC_22				2	9.938	0.669	7.621	6.743	5.684	3.241	5.068	3.698
SIC_23				3	9.552	6.552	8.802	11.042	9.797	11.400	4.591	9.698
SIC_31		I3	8 × 3	1	0.000	0.230	0.058	0.000	1.202	0.301	0.466	0.992
SIC_32				2	5.970	1.774	4.921	3.025	3.076	6.654	3.312	5.819
SIC_33				3	7.135	2.047	5.863	5.528	5.629	5.092	5.554	5.208
SIT_11	(0.25,0.75)	I1	4 × 3	1	4.111	1.702	2.304	7.415	3.742	5.355	2.254	3.029
SIT_12				2	7.899	3.869	4.877	4.185	5.794	5.392	5.431	5.736
SIT_13				3	1.983	4.182	3.632	2.037	8.805	7.113	7.076	6.889
SIT_21		I2	5 × 3	1	5.302	0.188	1.467	7.706	3.186	8.441	1.497	3.233
SIT_22				2	3.941	1.946	2.445	7.125	5.158	1.042	5.946	4.720
SIT_23				3	6.624	5.025	5.425	6.078	8.695	4.673	9.067	7.969
SIT_31		I3	8 × 3	1	0.000	0.451	0.338	0.000	2.147	0.000	2.905	2.179
SIT_32				2	3.877	2.509	2.851	4.310	4.996	4.310	5.475	5.184
SIT_33				3	1.854	4.432	3.788	2.512	9.476	7.735	10.415	8.333
SIF_11	(0.5,0.5)	I1	4 × 3	1	1.565	1.627	1.596	2.954	3.228	2.214	3.232	2.723
SIF_12				2	2.566	4.352	3.459	4.713	5.640	6.402	6.332	6.367
SIF_13				3	2.693	3.845	3.269	4.445	5.388	4.744	5.598	5.171
SIF_21		I2	5 × 3	1	0.512	1.349	0.931	1.665	2.650	4.554	1.902	3.228
SIF_22				2	4.690	1.393	3.042	10.454	7.049	1.298	6.358	3.828
SIF_23				3	5.804	5.906	5.855	5.232	8.897	7.065	7.152	7.460
SIF_31		I3	8 × 3	1	0.491	0.141	0.316	0.757	0.778	0.235	2.000	1.118
SIF_32				2	5.607	1.978	3.793	5.212	2.695	3.954	3.324	4.596
SIF_33				3	6.625	2.733	4.679	4.599	6.224	8.722	6.171	7.447

SIC_11 : le S correspond aux petites instances, I aux mot instances et C a l'objectif Cmax. Le numéro 1 fait référence à la première instance, et le second 1 signifie l'axe 1.

PF% : Le pourcentage d'écart de la fonction de pondération PF.

Chaque groupe est traité comme un job unique, et son temps de traitement est calculé comme la somme des temps de traitement de tous les jobs individuels au sein du groupe, utilisant la formule précédente (4.55). Pour la borne inférieure du makespan, la valeur de v_k est fixée à 1 (vitesse rapide) pour maximiser la réduction attendue des temps de traitement. En revanche, pour la borne inférieure de la consommation totale d'énergie, v_k est fixé à 3 (vitesse lente) pour minimiser la consommation d'énergie de traitement.

Concernant le temps de configuration entre les groupes, les valeurs minimales sont prises en compte pour générer la borne inférieure. Pour chaque groupe g et chaque machine i , nous définissons les termes suivants :

$$MinS_{gi} = \min_{h, h \neq g} S_{hgi} \quad g = 1, 2, \dots, G \quad i = 1, 2, \dots, m \quad (4.61)$$

$$Dd_i = \max\{Dd_{i-1} + \min_g \{P_{gi-1}\} + T_{(i-1)s}, \min_g \{S_{gi}\}\} \quad i = 2, 3, \dots, m \quad (4.62)$$

Avec

$$Dd_1 = \max\{\min_g S_{g1}, T_{0s}\} \quad (4.63)$$

Où $MinS_{gi}$ est le temps de configuration minimum pour traiter le groupe g sur la machine i , et Dd_i est le temps de démarrage au plus tôt du premier groupe traité sur la machine i . Par conséquent, en fixant les vitesses à un régime rapide, la borne inférieure du C_{\max} est définie de la manière suivante :

$$LB2_{C_{\max}} = \max_{1 \leq i \leq m} \{LB_i\} + T_{m3} \quad (4.64)$$

Où

$$LB_i = Dd_i + \sum_{g=1}^G AP_{gi} + \sum_{g=1}^G \min_h s_{hgi} \quad i = 1, 2, \dots, m \quad (4.65)$$

En agrégeant les temps de traitement des jobs dans les groupes, les temps d'inactivité de la machine sont négligés, et l'énergie consommée qui leur est associée n'est donc pas prise en compte. Par conséquent, pour déterminer la borne inférieure de la consommation totale d'énergie du système, il est nécessaire de déterminer d'abord les bornes inférieures de chaque phase de traitement.

$$LB_{EC_p} = \sum_{i=1}^m \sum_{g=1}^G AP_{gi} \cdot MP_{ik} \quad (4.66)$$

$$LB_{EC_s} = \sum_{i=1}^m \sum_{g=1}^G MinS_{gi} \cdot Sp_i \quad (4.67)$$

$$LB_{EC_i} = \sum_{i=1}^m (Dd_i - MinS_{1i}) \cdot Ip_i \quad (4.68)$$

$$LB_{EC_t} = \left(\sum_{i=1}^m Tp_s \cdot T_{is} \cdot Y_s + T_{0s} \cdot Tp_s \cdot Y_s + \sum_{i=1}^m R_i \cdot Tp_s \cdot Y_s \right) \times n \quad (4.69)$$

Et la borne inférieure de TEC est définie comme suit :

$$LB2_{TEC} = LB_{EC_p} + LB_{EC_s} + LB_{EC_i} + LB_{EC_t} \quad (4.70)$$

La borne inférieure de la fonction pondérée sera égale à :

$$LB2_{Fo} = w_1 \cdot LB2_{C_{\max}} + w_2 \cdot LB2_{TEC} \quad (4.71)$$

4.6.4 Validation de la deuxième borne inférieure $LB2$

Pour démontrer l'efficacité de la deuxième borne inférieure proposée, nous l'avons comparée aux moyennes instances où la solution optimale était connue. Nous avons calculé l'écart en pourcentage entre la borne inférieure $LB2$ et la solution optimale obtenue par le solveur, comme illustré dans la figure 4.12.

Comme on peut le voir à travers ces graphiques, le pourcentage d'erreur varie de 2% à 12% pour les instances I4 et I5, résolues de manière optimale. Ces valeurs augmentent jusqu'à un maximum de 16% pour l'instance I6. Cette augmentation est justifiée puisque, dans ce cas, la borne inférieure $LB2$ est comparée à la solution obtenue avec l'algorithme de résolution, entraînant une augmentation du pourcentage d'erreur. Cependant, ces valeurs restent raisonnables compte tenu de la variabilité des paramètres du problème. Salmasi et al. (2010) ont enregistré un pourcentage d'erreur de 16% pour la borne inférieure proposée dans un problème mono-objectif de FSDGS classique avec minimisation du makespan.

La deuxième borne inférieure proposée s'avère très efficace, surpassant la première borne et le modèle mathématique en termes de temps CPU, nécessitant moins d'une seconde pour résoudre les instances moyennes et grandes.

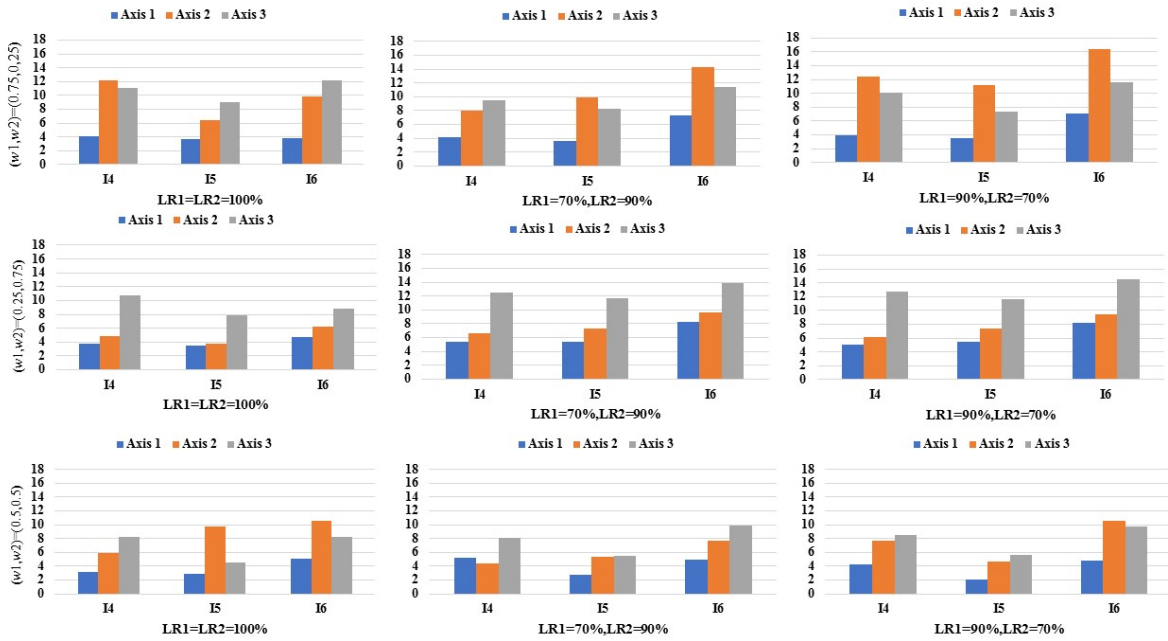


Figure 4.12: L'efficacité de la deuxième borne inférieure

4.7 Conclusion

Ce chapitre s'est initialement concentré sur l'intégration de la variante de transport pour le problème de FSDGS avec effet d'apprentissage. Cela a conduit à la création d'un nouveau modèle mathématique linéaire en nombres entiers mixtes, bi-objectif visant à minimiser à la fois le makespan et la consommation totale d'énergie du système. Ce modèle englobe toutes les contraintes prises en considération dans cette étude, liées à l'effet d'apprentissage et à son impact sur le processus d'ordonnancement des jobs. Il prend en compte également l'influence des temps de configuration dépendant de la séquence sur l'agencement des groupes, le tout étayé par un exemple numérique. Dans un second temps, nous avons analysé l'effet de la contrainte de transport sur le séquençement des jobs sous diverses hypothèses (système multi-transports et mono-transport). Enfin, nous avons abordé le schéma décisionnel relatif à la sélection des différentes vitesses de traitement et de transport.

Le problème de FSDGS étant intrinsèquement difficile, l'introduction de ces variantes accroît sa complexité. De plus, une optimisation multiobjectif est abordée dans ces recherches. Par conséquent, nous nous sommes penchés sur l'adaptation de la méthode de pondération avec procédure de normalisation pour déterminer la solution la plus conforme aux deux objectifs considérés. À cet effet, un second exemple numérique est élaboré afin de démontrer l'influence des différents coefficients de pondération sur les deux critères

d'optimisation ainsi que sur l'ordonnancement des jobs et des groupes.

Ces recherches ont été renforcées par un cas concret portant sur la fabrication des bielles forgées dans une entreprise de métallurgie, car une étude théorique gagne en crédibilité lorsqu'elle est liée à la pratique et au contexte industriel actuel. Dans le cadre de la validation et de l'évaluation de l'efficacité des méthodes de résolution utilisées pour l'optimisation du problème de FSDGS, deux bornes inférieures ont été développées. La première consiste en une version relaxée du modèle mathématique, tandis que la seconde repose sur l'heuristique SPT pour le séquençement des jobs dans les groupes. Le tout s'appuie sur la stratégie de variation des vitesses de traitement et de transport.

Plusieurs expérimentations ont été réalisées dans le but de valider les deux bornes inférieures proposées. Les résultats obtenus ont confirmé l'efficacité de la première borne inférieure, en particulier pour les petites instances, en comparaison avec le modèle mathématique. Cependant, avec une augmentation significative de la taille des instances, le modèle mathématique de LB1 ne parvient pas à fournir des solutions en un temps raisonnable. Dans ce contexte, la seconde borne inférieure intervient et démontre son efficacité par rapport au MILP pour les instances de taille moyenne. Toutes ces avancées scientifiques, en lien avec notre sujet de recherche, ont ouvert la voie au développement d'une méthode de résolution approchée multiobjectif basée sur l'algorithme du recuit simulé MOSA, qui fera l'objet d'étude dans le chapitre suivant.

Chapitre 5

Algorithme MOSA pour l'ordonnancement du problème FSDGS avec effet d'apprentissage des opérateurs

5.1 Introduction

Le flow shop constitue un domaine crucial au sein des problématiques d'ordonnancement, offrant un cadre particulièrement adapté à l'environnement industriel. Sa complexité et sa pertinence pratique ont suscité un vif intérêt. Toutefois, il est fréquemment abordé ou étudié sous des hypothèses restrictives qui ne reflètent pas la réalité industrielle. Ces hypothèses vont de la négligence des temps de configuration à la supposition de temps de traitement constants, voire à l'ignorance des temps de transport.

Cette approche limitée conduit à une gestion défailante des systèmes de production, engendrant des solutions déconnectées des objectifs préalablement définis. Elle s'accompagne fréquemment de coûts supplémentaires et de l'insatisfaction des clients. La prise en compte complète de ces contraintes s'avère difficile, et la considération de l'ensemble des paramètres et caractéristiques du problème représente un défi.

Les problèmes d'ordonnancement avec des aspects d'apprentissage et des temps de transport ajoutent une complexité supplémentaire. Il en résulte un manque de méthodes analytiques développées, notamment en raison des nombreuses variables en jeu. Cependant, les modèles mathématiques permettent d'exploiter les propriétés polyédriques de certains modèles d'optimisation, y compris ceux liés à l'ordonnancement. Cela ouvre la voie à l'élaboration de stratégies de solution efficaces.

Par ailleurs, une approche algorithmique alternative peut également être envisagée

pour instaurer des techniques d'optimisation robustes, offrant ainsi une diversité de méthodes pour aborder ces défis complexes.

En examinant les recherches menées dans le domaine de l'ordonnement, il a été clairement établi que le problème du flow shop de permutation est un problème NP-difficile (M. L. Pinedo & Pinedo, 2016). Par conséquent, le problème FSDGS avec effet d'apprentissage, visant à minimiser le makespan et la consommation d'énergie, est également NP-difficile. Pour résoudre ce problème, nous avons d'abord proposé une programmation linéaire en nombres entiers mixtes. Cependant, en raison de la complexité des défis rencontrés dans les entreprises manufacturières, qui font généralement face à des problèmes à grande échelle avec des temps de calcul souvent interminables, les méthodes exactes ne peuvent pas fournir de bonnes solutions dans un laps de temps raisonnable.

Par conséquent, une approche pratique consiste à recourir à des méthodes approchées pour obtenir des solutions satisfaisantes et de bonne qualité avec des temps de résolution réduits, sans garantir pour autant l'optimalité de la solution. Les métaheuristiques s'avèrent être des méthodes adéquates pour résoudre les problèmes d'optimisation difficiles.

Pour aborder cette problématique, nous adaptons la métaheuristique MOSA, qui combine les principes du recuit simulé avec des techniques d'optimisation multi-objectif pour explorer l'espace des solutions et identifier les meilleures solutions de compromis entre les objectifs contradictoires, à savoir le makespan et la consommation d'énergie. Au cours du processus d'optimisation, l'algorithme utilise un schéma de refroidissement pour contrôler l'acceptation des nouvelles solutions en fonction des valeurs de leur fonction objectif et de la température actuelle. Il exploite également diverses stratégies de recherche, telles que la mutation, pour générer de nouvelles solutions candidates et échapper aux optima locaux. Dans ce chapitre, une analyse de sensibilité est réalisée afin de déterminer les valeurs optimales des paramètres de cet algorithme et d'examiner leurs évolutions. Par la suite, une procédure de recherche locale est suggérée dans le dessein de surmonter les limitations inhérentes aux éléments individuels de l'algorithme MOSA et d'en améliorer le rendement.

La suite de ce chapitre est organisée comme suit : tout d'abord, nous exposons la métaheuristique MOSA. Dans la deuxième section, nous détaillons la mise en œuvre de l'algorithme que nous proposons, incluant ses caractéristiques spécifiques et les différents paramètres influant sur son bon fonctionnement. Nous abordons également l'application de l'algorithme, accompagnée d'une procédure de recherche locale visant à améliorer le fonctionnement de la métaheuristique. Dans la troisième section, nous réalisons une analyse de sensibilité de l'algorithme MOSA proposé afin de déterminer les valeurs adéquate de ces paramètres. Ensuite, la quatrième section présente les résultats issus d'expérimentations.

tations sur la méthode de résolution, en variant les différents paramètres de l'étude, et les enrichit par des comparaisons entre l'algorithme MOSA, MOSA avec recherche locale et les bornes inférieures proposées. Nous clôturons ce chapitre par une brève synthèse des résultats obtenus et une conclusion.

5.2 Présentation de l'algorithme MOSA

MOSA a été proposée par Ulungu et al. (1999) et repose sur le principe du recuit simulé pour trouver une surface de compromis. Cette métaheuristique se distingue par sa facilité de mise en œuvre et sa rapidité d'exécution par rapport à d'autres métaheursistiques. En analogie avec la thermodynamique, le recuit simulé est fondé sur le principe d'accepter une transformation locale qui génère une dégradation de la solution actuelle, en attendant une amélioration ultérieure. Cela évite ainsi de rester bloqué dans un minimum local. En métallurgie, des transformations élémentaires sont appliquées au système thermodynamique en réduisant lentement la température tout au long du processus de recuit, afin d'atteindre un état d'équilibre correspondant à une énergie minimale. Dans l'algorithme de recuit simulé, cette énergie correspond à la fonction objectif à optimiser, et l'état d'équilibre à la solution du problème.

La procédure de recuit simulé débute par une solution initiale générée de manière aléatoire. Ensuite, une modification locale est appliquée à cette solution pour créer une solution dans le voisinage de la solution existante. La fonction objectif de la nouvelle solution est comparée à celle de la solution actuelle pour déterminer si elle est meilleure ou non. Dans un problème de minimisation, si la différence entre les deux fonctions objectif est inférieure à zéro, alors la nouvelle solution est automatiquement acceptée et devient la solution courante pour l'itération suivante. De plus, l'algorithme peut accepter une solution dégradant la fonction objectif sous certaines conditions.

Le choix d'adapter cet algorithme se justifie à plusieurs égards, notamment sa capacité à traiter de manière agile des fonctions objectifs complexes, qu'elles soient linéaires, non linéaires ou multimodales, avec de nombreuses variables de décision. De plus, il peut converger rapidement vers un ensemble diversifié de solutions de Pareto optimales, offrant ainsi une large gamme d'options de compromis pour les décideurs. En outre, l'algorithme est facile à implémenter, et ses paramètres peuvent être ajustés pour répondre au problème spécifique en cours.

5.3 le codage de la solution

Le codage joue un rôle essentiel dans la détermination de la qualité de la solution et de la capacité de calcul. Pour optimiser un problème donné à l'aide d'une métaheuristique, on le représente sous la forme d'un chromosome numérique. Pour l'algorithme MOSA que nous proposons, les solutions sont exprimées sous forme de matrices de dimensions $(2G + 1)(Nmax + 1)$, où $Nmax = \max \{N_g, G\}$ et N_g représente le nombre de jobs dans chaque groupe $g = 1, 2, \dots, G$, tel que défini dans le chapitre 3.

Les G premières lignes de la matrice représentent les vecteurs de permutation π^g qui déterminent la séquence des jobs au sein de chaque groupe, suivies de $2G$ lignes contenant les vitesses de traitement de chaque job dans chaque groupe, représentées par le vecteur $[k]^g$. Enfin, la dernière ligne indique le vecteur de permutation Ω qui détermine la séquence des groupes et la vitesse de transport s . La matrice est structurée de la manière suivante :

$$\begin{bmatrix} \pi_1^1 & \dots & \pi_{N_1}^1 \\ \pi_1^g & \dots & \pi_{N_g}^g \\ \vdots & \ddots & \vdots \\ \pi_1^G & \dots & \pi_{N_G}^G \\ \hline [k]_1^1 & \dots & [k]_{N_1}^1 \\ [k]_1^g & \dots & [k]_{N_g}^g \\ \vdots & \ddots & \vdots \\ [k]_1^G & \dots & [k]_{N_G}^G \\ \hline \Omega_1 & \dots & \Omega_G, s \end{bmatrix} \quad (5.1)$$

Chaque ligne r ($r = 1, 2, \dots, G$) du sous-chromosome représente un ordonnancement spécifique correspondant à chaque groupe, les lignes suivantes $r = G + 1, \dots, 2G$ de la matrice partitionnée représentent les vitesses de traitement utilisées par chaque tâche dans chaque groupe, la dernière ligne $2G + 1$ identifie la séquence des groupes et la vitesse de transport s utilisée pour acheminer les tâches entre les machines. Pour éclaircir le schéma d'encodage utilisé, une solution réalisable du problème dans lequel $G = 3$ et $Nmax = 5$ est

représentée comme suit :

$$\begin{bmatrix} 2 & 1 & 3 & 0 & 0 & 0 \\ 5 & 3 & 2 & 4 & 1 & 0 \\ \hline 2 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 2 & 0 & 0 & 0 \\ 3 & 1 & 1 & 2 & 2 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 \\ \hline 2 & 3 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

Le sous-chromosomes 1 à 3 représentent l'ordonnement des tâches au sein des groupes (par exemple, l'ordonnement des tâches au sein du groupe 1 est 2-1-3, l'ordonnement des tâches au sein du groupe 2 est 5-3-2-4-1 et l'ordonnement du groupe 3 est 2-1), le sous-chromosome $r = G + 1, \dots, 2G$ définit la vitesse de traitement de chaque tâche dans chaque groupe (c'est-à-dire que pour le groupe 1, les tâches (2-1-3) sont exécutées à la vitesse (1,1,2) respectivement, pour le groupe 2, les tâches (5-3-2-4-1) sont exécutées à la vitesse (3,1,1,2,2) respectivement et pour le groupe 3, les vitesses (3,1) sont utilisées pour le traitement des tâches (2-1) respectivement). Enfin, la dernière ligne du chromosome $r = 2G + 1$ indique le séquençement du groupe $\Omega = 2 - 3 - 1$ et la vitesse de transport utilisée $s = 1$. Les nombres égaux à zéro ne sont pas utilisés dans le processus de recuit ou dans le décodage de la solution.

5.4 Solution initiale et voisinage

5.4.1 Solution initiale

La solution initiale est obtenue dans un premier temps en générant aléatoirement la séquence des groupes et les jobs dans chaque groupe, et dans un deuxième temps en sélectionnant aléatoirement les vitesses utilisées pendant la phase de traitement et de transport des jobs, comme indiqué dans l'exemple précédent. Une fois le chromosome correctement formulé, une procédure de génération de voisinage est définie.

5.4.2 Génération du voisinage

Une solution voisine est obtenue en utilisant une opération de permutation ou d'insertion sur la séquence des groupes et des jobs. Une première procédure de génération de voisinage NGP1 consiste à sélectionner aléatoirement un groupe et un job appartenant à ce groupe et à les placer dans une autre position choisie aléatoirement. Dans le deuxième mécanisme NGP2, les positions des deux groupes choisis au hasard sont inversées, et la

même technique est utilisée pour deux jobs appartenant aux deux groupes choisis au hasard. Pour le troisième mécanisme NGP3, un point de coupe généré aléatoirement est appliqué entre les groupes créant deux sous-ensembles positionnés à gauche et à droite de ce point. Dans ce cas, le mouvement est effectué en inversant la position des deux sous-ensembles, les groupes des sous-ensembles situés à droite du point de coupe sont placés à gauche du point et vice versa. Ensuite, une position du job choisie au hasard dans chaque groupe du sous-ensemble le plus petit est à nouveau définie.

Quel que soit le mécanisme utilisé pour générer les solutions voisines, la vitesse de traitement du job sélectionné au hasard change comme suit : si le job est exécuté dans le mode de vitesse le plus rapide ($k = 1$), la vitesse sera réduite d'un niveau (de rapide à moyen, $k = 2$). Dans le cas où le job est exécuté à une vitesse moyenne ($k = 2$), nous avons deux possibilités, soit la vitesse est ralentie ($k = 3$) avec une probabilité de $p = 0.5$, soit la vitesse est augmentée ($k = 1$) avec une probabilité de $1 - p$. D'autre part, si au début la vitesse du job sélectionné aléatoirement est lente ($k = 3$), alors elle est augmentée d'un niveau (de lente à moyenne, $k = 2$). La même procédure est appliquée pour déterminer les vitesses de transport utilisées pour une solution voisine.

Compte tenu du schéma de codage par permutation utilisé, nous mettons en œuvre la deuxième procédure de génération du voisinage (NGP2). Le premier mouvement consiste à sélectionner aléatoirement deux groupes et à inverser leurs positions, comme le montre la figure . Le deuxième mouvement consiste à sélectionner deux jobs au hasard dans les deux groupes et à appliquer la même procédure que pour le premier mouvement. Le troisième mouvement consiste à ajuster les vitesses de traitement des jobs sélectionnés aléatoirement en utilisant la procédure décrite précédemment. Enfin, le quatrième mouvement consiste à ajuster la vitesse de transport, en utilisant la même procédure que pour les vitesses de traitement.

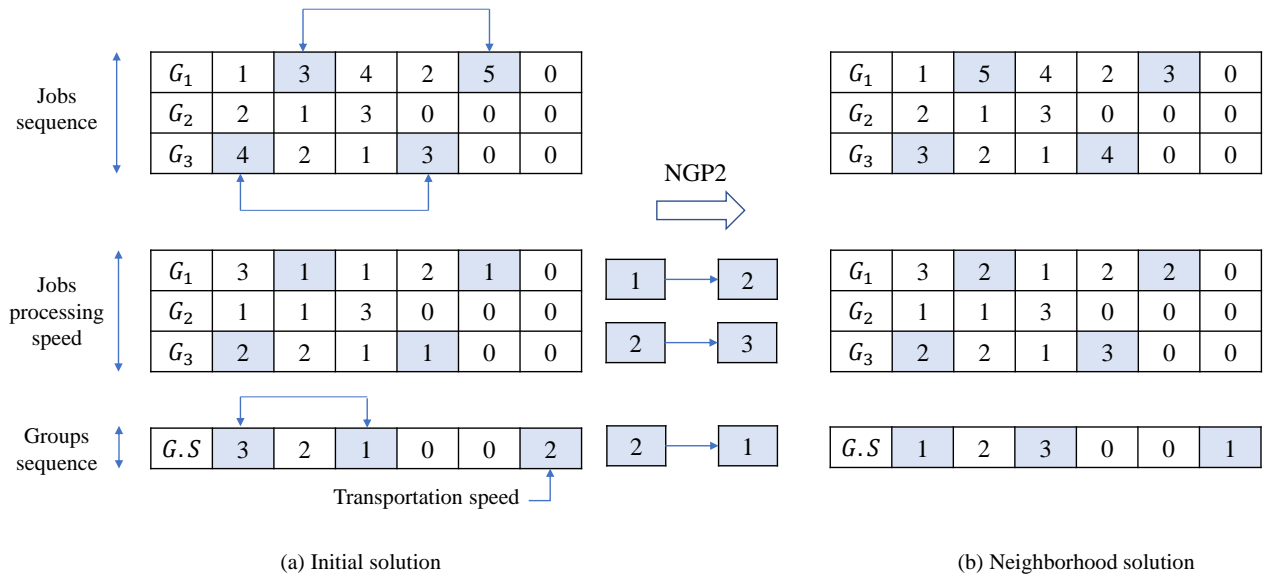


Figure 5.1: La procédure de génération du voisinage (NGP2)

5.5 Les paramètres de l'algorithme MOSA

Autre que la détermination de la solution initiale et de la procédure de voisinage, l'algorithme dépend de ces deux principales procédures :

- Le processus de refroidissement : ce mécanisme permet de définir la température à chaque itération de l'algorithme, son importance est assez élevée, car il joue un rôle important dans l'efficacité et l'efficience de l'algorithme.
- La probabilité d'acceptation : elle présente un élément majeur pour l'algorithme qui octroie la possibilité de sélectionner une solution voisine dégradant la fonction objective.

Ces procédures indispensables au bon fonctionnement de l'algorithme dépendent de plusieurs paramètres, qui vont être détaillés dans ce qui suit.

5.5.1 Le processus de refroidissement

La définition de la température T_i à chaque étape de l'algorithme, élaborée par le programme de refroidissement, exerce une influence déterminante sur le succès de l'algorithme d'optimisation. Le choix judicieux du programme de refroidissement revêt ainsi une importance cruciale pour les performances du recuit simulé.

L'élaboration d'un programme de refroidissement nécessite la considération attentive de plusieurs paramètres tels que la température initiale, l'état d'équilibre, une fonction de refroidissement, et la température finale, qui fixe les critères d'arrêt. L'initialisation appropriée de chacun de ces paramètres joue un rôle essentiel dans l'efficacité globale du recuit simulé. Des directives précises pour cette initialisation sont présentées ci-dessous.

5.5.1.1 La température initiale

Si la température initiale T_0 est très élevée, la recherche tendra à adopter une approche locale aléatoire. En revanche, une température initiale très basse orientera la recherche vers un schéma similaire à celui d'un algorithme local d'amélioration initiale. Ainsi, il est impératif de trouver un juste équilibre entre ces deux extrêmes. La température initiale ne doit pas être excessivement élevée afin d'éviter une période prolongée de recherche aléatoire, mais elle doit rester suffisamment élevée pour autoriser des déplacements vers des états voisins.

5.5.1.2 Le nombre d'itérations à chaque température

Afin d'atteindre un état d'équilibre à chaque température, il est nécessaire d'effectuer un nombre suffisant de transitions ou mouvements. Selon la théorie, le nombre d'itérations requis à chaque température peut être exponentiel par rapport à la taille du problème, ce qui représente une stratégie difficile à mettre en œuvre dans la pratique. Ainsi, le nombre d'itérations doit être déterminé en fonction de la taille de l'instance du problème, en particulier en proportion de la taille du voisinage.

5.5.1.3 La fonction de refroidissement

L'algorithme du recuit simulé implique une diminution progressive de la température, assurant que $T_i > 0, \forall i$. Un compromis constant entre la qualité des solutions obtenues et la rapidité du schéma de refroidissement doit être trouvé. Si la réduction de la température est effectuée lentement, des solutions de meilleure qualité émergent, mais cela entraîne une augmentation du temps de calcul. Diverses méthodes peuvent être employées pour abaisser la température, parmi lesquelles la distribution géométrique est la plus couramment utilisée. Dans cette étude, la réduction de la température est formulée selon l'équation $T_{i+1} = \alpha.T_i$, où α représente le taux de diminution, généralement choisi dans l'intervalle $[0.5, 0.99]$.

5.5.1.4 Le critère d'arrêt

Les paramètres définissant la cessation de la recherche ou l'arrêt de l'algorithme sont associés au point où aucune transformation n'est jugée acceptable, une fois que le critère

est atteint. Dans le cadre de l'algorithme présenté dans cette étude, l'arrêt intervient lorsque T_i devient inférieur à la température finale prédéterminée TF qui doit être basse (par exemple, $T_{min} = 0,01$), ou après un nombre spécifié d'itérations sans amélioration N .

5.5.2 La probabilité d'acceptation

Cet algorithme se distingue par sa capacité à accepter une solution non améliorée, permettant d'éviter de rester piégé dans des minima locaux. La probabilité d'accepter un voisin non amélioré est proportionnelle à la température T et inversement proportionnelle à la variation de la fonction objectif ΔE . Deux fonctions sont employées pour calculer cette probabilité : la fonction de distribution de Cauchy et la fonction de distribution de Boltzmann. Dans cette étude, la fonction de Boltzmann est adoptée, où la probabilité de remplacer la solution initiale par une solution voisine, lorsque $\Delta E > 0$, est donnée par $\exp(-\Delta E)/T$.

À des températures élevées, la probabilité d'accepter des mouvements détériorant la solution est élevée. Lorsque $T = \infty$, tous les mouvements sont acceptés, ce qui équivaut à une exploration locale aléatoire du paysage. En revanche, à basse température, la probabilité d'accepter des mouvements détériorant la solution diminue. Si $T = 0$, aucun déplacement défavorable n'est accepté, et la recherche équivaut à une recherche locale, c'est-à-dire une ascension de collines. De plus, la probabilité d'accepter une détérioration significative de la qualité de la solution décroît de manière exponentielle vers 0 selon la distribution de Boltzmann.

5.6 La procédure de l'algorithme MOSA

Au départ, nous commençons avec une configuration aléatoire, sélectionnée de manière intelligente en fonction du problème. La procédure opère sur une solution initiale S_i générée avec une température initiale T_i en tant que première étape. Ensuite, nous passons à une solution S_j générée dans le voisinage de S_i en utilisant l'une des méthodes mentionnées ci-dessus. La valeur de la fonction objective de la solution S_j est représentée par $OF(S_j)$ et comparée à l'objectif de la solution initiale $OF(S_i)$. Soit Δ la différence entre les deux objectifs, formulée comme suit : $\Delta S = OF(S_j) - OF(S_i)$. Si ΔS est inférieur ou égal à zéro, la solution S_j remplace automatiquement la solution S_i et représente la solution de départ pour la prochaine itération. Sinon, S_j est accepté avec une probabilité de $e^{(-\Delta E/T)}$, selon la fonction de distribution de Boltzmann, communément connue sous le nom de critère d'acceptation de Metropolis. Ensuite, nous comparons cette probabilité à un nombre aléatoire r généré dans l'intervalle $[0,1]$. Si $r \leq e^{(-\Delta E/T)}$, alors la solution

S_i est remplacée par la solution S_j ; sinon, nous conservons la solution initiale S_i . Une fois que le nombre d'itérations de la diminution précédente est terminé, la température T diminue selon la formule $T_{i+1} = \alpha \cdot T_i$, où $0.5 \leq \alpha \leq 0.99$. L'algorithme s'arrête lorsque la température est inférieure ou égale à la température finale TF , la procédure est également interrompue si la solution trouvée n'est pas améliorée après un nombre prédéterminé de N itérations. Les étapes de SA sont détaillées dans l'Algorithme 5.1.

La procédure de l'algorithme MOSA proposé reste la même que celle de l'algorithme SA, la différence réside dans le calcul de la probabilité d'acceptation de la solution de moindre qualité. Dans cet algorithme, la première étape consiste à définir une séquence de fonctions fournissant la probabilité d'acceptation de la solution de moindre qualité pour chaque fonction objective.

$$P_i(k) = \begin{cases} \exp^{-(\Delta)/T} & \text{if } \Delta S > 0 \\ 1 & \text{else} \end{cases} \quad (5.3)$$

Où $P_i(k)$ est la probabilité de la K_i ème fonction objective, à la i ème itération. Une fois que toutes les probabilités ont été déterminées, il existe plusieurs solutions pour les agréger :

- La première solution consiste à calculer le produit pondéré des probabilités :

$$P = \prod_{k=1}^K P_i(k)^{w_k}$$

- La seconde consiste à prendre la plus petite probabilité :

$$P = \min \{P_i(k)\}$$

Algorithme 5.1 : L'algorithme MOSA

Step 1. Initialize $(S_i, T_0, TF, \alpha, N, I_{iter})$.

Step 2. Generating initial solution S_i .

```
while Termination condition not satisfied do
  for  $I = 1, \dots, I_{iter}$  do
    Generating a solution  $S_j$  based on  $S_i$ .
    Calculate  $\Delta S = Fo(S_j) - Fo(S_i)$  .
    if  $\Delta S \leq 0$  then
       $S_i = S_j$  {solution accepté}
    else
      generating random number  $r \sim U[0, 1]$ 
      calculate a probability  $P = \exp^{-(\Delta E)/T}$ 
      if  $r \leq P$  then
         $S_i = S_j$  {solution accepté}
      else
         $S_i = S_i$  {solution refusé}
      end if
    end if
  end for
   $T_{i+1} = \alpha.T_i$ 
end while
```

5.7 La recherche locale

En raison des variations induites par l'effet d'apprentissage, les cadences des machines et la vitesse de transports, le problème étudié présente une diversité de scénarios. Ainsi, l'intégration de la recherche locale dans le processus de résolution s'avère essentielle pour améliorer la qualité de la solution élite fournie par l'algorithme MOSA. La procédure de recherche locale représentée dans l'algorithme 5.2 est la suivante : après un certain nombre d'itérations, la solution actuelle obtenue est examinée en sélectionnant aléatoirement deux jobs du premier groupe de la séquence et en échangeant leurs positions ainsi que leurs vitesses de traitement. Si cette action améliore la fonction objective, la procédure se poursuit pour les autres groupes ; sinon, elle est interrompue. À chaque étape, si la nouvelle solution obtenue est meilleure que la précédente, elle remplace la solution précédente ; dans le cas contraire, la solution précédente est conservée. La recherche locale est intégrée dans l'algorithme avant l'actualisation de la température.

Algorithme 5.2 : Procédure de la recherche Locale

Entrées : Solution actuelle S_a , Nombre de groupes G , Fonction objective F_o

Continuer = **Vrai**

while Continuer **do**

for $g = 1, \dots, G$ **do**

 Sélectionner au hasard deux jobs j_1 et j_2 du groupe $g(S_a)$.

 Échanger les positions et les vitesses de traitement de j_1 et j_2 dans la solution actuelle $g(S_a)$.

if $F_o(S_n) < F_o(S_a)$ **then**

$S_a = S_n$ {solution accepté}

else

 Continuer = **Faux**

$S_a = S_a$ {solution refusé}

end if

end for

end while

5.8 Analyse de sensibilité de l'algorithme MOSA proposé

L'algorithme MOSA est la version multi-objectif du recuit simulé, cette métaheuristique s'impose comme une technique d'optimisation générale offrant une solution adaptée aux modèles fortement non-linéaires, aux données bruitées ou soumises à des contraintes significatives. Son avantage notable par rapport aux méthodes de régression réside dans sa polyvalence, étant donné que l'algorithme Metropolis-Hastings qui la sous-tend ne repose sur aucune propriété intrinsèque au modèle. Ainsi, cette méthode se révèle aussi efficace pour ajuster des fonctions multi-paramètres que pour optimiser le parcours du voyageur de commerce.

Néanmoins, il existe un compromis évident entre la qualité de la solution trouvée et le temps nécessaire à sa détermination. Ce compromis émerge en raison des difficultés inhérentes aux métaheuristiques, notamment leur sensibilité aux paramètres, ce qui complique leur adaptation pour atteindre une solution quasi optimale. La performance de chaque métaheuristique est donc influencée par le choix et le réglage de ses paramètres, impactant directement le comportement du système.

Dans cette optique, une analyse de sensibilité est entreprise pour varier les paramètres de l'algorithme MOSA, permettant ainsi d'identifier le meilleur paramétrage. Cette démarche vise à analyser leur impact sur le système, fournissant ainsi des insights cruciaux

pour optimiser l'utilisation de l'algorithme dans divers contextes.

5.8.1 Étude de variation des paramètres de MOSA

L'analyse de sensibilité a été réalisée à travers plusieurs expérimentations, en utilisant la comparaison des résultats obtenus par MOSA avec ceux du solveur CPLEX comme indicateur de performance. Trois paramètres jugés cruciaux pour le bon fonctionnement de l'algorithme ont été variés au cours de ces tests. Il s'agit du taux de refroidissement, noté α , qui influence significativement la qualité de la solution, du nombre d'itérations à chaque température, noté I_{iter} , qui impacte le temps de calcul de l'algorithme, et de la température finale, notée TF , utilisée comme condition d'arrêt de l'algorithme.

Ainsi, les paramètres de MOSA ont été ajustés de la manière suivante :

- Le taux de refroidissement varie entre l'ensemble des variables suivantes : $\alpha = \{0.9, 0.93, 0.96\}$
- Le nombre d'itérations à chaque température peut prendre les valeurs de l'ensemble suivant : $I_{iter} = \{(G + n) \times 500, (G + n) \times 1000\}$ où G est le nombre de groupes et n est le nombre total de jobs dans le système $\sum_{g=1}^G N_g = n$.
- La température finale qui est testée suivant plusieurs valeurs appartenant à l'ensemble suivant : $TF = \{1, 5, 10\}$

Dans l'ensemble des tests, la température initiale T_0 est fixée à 1000. Il est important de noter que plusieurs configurations de paramètres ont été évaluées, avec diverses combinaisons dans lesquelles les paramètres sont ajustés à tour de rôle. Pour chaque combinaison, entre 10 et 20 répliquions ont été effectuées, et la valeur moyenne est considérée comme un indicateur. Le Tableau 5.1 résume l'écart moyen entre les résultats de MOSA et la solution optimale.

Tableau 5.1: L'écart moyen entre MOSA et l'optimum pour différents niveaux de facteurs contrôlables..

(α, T_F, I_{iter})	GAP%	(α, T_F, I_{iter})	GAP%	(α, T_F, I_{iter})	GAP%
$(0.9, 1, (G + n) \times 500)$	1.591	$(0.93, 1, (G + n) \times 500)$	1.779	$(0.96, 1, (G + n) \times 500)$	1.746
$(0.9, 1, (G + n) \times 1000)$	1.808	$(0.93, 1, (G + n) \times 1000)$	1.726	$(0.96, 1, (G + n) \times 1000)$	1.646
$(0.9, 5, (G + n) \times 500)$	1.585	$(0.93, 5, (G + n) \times 500)$	1.753	$(0.96, 5, (G + n) \times 500)$	1.839
$(0.9, 5, (G + n) \times 1000)$	1.930	$(0.93, 5, (G + n) \times 1000)$	1.816	$(0.96, 5, (G + n) \times 1000)$	1.467
$(0.9, 10, (G + n) \times 500)$	1.810	$(0.93, 10, (G + n) \times 500)$	1.831	$(0.96, 10, (G + n) \times 500)$	1.575
$(0.9, 10, (G + n) \times 1000)$	1.554	$(0.93, 10, (G + n) \times 1000)$	1.566	$(0.96, 10, (G + n) \times 1000)$	1.724

Selon cette analyse, les écarts minimaux sont obtenus avec deux combinaisons spécifiques : $(1000, 0.96, 5, (G + n) \times 1000)$ and $(1000, 0.9, 10, (G + n) \times 1000)$. La première

combinaison, caractérisée par les paramètres $T_0 = 1000$, $\alpha = 0.96$, $TF = 5$ et $I_{iter} = (G+n) \times 1000$, affiche un écart particulièrement réduit. Ces deux configurations optimales représentent les paramètres de contrôle de l'algorithme. En conséquence, nous retenons ces valeurs pour l'ensemble des expérimentations à venir.

5.8.2 Étude de variation des vitesses de traitement

Dans cette étude, les résultats obtenus sont influencés par deux facteurs principaux : l'effet d'apprentissage d'une part, et les vitesses de traitement et de transport d'autre part. Afin d'évaluer l'impact de ces vitesses sur la solution obtenue, plusieurs tests ont été réalisés dans le cadre de cinq scénarios distincts.

Le premier scénario suppose une vitesse unique pour le traitement et le transport, servant ainsi de point de comparaison entre un problème d'ordonnement de base avec des temps constants et un problème où les temps de traitement dépendent des vitesses sélectionnées.

Dans les deux scénarios suivants, trois vitesses de production et de transport (rapide, normale, lente) ont été prises en compte. Dans le deuxième scénario, pendant la génération des voisins, seuls deux jobs dans deux groupes voient leur vitesse modifiée, comme indiqué précédemment. Tandis que dans le troisième scénario, les vitesses de tous les jobs dans tous les groupes sont modifiés.

Pour les deux derniers scénarios, une procédure de recherche locale a été mise en œuvre. Le tableau 5.2 présente l'écart moyen entre les résultats de MOSA et la solution optimale pour ces scénarios, basé sur certaines instances du problème étudié. Seules deux machines sont considérées, la première opérant à trois cadences tandis que le temps de traitement sur la seconde dépend de l'effet d'apprentissage de l'opérateur.

Tableau 5.2: Protocole d'intégration de la vitesse et son impact sur la solution finale

Instances	Scénario 1	Scénario 2	Scénario 3	Scénario 2/LS	Scénario 3/LS
2(3,4)	0	0.966	0.445	0.566	0.297
2(5,6)	0.418	1.202	1.023	1.09	0.857
3(2,3,4)	0	0.984	0.779	0.539	0.358
3(3,3,5)	0.65	0.735	0.544	0.685	0.39
3(4,3,4)	0.6	1.101	0.779	0.787	0.737
4(2,3,3,4)	0.377	1.02	0.887	0.847	0.669
Moyenne	0.34	1.001	0.742	0.752	0.551

Dans le scénario 1, l'écart moyen pour les petites instances est nul. Cependant, à mesure que le nombre de jobs et de groupes augmente, l'écart moyen atteint 0,65%, bien qu'il reste relativement faible. En revanche, dans les deux scénarios suivants, qui intègrent

les vitesses dans le modèle, l'écart moyen augmente par rapport au premier scénario. En comparant les scénarios 2 et 3, les résultats sont meilleurs dans le scénario 3, où les vitesses de tous les jobs de tous les groupes sont modifiés lors de la phase de génération du voisinage. De plus, une diminution significative est observée dans les scénarios 2 et 3 lors de l'intégration de la procédure de recherche locale.

5.9 Expérimentations et résultats

Plusieurs expérimentations computationnelles ont été réalisées afin d'évaluer les performances de l'algorithme MOSA proposé. L'algorithme est implémenté dans l'environnement Java NetBeans, tandis que le modèle mathématique et les bornes inférieures sont résolues à l'aide du solveur CPLEX d'IBM ILOG (version 12.8). Les expériences ont été menées sur un ordinateur équipé d'un processeur Intel Core i3 avec 2.13 GHz et 8 Go de RAM.

5.9.1 Génération des instances

En fonction du cas d'étude, les instances sont générées selon les spécifications présentées dans le tableau 5.3.

5.9.1.1 Environnements de tests pour les machines

Pour la fabrication de bielles forgées, quatre machines sont nécessaires, comme mentionné précédemment. La première et la dernière machine effectuent des tâches en variant les vitesses v_k , ($k = 1, 2, 3$), tandis que les pièces semi-finies sont traitées en fonction des capacités des travailleurs sur la deuxième et la troisième machine, rendant ainsi l'effet d'apprentissage nécessaire.

5.9.1.2 Environnement de tests pour les groupes

Les instances utilisées sont les mêmes que celles de Naderi et Salmasi (2012), avec un nombre de groupes variant entre 2 et 16. Les tailles des groupes sont générées selon une distribution uniforme d'entiers de $U[1, 5]$, $U[6, 10]$ et $U[11, 16]$ pour les problèmes de petite, moyenne et grande taille, respectivement.

5.9.1.3 Environnement de tests pour les jobs

Le nombre de jobs dans chaque groupe varie entre 2 et 10. Il est essentiel de classifier les expériences en fonction du nombre maximal de jobs dans un groupe. Une instance composée de trois groupes, avec respectivement quatre, sept et neuf jobs, est ainsi classée

comme un problème à 9 jobs. Les problèmes, en fonction de leur nombre maximal de jobs, sont regroupés en trois catégories : la catégorie de petite taille, comprenant de 2 à 4 jobs dans le groupe ; la catégorie de taille moyenne, composée de 5 à 7 jobs dans le groupe ; et la catégorie de grande taille, avec 8 à 10 jobs dans le groupe.

5.9.1.4 Environnement de tests pour les temps de configuration

En ce qui concerne le temps de configuration entre les groupes, il varie selon trois axes, comme illustré dans le tableau 5.4. Sur le premier axe, le temps de réglage de chaque machine augmente de manière séquentielle. Sur le deuxième axe, le temps de réglage est presque identique pour chaque machine. Sur le troisième axe, le temps de configuration diminue progressivement de la première à la dernière machine, ce qui signifie que le temps de configuration de la machine de matriçage est soit plus élevé que celui de la machine de laminage, ou vice versa, ou encore qu'ils sont identiques. Le temps de traitement est généré à partir d'une loi de distribution uniforme d'entiers $U[1, 50]$.

5.9.1.5 Environnement de tests pour les caractéristiques de l'étude

Deux vecteurs de poids (w_1 et w_2) sont générés en se basant sur ceux proposés par Doğan et Özgüner (2005) et Tigane et al. (2019), où (w_1 et w_2) prennent les valeurs (0.75,0.25), (0.25,0.75) et (0.5,0.5). Trois taux d'apprentissage sont considérés selon les travaux de Pargar et al. (2018) : $LR = 100\%$ (pas d'apprentissage), $LR = 90\%$, ($a = -0.152$) et $LR = 70\%$, ($a = -0.514$). Plus le taux d'apprentissage est faible, plus l'effet de l'apprentissage est important.

Tableau 5.3: Caractéristiques des instances générées

Instances	Taille des groupes	Nombre de groupes	Taille des jobs	Nombre de jobs
I1			Petite	(2,3,4)
I2	Petite	3	Moyenne	(4,5,4)
I3			Grande	(8,3,5)
I4			Petite	(4,4,2,3,2,3,4,3)
I5	Moyenne	8	Moyenne	(3,7,4,5,3,6,4,5)
I6			Grande	(4,6,3,9,5,4,3,7)
I7			Petite	(3,2,2,3,4,4,3,3,4,4,2,3,4)
I8	Grande	13	Moyenne	(5,3,4,7,4,6,6,3,5,5,6,6,7)
I9			Grande	(6,4,9,3,5,10,4,5,8,6,7,4,6)

Tableau 5.4: Le temps de configuration de chaque machine

Machines	Axe 1	Axe 2	Axe 3
M1	$U[1, 50]$	$U[1, 50]$	$U[92, 142]$
M2	$U[17, 67]$	$U[1, 50]$	$U[45, 95]$
M3	$U[45, 95]$	$U[1, 50]$	$U[17, 67]$
M4	$U[92, 142]$	$U[1, 50]$	$U[1, 50]$

5.9.2 Résultats numériques

L'objectif des expériences menées est double : premièrement, démontrer l'efficacité de l'algorithme MOSA et de la procédure de recherche locale associée par rapport au modèle mathématique formulé pour les petites instances, ainsi que par rapport aux bornes inférieures proposées pour les moyennes et grandes instances ; deuxièmement, mettre en évidence l'impact des différents taux d'apprentissage, coefficients de pondération et niveaux de ratio de temps de configuration sur les performances de l'algorithme.

Avant d'exposer et d'examiner les résultats des expérimentations, nous allons illustrer la robustesse du modèle mathématique proposé par une brève analyse. Cette démonstration se concentrera particulièrement sur le pourcentage d'instances résolues dans un laps de temps de 30 minutes.

5.9.2.1 Proportion d'instances résolues avant l'atteinte de la condition d'arrêt.

Dans un premier temps, nous illustrons l'efficacité du modèle en examinant le pourcentage d'instances qui ont été résolues de manière optimale avant d'atteindre la condition d'arrêt de 30 minutes. Les résultats obtenus sont présentés dans le tableau 5.5 pour les petites instances. Sans surprise, on observe que toutes ces instances ont été optimalement résolues dans le délai imparti, indépendamment du nombre de jobs dans le groupe et du taux d'apprentissage.

Tableau 5.5: Pourcentage des petites instances résolues avant 30 minutes

Taux d'apprentissage	Groupe	Le nombre de jobs maximal dans les groupes		
		4	5	8
$LR2 = LR3 = 100\%$		100%	100%	100%
$LR2 = 70\%, LR3 = 90\%$	3	100%	100%	100%
$LR2 = 90\%, LR3 = 70\%$		100%	100%	100%

Par ailleurs, en augmentant la taille des instances composées de 8 groupe, ce pourcentage va être amené à être réduit comme on peut le constater dans le tableau 5.6.

Tableau 5.6: Pourcentage des moyennes instances résolues avant 30 minutes

Taux d'apprentissage	Groupe	Le nombre de jobs maximale dans les groupes		
		4	6	9
$LR2 = LR3 = 100\%$		100%	100%	0%
$LR2 = 70\%, LR3 = 90\%$	8	78%	100%	10%
$LR2 = 90\%, LR3 = 70\%$		100%	100%	33%

Le pourcentage reste assez élevé pour les instances composées de 8 groupes, demeurant égal à 100% pour la quasi-totalité des problèmes composés de 4 et 6 jobs maximum dans le groupe, et ce, quel que soit le taux d'apprentissage appliqué. En revanche, en augmentant le nombre de jobs à 9, les valeurs du pourcentage se réduisent considérablement, atteignant 0% avec $LR = 100\%$. Par ailleurs, elle augmente légèrement en intégrant le taux d'apprentissage ($LR2 = 70\%$ et $LR2 = 90\%$). Cela peut s'expliquer par le fait qu'en introduisant l'effet d'apprentissage, les temps de traitement vont être réduits de sorte que les jobs sont séquencés dans l'ordre croissant de leurs temps de traitement (suivant la règle SPT), générant un séquençement dans les limites du temps imparti de 30 minutes. Face aux grandes instances, le pourcentage est égal à 0% pour l'ensemble des problèmes générés, comme on peut le voir dans le tableau 5.7.

Tableau 5.7: Pourcentage des grandes instances résolues avant 30 minutes

Taux d'apprentissage	Groupe	Le nombre de jobs maximal dans les groupes		
		4	7	10
$LR2 = LR3 = 100\%$		0%	0%	0%
$LR2 = 70\%, LR3 = 90\%$	13	0%	0%	0%
$LR2 = 90\%, LR3 = 70\%$		0%	0%	0%

Cette analyse vient confirmer la robustesse du modèle mathématique proposé dans cette étude, capable de résoudre de manière optimale la quasi-totalité des instances moyennes générées dans la limite du temps fixé. Cependant, en augmentant la taille des problèmes, une méthode exacte ne peut pas fournir des solutions dans un temps de calcul raisonnable, d'où l'application de MOSA.

5.9.2.2 Temps de calcul et performances

Dans cette section, nous examinons et comparons divers résultats obtenus liés au temps de calcul ainsi qu'à la performance de l'algorithme. Les tableaux 5.8 à 5.16 résument les valeurs obtenues par les algorithmes MOSA et MOSA/LS, incluant le pourcentage d'écart relatif et les temps de calcul CPU_{time} . Par ailleurs, un indicateur essentiel à la portée de cette étude et à son importance est l'efficacité du modèle mathématique proposé. Comme

expliqué dans l'analyse précédente, ce modèle a permis de résoudre les deux premières moyennes instances (I4 et I5) en un temps de calcul d'environ une heure, ce qui est considéré comme assez raisonnable, compte tenu de la difficulté du problème étudié. Cependant, pour les grandes instances, nous comparons le pourcentage d'écart moyen de MOSA à la borne inférieure LB2 proposée.

Les simulations révèlent que le CPU_{time} augmente en augmentant la taille des instances, atteignant un maximum de 22 secondes pour l'algorithme MOSA et deux heures pour le solveur. De plus, les temps de calcul du solveur restent relativement faibles pour les petites instances, atteignant une valeur maximale de 12 secondes, mais demeure supérieur au CPU_{time} de l'algorithme, qui ne dépasse pas 4 secondes. Pour les moyennes et grandes instances, les temps de calcul de l'algorithme sont relativement inférieurs à ceux du solveur, qui pour ces instances a été arrêté au bout de deux heures de simulation. Concernant le problème de recherche proposé, le pourcentage d'écart relatif (RPD) est calculé selon la formule :

$$RPD = \frac{Fo(MOSA) - Fo(opt)}{Fo(opt)} \times 100 \quad (5.4)$$

Où $Fo(MOSA)$ représente la fonction objectif (la fonction pondérée normalisée) obtenue avec l'algorithme MOSA, et $Fo(opt)$ représente la fonction objectif obtenue avec le solveur Cplex. Les tableaux 5.8 à 5.16 présentent le RPD moyen pour le makespan, le RPD moyen pour le TEC , le PF (la fonction pondérée des deux objectifs combinés), ainsi que le temps CPU_{time} obtenu avec différents paramètres en utilisant l'algorithme MOSA. De nombreux tests ont été effectués pour évaluer les performances des bornes inférieures et de l'algorithme proposé. L'analyse des résultats obtenus est présentée dans les paragraphes suivants.

— Analyse des performances des bornes inférieures proposées

La première borne inférieure proposée constitue une version relaxée du modèle mathématique présenté dans le chapitre 4. Elle a été conçue pour évaluer la qualité des solutions générées par l'algorithme MOSA pour les instances de taille moyenne à grande. Les tableaux 5.11 à 5.13 indiquent que le pourcentage d'erreur entre LB1 et MOSA est d'environ 11% pour la fonction pondérée PF , avec un CPU_{time} raisonnable ne dépassant pas une heure.

En raison de la complexité inhérente du problème de recherche, une deuxième borne inférieure, LB2, a été développée et comparée à l'algorithme MOSA pour les grandes instances. Les résultats, présentés dans la figure 5.2, révèlent que le pourcentage d'erreur atteint un maximum de 19%, avec un temps de calcul ne dépassant pas une seconde. Bien que ce pourcentage puisse sembler élevé, il est crucial de noter que, comme le montre

la figure 4.12, même pour les instances de taille moyenne, LB2 s'écarte des solutions optimales de 2 à 12%. Par conséquent, cette valeur pourrait être attribuée à l'écart de LB2.

Par ailleurs, il est à noter que dans les recherches antérieures menées par Keshavarz et al. (2019), un écart de 15% a été constaté entre la borne inférieure et l'algorithme d'optimisation hybride par essaim de particules, proposé pour les grandes instances avec des dates d'échéance serrées. Ceci vient confirmer la qualité des bornes inférieures développées dans cette étude.

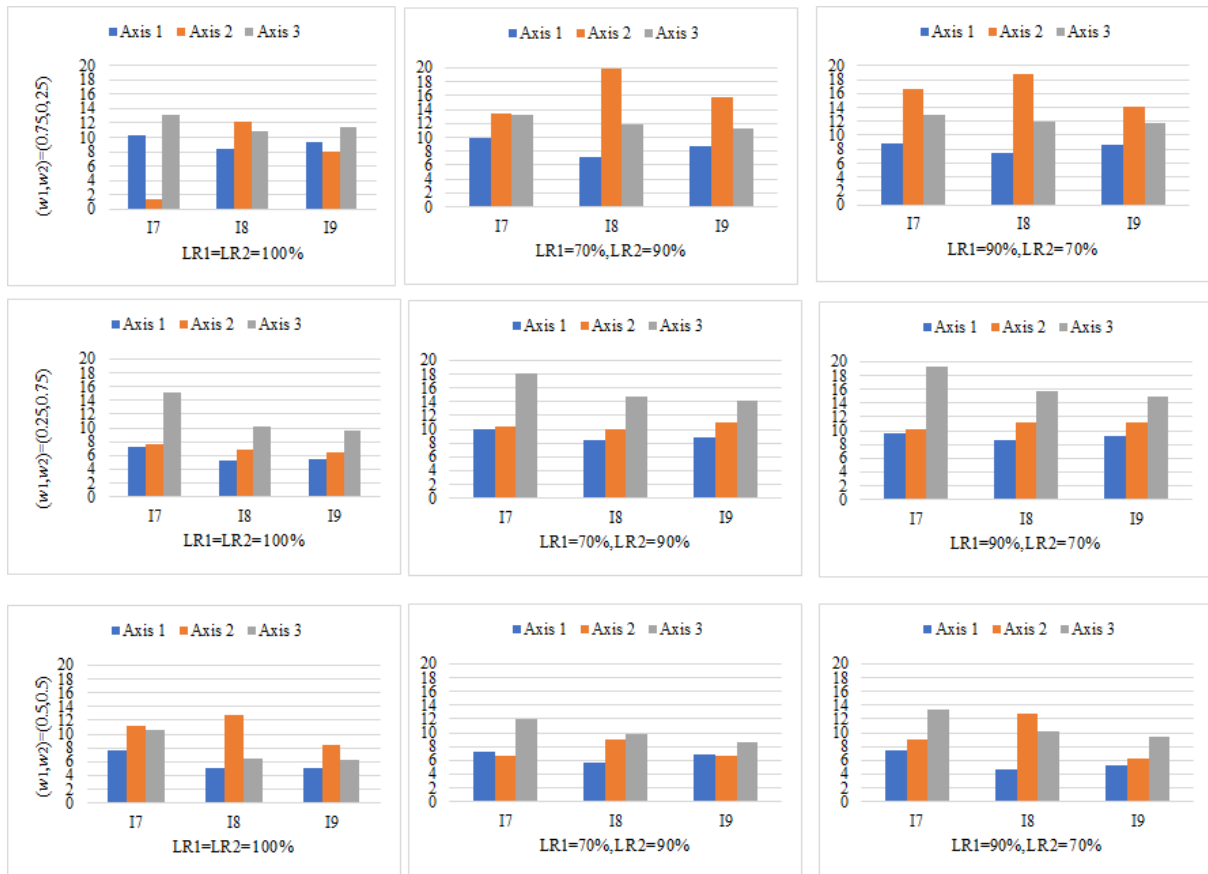


Figure 5.2: Le pourcentage d'erreur entre MOSA et LB2 pour les grandes instances

— Analyse des performances de MOSA

Conformément aux objectifs fixés par l'entreprise, qui portent sur le makespan et la consommation d'énergie, une solution à ce problème consiste à établir un ordonnancement pour l'exécution des tâches au sein de chaque groupe, en tenant compte des différents taux d'apprentissage. Le séquençement des groupes, quant à lui, dépendra du temps de configuration et de la consommation d'énergie associée. Le transport des produits entre les différentes machines se fait en utilisant des vitesses différentes, le tout étant déterminé par le poids accordé aux deux objectifs. Cette approche permet une organisation plus efficace de l'atelier de production, une meilleure affectation des travailleurs selon

leurs compétences, et le choix des vitesses de production appropriées, contribuant ainsi à l'augmentation de la productivité, à la fois en réduisant le temps de production et en économisant la consommation d'énergie.

L'algorithme évolutionnaire proposé est sensible aux paramètres préalablement sélectionnés. Par conséquent, un réglage précis de ces paramètres influe considérablement sur ses performances. Pour les résultats présentés dans cette étude, les paramètres suivants ont été utilisés sur la base d'expériences préliminaires : $T_i = 1000$, $\alpha = 0,96$, $TF = 5$, $I_{iter} = (G + n) \times 1000$. Dans ce qui suit, nous allons analyser les performances de l'algorithme sur le système étudié pour chaque famille d'instances.

• Les petites instances

Les tableaux 5.8, 5.9 et 5.10 représentent les résultats obtenus par MOSA et MOSA/LS pour les petites instances, démontrant leur efficacité par rapport au modèle mathématique MILP. Les cinq premières colonnes des tableaux indiquent les caractéristiques relatives à la génération des instances et la variation des paramètres utilisés dans les simulations. Les trois colonnes suivantes présentent le pourcentage d'écart relatif ($RPD\%$) moyen obtenu pour le Cmax et le TEC, ainsi que le pourcentage d'écart relatif de la fonction de pondération ($PF\%$), respectivement. Ensuite, la colonne suivante indique le (CPU_{time}) obtenu par MOSA sans la recherche locale. Les colonnes suivantes indiquent les mêmes valeurs obtenues par MOSA/LS, et la toute dernière colonne présente le (CPU_{time}) obtenu par le solveur.

D'après ces résultats, on remarque qu'en augmentant le nombre de jobs dans le groupe, le pourcentage d'erreur augmente, ce qui signifie que le nombre de jobs appartenant aux groupes influence grandement le RPD .

a. Les résultats obtenus avec $LR2 = LR3 = 100\%$:

Lorsque les coefficients de pondération (w_1, w_2) sont respectivement fixés à (0.75, 0.25), la valeur maximale du $RPD\%$ du makespan et le RPD associé du TEC , obtenus quel que soit l'axe de temps de configuration sont égaux à [4.039%, 0.896%], [2.919%, 1.886%] et [3.968%, 2.541%] pour les instances composées de (4 jobs et 3 groupes), (5x3) et (8x3) respectivement. Lorsque les poids sont inversés (0.25, 0.75), le RPD du TEC et le RPD du makespan associé sont égaux aux valeurs suivantes [2.941%, 0.831%], [-0.835%, 0.723%] et [3.221%, 1.122%] pour les instances I1, I2 et I3 respectivement. En fixant les poids w_i à 0.5, les valeurs maximales obtenues dans ce cas sont de 5.402% et 1.674% pour le RPD du makespan et le TEC respectivement.

Tableau 5.8: Solutions détaillées pour les petites instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de 100%

Code	(w_1, w_2)	Instances	$N_g \times G$	Axe	MOSA					MOSA/LS					Solver	
					Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}	CPU_{time} (sec)	CPU_{time} (sec)	
SIC_11	(0.75,0.25)	I1	4 × 3	1	2.557	0.479	2.038	1.444	2.429	-0.168	1.780	1.491	1.53	1.53		
SIC_12				2	4.039	0.296	3.103	1.269	3.731	0.002	2.799	1.843	4.93	4.93		
SIC_13				3	0.362	1.024	0.528	2.036	-0.082	0.976	0.183	2.333	1.04	1.04		
SIC_21		I2	53	1	0.696	1.579	0.917	2.242	0.209	1.392	0.505	1.843	3.96	3.96		
SIC_22				2	2.919	1.886	2.661	3.199	2.692	1.876	2.488	2.657	4.26	4.26		
SIC_23				3	0.242	1.207	0.483	2.474	-0.104	1.290	0.245	2.441	2.73	2.73		
SIC_31		I3	83	1	0.928	0.123	0.727	2.392	0.719	0.018	0.544	1.282	5.09	5.09		
SIC_32				2	3.968	2.541	3.611	2.335	3.714	2.700	3.461	2.329	2.40	2.40		
SIC_33				3	0.467	2.427	0.957	1.368	0.183	2.351	0.725	2.266	6.38	6.38		
SIT_11	(0.25,0.75)	I1	43	1	0.796	0.781	0.785	1.693	2.132	0.352	0.797	1.686	2.42	2.42		
SIT_12				2	2.941	0.831	1.359	1.880	4.332	0.531	1.481	1.959	3.54	3.54		
SIT_13				3	1.837	0.390	0.752	2.630	2.620	0.015	0.666	1.875	1.45	1.45		
SIT_21		I2	53	1	-0.036	0.599	0.440	2.325	0	0.389	0.292	2.243	4.07	4.07		
SIT_22				2	-0.835	0.723	0.334	3.093	-0.254	0.452	0.276	2.095	6.37	6.37		
SIT_23				3	1.082	0.471	0.624	2.756	1.472	0.164	0.491	2.082	3.74	3.74		
SIT_31		I3	83	1	0.881	0.661	0.716	1.281	0.198	0.617	0.512	2.490	9.36	9.36		
SIT_32				2	3.221	1.122	1.647	2.397	2.720	0.972	1.409	2.327	2.86	2.86		
SIT_33				3	1.992	0.911	1.181	2.587	1.188	0.740	0.852	2.590	3.82	3.82		
SIF_11	(0.5,0.5)	I1	43	1	-0.203	1.446	0.622	2.172	2.182	0.251	1.217	1.886	2.15	2.15		
SIF_12				2	-0.040	1.352	0.656	2.172	0.734	0.275	0.505	1.875	4.56	4.56		
SIF_13				3	1.861	0.482	1.172	2.241	1.328	0.281	0.805	2.955	0.98	0.98		
SIF_21		I2	53	1	-1.459	1.265	-0.097	2.545	-0.886	0.791	-0.048	3.199	5.36	5.36		
SIF_22				2	5.402	0.302	2.852	2.002	4.713	0.090	2.402	2.440	4.00	4.00		
SIF_23				3	0.750	1.292	1.021	1.263	0.782	0.600	0.691	2.195	2.62	2.62		
SIF_31		I3	83	1	1.130	1.199	1.165	1.372	0.704	1.015	0.860	2.253	7.96	7.96		
SIF_32				2	3.301	0.854	2.078	2.608	2.367	1.531	1.949	2.544	5.29	5.29		
SIF_33				3	0.718	1.674	1.196	2.332	0.946	1.356	1.151	2.559	11.09	11.09		

b. Les résultats obtenus avec $LR2 = 70\%$, $LR3 = 90\%$:

En intégrant l'effet d'apprentissage sur les deux machines nécessitant l'intervention des opérateurs (tableaux 5.9 et 5.10), on observe une considérable diminution du pourcentage d'écart relatif. Dans ce cas, le RPD maximum du makespan et le RPD du TEC qui lui est associé pour la paire (0.75, 0.25) sont de [1.118%, 0.099%], [3.76%, 1.401%], et [1.095%, 2.53%] pour les instances I1, I2 et I3 respectivement. Pour la paire (0.25, 0.75), la valeur maximale du RPD pour le TEC et le RPD pour le makespan sont de [0.725%, 0.876%], [0.14%, 0.978%], et [0.868%, 1.811%] pour les instances I1, I2 et I3 respectivement. Pour la paire (0.5, 0.5), le maximum des deux RPD est de 3.501% et 1.851% pour le makespan et le TEC respectivement. Ces valeurs sont obtenues lorsque l'effet d'apprentissage est élevé au niveau de la machine 2 ($LR2 = 70\%$).

c. Les résultats obtenus avec $LR2 = 90\%$, $LR3 = 70\%$:

En inversant les valeurs du taux d'apprentissage sur les deux machines M2 et M3 [90%, 70%], les RPDs diminuent encore, atteignant un maximum de 1.682% pour le makespan pour la paire (0.75, 0.25) et 1.862% pour le TEC pour la paire (0.25, 0.75). Pour la paire (0.5, 0.5), les valeurs maximales obtenues sont de 1.955% pour le RPD du makespan et de 2.791% pour le RPD du TEC, respectivement.

Les résultats obtenus pour les petites instances indiquent que l'écart relatif moyen obtenu par MOSA est d'environ 1.79% lorsque l'importance est accordée à la minimisation du $Cmax$ ($w_1 = 0.75$), avec un taux d'apprentissage égal à $LR2 = LR3 = 100\%$ (pas d'effet d'apprentissage). Cela est principalement dû à la variation des temps de production et de transport, qui dépendent de différentes vitesses. Ainsi, MOSA doit jongler entre ses différentes perturbations pour générer des solutions satisfaisantes qui se rapprochent au mieux de l'optimalité.

Par ailleurs, en accordant la priorité au second objectif, qui est la minimisation de la consommation d'énergie totale du système ($w_2 = 0.75$), le RPD obtenu par MOSA est moins important que celui du $Cmax$, atteignant une valeur de 0.721%. Ainsi, le second objectif est moins sensible à la variation des paramètres de l'étude, ce qui entraîne plusieurs ordonnancements satisfaisants, se rapprochant de ceux obtenus par le solveur.

Enfin, lorsque l'importance accordée aux deux objectifs est la même, le RPD recensé est de 1.27% pour le $Cmax$ et de 1.09% pour le TEC . Dans ce cas, MOSA tente d'apporter des solutions satisfaisantes pour les deux critères d'optimisation.

Tableau 5.9: Solutions détaillées pour les petites instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [70%,90%].

Code	(w_1, w_2)	Instances	$N_g \times G$	Axe	MOSA					MOSA/LS					Solver
					Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}		
SIC_11	(0.75,0.25)	I1	43	1	0.987	1.383	1.086	1.693	-0.219	1.131	0.119	1.962	2.46		
SIC_12				2	1.118	0.099	0.863	2.756	0.716	0.154	0.576	2.545	1.09		
SIC_13				3	-0.069	1.506	0.325	2.344	-0.151	1.208	0.189	2.693	1.17		
SIC_21		I2	53	1	-0.182	1.430	0.221	3.093	-0.149	0.975	0.132	2.325	2.37		
SIC_22				2	3.760	1.401	3.170	2.137	3.338	1.068	2.771	2.489	2.04		
SIC_23				3	0.007	2.473	0.624	2.462	0.012	2.193	0.557	1.921	1.56		
SIC_31		I3	83	1	0.393	2.192	0.843	3.066	0.284	2.031	0.721	2.527	2.90		
SIC_32				2	1.095	2.530	1.454	3.029	0.722	2.520	1.172	2.633	6.04		
SIC_33				3	0.760	1.514	0.949	2.448	0.537	1.562	0.793	2.503	2.85		
SIT_11	(0.25,0.75)	I1	43	1	0.725	0.876	0.838	2.693	0.088	0.607	0.477	2.078	1.70		
SIT_12				2	2.991	0.335	0.999	2.577	3.315	0.087	0.894	1.875	2.18		
SIT_13				3	1.275	0.513	0.704	3.051	1.068	0.355	0.533	3.093	1.23		
SIT_21		I2	53	1	-0.070	0.859	0.627	2.855	-0.093	0.622	0.443	2.739	3.88		
SIT_22				2	0.140	0.978	0.769	2.169	0.120	0.745	0.589	1.982	5.14		
SIT_23				3	1.469	0.354	0.633	2.457	1.523	0.221	0.547	1.931	7.59		
SIT_31		I3	83	1	0.571	1.233	1.068	2.606	0.314	0.963	0.801	3.507	3.36		
SIT_32				2	0.820	1.205	1.109	2.907	0.605	1.011	0.910	2.546	6.34		
SIT_33				3	0.868	1.811	1.575	2.648	0.515	1.623	1.346	2.530	2.67		
SIF_11	(0.5,0.5)	I1	43	1	0.668	1.653	1.161	2.589	0.806	0.893	0.850	2.325	6.37		
SIF_12				2	1.615	1.200	1.408	2.739	-0.146	1.285	0.570	2.901	2.23		
SIF_13				3	0.022	1.305	0.664	3.051	-0.005	1.158	0.577	2.855	1.18		
SIF_21		I2	53	1	-1.484	1.846	0.181	2.151	-1.554	1.495	-0.029	2.234	2.87		
SIF_22				2	-1.696	2.011	0.158	2.011	-1.344	1.696	0.176	2.309	6.57		
SIF_23				3	0.401	1.196	0.799	1.971	0.724	0.867	0.796	2.011	5.65		
SIF_31		I3	83	1	1.396	1.683	1.540	2.746	1.566	1.316	1.441	2.467	2.09		
SIF_32				2	3.501	1.851	2.676	2.646	2.934	1.880	2.407	2.852	6.04		
SIF_33				3	1.058	2.257	1.658	2.951	1.231	1.970	1.601	2.530	5.48		

En introduisant l'effet d'apprentissage au niveau de M2 et M3, avec une incidence plus marquée sur M2 ($LR2 = 70\%$), on observe une réduction du pourcentage d'écart relatif pour la quasi-totalité des paires de coefficients de pondération par rapport au premier scénario dans lequel l'effet d'apprentissage est négligé. Pour la paire (0.75,0.25), l'écart diminue d'environ 33%, et pour la paire (0.25,0.75), une réduction de 49% est constatée. Quant à la paire (0.5,0.5), une réduction de 16% est enregistrée pour le $Cmax$, accompagnée d'une augmentation de 35% pour le TEC . Cela indique que dans ce cas, l'algorithme a privilégié davantage la minimisation du makespan, qui est plus sensible aux différents taux d'apprentissage.

En optant pour un taux d'apprentissage plus élevé pour M3 ($LR3 = 70\%$), le pourcentage d'écart est réduit davantage, atteignant les valeurs de 64%, 44%, et 20% pour les paires (0.75,0.25), (0.25,0.75), et (0.5,0.5) respectivement. Ces résultats peuvent s'expliquer par le fait que cette machine consomme davantage d'énergie par rapport à M2 et que les temps de traitement sont assez élevés sur cette machine (machine goulot). Ainsi, l'affectation d'un opérateur compétent sur cette machine joue un rôle important dans la minimisation des deux objectifs.

- **Les moyennes instances**

Les résultats obtenus pour les moyennes instances, en fixant la taille des groupes à 8, sont représentés dans les tableaux 5.11 à 5.13.

- a. Les résultats obtenus avec $LR2 = LR3 = 100\%$:

Lorsque le taux d'apprentissage est de 100% (tableau 5.11), le RPD maximum obtenu pour le makespan pour la paire (0.75,0.25) est de 7.211%, tandis que le RPD maximum pour le TEC atteint 2.800% lorsque les poids des deux objectifs sont fixés à (0.25,0.75). Pour la paire (0.5,0.5), les valeurs maximales atteintes pour le makespan et le TEC sont de 6.364% et 4.072%, respectivement. Les pourcentages d'écart obtenus pour les instances de taille moyenne sont élevés par rapport aux petites instances mais restent raisonnables, ne dépassant pas les 7%. D'autre part, le RPD obtenu pour le $Cmax$ reste majoritairement élevé par rapport au TEC, car la métaheuristique tend à minimiser davantage cet objectif qui n'évolue pas à la même échelle que le makespan.

Tableau 5.10: Solutions détaillées pour les petites instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [90%.70%].

Code	(w_1, w_2)	Instances $N_g \times G$	MOSA					MOSA/LS					Solver
			Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time}		
SIC_11	(0.75,0.25)	I1	43	1	0.346	1.101	0.535	2.955	-0.137	0.753	0.086	2.855	1.87
SIC_12				2	1.682	1.188	1.559	3.051	1.340	0.896	1.229	2.901	1.17
SIC_13				3	-0.122	1.266	0.225	3.157	-0.159	1.176	0.175	2.078	1.04
SIC_21		I2	53	1	-0.066	2.179	0.495	2.390	-0.166	1.727	0.307	2.169	1.32
SIC_22				2	0.985	1.411	1.092	2.584	0.428	1.043	0.582	2.637	1.52
SIC_23				3	-0.153	2.831	0.593	1.997	-0.287	2.774	0.478	2.177	2.54
SIC_31		I3	83	1	-0.233	3.395	0.674	2.896	-0.839	3.250	0.183	3.680	2.28
SIC_32				2	0.889	2.007	1.169	2.773	0.703	1.964	1.018	2.796	4.24
SIC_33				3	0.827	2.722	1.301	3.223	0.826	2.372	1.213	2.718	5.01
SIT_11	(0.25,0.75)	I1	43	1	-1.509	0.960	0.343	2.756	-1.978	0.584	-0.057	2.545	2.09
SIT_12				2	3.096	0.504	1.152	2.241	3.465	0.357	1.134	2.951	1.28
SIT_13				3	2.174	0.335	0.795	3.116	1.823	0.191	0.599	2.901	1.90
SIT_21		I2	53	1	1.532	0.892	1.052	2.098	1.760	0.615	0.901	2.348	2.37
SIT_22				2	3.826	0.771	1.535	2.884	3.878	0.474	1.325	2.111	3.01
SIT_23				3	1.913	0.702	1.005	1.997	2.401	0.456	0.942	2.597	2.15
SIT_31		I3	83	1	-0.283	1.862	1.326	1.965	-0.875	1.542	0.938	3.042	6.81
SIT_32				2	2.321	1.841	1.961	2.444	1.570	1.419	1.457	2.280	4.48
SIT_33				3	1.903	1.219	1.390	3.776	1.260	1.058	1.109	3.746	3.80
SIF_11	(0.5,0.5)	I1	43	1	0.651	0.566	0.609	2.955	0.662	0.456	0.559	3.093	8.72
SIF_12				2	-0.029	1.318	0.645	2.485	0.045	1.142	0.594	2.934	3.62
SIF_13				3	0.078	1.113	0.596	2.325	-0.104	0.966	0.431	2.983	1.40
SIF_21		I2	53	1	-2.358	2.791	0.217	2.355	-2.885	2.578	-0.154	2.566	6.87
SIF_22				2	1.196	2.068	1.632	2.501	0.995	1.806	1.401	2.002	5.50
SIF_23				3	1.955	1.803	1.879	2.857	1.765	1.539	1.652	2.009	3.45
SIF_31		I3	83	1	1.090	1.492	1.291	2.555	1.093	1.037	1.065	2.763	5.68
SIF_32				2	1.568	1.883	1.726	2.633	1.671	1.631	1.651	2.825	7.20
SIF_33				3	1.058	1.668	1.363	2.857	1.335	1.379	1.357	2.903	3.58

b. Les résultats obtenus avec $LR2 = 70\%$, $LR3 = 90\%$:

En intégrant un effet d'apprentissage plus important sur la machine M2 (tableaux 5.12), le RPD maximal obtenu augmente par rapport aux tests réalisés avec un $LR = 100\%$, et cela pour les deux objectifs, atteignant les valeurs maximales de 8,156% et 4.689% pour le C_{max} et le TEC respectivement. Dans ce cas, les objectifs sont considérablement réduits par rapport aux petites instances, creusant ainsi davantage le fossé entre MOSA et le solveur.

c. Les résultats obtenus avec $LR2 = 90\%$, $LR3 = 70\%$:

Lorsque le taux d'apprentissage est inversé entre les machines M2 et M3 ($LR3 = 70\%$), le pourcentage d'écart se creuse davantage par rapport au scénario précédent. Cette augmentation est enregistrée à 8,979% pour le makespan et à 5,5% pour la consommation totale d'énergie, respectivement. Cette observation peut s'expliquer par une diminution plus conséquente des temps de traitement sur la troisième machine. Ainsi, un séquençement approprié des tâches dans les groupes devient primordial pour la minimisation du makespan, créant ainsi un fossé entre la solution proposée par MOSA et celle du solveur.

À travers les expériences réalisées pour les instances de taille moyenne, on observe une augmentation du pourcentage d'écart par rapport aux petites instances. Cette augmentation du RPD est d'environ 2% pour la quasi-totalité des instances générées dans le tableau 5.11, soit lorsque l'effet d'apprentissage est négligé. Cette évolution s'explique par le fait que les instances sont assez grandes, atteignant un nombre total de 40 jobs. Par conséquent, le nombre de combinaisons augmente, rendant ardue la recherche d'un séquençement approprié des jobs et des groupes simultanément. De plus, la moitié des instances de taille moyenne générées ont pu être comparées à la solution optimale donnée par le solveur, démontrant l'efficacité du modèle mathématique proposé. Ainsi, la concordance entre la solution trouvée par MOSA et celle du solveur demeure complexe en raison du comportement variable du système, fortement influencé par la régulation de ses différents paramètres, ainsi que par des temps de traitement variables dus à l'effet d'apprentissage et aux cadences des machines.

Tableau 5.11: Solutions détaillées pour les moyennes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de 100%

Code	(w_1, w_2)	Instances $N_g \times G$	MOSA						MOSA/LS						LB1						Solver
			Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF %	CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF %	CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF %	CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF %	CPU_{time} (sec)			
MIC_41	(0.75,0.25)	14	48	1	5.221	6.266	3.495	1.275	4.235	10.930	4.079	4.837	2.801	3.233	8.467	1.271	7.113	0.925	1.586	1.350	935.79
MIC_42			2																		1573.98
MIC_43			3																		1176.04
MIC_51		15	68	1	3.655	4.713	1.877	4.004	10.295	11.089	3.297	4.290	1.897	3.692	9.803	8.337	4.610	7.405	1.071	4.463	3158.41
MIC_52			2																		5133.69
MIC_53			3																		4992.61
MIC_61		16	98	1	5.238	4.217	2.960	3.903	16.138	18.228	3.442	4.219	1.768	3.280	15.206	5.593	4.763	2.704	4.248	1.741	673.20
MIC_62			2																		844.52
MIC_63			3																		1196.28
MIT_41	(0.25,0.75)	14	48	1	5.454	2.373	1.516	1.730	12.119	14.821	3.686	4.978	1.373	3.275	15.969	9.132	4.795	8.048	1.758	1.741	1590.9
MIT_42			2																		806.64
MIT_43			3																		1454.08
MIT_51		15	68	1	2.373	8.629	1.279	3.117	13.992	14.969	7.099	8.231	1.355	1.409	12.206	2.589	0.301	0.873	0.301	0.873	3591.36
MIT_52			2																		4658.03
MIT_53			3																		4526.81
MIT_61		16	98	1	4.498	3.371	1.689	2.391	14.969	17.788	3.495	4.978	1.479	1.983	14.788	1.716	1.953	1.894	1.758	1.741	830.88
MIT_62			2																		1017.25
MIT_63			3																		924.42
MIF_41	(0.5,0.5)	14	48	1	3.507	6.364	2.614	3.061	11.856	14.821	2.494	4.451	1.678	2.371	12.162	9.513	0.012	0.601	0.601	0.307	1326.26
MIF_42			2																		1205.06
MIF_43			3																		1627.9
MIF_51		15	68	1	2.179	3.371	2.043	2.111	16.119	16.119	1.644	1.644	1.696	3.075	13.025	0.816	0.187	0.502	0.187	0.502	4176.42
MIF_52			2																		5537.83
MIF_53			3																		6120.20
MIF_61		16	98	1	2.745	3.029	3.807	3.856	15.985	16.938	2.785	3.457	3.082	3.270	15.311	2.907	2.974	2.941	2.899	3.107	666.72
MIF_62			2																		1134.62
MIF_63			3																		1336.955

- **Les grandes instances**

Ces instances se composent de 13 groupes, du nombre de jobs allant de 4 à 10, générant ainsi des problèmes composés de 77 jobs organisés dans 13 groupes distincts. Les résultats obtenus par MOSA pour les différentes configurations sont représentés dans les tableaux 5.14 à 5.16.

- a. Les résultats obtenus avec $LR2 = LR3 = 100\%$:

Au niveau des grandes instances lorsque l'effet d'apprentissage est négligé, le RPD augmente, mais relativement peu par rapport aux moyennes instances, la valeur maximale obtenue avec l'effet d'apprentissage de 100% est égale à 8.979% pour le makespan pour la paire (0.75,0.25) et 3.944% pour le TEC pour la paire (0.25,0.75), alors que pour la paire (0.5,0.5) les valeurs maximales obtenues sont 8.041% et 5.556% pour le makespan et le TEC respectivement.

- b. Les résultats obtenus avec $LR2 = 70\%$, $LR3 = 90\%$:

En introduisant l'effet d'apprentissage (tableaux 5.15 et 5.16) le pourcentage d'écart restent élevées pour la paire (0.75,0.25) et (0.25,0.75) par contre le RPD du makespan diminue pour la paire (0.5,0.5) pour atteindre la valeur de 5.281%, ce qui signifie qu'en augmentant la taille des instances le makespan augmente et le RPD entre les deux fonctions objectifs est moins important lors de la normalisation de la fonction pondérée.

En présence d'un taux d'apprentissage de 70% pour la machine M2, on observe que le RPD demeure relativement constant par rapport au scénario précédent, atteignant une valeur de 8,901% pour le $Cmax$ et 4,382% pour la consommation d'énergie. En revanche, il diminue dans le cas de la paire (0.5, 0.5), prenant les valeurs suivantes : [5.281%, 5.953%] pour la minimisation simultanée des deux objectifs [$Cmax$, TEC], avec une priorité équivalente.

- c. Les résultats obtenus avec $LR2 = 90\%$, $LR3 = 70\%$:

Les mêmes observations peuvent être remarquées lorsque l'effet d'apprentissage est plus prononcé au niveau de M3 ($LR3 = 70\%$). Les valeurs recensées dans ce cas sont de 8.974% pour la paire (0.75,0.25). Par ailleurs, une légère augmentation est enregistrée avec la paire (0.25,0.75), atteignant 5.016%. En ce qui concerne la paire (0.5,0.5), les RPD associés sont de 5.875% pour le $Cmax$ et de 6.294% pour le TEC , respectivement.

Les résultats obtenus pour les grandes instances sont assez concluants et ne dépassent pas les 9% de pourcentage d'écart relatif pour la minimisation du makespan et les 7% pour le TEC , ce qui représente une augmentation minime par rapport aux moyennes instances. Ceci prône que l'algorithme MOSA est très efficace et fonctionne d'une excellente manière lorsque la taille des instances augmentent.

Tableau 5.12: Solutions détaillées pour les moyennes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [70%.90%].

Code	(w_1, w_2)	Instances $N_g \times G$	MOSA						MOSA/LS						LB1						Solver
			Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF %	CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF %	CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF %	CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF %	CPU_{time} (sec)			
MIC_41	(0.75,0.25)	14	4.695	2.655	4.185	14.263	3.456	2.631	3.250	9.352	1.154	0.322	0.946	1583.48							
MIC_42		2	6.980	1.736	5.669	10.525	5.380	1.672	4.453	10.352	6.046	0.483	4.655	1169.32							
MIC_43		3	4.265	2.820	3.904	10.242	3.563	2.381	3.268	9.442	8.398	6.043	7.809	1628.7							
MIC_51		15	3.275	2.558	3.096	16.119	2.707	2.162	2.571	17.228	0.938	1.012	0.957	4269.26							
MIC_52		2	7.268	2.493	6.074	17.880	6.465	2.100	5.374	18.465	6.279	1.092	4.982	3407.39							
MIC_53		3	3.712	4.662	3.950	19.143	2.923	4.340	3.277	16.080	5.182	4.043	4.897	3420.58							
MIC_61		16	5.192	2.973	4.637	15.046	4.544	3.210	4.211	16.277	4.846	4.924	4.866	654.63							
MIC_62		2	8.806	2.110	7.132	17.672	7.636	2.031	6.235	17.087	11.172	3.680	9.299	844.52							
MIC_63		3	2.954	3.751	3.153	15.162	2.126	3.415	2.448	14.434	8.134	9.286	8.422	1005.47							
MIT_41	(0.25,0.75)	14	4.151	1.695	2.309	8.938	3.372	1.419	1.907	9.287	1.383	2.104	1.924	1299.53							
MIT_42		2	4.385	3.110	3.429	9.836	2.712	2.920	2.868	10.540	5.636	3.531	4.057	1138.62							
MIT_43		3	4.417	2.432	2.928	9.227	4.941	1.913	2.670	11.258	3.957	11.659	9.734	2107.30							
MIT_51		15	2.529	2.562	2.554	15.206	2.029	2.350	2.270	13.948	1.626	2.850	2.544	3512.28							
MIT_52		2	2.606	2.566	2.576	16.143	5.324	2.294	3.052	11.220	4.557	4.916	4.826	2108.30							
MIT_53		3	6.276	2.418	3.383	19.673	5.867	2.112	3.051	19.867	0.903	7.938	6.179	3635.94							
MIT_61		16	3.142	3.358	3.304	18.866	4.840	3.001	3.461	17.393	5.047	4.703	4.789	563.52							
MIT_62		2	1.913	3.396	3.025	15.996	1.834	2.817	2.571	16.028	7.589	7.638	7.626	726.11							
MIT_63		3	7.971	3.065	4.292	16.342	8.142	2.439	3.865	15.608	9.795	10.939	10.653	915.34							
MIF_41	(0.5,0.5)	14	1.727	2.907	2.317	11.580	1.577	2.671	2.124	11.699	3.953	0.266	2.110	1819.15							
MIF_42		2	4.276	3.206	3.741	9.970	3.027	3.402	3.215	9.322	1.868	2.220	2.044	1411.22							
MIF_43		3	4.413	4.115	4.264	9.713	3.781	3.497	3.639	11.009	5.774	7.902	6.838	1624.80							
MIF_51		15	3.035	2.937	2.986	14.581	2.093	2.711	2.402	15.328	0.903	1.802	1.353	2809.91							
MIF_52		2	3.576	3.462	3.519	11.357	3.064	3.105	3.085	12.069	3.297	2.871	3.084	1239.33							
MIF_53		3	4.154	4.343	4.249	15.418	3.142	3.964	3.553	18.488	2.061	6.276	4.169	4181.67							
MIF_61		16	3.492	3.472	3.482	16.277	3.198	3.008	3.103	15.032	4.410	3.452	3.931	966.73							
MIF_62		2	4.611	4.288	4.450	15.466	3.178	4.066	3.622	17.248	3.416	7.670	5.543	1044.94							
MIF_63		3	3.899	4.165	4.032	14.952	3.484	3.885	3.685	17.852	7.122	11.892	9.507	1068.26							

Tableau 5.13: Solutions détaillées pour les moyennes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [90%.70%].

Code	(w_1, w_2)	Instances $N_g \times G$	MOSA						MOSA/LS						LB1			Solver
			Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)				
MIC_41	(0.75,0.25)	14	48	1	5.133	2.550	4.487	9.836	4.256	2.706	3.869	11.488	0.737	0.539	0.688	1072.77		
MIC_42			2	6.756	4.127	6.099	11.411	6.138	3.772	5.547	9.221	5.003	2.494	4.376	1395.6			
MIC_43			3	3.770	3.285	3.649	11.068	2.975	2.824	2.937	10.394	7.115	7.004	7.087	1126.62			
MIC_51		15	68	1	3.314	3.250	3.298	18.192	2.844	2.982	2.879	19.413	0.733	1.082	0.820	1545.99		
MIC_52			2	7.061	3.234	6.104	16.119	5.892	2.695	5.093	15.487	5.216	1.982	4.408	3750.89			
MIC_53			3	4.611	3.858	4.423	17.089	4.188	3.944	4.127	18.209	2.750	5.527	3.444	4470.30			
MIC_61		16	98	1	5.610	3.154	4.996	19.122	4.666	3.287	4.321	18.870	5.217	3.694	4.836	731.00		
MIC_62			2	8.156	2.786	6.814	18.569	6.976	2.578	5.877	17.788	12.946	5.268	11.027	900.95			
MIC_63			3	3.729	3.059	3.562	17.277	2.864	2.849	2.86	16.672	8.260	9.582	8.591	1336.95			
MIT_41	(0.25,0.75)	14	48	1	3.431	3.013	3.118	8.763	2.929	2.549	2.644	9.401	3.119	1.111	1.613	1093.07		
MIT_42			2	6.724	3.332	4.18	9.375	6.880	2.998	3.969	10.209	2.309	3.894	3.498	925.31			
MIT_43			3	5.594	2.104	2.977	8.041	5.094	1.639	2.503	11.065	5.581	11.018	9.659	1733.93			
MIT_51		15	68	1	0.093	3.810	2.881	10.378	-0.246	3.607	2.644	15.292	4.719	1.771	2.508	1319.07		
MIT_52			2	2.698	3.209	3.081	16.390	3.593	2.967	3.124	16.209	4.172	5.156	4.910	4243.31			
MIT_53			3	8.478	3.530	4.767	15.306	7.153	3.138	4.142	14.788	3.446	6.514	5.747	4417.34			
MIT_61		16	98	1	4.026	3.660	3.752	15.172	4.049	3.282	3.474	16.176	5.047	5.660	5.507	355.27		
MIT_62			2	2.387	3.587	3.287	16.938	1.253	3.144	2.671	17.964	7.574	7.556	7.561	934.52			
MIT_63			3	8.711	3.313	4.663	15.552	8.979	2.915	4.431	17.089	8.704	12.208	11.332	1122.29			
MIF_41	(0.5,0.5)	14	48	1	3.336	3.074	3.205	11.488	2.645	2.854	2.75	10.278	1.824	0.699	1.262	1501.93		
MIF_42			2	4.959	4.193	4.576	9.836	3.618	4.049	3.834	11.225	0.600	2.976	1.788	1779.88			
MIF_43			3	3.295	4.254	3.775	10.119	3.003	4.015	3.509	11.028	6.535	7.107	6.821	1340.67			
MIF_51		15	68	1	3.130	4.047	3.589	18.567	3.346	3.804	3.575	12.965	1.002	0.964	0.983	1600.16		
MIF_52			2	5.019	4.455	4.737	15.049	4.455	3.950	4.203	14.201	9.595	0.887	5.241	6087.92			
MIF_53			3	4.299	4.689	4.494	15.948	3.784	4.380	4.082	16.280	5.426	4.4920	4.959	3137.63			
MIF_61		16	98	1	4.246	4.088	4.167	17.965	3.692	3.514	3.603	15.823	7.095	4.054	5.575	506.55		
MIF_62			2	3.747	4.208	3.978	18.046	4.168	3.674	3.921	16.122	6.697	7.984	7.341	888.60			
MIF_63			3	4.204	4.517	4.361	15.162	3.600	4.196	3.898	17.588	7.493	12.181	9.837	1107.50			

Tableau 5.14: Solutions détaillées pour les grandes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de 100%.

Code	(w_1, w_2)	Instances	$N_g \times G$	Axe	MOSA					MOSA/LS					Solver
					Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)		
LIC_71	(0.75,0.25)	17	413	1	3.876	1.984	3.403	10.690	3.045	1.760	2.724	11.147	7200		
LIC_72				2	8.772	3.394	7.428	10.869	7.627	3.267	6.537	10.987	7200		
LIC_73				3	3.131	2.088	2.870	11.989	2.368	1.648	2.188	11.726	7200		
LIC_81		18	713	1	6.410	1.849	5.270	17.770	5.894	1.737	4.855	16.678	7200		
LIC_82				2	8.979	3.038	7.494	15.964	8.384	3.058	7.053	16.759	7200		
LIC_83				3	4.322	1.164	3.533	16.560	3.647	0.939	2.970	16.632	7200		
LIC_91		19	1013	1	4.409	1.821	3.762	18.929	3.606	1.895	3.178	19.085	7200		
LIC_92				2	3.766	5.783	4.270	19.384	2.675	5.407	3.358	19.589	7200		
LIC_93				3	6.044	3.230	5.341	17.925	5.253	2.999	4.690	19.559	7200		
LIT_71	(0.25,0.75)	17	413	1	5.360	2.037	2.868	11.124	5.229	1.687	2.573	11.562	7200		
LIT_72				2	7.049	2.631	3.736	12.726	6.955	2.313	3.474	11.989	7200		
LIT_73				3	8.528	2.694	4.153	13.174	7.072	2.465	3.617	12.069	7200		
LIT_81		18	713	1	-0.343	2.657	1.907	16.929	0.312	2.357	1.846	16.857	7200		
LIT_82				2	7.524	2.495	3.752	16.976	6.875	2.221	3.385	17.008	7200		
LIT_83				3	8.342	3.267	4.536	15.164	8.468	2.783	4.204	16.934	7200		
LIT_91		19	1013	1	-0.282	2.361	1.700	19.440	0.972	2.003	1.745	19.405	7200		
LIT_92				2	5.489	1.801	2.723	18.464	6.366	0.965	2.315	19.560	7200		
LIT_93				3	8.439	3.944	5.068	18.872	8.350	3.644	4.821	18.579	7200		
LIF_71	(0.5,0.5)	17	413	1	3.451	3.882	3.667	10.573	3.119	3.441	3.280	11.810	7200		
LIF_72				2	8.041	4.801	6.421	11.726	6.982	5.549	6.266	13.147	7200		
LIF_73				3	4.009	3.874	3.942	12.697	3.611	3.312	3.462	11.628	7200		
LIF_81		18	713	1	3.531	3.123	3.327	16.862	2.639	2.923	2.781	16.879	7200		
LIF_82				2	6.944	5.556	6.250	16.946	6.308	5.220	5.764	17.213	7200		
LIF_83				3	4.713	3.511	4.112	17.828	4.024	3.626	3.825	16.535	7200		
LIF_91		19	1013	1	2.361	1.773	2.067	18.582	1.342	2.290	1.816	19.208	7200		
LIF_92				2	3.800	4.414	4.107	19.995	3.794	3.554	3.674	20.139	7200		
LIF_93				3	4.599	3.897	4.248	19.872	3.891	3.728	3.810	17.088	7200		

Tableau 5.15: Solutions détaillées pour les grandes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [70%, 90%].

Code	(w_1, w_2)	Instances	$N_g \times G$	Axe	MOSA					MOSA/LS					Solver
					Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)		
LIC_71	(0.75,0.25)	17	413	1	8.239	2.490	6.802	12.698	7.948	2.262	6.527	11.067	7200		
LIC_72				2	8.901	1.284	6.997	11.025	8.226	1.285	6.491	13.028	7200		
LIC_73				3	4.410	4.837	4.517	12.810	3.677	4.568	3.900	14.500	7200		
LIC_81		18	713	1	3.004	2.554	2.892	16.770	2.023	2.236	2.076	17.689	7200		
LIC_82				2	3.259	3.449	3.307	15.569	2.567	3.241	2.736	16.208	7200		
LIC_83				3	4.751	4.944	4.799	16.039	3.921	4.521	4.071	17.123	7200		
LIC_91		19	1013	1	4.671	3.198	4.303	18.745	4.133	2.949	3.837	18.578	7200		
LIC_92				2	4.769	3.745	4.513	21.589	3.658	3.453	3.607	18.698	7200		
LIC_93				3	3.216	4.395	3.511	20.320	1.925	4.017	2.448	19.687	7200		
LIT_71	(0.25,0.75)	17	413	1	3.388	4.382	4.134	12.741	4.995	3.943	4.206	11.257	7200		
LIT_72				2	8.496	3.777	4.957	13.698	6.519	3.470	4.232	11.789	7200		
LIT_73				3	8.640	3.600	4.860	13.069	7.752	3.068	4.239	12.987	7200		
LIT_81		18	713	1	1.296	3.441	2.905	16.928	2.317	3.179	2.964	16.247	7200		
LIT_82				2	1.086	3.332	2.771	17.962	1.036	2.976	2.491	16.951	7200		
LIT_83				3	5.383	4.270	4.548	16.752	5.090	3.980	4.258	17.456	7200		
LIT_91		19	1013	1	0.043	3.647	2.746	19.814	0.295	3.392	2.618	18.475	7200		
LIT_92				2	5.220	3.529	3.952	17.789	6.743	3.076	3.993	19.638	7200		
LIT_93				3	5.404	2.871	3.504	18.457	4.544	2.974	3.367	19.032	7200		
LIF_71	(0.5,0.5)	17	413	1	4.540	4.220	4.380	11.124	4.595	3.782	4.189	12.571	7200		
LIF_72				2	3.947	4.479	4.213	12.562	3.459	3.909	3.684	11.537	7200		
LIF_73				3	5.281	5.953	5.617	13.269	4.529	5.500	5.015	12.067	7200		
LIF_81		18	713	1	1.992	2.021	2.007	17.895	1.709	1.783	1.746	16.258	7200		
LIF_82				2	3.851	4.094	3.973	16.987	3.751	3.610	3.681	15.217	7200		
LIF_83				3	4.053	4.975	4.514	15.827	4.166	4.484	4.325	15.847	7200		
LIF_91		19	1013	1	2.650	3.028	2.839	18.562	2.602	2.875	2.739	17.384	7200		
LIF_92				2	4.832	4.388	4.610	19.062	4.011	4.185	4.098	18.823	7200		
LIF_93				3	3.633	4.138	3.886	18.587	3.184	4.236	3.710	21.038	7200		

Tableau 5.16: Solutions détaillées pour les grandes instances basées sur différents coefficients de pondération pour le makespan et le critère de consommation d'énergie pour un taux d'apprentissage de [90%, 70%].

Code	(w_1, w_2)	Instances	$N_g \times G$	Axe	MOSA					MOSA/LS					Solver
					Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	Ave % RPD_{Cmax}	Ave % RPD_{TEC}	PF % CPU_{time} (sec)	CPU_{time} (sec)	
LIC_71	(0.75,0.25)	17	413	1	7.147	4.382	6.456	11.627	5.702	3.787	5.223	13.058	7200		
LIC_72				2	8.974	4.596	7.880	13.026	8.568	4.468	7.543	12.278	7200		
LIC_73				3	6.714	3.544	5.922	13.147	6.091	3.264	5.384	12.697	7200		
LIC_81		18	713	1	2.871	1.736	2.587	16.023	1.910	1.953	1.921	15.545	7200		
LIC_82				2	7.580	2.868	6.402	16.899	6.511	2.436	5.492	16.623	7200		
LIC_83				3	5.435	5.244	5.387	16.239	4.678	4.791	4.706	14.294	7200		
LIC_91		19	1013	1	2.832	4.061	3.139	18.837	2.325	3.932	2.727	18.017	7200		
LIC_92				2	8.281	4.479	7.331	19.125	7.749	4.189	6.859	19.368	7200		
LIC_93				3	4.308	5.671	4.649	20.904	3.347	5.244	3.821	19.530	7200		
LIT_71	(0.25,0.75)	17	413	1	6.303	3.949	4.538	12.058	6.289	3.692	4.341	11.455	7200		
LIT_72				2	8.889	3.766	5.047	10.998	8.432	2.934	4.309	12.068	7200		
LIT_73				3	8.471	3.194	4.513	13.264	8.622	2.740	4.211	13.368	7200		
LIT_81		18	713	1	-2.802	3.752	2.114	14.964	-2.259	3.432	2.009	15.369	7200		
LIT_82				2	3.198	5.016	4.562	16.608	3.043	4.773	4.341	14.987	7200		
LIT_83				3	6.378	4.002	4.596	17.022	6.237	3.586	4.249	14.569	7200		
LIT_91		19	1013	1	-0.248	4.334	3.189	18.554	-0.456	3.932	2.835	18.887	7200		
LIT_92				2	3.861	4.327	4.211	17.874	4.519	3.954	4.095	19.023	7200		
LIT_93				3	8.652	4.658	5.657	18.281	8.341	4.348	5.346	17.931	7200		
LIF_71	(0.5,0.5)	17	413	1	3.969	4.138	4.054	11.120	3.163	3.632	3.398	12.659	7200		
LIF_72				2	5.875	6.294	6.085	13.489	4.901	5.583	5.242	11.726	7200		
LIF_73				3	2.497	4.362	3.430	13.690	2.188	3.871	3.030	12.698	7200		
LIF_81		18	713	1	4.313	3.836	4.075	15.684	3.985	3.621	3.803	16.324	7200		
LIF_82				2	4.309	4.415	4.362	16.911	3.844	4.147	3.996	17.023	7200		
LIF_83				3	4.393	4.809	4.646	15.697	3.916	4.899	4.408	14.951	7200		
LIF_91		19	1013	1	2.882	3.228	3.055	18.437	2.942	2.687	2.815	20.836	7200		
LIF_92				2	5.745	5.941	5.843	19.599	5.000	6.163	5.582	18.333	7200		
LIF_93				3	5.564	6.025	5.795	18.486	4.533	5.922	5.228	17.956	7200		

— Analyse des performances de MOSA/LS

Les expérimentations réalisées pour analyser les performances de l'algorithme MOSA ont été reproduites en prenant en considération la recherche locale MOSA/LS. Ces tests ont été réalisés sous les mêmes conditions définies dans les sections précédentes, en utilisant les mêmes instances et paramètres d'étude. Cela a généré un total d'environ 3240 tests pour l'algorithme MOSA et MOSA/LS. Les résultats obtenus pour MOSA/LS sont répertoriés dans les tableaux 5.8 à 5.16.

Pour les petites instances avec un taux d'apprentissage de 100% ($LR2 = LR3 = 100\%$), le pourcentage d'écart relatif de la fonction de pondération $PF\%$ est réduit d'environ 16% pour le makespan lorsque les coefficients de pondération (w_1, w_2) sont égaux à $(0.75, 0.25)$. Pour le TEC avec la paire (w_1, w_2) égale à $(0.75, 0.25)$, une réduction de 14% est enregistrée pour le $PF\%$. Par ailleurs, lorsque le coefficient de pondération est le même pour les deux objectifs ($w_1 = w_2 = 0.5$), cette réduction atteint la valeur de 11%.

En tenant compte de l'effet d'apprentissage au niveau de M2 ($LR2 = 70\%$), le $PF\%$ est réduit d'environ 27% pour le makespan, de 22% pour le TEC , et de 19% pour la paire $(0.5, 0.5)$. Par ailleurs, en inversant le taux d'apprentissage entre les machines M2 et M3 ($LR3 = 70\%$), le $PF\%$ du $Cmax$ est réduit de 32%, du TEC de 21%, et pour la paire $(0.5, 0.5)$, une diminution de 15% est recensée.

En présence des moyennes instances avec $LR = 100\%$, le $PF\%$ est réduit d'environ 17% pour la quasi-totalité des paires de coefficients de pondération. Parallèlement, cette réduction est constatée à 17% pour le $Cmax$, 10% pour le TEC , et 14% pour la paire $(0.5, 0.5)$, en tenant compte de l'effet d'apprentissage au niveau de M2 et M3.

Face aux grandes instances, le pourcentage d'écart relatif pour $PF\%$ obtenu par MOSA/LS est réduit de 12% en présence d'un taux d'apprentissage de 100%, pour l'ensemble des paires de coefficients de pondération. Cependant, en intégrant l'effet d'apprentissage, cette valeur est ramenée à 10%.

Par conséquent, l'analyse des résultats obtenus révèle que l'intégration de la procédure de recherche locale dans l'algorithme réduit considérablement le pourcentage d'écart relatif de ce dernier. Ce constat est clairement illustré dans les figures 5.3, 5.4 et 5.5. En examinant ces trois graphiques, on observe que la métaheuristique MOSA, renforcée par la recherche locale, fournit de meilleurs résultats en réduisant significativement les deux fonctions objectifs, ainsi que la fonction pondérée PF , pour toutes les instances générées. Cela est vrai quel que soit le taux d'apprentissage appliqué, les coefficients de pondération fixés ou l'axe sur lequel repose le temps de configuration. Dans certains cas, cette réduction peut atteindre jusqu'à 97% de la valeur initiale trouvée (la valeur de $PF\%$ obtenue sans l'application de la recherche locale).

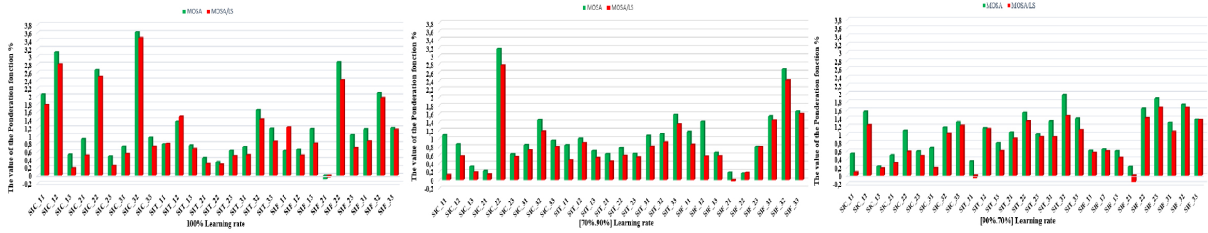


Figure 5.3: Impact de la recherche locale sur la valeur de la fonction pondérée en considérant différents taux d'apprentissage et différents coefficients de pondération pour les petites instances.

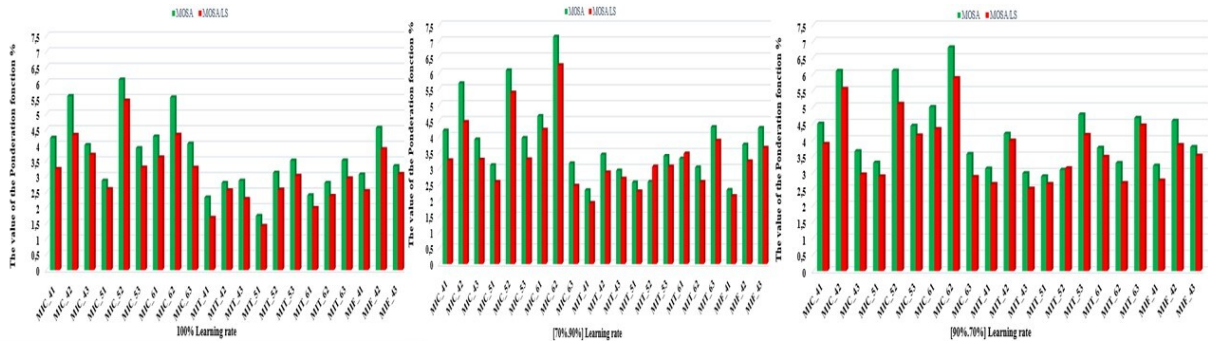


Figure 5.4: Impact de la recherche locale sur la valeur de la fonction pondérée en considérant différents taux d'apprentissage et différents coefficients de pondération pour les moyennes instances.

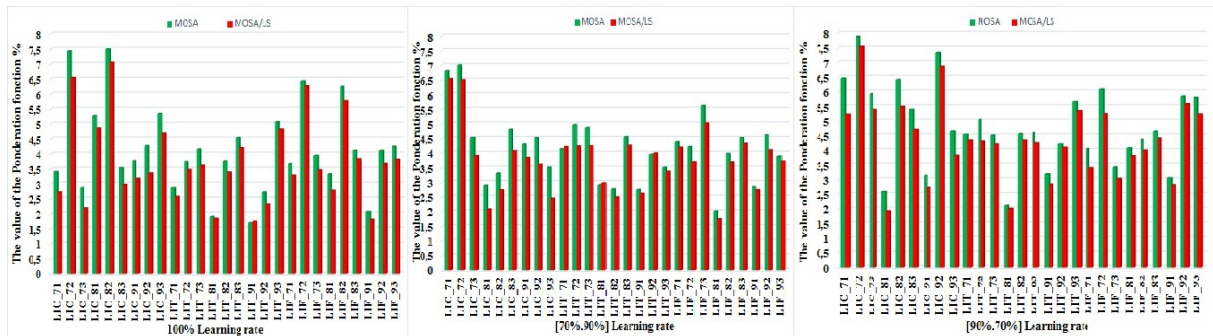


Figure 5.5: Impact de la recherche locale sur la valeur de la fonction pondérée en considérant différents taux d'apprentissage et différents coefficients de pondération pour les grandes instances.

En observant les résultats obtenus, il est essentiel de retenir les points suivants :

- Le premier point concerne le temps de calcul de l'algorithme MOSA, qui se révèle nettement inférieur à celui du solveur, notamment pour les instances de taille moyenne et grande.
- Le deuxième point à analyser est en rapport avec la prise en compte de différents niveaux de ratios de temps de configuration. Ce facteur a un impact significatif sur

l'ordonnement des groupes, et par conséquent, sur les critères d'optimisation. Les expériences réalisées révèlent que le pourcentage d'écart relatif le plus élevé obtenu est observé avec l'axe 2 et l'axe 3. Dans ces cas, les temps de configuration entre les groupes sur les machines sont presque identiques pour l'axe 2 et suivent un ordre décroissant pour l'axe 3. En revanche, le *RPD* obtenu avec l'axe 1 (ordre croissant des temps de configuration) est significativement plus bas que les deux autres, principalement en raison de la réduction des temps d'inactivité de la machine.

- Étant donné les variations des temps de traitement entre les différentes machines, considérer un temps de configuration identique sur toutes les machines (axe 2) génère des périodes d'inactivité. Pendant ces périodes, les machines restent inoccupées en attendant l'arrivée d'un nouveau job, entraînant une augmentation de la consommation d'énergie. Dans le cas de l'axe 3, les temps de configuration entre les machines suivent un ordre décroissant, entraînant une perte significative de temps pour démarrer le processus de fabrication sur la première machine. En conséquence, les autres machines restent inactives pendant un certain temps, ce qui augmente la consommation d'énergie. En revanche, avec l'axe 1, le traitement des jobs débute rapidement sur la première machine, tandis que la phase d'installation est effectuée sur les autres machines. Cela se traduit par une réduction du temps d'exécution global (Makespan) et de la consommation d'énergie, grâce à une minimisation des périodes d'inactivité.
- Le troisième et dernier aspect à considérer concerne la variation des taux d'apprentissage. Les résultats obtenus révèlent une augmentation du pourcentage d'écart relatif lors de la prise en compte de l'effet d'apprentissage. Ceci est attribuable à une diminution significative des critères d'optimisation. Donc, pour atteindre la valeur optimale obtenue par le solveur, un ordonnancement précis des tâches et des groupes devient nécessaire.
- De plus, la considération d'un taux d'apprentissage de 70% au niveau de M3 réduit considérablement les deux fonctions objectives et crée un écart plus important entre MOSA et la solution optimale. Dans ce cas, il est impératif de trouver les vitesses de production et de transport appropriées pour parvenir à une réduction encore plus conséquente des fonctions objectives.

Pour conclure, les algorithmes MOSA et MOSA/LS démontrent une meilleure performance avec l'augmentation de la taille du problème. Ainsi, l'algorithme MOSA se

révèle efficace pour l'ordonnement des tâches et des groupes dans le problème FSDGS bi-objectif avec effet d'apprentissage. De plus, cela est réalisé dans un laps de temps relativement court par rapport au modèle mathématique.

5.10 Conclusion

Dans ce chapitre, notre focalisation s'est dirigée vers l'adaptation de la version multi-objectif de l'algorithme du recuit simulé MOSA pour la minimisation simultanée du makespan et de la consommation totale d'énergie du système dans un atelier de machines en série. Cet environnement implique des temps de configuration dépendant de la séquence, un effet d'apprentissage, ainsi que des contraintes de transport à capacité limitée, ajoutant une complexité au processus d'ordonnement des tâches qui peut nécessiter plusieurs transporteurs simultanément.

Nous avons montré l'efficacité de l'algorithme MOSA sur un grand nombre de configurations en le comparant au modèle mathématique du problème FSDGS pour les petites et quelques moyennes instances. Cependant, le problème étudié est assez complexe, et une méthode de résolution exacte ne peut fournir des solutions optimales en un temps raisonnable. Donc, l'efficacité de l'algorithme pour les grandes instances a été approuvée en comparant les résultats aux bornes inférieures développées.

Nous avons également analysé l'impact des méthodes de recherche locales sur les performances de MOSA, en variant les taux d'apprentissage et les coefficients de pondération associés à chaque objectif. Cette analyse nous a permis de dégager des résultats préliminaires significatifs pour évaluer le système décisionnel d'ordonnement basé sur l'algorithme MOSA. Voici les principales conclusions :

- Le modèle mathématique proposé s'est avéré robuste, résolvant de manière optimale plusieurs instances de taille moyenne.
- L'étude réalisée est assez innovante et touche plusieurs aspects concrets de l'industrie, tels que la prise de décision reflétant l'expertise des employés, l'intégration des temps de configuration dépendant de la séquence avec différents ratios, et l'intégration de la variante de transport dans le problème de FSDGS.
- Nous avons démontré que MOSA fournit des solutions satisfaisantes dans un temps de calcul assez réduit par rapport au modèle mathématique, surtout en présence des grandes instances.
- Nous avons analysé l'impact des différents taux d'apprentissage sur l'ordonnement des jobs et des groupes, ainsi que sur l'affectation des résultats obtenus par MOSA, soulignant les répercussions significative d'une affectation adéquate des employés aux machines.

- L'influence des ratios de temps de configuration sur les performances de l'algorithme MOSA et sur la minimisation des deux objectifs a aussi été analysée, et nous sommes arrivés à la conclusion que si ces temps de changement ne sont pas idéalement gérés, cela peut perturber grandement le séquençement des groupes et engendrer une augmentation du makespan.
- La variante du transport ajoute une note de complexité élevée au modèle proposé, imposant des restrictions de traitement sur les machines et un calibrage idéal des vitesses de transport et de traitement, car la non-gérance de ces variations endommage les solutions de MOSA et les éloigne complètement du voisinage des solutions de bonne qualité.
- En outre, l'amélioration des résultats par le renforcement de MOSA avec une procédure de recherche locale a été démontrée à travers l'ensemble des expérimentations, montrant des pourcentages d'amélioration considérables, atteignant jusqu'à 40% par rapport aux résultats initiaux sans ce mécanisme évolutif.

Conclusion Générale et Perspectives

Cette thèse aborde le problème d’ordonnancement à machines en série (flow shop) en se basant sur une étude de cas industrielle réelle, et se structure autour de trois principaux axes : l’ordonnancement en tenant compte de la technologie de groupe, l’effet d’apprentissage influençant le temps de traitement des jobs et leurs séquençages. Les temps de configuration, dépendant de la séquence, impactant chaque paire consécutive de groupes devant être réalisées sur une même machine, tandis que le temps de transport a des répercussions directes sur les temps de production totale. L’objectif est l’optimisation d’un ordonnancement éco-énergétique soucieux de l’environnement.

En nous basant sur les problématiques évoquées précédemment, nous avons étudié et résolu divers problèmes, intégrant progressivement l’ensemble des axes qui constituent les piliers de cette étude, détaillés dans les différents chapitres de cette thèse. Ce manuscrit compile nos recherches sur plusieurs méthodes de résolution, incluant divers modèles mathématiques fondés sur la programmation linéaire en nombres entiers mixtes. Il inclut également des méthodes approchées, telles que les métaheuristiques, capables de fournir des solutions de qualité en un temps très court, contournant ainsi les limites des modèles mathématiques face à des problèmes d’ordonnancement de plus en plus complexes. Les trois piliers de notre recherche, représentés par trois problèmes d’ordonnancement, nous ont conduits à proposer plusieurs contributions et à définir des axes de recherche futurs, résumés ci-dessous et illustrés par la figure 5.6.

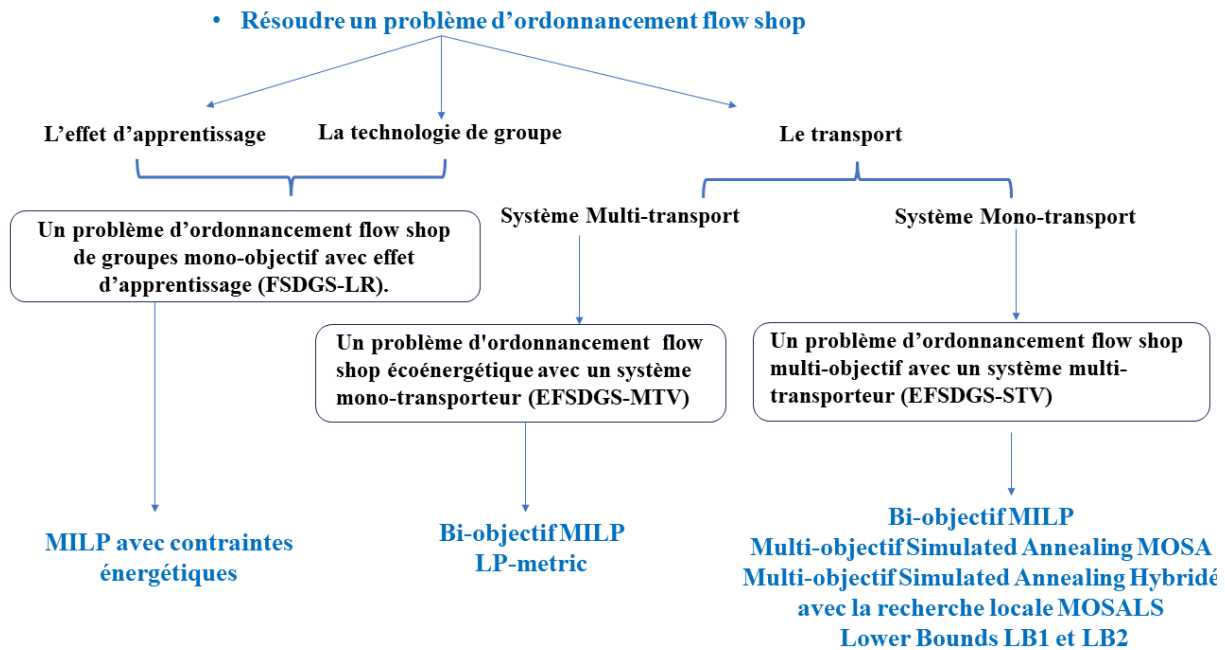


Figure 5.6: Les trois principaux axes de la thèse

Le premier axe établi les fondations sur lesquelles repose cette étude en exposant en détail l'approche intégrative de l'effet d'apprentissage. Dans un premier temps, nous avons proposé un modèle mathématique pour la résolution mono-objectif du problème. Ensuite, nous avons pris en considération la seconde problématique constituant le second pilier de cette étude, basée sur la mise en œuvre et l'intégration de la technologie de groupe ainsi que des temps de configuration dépendant de la séquence dans le processus décisionnel de notre problème de flow shop avec effet d'apprentissage. En conséquence, un second modèle mathématique a été développé et appliqué pour résoudre divers scénarios visant à minimiser le makespan tout en tenant compte des contraintes énergétiques liées aux émissions de CO₂. Cette approche de résolution s'inscrit dans une perspective d'optimisation éco-énergétique.

Le second axe expose le dernier pilier sur lequel reposent ces travaux, à savoir le temps de transport. Ce dernier a été intégré au modèle précédent avec un effet d'apprentissage pour les problèmes de flow shop avec groupes et temps de configuration de la séquence. Dans la littérature, cette variante du transport est rarement abordée, et les quelques travaux qui le font le considèrent sous différentes hypothèses dans le domaine de l'ordonnancement. À cette fin, un premier modèle mathématique aborde la première hypothèse supposée d'un système multi-transport à capacité limitée. La résolution de ce premier modèle s'est faite à travers l'adaptation de la méthode LP-metric, qui, à travers les différents scénarios et tests réalisés, a démontré l'efficacité du modèle mathématique, capable de résoudre optimalement un grand nombre d'instances moyennes, démontrant

ainsi sa robustesse. Cependant, elle met également en évidence le caractère conflictuel entre les deux objectifs. Le second modèle mathématique, intégrant l'hypothèse d'un système mono-transport à capacité limitée, représentera le dernier modèle modélisé et servira de point de comparaison pour la méthode de résolution approchée. Enfin, deux bornes inférieures sont proposées et plusieurs tests ont été réalisés afin de démontrer leurs validités.

Le problème d'ordonnancement FSDGS avec effet d'apprentissage et temps de transport est un problème NP-difficile. La résolution de ce problème à travers des méthodes exactes s'avère être vaine lorsque la taille des instances augmente, générant ainsi des temps de calcul interminables. Par conséquent, l'adaptation d'une métaheuristique sonne comme une évidence, car elle permet de fournir des solutions de bonne qualité dans un laps de temps considérablement réduit.

Dans cette troisième contribution, nous avons présenté la structure de l'algorithme MOSA proposé, en définissant les paramètres appropriés grâce à une analyse de sensibilité. L'algorithme s'est avéré performant dans sa prise de décision concernant l'ordonnancement des tâches et des groupes dans un environnement de machines en série avec effet d'apprentissage et temps de transport. L'évaluation de l'algorithme s'est basée sur deux critères : le pourcentage d'erreur relative ($RPD\%$) et le temps de calcul CPU pour différentes instances.

Pour les petites instances, l'algorithme MOSA a produit des résultats satisfaisants avec des temps de calcul CPU très courts, tandis que le temps CPU du MILP était plus important, mais toujours de l'ordre des secondes. Pour les instances de taille moyenne et grande, le pourcentage d'erreur a augmenté, mais est resté dans des limites acceptables, ne dépassant pas 9%. De plus, le temps de calcul était négligeable par rapport au temps de calcul du MILP. Les bornes inférieures ont également donné des résultats prometteurs, avec un pourcentage d'erreur maximal de 19% pour les grandes instances, dans un temps de calcul négligeable d'environ une seconde.

D'autre part, l'inclusion des différentes variantes de cette étude complexifie davantage la situation. Dans ce contexte, une grande partie de l'analyse a été dédiée à l'application d'une méthode de recherche locale. Cette approche présente l'avantage de se concentrer intensivement sur l'exploration et l'exploitation de l'espace de recherche pour améliorer les performances de l'algorithme MOSA. Les résultats obtenus sont prometteurs, avec l'algorithme MOSA/LS dépassant l'algorithme MOSA en termes de qualité de solution. Ceci est attribuable à l'impact positif de la procédure de recherche locale sur les performances de ce dernier.

Perspectives

L'étude présentée dans cette thèse offre un potentiel considérable pour améliorer le processus décisionnel des opérations d'ordonnancement, en intégrant la technologie de groupe et l'effet d'apprentissage. Ce domaine, vaste et en constante expansion, attire de plus en plus l'intérêt des chercheurs en raison de son importance pour les entreprises manufacturières et de la croissance significative du nombre de publications.

Cependant, l'effet d'apprentissage nécessite une réflexion approfondie sur ses implications pour la gestion opérationnelle et l'utilisation optimale des capacités de production. Par conséquent, il est essentiel de poursuivre les efforts pour enrichir cette recherche, qui peut ouvrir de nouvelles pistes d'investigation et avoir des implications importantes dans les domaines suivants :

Perspectives pour le modèle d'effet d'apprentissage

Dans ces travaux, une hypothèse forte est faite en rapport avec les temps de traitement qui dépendent de l'effet d'apprentissage, qui varie à son tour en fonction de la position l , avec $P(j, r)$ représentant la durée de la tâche j à la position l . La littérature montre qu'un deuxième modèle pertinent existe, où la durée de la tâche dépend de sa date de début t . Il serait donc intéressant d'étendre notre approche globale pour inclure ce modèle temporel $p(j, t)$. Dans ce cas, l'effet d'apprentissage dépend de la somme des temps de traitement des tâches déjà ordonnancé $t = \sum_{s=1}^{j-1} p[s]$.

Pour approfondir davantage cette piste et après une analyse détaillée du cas d'étude de bielles forgées, on a constaté que la dégradation des machines a un impact significatif sur les temps de traitement. Ainsi, l'exploration des effets conjugués de la détérioration des machines et de l'apprentissage, ainsi que l'évaluation des compromis entre ces facteurs, pourraient s'avérer bénéfiques.

Perspectives pour le problème flow shop

L'incorporation des ressources consommables comme les lubrifiants pourrait constituer une piste de recherche intéressante, surtout compte tenu de leur consommation importante dans l'industrie métallurgique. Pouvoir également analyser les situations de compromis résultantes de l'intégration simultanée de l'effet d'apprentissage et des ressources consommables serait assez intéressant et prometteur.

L'intégration des différents paramètres de l'étude et son adaptation dans le domaine hospitalier révèle un aspect crucial en rapport avec l'impact significatif de l'effet d'ap-

prentissage sur l'efficacité du personnel hospitalier. En milieu hospitalier, où la prise en charge des patients nécessite une combinaison complexe de compétences techniques, de gestion du temps et de compétences interpersonnelles, l'apprentissage continu joue un rôle déterminant.

Premièrement, l'intégration des paramètres de l'étude implique la compréhension et l'analyse de divers facteurs tels que les protocoles médicaux, les nouvelles technologies, les dynamiques d'équipe et les besoins des patients. Ces éléments interagissent de manière complexe pour influencer la prestation des soins de santé.

Deuxièmement, dans un environnement aussi dynamique et stressant que l'hôpital, où les conditions peuvent évoluer rapidement, l'adaptation des pratiques et des compétences est essentielle. L'effet d'apprentissage se manifeste lorsque le personnel hospitalier acquiert de nouvelles connaissances, compétences et stratégies à travers leur expérience quotidienne et la formation continue. Cette capacité à s'adapter et à apprendre en continu est ce qui permet aux professionnels de santé de s'améliorer et d'optimiser leurs performances au fil du temps.

Troisièmement, l'efficacité du personnel hospitalier est directement liée à leur capacité à intégrer ces nouveaux apprentissages dans leur pratique quotidienne. Cela inclut non seulement l'application des meilleures pratiques médicales, mais aussi l'amélioration des processus de communication et de collaboration au sein des équipes multidisciplinaires. Lorsque les professionnels de la santé sont encouragés à apprendre et à s'adapter, cela se traduit souvent par une meilleure qualité des soins, une réduction des erreurs médicales et une plus grande satisfaction des patients.

Perspectives d'ordre général

Plusieurs orientations de recherche potentielles peuvent être envisagées de manière générale :

- L'ensemble des paramètres considérés dans cette étude peut également être adapté à une configuration d'atelier flow shop flexible composé de plusieurs étages. Dans ce contexte, l'analyse du compromis entre la réduction du temps total d'exécution et la consommation d'énergie résultant de cette adaptation, ainsi que la variation des effets de la détérioration des différentes ressources à chaque étage, serait très intéressante.
- Intégrer des actions de maintenance au niveau des machines pour prévenir les détériorations causées par une forte utilisation est également envisageable pour assurer le bon fonctionnement du système.
- Améliorer davantage les bornes inférieures serait également possible en intégrant des

heuristiques dédiées pour les caractéristiques du système.

- Une étude du système, présenté dans cette thèse, avec d'autres critères d'optimisation tels que la minimisation des retards.
- Adaptation d'autre métaheuristiques comme le NSGAIII au problème considéré, en le comparant à l'algorithme MOSA proposé.

Références

- Abadi, I. K., Hall, N. G., & Sriskandarajah, C. (2000). Minimizing cycle time in a blocking flowshop. *Operations Research*, 48(1), 177–180.
- Adressi, A., Hassanpour, S., & Azizi, V. (2016). Solving group scheduling problem in no-wait flexible flowshop with random machine breakdown. *Decision Science Letters*, 5(1), 157–168.
- Aghelinejad, M., Ouazene, Y., & Yalaoui, A. (2019). Complexity analysis of energy-efficient single machine scheduling problems. *Operations Research Perspectives*, 6, 100105.
- Alfares, H., Mohammed, A., & Ghaleb, M. (2021). Two-machine scheduling with aging effects and variable maintenance activities. *Computers & Industrial Engineering*, 160, 107586.
- Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 246(2), 345–378.
- Allahverdi, A., Gupta, J. N., & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, 27(2), 219–239.
- Allahverdi, A., Ng, C. T., Cheng, T. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European journal of operational research*, 187(3), 985–1032.
- Allali, K., Aqil, S., & Belabid, J. (2022). Distributed no-wait flow shop problem with sequence dependent setup time : Optimization of makespan and maximum tardiness. *Simulation Modelling Practice and Theory*, 116, 102455.
- Allison, J. D. (1990). Combining petrov’s heuristic and the cds heuristic in group scheduling problems. *Computers & Industrial Engineering*, 19(1-4), 457–461.
- Andrés, C., Albarracín, J. M., Tormo, G., Vicens, E., & Garcia-Sabater, J. P. (2005). Group technology in a hybrid flowshop environment : A case study. *European Journal of Operational Research*, 167(1), 272–281.
- Aqil, S., & Allali, K. (2021). Two efficient nature inspired meta-heuristics solving blocking hybrid flow shop manufacturing problem. *Engineering Applications of Artificial Intelligence*, 100, 104196.
- Arabzad, S. M., Ghorbani, M., & Zolfani, S. H. (2015). A multi-objective robust opti-

- mization model for a facility location-allocation problem in a supply chain under uncertainty. *Engineering Economics*, 26(3), 227–238.
- Artiba, A., Dhaevers, V., Duvivier, D., & Elmaghraby, S. E. (2011). A multi-model approach for production planning and scheduling in an industrial environment. In *Planning production and inventories in the extended enterprise : A state-of-the-art handbook, volume 2* (pp. 489–530). Springer.
- Avalos-Rosales, O., Angel-Bello, F., & Alvarez, A. (2015). Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, 76, 1705–1718.
- Azizi, V., & Hu, G. (2020). A branch and bound algorithm to solve a two-machine no-wait flowshop scheduling problem with truncated learning function. *International Journal of Management Science and Engineering Management*, 15(2), 89–95.
- Bachman, A., & Janiak, A. (2004). Scheduling jobs with position-dependent processing times. *Journal of the Operational Research Society*, 55(3), 257–264.
- Bai, D., Bai, X., Yang, J., Zhang, X., Ren, T., Xie, C., & Liu, B. (2021). Minimization of maximum lateness in a flowshop learning effect scheduling with release dates. *Computers & Industrial Engineering*, 158, 107309.
- Baker, K. R. (1974). Introduction to sequencing and scheduling. (*No Title*).
- Baptiste, P. (2005). *Gestion de production et ressources humaines : méthodes de planification dans les systèmes productifs*. Presses inter Polytechnique.
- Bassem, J., Patrick, S., & Jacques, T. (2013). *Métaheuristiques pour l’ordonnancement monocritère des ateliers de production*. Lavoisier.
- Bektur, G. (2022). Distributed flow shop scheduling problem with learning effect, setups, non-identical factories, and eligibility constraints. *International Journal of Industrial Engineering*, 29(1).
- Biskup, D. (1999). Single-machine scheduling with learning considerations. *European Journal of Operational Research*, 115(1), 173–178.
- Biskup, D. (2008). A state-of-the-art review on scheduling with learning effects. *European Journal of Operational Research*, 188(2), 315–329.
- Bosio, A., & Righini, G. (2006). A dynamic programming algorithm for the single-machine scheduling problem with deteriorating processing times. *Electronic Notes in Discrete Mathematics*, 25, 139–142.
- Bozorgirad, M. A., & Logendran, R. (2016). A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. *The international journal of advanced manufacturing technology*, 83, 1135–1151.
- Browne, S., & Yechiali, U. (1990). Scheduling deteriorating jobs on a single processor.

- Operations Research*, 38(3), 495–498.
- Brucker, P. (2007). Classification of scheduling problems. *Scheduling Algorithms*, 1–10.
- Burbidge, J. L. (1996). The first step in planning group technology. *International Journal of Production Economics*, 43(2-3), 261–266.
- Carlier, J., & Chrétienne, P. (1988). *Problèmes d’ordonnancement : modélisation, complexité, algorithmes*. Masson.
- Carlier, J., Chrétienne, P., Erschler, J., Hanen, C., Lopez, P., Munier, A., . . . others (1993). Les problèmes d’ordonnancement. *RAIRO-Operations Research*, 27(1), 77–150.
- Caumont, A. (2006). *Le problème de jobshop avec contraintes : modélisation et optimisation* (Thèse de doctorat non publiée). Université Blaise Pascal-Clermont-Ferrand II.
- Celano, G., Costa, A., & Fichera, S. (2010). Constrained scheduling of the inspection activities on semiconductor wafers grouped in families with sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, 46(5), 695–705.
- Chen, H., Zhou, S., Li, X., & Xu, R. (2014). A hybrid differential evolution algorithm for a two-stage flow shop on batch processing machines with arbitrary release times and blocking. *International Journal of Production Research*, 52(19), 5714–5734.
- Chen, W.-J. (2008). Single-machine scheduling with family setup times in a manufacturing system. *Engineering Optimization*, 40(6), 579–589.
- Cheng, H.-M., & Ying, K.-C. (2011). Minimizing makespan in a flow-line manufacturing cell with sequence dependent family setup times. *Expert Systems with Applications*, 38(12), 15517–15522.
- Cheng, M., Tadikamalla, P. R., Shang, J., & Zhang, B. (2015). Two-machine flow shop scheduling with deteriorating jobs : minimizing the weighted sum of makespan and total completion time. *Journal of the Operational Research Society*, 66(5), 709–719.
- Cheng, M., Xiao, S., Luo, R., & Lian, Z. (2018). Single-machine scheduling problems with a batch-dependent aging effect and variable maintenance activities. *International Journal of Production Research*, 56(23), 7051–7063.
- Cheng, T., & Wang, G. (2000). Single machine scheduling with learning effect considerations. *Annals of Operations Research*, 98(1), 273–290.
- Cheng, T. E., Cheng, S.-R., Wu, W.-H., Hsu, P.-H., & Wu, C.-C. (2011). A two-agent single-machine scheduling problem with truncated sum-of-processing-times-based learning considerations. *Computers & Industrial Engineering*, 60(4), 534–541.
- Cheng, T. E., Ding, Q., & Lin, B. M. (2004). A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research*, 152(1), 1–13.
- Cheng, T. E., Janiak, A., & Kovalyov, M. Y. (2001). Single machine batch scheduling with

- resource dependent setup and processing times. *European Journal of Operational Research*, 135(1), 177–183.
- Cheng, T. E., Wu, C.-C., & Lee, W.-C. (2008). Some scheduling problems with sum-of-processing-times-based and job-position-based learning effects. *Information Sciences*, 178(11), 2476–2487.
- Chyu, C.-C., & Chang, W.-S. (2010). A pareto evolutionary algorithm approach to bi-objective unrelated parallel machine scheduling problems. *The International Journal of Advanced Manufacturing Technology*, 49, 697–708.
- Costa, A., Cappadonna, F., & Fichera, S. (2020). Minimizing makespan in a flow shop sequence dependent group scheduling problem with blocking constraint. *Engineering Applications of Artificial Intelligence*, 89, 103413.
- Costa, A., Cappadonna, F. A., & Fichera, S. (2014). Joint optimization of a flow-shop group scheduling with sequence dependent set-up times and skilled workforce assignment. *International Journal of Production Research*, 52(9), 2696–2728.
- Costa, A., Cappadonna, F. A., & Fichera, S. (2017). A hybrid genetic algorithm for minimizing makespan in a flow-shop sequence-dependent group scheduling problem. *Journal of Intelligent Manufacturing*, 28(6), 1269–1283.
- Dai, M., Tang, D., Giret, A., Salido, M. A., & Li, W. D. (2013). Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 29(5), 418–429.
- Delgado-Arana, E. M., Burtseva, L., Flores-Rios, B., Ibarra, R., & Werner, F. (2017). A batch sequencing model for a semiconductor packaging company. *Engineering Letters*, 25(2).
- Ding, J.-Y., Song, S., & Wu, C. (2016). Carbon-efficient scheduling of flow shops by multi-objective optimization. *European Journal of Operational Research*, 248(3), 758–771.
- Doğan, A., & Özgüner, F. (2005). Biobjective scheduling algorithms for execution time–reliability trade-off in heterogeneous computing systems. *The Computer Journal*, 48(3), 300–314.
- Dolgui, A., Gordon, V., & Strusevich, V. (2012). Single machine scheduling with precedence constraints and positionally dependent processing times. *Computers & Operations Research*, 39(6), 1218–1224.
- Eddaly, M., Jarboui, B., Bouabda, R., & Rebai, A. (2009). Hybrid estimation of distribution algorithm for permutation flowshop scheduling problem with sequence dependent family setup times. In *2009 international conference on computers & industrial engineering* (pp. 217–220).
- Eren, T., & Güner, E. (2007). Minimizing total tardiness in a scheduling problem with a learning effect. *Applied Mathematical Modelling*, 31(7), 1351–1361.

- Esquirol, P., Lopez, P., & Lopez, P. (s. d.). *L'ordonnancement*.
- Fan, K., Zhai, Y., Li, X., & Wang, M. (2018). Review and classification of hybrid shop scheduling. *Production Engineering*, *12*, 597–609.
- Fang, K., Uhan, N., Zhao, F., & Sutherland, J. W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, *30*(4), 234–240.
- Fang, K., Uhan, N. A., Zhao, F., & Sutherland, J. W. (2013). Flow shop scheduling with peak power consumption constraints. *Annals of Operations Research*, *206*(1), 115–145.
- Fasihi, M., Tavakkoli-Moghaddam, R., & Jolai, F. (2023). A bi-objective re-entrant permutation flow shop scheduling problem : minimizing the makespan and maximum tardiness. *Operational Research*, *23*(2), 29.
- Feng, H., Xi, L., Xiao, L., Xia, T., & Pan, E. (2018). Imperfect preventive maintenance optimization for flexible flowshop manufacturing cells considering sequence-dependent group scheduling. *Reliability Engineering & System Safety*, *176*, 218–229.
- Fichera, S., Costa, A., & Cappadonna, F. (2015). Scheduling jobs families with learning effect on the setup. *Advances in Operations Research*, *2015*.
- Figielska, E. (2018). Scheduling in a two-stage flowshop with parallel unrelated machines at each stage and shared resources. *Computers & Industrial Engineering*, *126*, 435–450.
- Fisher, M. L., Jaikumar, R., & Van Wassenhove, L. N. (1986). A multiplier adjustment method for the generalized assignment problem. *Management science*, *32*(9), 1095–1103.
- Framinan, J. M., Gupta, J. N., & Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society*, *55*, 1243–1255.
- França, P. M., Gupta, J. N., Mendes, A. S., Moscato, P., & Veltink, K. J. (2005). Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers & Industrial Engineering*, *48*(3), 491–506.
- Fu, Y., Ding, J., Wang, H., & Wang, J. (2018). Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in industry 4.0-based manufacturing system. *Applied Soft Computing*, *68*, 847–855.
- Fu, Y., Wang, H., Wang, J., & Pu, X. (2020). Multiobjective modeling and optimization for scheduling a stochastic hybrid flow shop with maximizing processing quality and minimizing total tardiness. *IEEE Systems Journal*, *15*(3), 4696–4707.
- Gantt, H. L. (1913). *Work, wages, and profits*. Engineering magazine.
- Garde, S. (2015). Connecting rod's manufacturing improvements and process planning. *Tolerance chain analysis and training material, KTH Royal Institute of Technology*,

Stockholm.

- Gelogullari, C. A., & Logendran, R. (2010). Group-scheduling problems in electronics manufacturing. *Journal of scheduling*, *13*, 177–202.
- Ghorbanzadeh, M., & Ranjbar, M. (2023). Energy-aware production scheduling in the flow shop environment under sequence-dependent setup times, group scheduling and renewable energy constraints. *European Journal of Operational Research*, *307*(2), 519–537.
- Giard, V. (2003). *Gestion de la production et des flux*. Economica Paris.
- Giglio, D. (2015). Optimal control strategies for single-machine family scheduling with sequence-dependent batch setup and controllable processing times. *Journal of Scheduling*, *18*, 525–543.
- Giret, A., Trentesaux, D., & Prabhu, V. (2015). Sustainability in manufacturing operations scheduling : A state of the art review. *Journal of Manufacturing Systems*, *37*, 126–140.
- Gombinski, J. (1967). Group technology-an introduction. *Production engineer*, *46*(9), 557–564.
- Gong, H., Tang, L., & Duin, C. (2010). A two-stage flow shop scheduling problem on a batching machine and a discrete machine with blocking and shared setup times. *Computers & Operations Research*, *37*(5), 960–969.
- Gotha. (1993). Les problèmes d’ordonnancement. *RAIRO-Operations Research*, *27*(1), 77–150.
- Grabowski, J., & Pempera, J. (2000). Sequencing of jobs in some production system. *European journal of operational research*, *125*(3), 535–550.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling : a survey. In *Annals of discrete mathematics* (Vol. 5, pp. 287–326). Elsevier.
- Greene, T. J., & Sadowski, R. P. (1984). A review of cellular manufacturing assumptions, advantages and design techniques. *Journal of Operations Management*, *4*(2), 85–97.
- Gupta, J. N., & Gupta, S. K. (1988). Single facility scheduling with nonlinear processing times. *Computers & Industrial Engineering*, *14*(4), 387–393.
- Gupta, J. N., & Schaller, J. E. (2006). Minimizing flow time in a flow-line manufacturing cell with family setup times. *Journal of the Operational Research Society*, *57*(2), 163–176.
- Gupta, J. N., & Stafford Jr, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, *169*(3), 699–711.
- Hajinejad, D., Salmasi, N., & Mokhtari, R. (2011). A fast hybrid particle swarm optimization algorithm for flow shop sequence dependent group scheduling problem. *Scientia Iranica*, *18*(3), 759–764.

- Hall, N. G., & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations research*, 44(3), 510–525.
- Ham, I., Hitomi, K., Yoshida, T., Ham, I., Hitomi, K., & Yoshida, T. (1985). Group technology and other related topics. *Group Technology : Applications to Production Management*, 171–186.
- Han, W., Deng, Q., Gong, G., Zhang, L., & Luo, Q. (2021). Multi-objective evolutionary algorithms with heuristic decoding for hybrid flow shop scheduling problem with worker constraint. *Expert Systems with Applications*, 168, 114282.
- Han, Y., Gong, D., Jin, Y., & Pan, Q.-k. (2016). Evolutionary multi-objective blocking lot-streaming flow shop scheduling with interval processing time. *Applied Soft Computing*, 42, 229–245.
- Hendizadeh, S. H., Faramarzi, H., Mansouri, S. A., Gupta, J. N., & ElMekkawy, T. Y. (2008). Meta-heuristics for scheduling a flowline manufacturing cell with sequence dependent family setup times. *International journal of production economics*, 111(2), 593–605.
- Hitomi, K. (2017). *Manufacturing systems engineering : a unified approach to manufacturing technology, production management and industrial economics*. Routledge.
- Hitomi, K., & Ham, I. (1976). Operations scheduling for group technology applications. *Annals of the CIRP*, 25(1), 419–422.
- Hnaïen, F., & Arbaoui, T. (2023). Minimizing the makespan for the two-machine flow shop scheduling problem with random breakdown. *Annals of Operations Research*, 1–24.
- Ho, K. I., Leung, J. Y., & Wei, W. (1993). Complexity of scheduling tasks with time-dependent execution times. *Information Processing Letters*, 48(6), 315–320.
- Ho, M. H., Hnaïen, F., & Dugardin, F. (2021). Electricity cost minimisation for optimal makespan solution in flow shop scheduling under time-of-use tariffs. *International journal of production research*, 59(4), 1041–1067.
- Hosseini, N., & Tavakkoli-Moghaddam, R. (2013). Two meta-heuristics for solving a new two-machine flowshop scheduling problem with the learning effect and dynamic arrivals. *The International Journal of Advanced Manufacturing Technology*, 65(5), 771–786.
- İnce, N., Deliktaş, D., & Hakan Selvi, İ. (2023). A comprehensive literature review of the flowshop group scheduling problems : systematic and bibliometric reviews. *International Journal of Production Research*, 1–30.
- İşler, M. C., Toklu, B., & Çelik, V. (2012). Scheduling in a two-machine flow-shop for earliness/tardiness under learning effect. *The International Journal of Advanced Manufacturing Technology*, 61(9), 1129–1137.
- Jain, A., & Jain, A. (2022). An approach for optimisation of flexible flow shop group

- scheduling with sequence dependent set-up time and preventive maintenance. *International Journal of Computer Aided Engineering and Technology*, 16(1), 40–66.
- Javel, G., & Le Bert, J. (1993). *L'organisation et la gestion de production* (Vol. 138). Masson.
- Ji, M., & Cheng, T. E. (2007). An fptas for scheduling jobs with piecewise linear decreasing processing times to minimize makespan. *Information Processing Letters*, 102(2-3), 41–47.
- Ji, M., He, Y., & Cheng, T. E. (2007). Single-machine scheduling with periodic maintenance to minimize makespan. *Computers & operations research*, 34(6), 1764–1770.
- Ji, M., Yao, D., Yang, Q., & Cheng, T. (2015). Machine scheduling with dejong's learning effect. *Computers & Industrial Engineering*, 80, 195–200.
- Jiang, E.-d., & Wang, L. (2019). An improved multi-objective evolutionary algorithm based on decomposition for energy-efficient permutation flow shop scheduling problem with sequence-dependent setup time. *International Journal of Production Research*, 57(6), 1756–1771.
- Jiang, S., Liu, M., Hao, J., & Qian, W. (2015). A bi-layer optimization approach for a hybrid flow shop scheduling problem involving controllable processing times in the steelmaking industry. *Computers & Industrial Engineering*, 87, 518–531.
- Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval research logistics quarterly*, 1(1), 61–68.
- Jung, S., Woo, Y.-B., & Kim, B. S. (2017). Two-stage assembly scheduling problem for processing products with dynamic component-sizes and a setup time. *Computers & industrial engineering*, 104, 98–113.
- Kar, A. (2019). Connecting rod manufacturing. *Report Number E341076-2018*.
- Keshavarz, T., & Salmasi, N. (2013). Makespan minimisation in flexible flowshop sequence-dependent group scheduling problem. *International Journal of Production Research*, 51(20), 6182–6193.
- Keshavarz, T., & Salmasi, N. (2014). Efficient upper and lower bounding methods for flowshop sequence-dependent group scheduling problems. *European Journal of Industrial Engineering*, 8(3), 366–387.
- Keshavarz, T., Salmasi, N., & Varmazyar, M. (2019). Flowshop sequence-dependent group scheduling with minimisation of weighted earliness and tardiness. *European Journal of Industrial Engineering*, 13(1), 54–80.
- Khamseh, A., Jolai, F., & Babaei, M. (2015). Integrating sequence-dependent group scheduling problem and preventive maintenance in flexible flow shops. *The International Journal of Advanced Manufacturing Technology*, 77(1), 173–185.
- Khatami, M., Salehipour, A., & Hwang, F. (2019). Makespan minimization for the machine ordered flow shop scheduling problem. *Computers & Operations Research*,

- 111, 400–414.
- Kheirandish, O., Tavakkoli-Moghaddam, R., & Karimi-Nasab, M. (2015). An artificial bee colony algorithm for a two-stage hybrid flowshop scheduling problem with multilevel product structures and requirement operations. *International Journal of Computer Integrated Manufacturing*, 28(5), 437–450.
- Koh, S.-G., et al. (2013). *Heuristics for non-identical parallel machine scheduling with sequence-dependent setup times* (Thèse de doctorat non publiée). .
- Kong, M., Liu, X., Pei, J., Zhou, Z., & Pardalos, P. M. (2020). Parallel-batching scheduling of deteriorating jobs with non-identical sizes and rejection on a single machine. *Optimization Letters*, 14, 857–871.
- Koulamas, C., & Kyparisis, G. J. (2007). Single-machine and two-machine flowshop scheduling with general learning functions. *European Journal of Operational Research*, 178(2), 402–407.
- Kuo, W.-H., Hsu, C.-J., & Yang, D.-L. (2012). Worst-case and numerical analysis of heuristic algorithms for flowshop scheduling problems with a time-dependent learning effect. *Information Sciences*, 184(1), 282–297.
- Kuo, W.-H., & Yang, D.-L. (2006). Single-machine group scheduling with a time-dependent learning effect. *Computers & Operations Research*, 33(8), 2099–2112.
- Lai, K., Hsu, P.-H., Ting, P.-H., & Wu, C.-C. (2014). A truncated sum of processing-times-based learning model for a two-machine flowshop scheduling problem. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 24(2), 152–160.
- Lamy, D. (2017). *Université clermont auvergne* (Thèse de doctorat non publiée). Université de Valenciennes et du Hainaut-Cambrésis.
- Lee, C.-Y. (1999). Two-machine flowshop scheduling with availability constraints. *European Journal of Operational Research*, 114(2), 420–429.
- Lenstra, J. K., Shmoys, D. B., & Tardos, É. (1990). Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46, 259–271.
- Li, G., Wang, X.-Y., Wang, J.-B., & Sun, L.-Y. (2013). Worst case analysis of flow shop scheduling problems with a time-dependent learning effect. *International Journal of Production Economics*, 142(1), 98–104.
- Li, J.-q., Sang, H.-y., Han, Y.-y., Wang, C.-g., & Gao, K.-z. (2018). Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *Journal of Cleaner Production*, 181, 584–598.
- Li, M.-H., Lv, D.-Y., Zhang, L.-H., & Wang, J.-B. (2024). Permutation flow shop scheduling with makespan objective and truncated learning effects. *Journal of Applied Mathematics and Computing*, 1–33.
- Li, P., Xue, Q., Zhang, Z., Chen, J., & Zhou, D. (2023). Multi-objective energy-efficient hybrid flow shop scheduling using q-learning and gvns driven nsga-ii. *Computers &*

- Operations Research*, 159, 106360.
- Li, S. (1997). A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. *European Journal of Operational Research*, 102(1), 142–156.
- Li, Y., Li, X., Gao, L., & Meng, L. (2020). An improved artificial bee colony algorithm for distributed heterogeneous hybrid flowshop scheduling problem with sequence-dependent setup times. *Computers & Industrial Engineering*, 147, 106638.
- Lin, C.-C., Liu, W.-Y., & Chen, Y.-H. (2020). Considering stockers in reentrant hybrid flow shop scheduling with limited buffer capacity. *Computers & Industrial Engineering*, 139, 106154.
- Lin, H.-T., & Liao, C.-J. (2003). A case study in a two-stage hybrid flow shop with setup time and dedicated machines. *International Journal of Production Economics*, 86(2), 133–143.
- Lin, S.-W., Gupta, J. N., Ying, K.-C., & Lee, Z.-J. (2009). Using simulated annealing to schedule a flowshop manufacturing cell with sequence-dependent family setup times. *International Journal of Production Research*, 47(12), 3205–3217.
- Lin, S.-W., & Ying, K.-C. (2012). Scheduling a bi-criteria flowshop manufacturing cell with sequence-dependent family setup times. *European journal of industrial engineering*, 6(4), 474–496.
- Lin, S.-W., Ying, K.-C., Lu, C.-C., & Gupta, J. N. (2011). Applying multi-start simulated annealing to schedule a flowline manufacturing cell with sequence dependent family setup times. *International Journal of Production Economics*, 130(2), 246–254.
- Lin, W.-C., Wu, C.-C., Yu, K., Zhuang, Y.-H., & Liu, S.-C. (2017). On the use of genetic algorithm for solving re-entrant flowshop scheduling with sum-of-processing-times-based learning effect to minimize total tardiness. *Intelligent Automation & Soft Computing*, 1–11.
- Lin, Y.-K., & Hsieh, F.-Y. (2014). Unrelated parallel machine scheduling with setup times and ready times. *International Journal of Production Research*, 52(4), 1200–1214.
- Liou, C.-D., & Hsieh, Y.-C. (2015). A hybrid algorithm for the multi-stage flow shop group scheduling with sequence-dependent setup and transportation times. *International Journal of Production Economics*, 170, 258–267.
- Liu, C.-H., & Tsai, W.-N. (2016). Multi-objective parallel machine scheduling problems by considering controllable processing times. *Journal of the Operational Research Society*, 67(4), 654–663.
- Liu, G.-S., Zhou, Y., & Yang, H.-D. (2017). Minimizing energy consumption and tardiness penalty for fuzzy flow shop scheduling with state-dependent setup time. *Journal of cleaner production*, 147, 470–484.
- Liu, L., He, H., & Shi, L. (2019). A new solution approach for flow shop scheduling with an exponential time-dependent learning effect. In *2019 IEEE 15th International*

- conference on automation science and engineering (case)* (pp. 468–473).
- Liu, L., Shi, Z., & Shi, L. (2018). Minimization of total energy consumption in an m-machine flow shop with an exponential time-dependent learning effect. *Frontiers of Engineering Management*, 5(4), 487–498.
- Liu, S.-C. (2020). A two-stage three-machine flow shop assembly problem mixed with a controllable number and sum-of-processing times-based learning effect by simulated annealing algorithms. *Discrete Dynamics in Nature and Society*, 2020(1), 4085718.
- Liu, S. Q., & Kozan, E. (2009). Scheduling a flow shop with combined buffer conditions. *International Journal of Production Economics*, 117(2), 371–380.
- Liu, Y., Yuen, C., Hassan, N. U., Huang, S., Yu, R., & Xie, S. (2014). Electricity cost minimization for a microgrid with distributed energy resource under different information availability. *IEEE Transactions on Industrial Electronics*, 62(4), 2571–2583.
- Logendran, R., Mai, L., & Talkington, D. (1995). Combined heuristics for bi-level group scheduling problems. *International Journal of Production Economics*, 38(2-3), 133–145.
- Low, C., & Lin, W.-Y. (2013). Some scheduling problems with time-dependent learning effect and deteriorating jobs. *Applied Mathematical Modelling*, 37(20-21), 8865–8875.
- Lu, C., Gao, L., Li, X., Pan, Q., & Wang, Q. (2017). Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *Journal of Cleaner Production*, 144, 228–238.
- Lu, C., Li, X., Gao, L., Liao, W., & Yi, J. (2017). An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times. *Computers & Industrial Engineering*, 104, 156–174.
- Lu, S., Liu, X., Pei, J., & Pardalos, P. M. (2021). Permutation flowshop manufacturing cell scheduling problems with deteriorating jobs and sequence dependent setup times under dominant machines. *Optimization Letters*, 15(2), 537–551.
- Lu, Y.-Y. (2016). Research on no-idle permutation flowshop scheduling with time-dependent learning effect and deteriorating jobs. *Applied Mathematical Modelling*, 40(4), 3447–3450.
- Luo, K. (2015). A scheduling model with a more general function of learning effects. *Computers & Industrial Engineering*, 82, 159–166.
- Mansouri, S. A., Aktas, E., & Besikci, U. (2016). Green scheduling of a two-machine flowshop : Trade-off between makespan and energy consumption. *European Journal of Operational Research*, 248(3), 772–788.
- Martinez, S., Dauzère-Pérès, S., Gueret, C., Mati, Y., & Sauer, N. (2006). Complexity of flowshop scheduling problems with a new blocking constraint. *European Journal of*

- Operational Research*, 169(3), 855–864.
- Mattsson, S.-A., & Jonsson, P. (2003). *Produktionslogistik*. Studentlitteratur AB.
- Meng, L., Zhang, C., Shao, X., Zhang, B., Ren, Y., & Lin, W. (2020). More milp models for hybrid flow shop scheduling problem and its extended problems. *International journal of production research*, 58(13), 3905–3930.
- Mitrofanov, S. P., Harris, E., & Grayson, T. (1966). Scientific principles of group technology (nauchnye osnovy gruppovoi tekhnologii). (*No Title*).
- Miyata, H. H., & Nagano, M. S. (2019). The blocking flow shop scheduling problem : A comprehensive and conceptual review. *Expert Systems with Applications*, 137, 130–156.
- Moccellin, J. V., Nagano, M. S., Pitombeira Neto, A. R., & de Athayde Prata, B. (2018). Heuristic algorithms for scheduling hybrid flow shops with machine blocking and setup times. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 40, 1–11.
- Mollaei, A., Mohammadi, M., & Naderi, B. (2019). A bi-objective milp model for blocking hybrid flexible flow shop scheduling problem : robust possibilistic programming approach. *International Journal of Management Science and Engineering Management*, 14(2), 137–146.
- Mor, B., & Mosheiov, G. (2014). Batch scheduling of identical jobs with controllable processing times. *Computers & Operations Research*, 41, 115–124.
- Mosheiov, G., & Sidney, J. B. (2003). Scheduling with general job-dependent learning curves. *European Journal of Operational Research*, 147(3), 665–670.
- Mousavi, S. M., Mahdavi, I., Rezaeian, J., & Zandieh, M. (2018). An efficient bi-objective algorithm to solve re-entrant hybrid flow shop scheduling with learning effect and setup times. *Operational Research*, 18, 123–158.
- Mouzon, G., Yildirim, M. B., & Twomey, J. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of production research*, 45(18-19), 4247–4271.
- Naderi, B., & Salmasi, N. (2012). Permutation flowshops in group scheduling with sequence-dependent setup times. *European Journal of Industrial Engineering*, 6(2), 177–198.
- Nawaz, M., Ensore Jr, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1), 91–95.
- Neufeld, J. S., Gupta, J. N., & Buscher, U. (2016). A comprehensive review of flowshop group scheduling literature. *Computers & Operations Research*, 70, 56–74.
- Neufeld, J. S., Schulz, S., & Buscher, U. (2023). A systematic review of multi-objective hybrid flow shop scheduling. *European Journal of Operational Research*, 309(1), 1–23.

- Nikjo, B., & Rezaeian, J. (2014). Meta heuristic for minimizing makespan in a flow-line manufacturing cell with sequence dependent family setup times. *Journal of Optimization in Industrial Engineering*, 7(16), 21–29.
- Nikzad, F., Rezaeian, J., Mahdavi, I., & Rastgar, I. (2015). Scheduling of multi-component products in a two-stage flexible flow shop. *Applied Soft Computing*, 32, 132–143.
- Niu, W., & Li, J.-q. (2022). A two-stage cooperative evolutionary algorithm for energy-efficient distributed group blocking flow shop with setup carryover in precast systems. *Knowledge-Based Systems*, 257, 109890.
- Pan, Q.-K., Gao, L., & Wang, L. (2020). An effective cooperative co-evolutionary algorithm for distributed flowshop group scheduling problems. *IEEE Transactions on Cybernetics*, 52(7), 5999–6012.
- Panwalker, S., Dudek, R., & Smith, M. (1973). *Theory of scheduling and its application*. Springer-Verlag : New York.
- Paredes-Astudillo, Y. A., Botta-Genoulaz, V., & Montoya-Torres, J. R. (2022). Comparing linear and non-linear modelling approaches of learning effects in 2-stage flow-shop scheduling problems. *IFAC-PapersOnLine*, 55(10), 842–847.
- Pargar, F., & Zandieh, M. (2012). Bi-criteria sdst hybrid flow shop scheduling with learning effect of setup times : water flow-like algorithm approach. *International Journal of Production Research*, 50(10), 2609–2623.
- Pargar, F., Zandieh, M., Kauppila, O., & Kujala, J. (2018). The effect of worker learning on scheduling jobs in a hybrid flow shop : A bi-objective approach. *Journal of Systems Science and Systems Engineering*, 27(3), 265–291.
- Pei, J., Liu, X., Fan, W., Pardalos, P. M., Migdalas, A., & Yang, S. (2016). Scheduling jobs on a single serial-batching machine with dynamic job arrivals and multiple job types. *Annals of Mathematics and Artificial Intelligence*, 76, 215–228.
- Peña Tibaduiza, E., Garavito Hernández, E. A., Perez Figueredo, L. E., & Moratto Chimenty, E. (2017). Literature review on the hybrid flow shop scheduling problem with unrelated parallel machines. *Ingeniería*, 22(1), 9–22.
- Pinedo, M. (2005). *Planning and scheduling in manufacturing and services*. Springer.
- Pinedo, M., & Chao, X. (1999). Operations scheduling with applications in manufacturing and services. (*No Title*).
- Pinedo, M. L. (2012). *Scheduling* (Vol. 29). Springer.
- Pinedo, M. L., & Pinedo, M. L. (2016). Design and implementation of scheduling systems : More advanced concepts. *Scheduling : Theory, Algorithms, and Systems*, 485–508.
- Potts, C. N., & Kovalyov, M. Y. (2000). Scheduling with batching : A review. *European journal of operational research*, 120(2), 228–249.
- Prasad, S. D., Rajendran, C., & Chetty, O. K. (2006). A genetic algorithmic approach to multi-objective scheduling in a kanban-controlled flowshop with intermediate buffer

- and transport constraints. *The International Journal of Advanced Manufacturing Technology*, 29, 564–576.
- Qin, H., Han, Y., Wang, Y., Liu, Y., Li, J., & Pan, Q. (2022). Intelligent optimization under blocking constraints : A novel iterated greedy algorithm for the hybrid flow shop group scheduling problem. *Knowledge-Based Systems*, 258, 109962.
- Qin, W., Zhuang, Z., Liu, Y., & Tang, O. (2019). A two-stage ant colony algorithm for hybrid flow shop scheduling with lot sizing and calendar constraints in printed circuit board assembly. *Computers & Industrial Engineering*, 138, 106115.
- Rachaniotis, N. P., & Pappis, C. P. (2006). Scheduling fire-fighting tasks using the concept of "deteriorating jobs". *Canadian Journal of Forest Research*, 36(3), 652–658.
- Ribas, I., Companys, R., & Tort-Martorell, X. (2015). An efficient discrete artificial bee colony algorithm for the blocking flow shop problem with total flowtime minimization. *Expert Systems with Applications*, 42(15-16), 6155–6167.
- Rustogi, K., & Strusevich, V. A. (2012a). Simple matching vs linear assignment in scheduling models with positional effects : A critical review. *European Journal of Operational Research*, 222(3), 393–407.
- Rustogi, K., & Strusevich, V. A. (2012b). Single machine scheduling with general positional deterioration and rate-modifying maintenance. *Omega*, 40(6), 791–804.
- Rustogi, K., & Strusevich, V. A. (2015). Single machine scheduling with time-dependent linear deterioration and rate-modifying maintenance. *Journal of the Operational Research Society*, 66(3), 500–515.
- Safari, E., & Sadjadi, S. J. (2011). A hybrid method for flowshops scheduling with condition-based maintenance constraint and machines breakdown. *Expert Systems with Applications*, 38(3), 2020–2029.
- Salmasi, N., Logendran, R., & Skandari, M. R. (2010). Total flow time minimization in a flowshop sequence-dependent group scheduling problem. *Computers & Operations Research*, 37(1), 199–212.
- Sánchez-Herrera, S., Montoya-Torres, J. R., & Solano-Charris, E. L. (2019). Flow shop scheduling problem with position-dependent processing times. *Computers & Operations Research*, 111, 325–345.
- Schaller, J. E., Gupta, J. N., & Vakharia, A. J. (2000). Scheduling a flowline manufacturing cell with sequence dependent family setup times. *European Journal of Operational Research*, 125(2), 324–339.
- Sekkal, D. N., & Belkaid, F. (2020). Optimization of production system with environmental consideration : the effect of worker learning in a metallurgy factory. In *2020 IEEE 13th international colloquium of logistics and supply chain management (logistica)* (pp. 1–6).
- Sekkal, D. N., & Belkaid, F. (2022). A bi-objective optimization for flow shop group

- scheduling problem with sequence dependent setup time. In *2022 14th international colloquium of logistics and supply chain management (logistiqua)* (pp. 1–7).
- Sekkal, D. N., & Belkaid, F. (2023). A multi-objective optimization algorithm for flow shop group scheduling problem with sequence dependent setup time and worker learning. *Expert Systems with Applications*, 120878.
- Sekkal, N., & Belkaid, F. (2020). A multi-objective simulated annealing to solve an identical parallel machine scheduling problem with deterioration effect and resources consumption constraints. *Journal of Combinatorial Optimization*, 40(3), 660–696.
- Senyigit, E., & Atici, U. (2018). Scheduling with job dependent learning effect and ergonomic risk deterioration. In *2018 2nd international symposium on multidisciplinary studies and innovative technologies (ismsit)* (pp. 1–4).
- Shabtay, D., & Steiner, G. (2007). A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, 155(13), 1643–1666.
- Shahzad, A. M. (2011). *Une approche hybride de simulation-optimisation basée sur la fouille de données pour les problèmes d'ordonnancement* (Thèse de doctorat non publiée). Université de Nantes.
- Shao, W., Pi, D., & Shao, Z. (2017). Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms. *Knowledge-Based Systems*, 137, 163–181.
- Sharma, M., Sharma, M., & Sharma, S. (2022). Desert sparrow optimization algorithm for the bicriteria flow shop scheduling problem with sequence-independent setup time. *Operational Research*, 22(4), 4353–4396.
- Shingo, S. (2019). *A revolution in manufacturing : the smed system*. Routledge.
- Shokoufi, K., Rezaeian, J., Shirazi, B., & Mahdavi, I. (2019). Preemptive just-in-time scheduling problem on uniform parallel machines with time-dependent learning effect and release dates. *International Journal of Operational Research*, 34(3), 339–368.
- Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., & Ortega-Mier, M. (2014). Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *Journal of Cleaner Production*, 67, 197–207.
- Solaja, V., & Ursoevic, S. (1969). Optimization of group technology lines by methods developed in the institute for machine tools and tooling in belgrad. In *Group technology conference, turin, italy*.
- Solimanpur, M., & Elmi, A. (2011). A tabu search approach for group scheduling in buffer-constrained flow shop cells. *International Journal of Computer Integrated Manufacturing*, 24(3), 257–268.
- Song, H., Yi, S., & Shen, H. (2016). A study of job shop standard setup time quota. *Production Engineering*, 10, 185–196.
- Sridhar, J., & Rajendran, C. (1994). A genetic algorithm for family and job scheduling in

- a flowline-based manufacturing cell. *Computers & Industrial Engineering*, 27(1-4), 469–472.
- Sule, D. R. (2007). *Production planning and industrial scheduling : examples, case studies and applications*. CRC press.
- Sun, L., Ning, L., & Huo, J.-z. (2020). Group scheduling problems with time-dependent and position-dependent dejong's learning effect. *Mathematical Problems in Engineering*, 2020(1), 5161872.
- Sun, X., Liu, T., Geng, X.-N., Hu, Y., & Xu, J.-X. (2023). Optimization of scheduling problems with deterioration effects and an optional maintenance activity. *Journal of Scheduling*, 26(3), 251–266.
- Tahraoui, N., Sari-Triqui, L., & Bennkrouf, M. (2022). A bi-objective optimization approach based on lp-metric method in broiler production network : a case study. In *E3s web of conferences* (Vol. 336, p. 00025).
- Taillard, E. (1993). Benchmarks for basic scheduling problems. *European journal of operational research*, 64(2), 278–285.
- Talbi, E.-G. (2009). *Metaheuristics : from design to implementation*. John Wiley & Sons.
- Tigane, M., Dahane, M., & Boudhar, M. (2019). Multiobjective approach for deteriorating jobs scheduling for a sustainable manufacturing system. *The International Journal of Advanced Manufacturing Technology*, 101(5), 1939–1957.
- Tosun, Ö., Marichelvam, M. K., & Tosun, N. (2020). A literature review on hybrid flow shop scheduling. *International Journal of Advanced Operations Management*, 12(2), 156–194.
- Trabelsi, W., Sauvey, C., & Sauer, N. (2012). Heuristics and metaheuristics for mixed blocking constraints flowshop scheduling problems. *Computers & Operations Research*, 39(11), 2520–2527.
- Ulungu, E. L., Teghem, J., Fortemps, P., & Tuyttens, D. (1999). Mosa method : a tool for solving multiobjective combinatorial optimization problems. *Journal of multicriteria decision analysis*, 8(4), 221.
- Umam, M. S., Mustafid, M., & Suryono, S. (2022). A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *Journal of King Saud University-Computer and Information Sciences*, 34(9), 7459–7467.
- Vahedi Nouri, B., Fattahi, P., & Ramezani, R. (2013). Hybrid firefly-simulated annealing algorithm for the flow shop problem with learning effects and flexible maintenance activities. *International Journal of Production Research*, 51(12), 3501–3515.
- Vahedi-Nouri, B., Fattahi, P., & Ramezani, R. (2013). Minimizing total flow time for the non-permutation flow shop scheduling problem with learning effects and availability constraints. *Journal of Manufacturing Systems*, 32(1), 167–173.
- Vakharia, A. J., & Chang, Y.-L. (1990). A simulated annealing approach to scheduling a

- manufacturing cell. *Naval Research Logistics (NRL)*, 37(4), 559–577.
- Van Der Zee, D.-J., Gaalman, G. J., & Nomden, G. (2011). Family based dispatching in manufacturing networks. *International journal of production research*, 49(23), 7059–7084.
- Vickson, R. (1980a). Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine. *Operations Research*, 28(5), 1155–1167.
- Vickson, R. (1980b). Two single machine sequencing problems involving controllable job processing times. *AIIE transactions*, 12(3), 258–262.
- Wang, J.-B. (2006). A note on scheduling problems with learning effect and deteriorating jobs. *International Journal of Systems Science*, 37(12), 827–833.
- Wang, J.-B., Ng, C. T., Cheng, T., & Liu, L. (2008). Single-machine scheduling with a time-dependent learning effect. *International Journal of Production Economics*, 111(2), 802–811.
- Wang, Y., Han, Y., Wang, Y., Li, J., Gao, K., & Liu, Y. (2023). An effective two-stage iterated greedy algorithm for distributed flowshop group scheduling problem with setup time. *Expert Systems with Applications*, 233, 120909.
- Wang, Y.-C., & Wang, J.-B. (2023). Study on convex resource allocation scheduling with a time-dependent learning effect. *Mathematics*, 11(14), 3179.
- Wang, Z.-Y., Pan, Q.-K., Gao, L., & Wang, Y.-L. (2022). An effective two-stage iterated greedy algorithm to minimize total tardiness for the distributed flowshop group scheduling problem. *Swarm and Evolutionary Computation*, 74, 101143.
- Wang, Z.-Y., Yu, C.-M., Pan, Q.-K., & Li, Y.-Z. (2021). An iterated greedy algorithm for distributed flowshop group scheduling problem with total tardiness criterion. In *2021 40th chinese control conference (ccc)* (pp. 2009–2014).
- Wilson, A., King, R., & Hodgson, T. (2004). Scheduling non-similar groups on a flow line : multiple group setups. *Robotics and Computer-Integrated Manufacturing*, 20(6), 505–515.
- Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of the aeronautical sciences*, 3(4), 122–128.
- Wu, C.-C., Chen, J.-Y., Lin, W.-C., Lai, K., Liu, S.-C., & Yu, P.-W. (2018). A two-stage three-machine assembly flow shop scheduling with learning consideration to minimize the flowtime by six hybrids of particle swarm optimization. *Swarm and evolutionary computation*, 41, 97–110.
- Wu, C.-C., Shiau, Y.-R., & Lee, W.-C. (2008). Single-machine group scheduling problems with deterioration consideration. *Computers & Operations Research*, 35(5), 1652–1659.
- Wu, C.-C., Zhang, X., Azzouz, A., Shen, W.-L., Cheng, S.-R., Hsu, P.-H., & Lin, W.-C.

- (2021). Metaheuristics for two-stage flow-shop assembly problem with a truncation learning function. *Engineering optimization*, 53(5), 843–866.
- Xin, X., Jiang, Q., Li, C., Li, S., & Chen, K. (2023). Permutation flow shop energy-efficient scheduling with a position-based learning effect. *International Journal of Production Research*, 61(2), 382–409.
- Yang, D.-L., Cheng, T., Yang, S.-J., & Hsu, C.-J. (2012). Unrelated parallel-machine scheduling with aging effects and multi-maintenance activities. *Computers & Operations Research*, 39(7), 1458–1464.
- Yang, D.-L., Kuo, W.-H., & Chern, M.-S. (2008). Multi-family scheduling in a two-machine reentrant flow shop with setups. *European Journal of Operational Research*, 187(3), 1160–1170.
- Yang, S.-J., & Yang, D.-L. (2010). Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities. *Omega*, 38(6), 528–533.
- Yang, S.-J., & Yang, D.-L. (2013). Note on “a unique integer mathematical model for scheduling deteriorating jobs with rate-modifying activities on a single machine”. *The International Journal of Advanced Manufacturing Technology*, 64, 1759–1764.
- Yin, Y., Cheng, T., Wu, C.-C., et al. (2014). *Scheduling with time-dependent processing times* (Vol. 2014). Hindawi.
- Yin, Y., Wu, W.-H., Cheng, T., & Wu, C.-C. (2014). Due-date assignment and single-machine scheduling with generalised position-dependent deteriorating jobs and deteriorating multi-maintenance activities. *International Journal of Production Research*, 52(8), 2311–2326.
- Ying, K.-C., Gupta, J. N., Lin, S.-W., & Lee, Z.-J. (2010). Permutation and non-permutation schedules for the flowline manufacturing cell with sequence dependent family setups. *International Journal of Production Research*, 48(8), 2169–2184.
- Yuan, S., Li, T., & Wang, B. (2020). A co-evolutionary genetic algorithm for the two-machine flow shop group scheduling problem with job-related blocking and transportation times. *Expert Systems with Applications*, 152, 113360.
- Zahmani, M. H. (2018). *Contribution à l’ordonnancement dynamique : proposition d’une approche guidée par effet de simulation/datamining* (Thèse de doctorat non publiée). Université d’Oran.
- Zeddani, B., Belkaid, F., & Bennekrouf, M. (2020). An efficient approach for solving integrated production and distribution planning problems : Cost vs. energy. *International Journal of Applied Logistics (IJAL)*, 10(2), 25–44.
- Zhang, Y. (2013). Cellular manufacturing systems. *Methods in Product Design : New Strategies in Reengineering*, 1027.
- Zhang, Y., Lu, Y., & Zhang, C. (2018). Resource-reconfigured flow shop scheduling and lot-sizing problems in semiconductor test. In *2018 5th international conference on*

- industrial engineering and applications (iciea)* (pp. 326–332).
- Zhang, Z., Tang, Q., & Chica, M. (2021). Maintenance costs and makespan minimization for assembly permutation flow shop scheduling by considering preventive and corrective maintenance. *Journal of Manufacturing Systems*, 59, 549–564.
- Zhao, H., Pan, Q.-K., & Gao, K.-Z. (2023). A cooperative population-based iterated greedy algorithm for distributed permutation flowshop group scheduling problem. *Engineering Applications of Artificial Intelligence*, 125, 106750.
- Zolfaghari, S., & Liang, M. (1999). Jointly solving the group scheduling and machining speed selection problems : a hybrid tabu search and simulated annealing approach. *International Journal of Production Research*, 37(10), 2377–2397.

ملخص

يلعب تحسين الجدولة دوراً محورياً في إدارة أنظمة الإنتاج بكفاءة، مما يؤثر بشكل مباشر على الكفاءة التشغيلية للشركات. وبالتالي، هناك طلب حتمي على اتباع نهج متكامل يراعي بدقة جميع الخصائص والقيود المتأصلة في المشكلة. تشمل العناصر الجديرة بالملاحظة خبرة المشغلين، وتوليد تأثيرات التعلم، وتجميع المهام لتقليل أوقات التجهيز، والقيود اللوجستية المرتبطة بأوقات النقل التي لا يمكن تجنبها بين الآلات. تتعمق أطروحتنا في هذه المتغيرات المتنوعة ضمن مشكلة جدولة متجر التدفق، بهدف تقليل كل من فترة التصنيع واستهلاك الطاقة في النظام. ولتحقيق ذلك، قمنا بصياغة العديد من النماذج الرياضية الخطية المختلطة ذات الأعداد الصفرية المختلطة، واستيعاب جميع السمات ذات الصلة بالدراسة تدريجياً. بالإضافة إلى ذلك، قدمنا حدين أدنى وأسلوباً فائقاً متعدد الأهداف يعتمد على محاكاة التلدين بمحاكاة شصضا للتعامل مع التعقيد الفطري للمشكلة. وتثبت نتائج التجارب المختلفة فعالية طرق الحل المتقدمة هذه، حيث تظهر مهارتها في إدارة جميع الخصائص المحددة في الدراسة.

الكلمات المفتاحية: الجدولة، متجر التدفق، شصضا، الحدود الأدنى، تأثير التعلم، النقل

Abstract

The scheduling plays a pivotal role in efficiently managing production systems, directly impacting the operational efficiency of companies. Consequently, there is an imperative need for an integrated approach that meticulously considers all inherent characteristics and constraints of the problem. Noteworthy elements include the experience of operators, which generates learning effects, task grouping to reduce set-up times, and logistical constraints due to unavoidable transport times between machines. Our thesis explores these diverse variables within a flow shop scheduling problem, aiming to minimize both the makespan and energy consumption of the system. To achieve this, we have formulated several mixed-integer linear mathematical models, progressively incorporating all relevant features of the study. Additionally, we have introduced two lower bounds and a multi-objective simulated annealing algorithm MOSA to address the problem's inherent complexity. The outcomes from various experiments demonstrate the effectiveness of these advanced solution methods, showcasing their adeptness in managing all the characteristics identified in the study.

Keywords: Scheduling, flow shop, MOSA, lower bounds, learning effect, transport.

Résumé

L'ordonnancement joue un rôle crucial dans la gestion des systèmes de production, influençant directement l'efficacité opérationnelle des entreprises. Ainsi, la nécessité impérieuse d'une approche intégrée, prenant en compte l'ensemble des caractéristiques et contraintes du problème, se fait ressentir. Parmi les éléments à prendre en considération, on retrouve l'expérience des opérateurs générant des effets d'apprentissage, le regroupement des tâches visant à réduire les temps de configuration, ainsi que les contraintes logistiques liées aux temps de transport entre les machines, lesquelles se présentent comme incontournables. Notre thèse aborde ces diverses variantes dans un problème d'ordonnancement flow shop visant à minimiser le makespan et la consommation d'énergie du système. Nous avons développé plusieurs modèles mathématiques linéaires en nombres entiers mixtes, progressivement intégrant toutes les caractéristiques de l'étude. En outre, nous avons introduit deux bornes inférieures et une métaheuristique multi-objectif basée sur le recuit simulé MOSA pour faire face à la complexité inhérente du problème. Les résultats obtenus suite à diverses expérimentations attestent de l'efficacité des méthodes de résolution avancées, capables de gérer toutes les caractéristiques de l'étude.

Mots Clés : Ordonnancement, flow shop, MOSA, bornes inférieures, effet d'apprentissage, transport.