TABLE DES MATIÈRES

TABLE DES FIGURES

Liste des tableaux

INTRODUCTION GÉNÉRALE

Aujourd'hui, la situation du marché est caractérisée par une demande fluctuante, l'introduction fréquente de nouveaux produits, des technologies en évolution rapide et des réglementations gouvernementales strictes. Dans un tel environnement, les industriels ressentent de plus en plus le besoin d'intégrer la flexibilité et la reconfigurabilité dans leur système de production afin d'être en mesure de s'adapter rapidement et efficacement face à ces changements. En effet, les lignes de production dédiées, conçues pour ne fabriquer qu'un seul type de produit en grande quantité, sont rigides et ne peuvent donc pas évoluer dans le temps. Pour remédier à ce problème, des systèmes de production flexibles (FMS) ont été proposés comme alternative. Ces systèmes se composent principalement de machines à commande numérique qui offrent une flexibilité généralisée permettant la production d'une grande variété de produits et donc de s'adapter facilement aux changements et fluctuation. Cela dit, la faible productivité de ces systèmes et les coûts d'investissement élevés qu'ils entraînent font qu'ils ne se sont finalement pas révélés être une bonne solution pour de nombreux secteurs industriels. C'est ainsi que les systèmes de production reconfigurables (RMS) ont été introduits. Ces derniers visent à offrir un équilibre entre la capacité de production et la flexibilité. Contrairement aux systèmes flexibles, les RMS sont conçus avec une flexibilité personnalisée autour d'une famille de produits prédéfinie. La structure globale de ces systèmes est composée de machines reconfigurables qui peuvent être facilement ajoutées, retirées ou reconfigurées (en modifiant leur matériel ou en reprogrammant leur logiciel) afin de répondre aux demandes du marché ou de faire face à des changements externes.

Depuis son apparition, le concept de RMS fait l'objet d'un intérêt croissant de la part des industriels et des chercheurs. La littérature montre que les RMS sont en plein développement et que de nombreuses études sur les différents niveaux décisionnels sont menées. Cependant, plusieurs problèmes restent peu explorés quant à la conception et au pilotage de ces systèmes. Ainsi, c'est dans ce contexte que cette thèse est réalisée. Plus particulièrement, le but est de développer des approches d'aide à la décision pour la mise en place et la configuration dynamique des systèmes de production reconfigurables. L'un des avantages d'un RMS est le fait qu'il soit conçu pour être facilement reconfigurable, ce qui lui permet de fabriquer plusieurs produits et de s'adapter rapidement aux changements. Ceci étant dit, nous avons constaté que la notion de « reconfigurabilité » reste ambiguë dans la littérature des RMS. Même si la reconfiguration apporte beaucoup de flexibilité au système, celle-ci nécessite du temps, des ressources et des coûts supplémentaires. Par conséquent, il devient important de prendre ces aspects en considération

lors de la conception d'un RMS et de les optimiser de telle sorte que la reconfiguration se passe de manière rapide, efficace et à moindre coût.

Nous considérons dans ce mémoire la conception de lignes de production reconfigurables multi-produits. Ces lignes sont modulaires et peuvent fabriquer différents produits appartenant à la même famille. Pour chaque produit, une configuration adéquate de la ligne est nécessaire. Ainsi, passer de la production d'un produit vers un autre nécessite une reconfiguration de la ligne. Typiquement, une reconfiguration consiste à déplacer, ajouter ou retirer certains modules pour répondre aux nouvelles exigences de production. Par exemple, dans les centres d'usinage de pièces mécaniques, composés de machines-outils à haute précision, une reconfiguration consiste à relocaliser certains outils (ou modules) entre les différentes machines (reconfiguration liée à la partie matérielle) et à reprogrammer les ordinateurs de contrôle de ces dernières (reconfiguration liée à la partie logicielle). Nous nous limiterons dans le cadre de cette thèse uniquement aux aspects matériels. Par conséquent, l'objectif principal de ce mémoire est de présenter des techniques d'optimisation qui visent à concevoir pour chaque produit une configuration adéquate tout en optimisant le nombre de modules et le processus de reconfiguration lors du passage d'une configuration vers une autre.

Ce problème est similaire au problème d'équilibrage de lignes d'assemblage multi-modèles ou MuMALBP (*Multi-Model Assembly Line Balancing Problem*, en anglais) dans lequel différents produits sont fabriqués séparément par lots et la ligne doit être reconfigurée en conséquence. Malgré son importance grandissante dans l'industrie, ce problème est peu traité dans la littérature. Ceci est dû au fait que le MuMALBP est généralement réduit au problème d'équilibrage de lignes d'assemblage à un seul produit, qui est mieux connu dans la littérature. Cette réduction permet d'éviter la reconfiguration en générant une configuration de la ligne commune entre tous les produits. D'une part, ceci peut être vu comme un avantage lorsque la ligne conçue est rigide et donc difficilement reconfigurable. D'autre part, cette approche conduit à une configuration inefficace de la ligne puisqu'elle fournit un compromis de performance entre les différents produits, ce qui peut être particulièrement inapproprié de nos jours où les gammes de produits sont très variées. C'est dans ce contexte que le concept des RMS émerge comme solution permettant de concevoir des lignes facilement reconfigurables. Cela nous permet de considérer chaque produit séparément et de concevoir la meilleure configuration de la ligne pour chacun d'eux. C'est donc autour de cette vision que seront présentées les contributions étudiées dans cette thèse.

Cette thèse est basée sur un ensemble d'articles que nous avons publiés, et est divisée en cinq chapitres.

Le chapitre 1 vise à retracer l'évolution des systèmes de production, en commençant par les lignes de production dédiées, puis des systèmes de production flexibles et reconfigurables. Pour chacun de ces systèmes, nous fournissons une description de leur caractéristiques et composants,

et nous les comparons en fonction des avantages et inconvénients qu'ils présentent. Une attention particulière est portée aux différentes structures des systèmes de production reconfigurables et aux machines reconfigurables. De plus, nous expliquons brièvement les causes et évènements qui ont poussé les industriels à envisager des systèmes de production flexibles et reconfigurables.

Dans le chapitre 2, nous passons en revue la littérature des RMS. Pour ce faire, nous nous focalisons uniquement sur les problèmes d'optimisation qui surviennent lors des phases de conception et d'exploitation. Les différentes mesures de performance étudiées dans la littérature sont introduites et décrites. Puis, nous fournissons une brève description des approches de résolutions utilisées. Finalement, nous proposons une classification des problèmes d'optimisation étudiés dans la littérature de RMS.

Le chapitre 3 introduit l'une des étapes les plus importantes dans la conception de lignes de production, à savoir le problème d'équilibrage. Ce dernier est plus souvent associé aux lignes d'assemblage et est donc connu sous le nom de problème d'équilibrage de lignes d'assemblage ou ALBP (*Assembly Line Balancing Problem*, en anglais). Nous introduisons les 3 types du ALBP notamment le ALBP simple, mixte et multiple. Ensuite, le problème d'équilibrage correspondant à chacun de ces types est décrit et un aperçu des approches de résolution les plus utilisées dans la littérature est donné.

Le problème de conception de lignes de production multi-produits reconfigurables est présenté dans le chapitre 4. Le but est de concevoir pour chaque produit une configuration admissible de telle sorte à optimiser le nombre de tâches réaffectées lors du passage d'une configuration vers une autre. Pour ce faire, deux variantes de ce problème sont étudiées. Pour chacune de ces dernières, un modèle de programmation linéaire en nombres entiers (PLNE) est proposé et une nouvelle heuristique, appelée Halt-and-Fix, est également développée.

Dans le chapitre 5, nous nous intéressons aux aspects liés à la modularité dans la conception de lignes reconfigurables multi-produits. Ainsi, en considérant plusieurs produits à fabriquer et un nombre de machines modulaires donné, l'objectif vise à optimiser le nombre total de modules utilisés. Pour cela, nous avons développé deux approches : l'une est un PLNE compact, et l'autre est une approche de décomposition du problème qui se base sur un PLNE étendu.

Finalement, une conclusion générale sur les travaux présentés dans ce manuscrit et des perspectives de recherches seront données.

Chapitre 1

Évolution des systèmes de production

Dans ce chapitre, les différents types de systèmes de production sont présentés. Une attention particulière est accordée aux systèmes de production reconfigurables, dont les composants et la structure sont expliqués en détails.

1.1 Introduction

Avant l'apparition de l'industrie, la production était à petite échelle, destinée à des marchés limités, et à forte intensité de main-d'œuvre plutôt que de capital [Hopp and Spearman, 2000]. Le travail était effectué selon deux systèmes : le système domestique et les guildes artisanales. Dans le système domestique, les matériaux étaient distribués par les marchands dans les maisons où les gens effectuaient les opérations nécessaires. Par exemple, dans le secteur du textile, différentes familles filaient, blanchissaient, teignaient le matériel et le revendaient à la pièce. Dans les guildes artisanales, le travail était transmis d'un atelier à l'autre. Par exemple, le cuir était tanné par un tanneur, transmis aux corroyeurs, puis aux cordonniers et aux selliers. Il en résulte des marchés distincts et décentralisés pour le matériau à chaque étape du processus [Hopp and Spearman, 2000].

L'innovation la plus importante de cette époque est la machine à vapeur. Cette dernière a été mise au point par James Watt en 1765 et installée pour la première fois par John Wilkinson dans son usine de fer en 1776. En 1781, Watt a mis au point la technologie permettant de transformer le mouvement de haut en bas de la poutre motrice en mouvement rotatif [Hopp and Spearman, 2000]. La vapeur est ainsi devenue une source d'énergie pratique pour une multitude d'applications, notamment dans les usines, les navires, les trains et les mines. Suite à ça, de nombreuses nouvelles inventions et innovations ont vu le jour. L'une des plus importantes est la ligne d'assemblage. Celle-ci a révolutionné le monde et constitue, jusqu'à présent, un des piliers de l'industrie moderne.

1.2 Lignes d'assemblage et systèmes de production dédiées

Au début du 20ème siècle, la production à grand volume était courante dans les industries de transformation telles que l'acier, l'aluminium, le pétrole, les produits chimiques, les aliments et le tabac. La production de masse de produits mécaniques tels que les machines à coudre, machines à écrire, faucheuses et machines industrielles, basée sur de nouvelles méthodes de fabrication et d'assemblage de pièces métalliques interchangeables, était en plein essor. Cependant, c'est Henry Ford (1863-1947) qui a rendu possible la production de masse, à grande vitesse, de produits mécaniques complexes avec sa célèbre innovation, la ligne d'assemblage [Hopp and Spearman, 2000]. Pour ce faire, Ford a abandonné la pratique des ouvriers qualifiés assemblant des sous-ensembles substantiels et des ouvriers se rassemblant autour d'un châssis statique pour terminer l'assemblage. Au lieu de cela, il a cherché à amener le produit aux opérateurs dans un flux continu sans arrêt. La figure 1.1 schématise ce type de lignes. Après Ford, la production de masse est devenue presque synonyme de ligne d'assemblage. Cette dernière a marqué le début des systèmes de fabrication dédiés, ou DML (*Dedicated Manufacturing Lines*, en anglais).

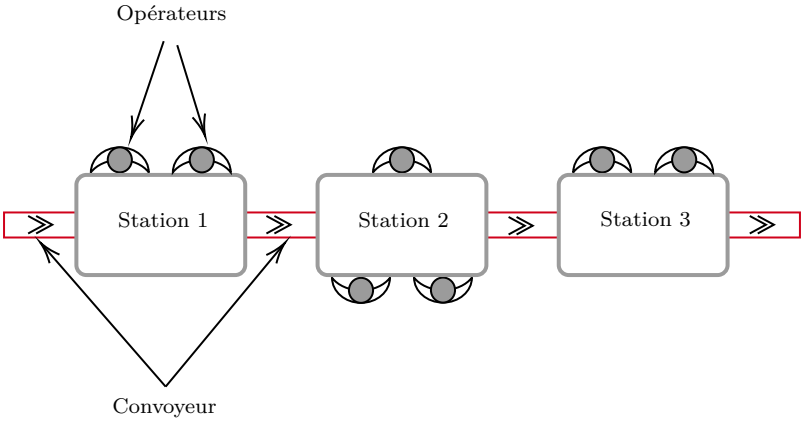


FIGURE 1.1 – Exemple d'une ligne d'assemblage

Les DML ont dominé la fabrication de produits et de pièces de haute précision dans l'industrie mondiale au cours de la seconde moitié du vingtième siècle [Koren, 2020]. Les DML se basent principalement sur un processus fixe et rigide et sont capables de produire des produits ou des pièces à haut volume. En effet, une ligne dédiée est généralement conçue pour produire un seul modèle de produit/pièce à une cadence de production assez élevée. Ces lignes se composent d'un ensemble de stations de travail, où de nombreuses opérations sont effectuées simultanément sur un produit. Un système de manutention, généralement un convoyeur, relie les stations entre elles et permet ainsi de transporter le produit tout au long de sa fabrication (voir figure 1.1). Ce

fonctionnement s'est avéré très efficace et a permis d'atteindre des taux de production élevés à un faible coût unitaire, les coûts devant diminuer à mesure que le volume augmente.

La production de masse, entraînée par l'invention des DML, s'est avérée efficace et a permis de faire basculer la balance du marché rendant l'offre supérieure à la demande. Par conséquent, et en raison de l'évolution rapide de la technologie et de la mondialisation qui a entraîné une compétitivité globale, la satisfaction du client et l'adaptation à ses attentes sont devenues une nécessité pour les entreprises de production. Or, un des problèmes majeurs des DML est qu'ils sont intrinsèquement inflexibles. En d'autres termes, les outils, machines, et postes de travail sont précisément adaptés pour produire, avec une efficacité maximale, un produit ou une pièce seulement. Ainsi, tout changement dans le produit (fluctuation de la demande ou évolution) est susceptible de rendre obsolètes des machines et outils coûteux et compliquer la réorganisation des tâches des travailleurs. Dans un tel environnement, une solution consiste à équiper la ligne de machines flexibles pour la rendre capable de s'adapter rapidement aux changements du marché.

1.3 Systèmes de production flexibles

Vers la fin du 20ème siècle, le besoin de concevoir un système de production flexible se faisait de plus en plus ressentir. En effet, le marché était (et est toujours) caractérisé par des fluctuations et des changements de plus en plus nombreux et imprévisibles entraînés principalement par : (1) l'augmentation de la fréquence d'introduction de nouveaux produits ; (2) des modifications de pièces pour des produits existants ; (3) des fluctuations importantes de la demande et de la gamme de produits ; (4) des changements dans les réglementations gouvernementales liés à la sécurité et l'environnement, et (5) des changements dans la technologie.

Les systèmes de production flexibles, ou FMS (*Flexible Manufacturing Systems*, en anglais), ont été introduits afin d'offrir aux industriels la possibilité d'améliorer leur technologie, leur compétitivité et leur rentabilité [Liu, 2008]. En effet, la flexibilité permet à une entreprise de s'adapter plus facilement à l'évolution du marché et aux exigences des clients, tout en maintenant des normes de qualité élevées pour ses produits [Shang and Sueyoshi, 1995]. Ceci est rendu possible grâce à l'apparition, dans les années 1960, des bras robotiques, des automates programmables et des machines-outils à commande numérique, communément appelées machines CNC (*Computer Numerical Controlled*, en anglais). Ces dernières sont cependant celles qui caractérisent le plus les FMS.

Les machines CNC sont principalement utilisées dans l'usinage de pièces mécaniques et permettent d'effectuer des opérations telles que le perçage, découpage, fraisage, etc. Les CNC sont des machines programmables pouvant ainsi usiner des pièces en toute autonomie et avec une grande précision. Contrairement aux machines standards des DML, qui sont pré-programmées

pour effectuer de façon répétitive les mêmes opérations, les machines CNC, quant à elles, peuvent être programmées et reprogrammées facilement et rapidement pour effectuer une grande variété d'opérations sur des produits et matériaux différents. De plus, ces machines disposent généralement d'un ensemble d'outils changeables permettant à la machine d'être reconfigurée.

Un FMS est donc composé de cellules d'usinage, chacune disposant d'une ou de plusieurs machines CNC. Les cellules sont interconnectées, via des stations de chargement\déchargement, par un système de transport automatisé [Kostal and Velisek, 2011]. Cette structure permet d'assurer la production de nombreux modèles de pièces en lots de petites tailles. La figure 1.2 montre un exemple d'un FMS composé d'un convoyeur central autour duquel sont disposés trois machines CNC, un système de stockage/déstockage automatisé, une station de chargement/déchargement, un bras robotique et un ordinateur qui permet de contrôler toutes ces ressources.

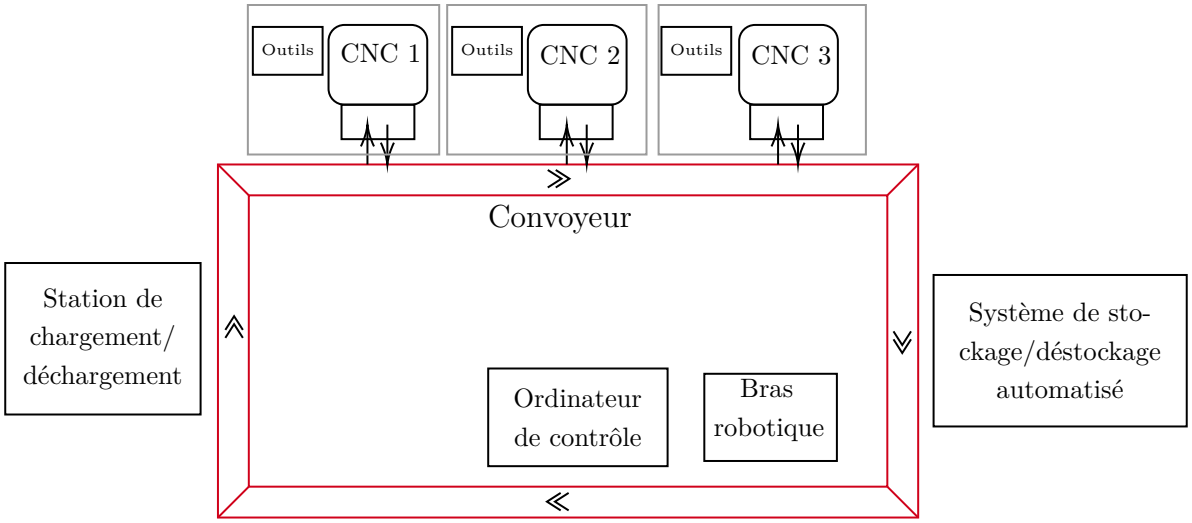


FIGURE 1.2 – Exemple d'un système de production flexible

Cependant, les FMS ne se sont pas diffusés comme prévu [Handfield and Pagell, 1995]. Les entreprises se sont montrées réticentes face à ces systèmes et ce pour plusieurs raisons. En effet, passer à un FMS représente une décision à fort impact financier et organisationnel. La mise de fonds initiale est si élevée qu'elle pèse souvent lourdement sur les ressources financières des entreprises, tandis que la flexibilité des FMS est souvent surdimensionnée par rapport aux besoins réels des entreprises. De plus, les FMS ne sont pas conçus pour la production de masse. Comme les machines CNC n'utilisent qu'un seul outil à la fois, la productivité des FMS est faible par rapport à celle des DML. Par conséquent, la combinaison d'un coût d'équipement important et d'un faible débit rend le coût par pièce relativement élevé [Koren et al., 1999].

Pour résumer, il est évident que la flexibilité est une caractéristique nécessaire pour les entreprises d'aujourd'hui en raison des exigences et fluctuations des marchés. Les FMS se sont

avérés être une solution capable de faire face à cette situation. Néanmoins, les coûts d'investissement élevés, la complexité de ces systèmes et les performances fournies, font que les FMS ne conviennent pas à de nombreux secteurs industriels. Par conséquent, la flexibilité doit être obtenue par d'autres moyens.

1.4 Systèmes de production reconfigurables

Nous avons introduit précédemment deux types de systèmes de production. A savoir, les DML qui sont conçus pour fabriquer un seul produit avec une capacité de production élevée, et les FMS qui sont des systèmes flexibles capables de s'adapter rapidement aux changements, mais qui offrent une faible productivité. Ainsi, en considérant les avantages et les inconvénients de chacun de ces systèmes, il semble que l'amélioration la plus naturelle soit une combinaison des deux. Autrement dit, un système à la fois flexible et capable d'offrir une productivité élevée.

Dans ce contexte, le concept de systèmes de production reconfigurables ou RMS (*Reconfigurable Manufacturing Sytems*, en anglais) a été introduit par Koren et al. [1999]. Contrairement aux FMS, les RMS sont conçus avec un niveau de flexibilité personnalisé autour d'une famille prédéfinie de produits. De plus, la structure de ces systèmes doit être conçue de telle sorte qu'elle soit évolutive et ajustable pour mieux faire face aux variations de la demande ou aux changements des produits. La figure 1.3 montre l'évolution d'un RMS en termes de fonctionnalité et de capacité de production, durant son cycle de vie. Cette figure met aussi en évidence la différence entre un DML, FMS et RMS. En effet, on peut voir que les RMS peuvent évoluer dans le temps et s'adapter, suivant le besoin et de façon progressive, aux variations de la demande et aux changements des produits.

Selon Koren and Shpitalni [2010], pour qu'un système de production soit reconfigurable, il doit satisfaire six caractéristiques principales décrites ci-dessous.

1. **La personnalisation** se traduit par une flexibilité personnalisée d'un système/machine autour d'une famille de produit seulement.
2. **La convertibilité** est la capacité d'un système/machine à adapter rapidement ses fonctionnalités pour répondre à de nouvelles exigences de production.
3. **L'évolutivité** est l'aptitude d'un système/machine à modifier facilement sa capacité de production en ajoutant de nouvelles machines par exemple.
4. **La modularité** consiste à subdiviser des fonctions opérationnelles en unités, ou modules, interdépendants de manière à pouvoir les arranger et les réorganiser facilement.
5. **L'intégrabilité** consiste à ajouter de nouvelles ressource (modules, machines, etc) rapidement et efficacement à travers des interfaces de contrôle standardisées facilitant l'intégration et la communication.

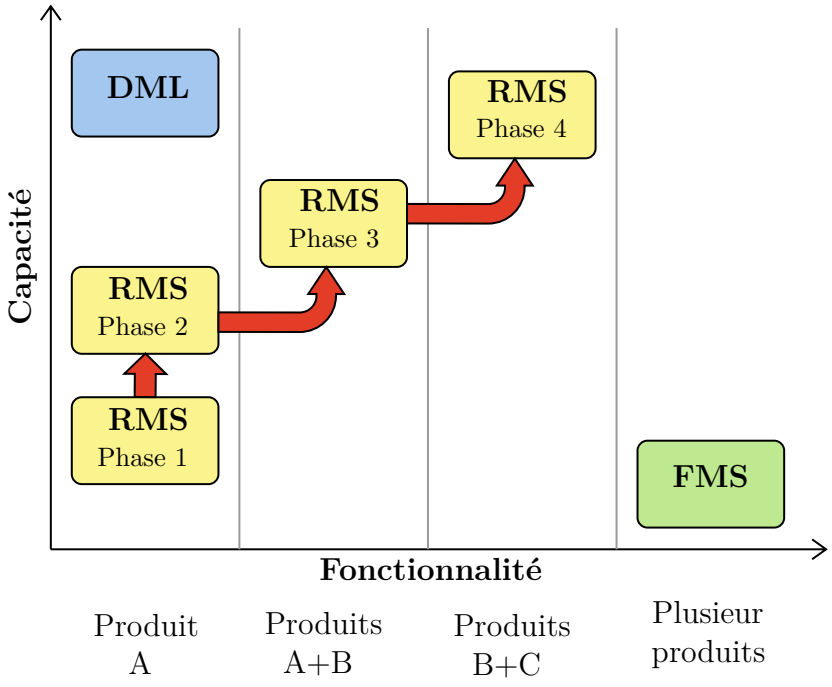


FIGURE 1.3 – Comparaison entre DML, FMS et RMS

6. **La diagnostiquabilité** est la capacité d'un système/machine à pouvoir être facilement diagnostiqué dans le but de détecter rapidement des anomalies telles que les pannes et les défauts.

Ces caractéristiques permettent d'améliorer la réactivité du système en réduisant le temps et l'effort de reconfiguration. De plus, celles-ci peuvent réduire de manière fiable les coûts associés au cycle de vie du système en lui permettant d'évoluer constamment au rythme de la demande des consommateurs, de la technologie des procédés et des variations du marché [Koren and Shpitalni, 2010].

Parmi les caractéristiques mentionnées ci-dessus, la personnalisation, l'évolutivité et la convertibilité sont les plus critiques. Quant aux autres, à savoir la modularité, l'intégrabilité et la diagnostiquabilité, elles permettent une reconfiguration rapide, mais ne garantissent pas les modifications de la capacité de production et des fonctionnalités.

1.5 Les machines reconfigurables

Les Machines Reconfigurables ou RM (*Reconfigurable Machines*, en anglais) sont les principaux composants qui confèrent aux RMS la capacité de s'adapter et de répondre rapidement aux fluctuations du marché et aux aléas. Elles sont essentiellement conçues avec une flexibi-

lité nécessaire pour traiter une famille particulière de produits. Ces machines ont une structure modulaire, c'est-à-dire qu'elles sont constituées de composants physiques qui peuvent être facilement remplacés et/ou réorientés, offrant ainsi une large gamme de fonctionnalités. Cela facilite la reconfiguration d'une RM lui permettant ainsi de s'adapter aux changements des produits ou de la demande. Selon Koren [2010], il existe quatre types de RM (illustrés dans la figure 1.4) :

1. Les machines-outils reconfigurables,
2. Les machines d'assemblage reconfigurables,
3. Les machines d'inspection reconfigurables,
4. Les fixations reconfigurables.

1.5.1 Machines-outils reconfigurables

Communément appelées RMT (*Reconfigurable Machine Tools*), ces machines sont utilisées dans les centres d'usinage. Elles se composent principalement de modules physiques (tels que des outils d'usinage, des têtes de broche, des tourelles, etc.), chacun d'entre eux étant capable d'effectuer une ou plusieurs tâches de fabrication. Ces modules peuvent être placés de plusieurs façons dans une machine, ce qui permet d'obtenir différentes configurations. Chaque configuration est caractérisée par un ensemble de fonctionnalités.

1.5.2 Machines d'assemblage reconfigurables

Les machines d'assemblage reconfigurables ou RAM (*Reconfigurable Assembly Machines*, en anglais), sont principalement utilisées dans les lignes d'assemblage. De la même façon que les RMT, celles-ci ont une structure modulaire et peuvent être modifiées pour assembler des produits ayant certaines caractéristiques communes. Katz [2006] ont fourni un exemple d'une RAM permettant l'assemblage de différents types d'échangeurs de chaleur automobile appartenant à la même famille de pièces.

1.5.3 Machines d'inspection reconfigurables

Contrairement aux RMT et RAM, ces machines sont utilisées pour inspecter et mesurer les pièces usinées en cours de fabrication. Elles sont composées de différents capteurs photoélectrique ou optique qui servent, entre autre, à mesurer les dimensions et à s'assurer de la bonne qualité de la pièce. Le nombre et l'emplacement de ces capteurs peuvent être modifiés en fonction de la pièce à mesurer.

1.5.4 Les fixations reconfigurables

Ces fixations sont utilisées dans l'usinage de pièces de grande taille ou complexes telles que les culasses de moteurs. Le processus d'usinage consiste à fixer la pièce sur des supports spécialement conçus pour effectuer différentes tâches d'usinage telles que la découpe et la soudure.

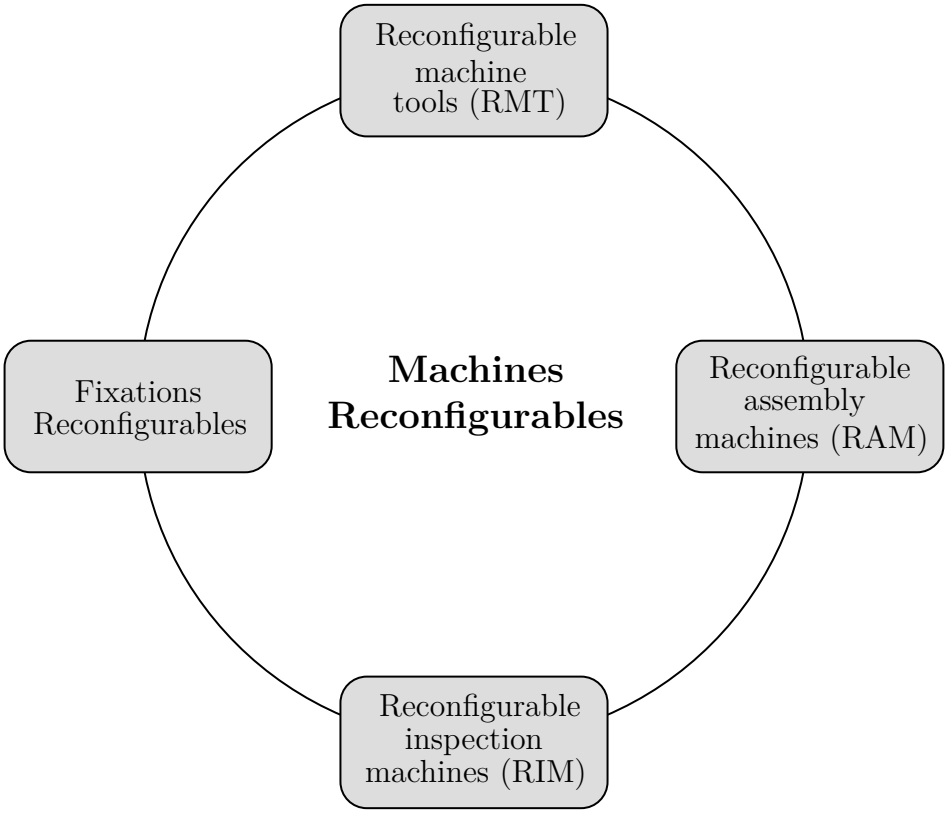


FIGURE 1.4 – Les types de machines reconfigurables (RM)

L'utilisation d'un type particulier de RM dépend du secteur industriel concerné. Cependant, la plupart des études menées sur les RMS n'ont considéré que les RMT [Bortolini et al., 2018]. Cela est dû au fait que le RMT a attiré l'attention des universitaires et des industriels (fabricants et usagers de machines-outils), qui ont encouragé leur recherche et leur développement [Gadalla and Xue, 2016].

1.6 Les structures d'un système reconfigurable

Un RMS est principalement composé d'un ensemble de RM, qui sont disposées d'une certaine manière pour fabriquer une famille donnée de produits tout en assurant une capacité de production souhaitée. La disposition de ces machines définit le type de système de fabrication,

dont le choix est un enjeu majeur dans la phase de conception des RMS. Il s'agit d'une décision stratégique, coûteuse en termes d'investissement, et qui a un impact direct sur la productivité et la qualité des produits [Koren, 2010]. Nous allons maintenant donner un aperçu des différents types de RMS, sur la base de l'étude des articles de recherche.

1.6.1 Lignes reconfigurables

Ce type de lignes est constitué d'un ensemble de postes de travail, aussi appelés stations. Chaque station est composée d'une ou plusieurs RM capables d'exécuter un ensemble donné de tâches de fabrication. La figure 1.5 montre la structure de base de ce type de lignes. Elles sont généralement utilisées pour l'assemblage ou l'usinage de pièces. Dans Bryan et al. [2013], les

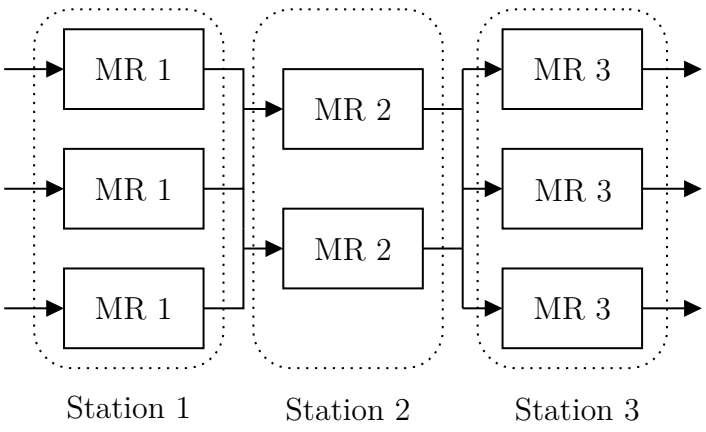


FIGURE 1.5 – Structure d'une ligne de production reconfigurable

auteurs ont considéré une ligne d'assemblage reconfigurable comprenant un ensemble de postes de travail composés de centres identiques disposés en parallèle. Chaque centre comprend des ressources reconfigurables qui exécutent des tâches. Cette structure permet au système d'être plus réactif lorsqu'un produit change ou évolue, en rendant possible la reconfiguration des centres ou le déplacement des ressources entre les différents postes de travail. Par exemple, un centre peut être composé de convoyeurs et de racks de stockage, alors que les ressources sont des pistolets de soudage et des dispositifs de fixation [Bryan et al., 2013]. De même, dans les lignes d'usinage reconfigurables, un poste de travail est constitué d'un ensemble de RMT et/ou de machines identiques disposées en parallèlement. Cette structure est, selon Koren et al. [2016], efficace car elle offre un certain niveau d'évolutivité qui permet d'ajouter de nouvelles machines si besoin. De plus, cette structure apporte un certain niveau de robustesse contre les pannes ou l'indisponibilité des machines [Spicer and Carlo, 2007, Dou et al., 2008, Carlo et al., 2012, Ashraf and Hasan, 2018].

Dans la même optique, les tables d'usinage rotatives reconfigurables utilisent une table tournante pour transporter des produits à travers différents postes de travail (ou positions de travail). Chaque poste de travail est constitué d'unités d'usinage modulaires (composées de têtes de broche ou de tourelles), qui peuvent effectuer différentes tâches sur un ensemble de pièces [Battaïa et al., 2016, Liu et al., 2018]. La figure 1.6 illustre une table d'usinage rotative avec six positions de travail reconfigurables, notées PTR.

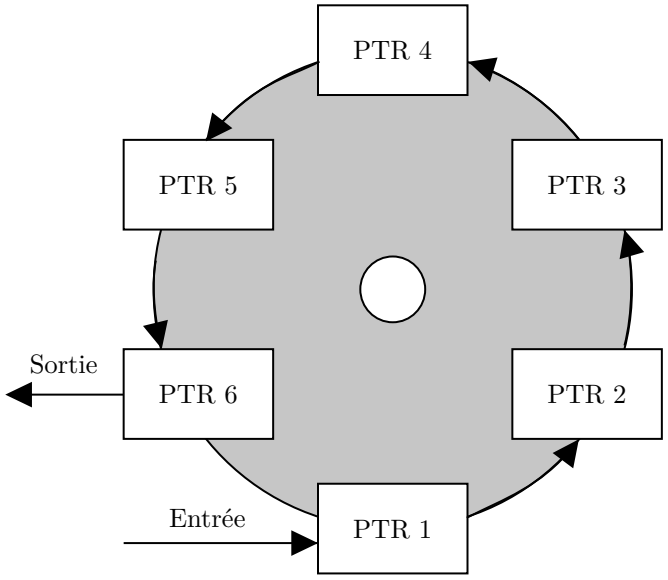


FIGURE 1.6 – Table d'usinage rotative reconfigurable

1.6.2 Systèmes de production cellulaires reconfigurables

Les systèmes de production cellulaires, ou CMS (*Cellular Manufacturing Systems*, en anglais), sont basés sur le concept de la technologie de groupe, qui vise à augmenter la productivité en regroupant des principes, des problèmes et des tâches [Greene and Sadowski, 1984]. Un CMS est constitué de cellules capables de produire un groupe de pièces ayant des tâches de production, des outils, ou des machines similaires. Selon Wemmerlov and Johnson [1997], un CMS permet de réduire le temps de production, le nombre de produits en cours de fabrication, le temps de réponse aux commandes et le temps/la distance de déplacement du flux de produits, tout en leur assurant une meilleure qualité. De la même façon, un CMS reconfigurable est défini comme un CMS dont les cellules sont composées de RM [Yu et al., 2012, Aljuneidi and Bulgak, 2016] comme le montre la figure 1.7.

Il existe dans la littérature un autre type de CMS qui peut aussi être considéré dans le contexte des RMS. Il s'agit des CMS dynamiques qui sont composés de machines mobiles pouvant

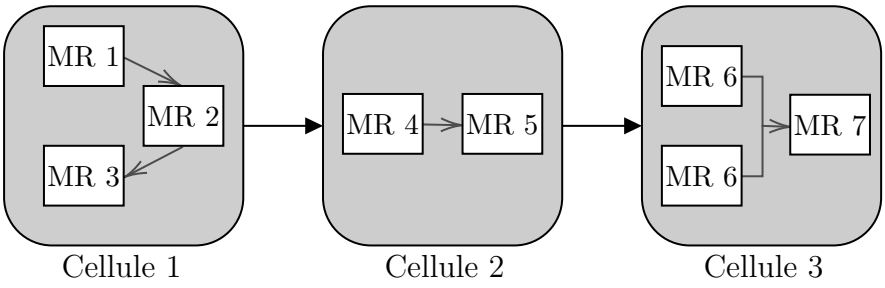


FIGURE 1.7 – Un système de production cellulaire reconfigurable

être déplacées parmi les cellules existantes afin de répondre aux exigences de production sur différentes fenêtres de temps [Rheault et al., 1995].

1.7 Conclusion

Dans ce chapitre, nous avons retracé l'évolution des systèmes de production en commençant par la première révolution industrielle jusqu'à aujourd'hui. Une attention particulière a cependant été accordée aux RMS. Pour cela, la première partie traitait de l'apparition des lignes d'assemblage et le commencement des DML qui ont principalement permis la production de masse. Dans la deuxième partie de ce chapitre, nous avons expliqué comment le besoin de flexibilité des systèmes de production se faisait de plus en plus ressentir dans un marché devenu volatile et imprévisible. Ceci nous a permis d'introduire et de décrire brièvement les FMS. Finalement, nous avons introduit les RMS et expliqué en quoi ces systèmes sont une meilleure solution comparés au DML et FMS. De plus, nous avons introduit, en se basant sur la littérature, les machines reconfigurables et les différentes structures des RMS.

Dans cette thèse, nous nous intéressons principalement à la conception des RMS. Ainsi, le chapitre suivant présente un état de l'art sur les différents problèmes d'optimisation liés aux RMS, les mesures de performances de ces systèmes et les approches de résolution utilisées.

Chapitre 2

Conception et optimisation des RMS : état de l'art

L'objectif de ce chapitre est de passer en revue la littérature de recherche relative aux problèmes d'optimisation des RMS et à leurs méthodes de résolution. À cette fin, les fonctions objectifs et les indicateurs de performance des RMS sont présentés. En outre, un aperçu des approches de résolution les plus utilisées et une classification des problèmes d'optimisation sont proposés. Enfin, une analyse détaillée, nos conclusions et des suggestions pour les recherches futures sont fournies.

2.1 Introduction

Comme mentionné dans le chapitre précédent, un système de production est dit "*reconfigurable*" s'il présente un certain nombre de caractéristiques telles que la modularité, l'intégrabilité, la personnalisation et l'évolutivité [Koren et al., 1999]. Chacune de ces caractéristiques confère à un RMS un certain degré de réactivité lui permettant de s'adapter à différentes circonstances. Le concept de RMS a reçu une attention croissante au sein de la communauté scientifique, avec plus de 60% des articles scientifiques publiés entre 2010 et 2017 selon Bortolini et al. [2018]. Dans ce dernier, une revue de l'état de l'art a été proposée. Celle-ci fournit une vue générale sur des différents sujets et applications liés aux RMS. Pour ce faire, les auteurs ont classé et cartographié les articles scientifiques et identifié cinq principaux axes de recherche. En outre, ils ont présenté une vision large reliant RMS à l'Industrie 4.0, et ont soulevé quelques questions intéressantes qui pourraient donner lieu à des pistes prometteuses pour des travaux futurs. De même, dans Andersen et al. [2015], les auteurs ont proposé un aperçu des tendances de la recherche sur les RMS. La classification qu'ils ont proposée est basée sur différents niveaux organisationnels dans une usine. Ceci a permis d'identifier les problèmes de recherche, abordés dans la littérature, pour chaque niveau. Ils ont conclu que peu de travaux recherches ont été menés sur le plan de la structuration supérieure d'une usine dans un environnement reconfigurable.

Les deux revues de la littérature évoquées ci-dessus donnent une vision globale des problèmes

liés aux RMS. Elles décrivent toutes deux les caractéristiques générales et les indicateurs de performance. En effet, nous avons pu constater, à travers ces articles, que la plupart des études liées aux RMS ont été formulées comme des problèmes d'optimisation combinatoires. Toutefois, l'objectif de Bortolini et al. [2018] et de Andersen et al. [2015] n'était pas de fournir une description détaillée de ces problèmes et de leurs approches de résolution, mais plutôt de présenter un aperçu global des travaux de recherche menés dans cette direction. Par conséquent, le but de cette section est de présenter, de façon plus approfondie, les différents problèmes d'optimisation qui sont traités dans la littérature des RMS. Pour ce faire, nous décrivons d'abord les fonctions objectifs les plus pertinentes utilisées pour évaluer les performances des RMS, puis nous proposons une classification des problèmes d'optimisation et de leurs approches de résolution.

La recherche des articles était fondée sur une fouille avancée des contributions dont le titre, le résumé ou les mots-clés contenaient les termes *« reconfigurable »*, *« RMS »* ou *« reconfigurability »* dans la base de données SCOPUS. Suite à ça, nous avons remarqué que la plupart des articles de conférences et chapitres de livres sont étendus en articles de journaux. Ainsi, nous avons préféré ne considérer que ces derniers dans notre état de l'art.

2.2 Les mesures de performance

Avant d'aborder les problèmes d'optimisation et leurs approches de résolution, il est important d'analyser d'abord les mesures utilisées pour évaluer les performances des RMS. Ces dernières sont généralement considérées comme des fonctions objectifs à optimiser. Ainsi, on distingue deux niveaux différents dans lesquels les mesures de performance peuvent être évaluées, à savoir le niveau machine et le niveau système.

2.2.1 Niveau machine

Comme mentionné précédemment, un RMS est principalement composé de Machines Reconfigurables (MR). Par conséquent, évaluer les performances des MR à travers différentes mesures s'avère une étape importante. Dans ce contexte, plusieurs études ont été menées sur les MR, et plus particulièrement les RMT (pour les raisons mentionnées précédemment). Dans cette section, nous nous limitons aux indicateurs de performance utilisés dans la phase de conception/sélection des RMT. Nous renvoyons le lecteur aux articles de Landers et al. [2001], Pasek [2006], et Gadalla and Xue [2016] pour des informations plus détaillées sur les problèmes liés aux RMT.

a. Objectifs axés sur les coûts

Le coût est l'un des objectifs le plus considéré dans la littérature. Celui-ci est généralement calculé comme étant la somme des coûts d'investissement et d'exploitation. Le premier inclut les coûts relatifs à l'achat des modules, axes, broches, etc. [Youssef and ElMaraghy, 2006, Maniraj et al., 2015], tandis que le second comprend les coûts engendrés durant la phase de production [Bensmaine et al., 2013] tels que les coûts d'utilisation et d'entretien des machines, et ceux liés à la consommation énergétique.

b. Objectifs axés sur le temps

Le temps d'exécution est souvent considéré dans les modèles d'optimisation des systèmes de production conventionnels. Au niveau machine, cet objectif se traduit par la somme de la durée des tâches effectuées par celle-ci. Dans un environnement composé de machines reconfigurables, il est également important de prendre en compte le temps de reconfiguration de la machine pour passer d'une configuration vers une autre.

c. Reconfigurabilité des machines

La reconfigurabilité de la machine est un indicateur de performance qui montre si une RMT dispose de configurations alternatives. En effet, celle-ci reflète la faculté d'adaptation d'une RMT à différents scénarios de production. Dans Goyal et al. [2012] et Ashraf and Hasan [2018], cet indicateur est calculé à partir du nombre de configurations alternatives possibles et de l'effort requis pour la reconfiguration. Ce dernier est exprimé en nombre de modules retirés, déplacés ou ajoutés.

d. Capacité opérationnelle

Cet indicateur est utilisé par Goyal et al. [2012], Goyal and Jain [2015], et Ashraf and Hasan [2018]. Il reflète la capacité d'une RMT à effectuer une variété de tâches de fabrication avec une configuration donnée.

e. Autres indicateurs de performance

Goyal and Jain [2015] vise à optimiser le taux d'utilisation des machines, considéré comme le rapport entre la demande et le taux de production des machines. D'un autre côté, Xie et al. [2012] a considéré la maximisation de la précision d'usinage, qui, selon les auteurs, est liée à la configuration d'une RMT.

Le tableau 2.1 montre les objectifs les plus considérés dans la littérature sur les RMT. On constate que le coût et le temps sont les objectifs les plus récurrents. De plus, la reconfigurabilité

des machines, la capacité opérationnelle et d'autres indicateurs de performance sont des objectifs tout aussi importants, car ils reflètent la capacité d'une RMT à faire face à des fluctuations telles que des changements de produits, des variations de la demande ou des pannes de machines.

Tableau 2.1 – Fonctions objectifs liées aux RMT

Références	Coût	Temps	Reconfiguration	CO	Autres
Ashraf and Hasan [2018]	✓		✓	✓	✓
Battaïa et al. [2016]	✓				
Benderbal et al. [2017a]		✓			
Benderbal et al. [2017b]	✓	✓			
Bensmaine et al. [2013]		✓			
Chaube et al. [2010]	✓	✓			
Choi and Xirouchakis [2014]	✓				
Dou et al. [2008]	✓				
Dou et al. [2016]	✓				
Goyal et al. [2012]	✓		✓	✓	
Goyal and Jain [2015]	✓		✓	✓	✓
Liu et al. [2018]	✓				
Maniraj et al. [2014]	✓				
Touzout and Benyoucef [2018]	✓	✓			
Touzout and Benyoucef [2019]		✓			
Xie et al. [2012]	✓				✓

CO : "Capacité Opérationnelle".

2.2.2 Niveau système

Le niveau système est composé de toutes les ressources qui constituent un RMS telles que les machines reconfigurables, les postes de travail, les cellules, les systèmes de manutention, etc. Dans cette sous-section, nous présentons une description des fonctions objectifs utilisées dans le contexte des RMS.

a. Objectifs axés sur les coûts

En plus des coûts liés aux RMT mentionnés plus haut, il existe également des coûts spécifiques aux lignes qui doivent être pris en compte pour l'ensemble du système. Ces coûts peuvent être classés en trois groupes principaux, à savoir les coûts d'investissement, les coûts d'exploitation et les coûts de reconfiguration du RMS. Il convient de noter que les coûts d'exploitation et de reconfiguration représentent le coût de production [Abbasi and Houshmand, 2009].

b. Objectifs axés sur le temps

En plus des temps de traitement et de reconfiguration des machines, un temps de réalisation au niveau du système, qui comprend le temps nécessaire pour modifier la configuration d'une ligne (c'est-à-dire changer la disposition de la ligne, ajouter de nouvelles stations, etc.) et le temps de changement de machines ont été pris en compte par Benderbal et al. [2017a]. D'autres auteurs tels que Chaube et al. [2010], Bensmaine et al. [2014] et Benderbal et al. [2017b] ont également considéré le temps de transport entre les différentes machines.

La productivité est un autre indicateur de performance couramment utilisé dans les système de production. Celui-ci fournit des informations concernant le nombre de pièces produites par unité de temps [Musharavati and Hamouda, 2012b, Choi and Xirouchakis, 2014]. Dans le contexte des RMS, la productivité n'est pas autant traitée dans la littérature comparée au temps de réalisation. De la même manière, Liu et al. [2018] a considéré le temps de cycle, qui est défini comme l'intervalle de temps entre la sortie de deux produits finis. Contrairement au temps de réalisation, le temps de cycle ne tient compte que des temps de traitement et non du temps de reconfiguration.

c. Consommation énergétique

La réduction de la consommation d'énergie et de l'empreinte carbone est devenue un défi majeur dans le secteur industriel durant ces dernières décennies [Touzout and Benyoucef, 2018]. Ces sujets ont donc fait l'objet d'une attention considérable dans la recherche sur les RMS. On peut citer par exemple Choi and Xirouchakis [2014], Liu et al. [2018], et Touzout and Benyoucef [2018] qui visent à optimiser la consommation énergétique d'un RMS. En effet, dans Choi and Xirouchakis [2014] et Touzout and Benyoucef [2018], les auteurs ont considéré l'énergie consommée par un RMS composé de RMT. Elle comprend l'énergie liée à l'usinage des pièces, l'énergie induite par le changement de la configuration et d'outils et l'énergie nécessaire au transport et à la manutention. Liu et al. [2018] ont considéré un RMS composé d'une table d'usinage rotative. Les auteurs ont calculé la consommation énergétique comme la somme de l'énergie de démarrage de la machine, de l'énergie liée à la table et aux tourelles, et de l'énergie requise pour effectuer les opérations d'usinage.

d. Autres indicateurs de performance

Afin de quantifier les différentes caractéristiques d'un RMS, plusieurs auteurs ont proposé divers indicateurs de performance. Ces derniers permettent de mesurer le niveau de flexibilité et de réactivité du système dans diverses circonstances. On citera par exemple la modularité et la flexibilité du système qui ont été considérés par Benderbal et al. [2017b] et Benderbal et al.

[2017a]. Aussi, Goyal and Jain [2015] a introduit la convertibilité d'une configuration qui reflète la capacité du système à être reconfiguré.

Le tableau 2.2 résume les fonctions objectifs utilisées dans les RMS. On peut remarquer que le coût et le temps restent les aspects les plus traitées. Peu de travaux ont été menés pour optimiser la consommation énergétique (notée "Énergie") et la reconfigurabilité de ce type de lignes.

Tableau 2.2 – Fonctions objectifs liées aux RMS

Références	Coût	Temps	Énergie	Autres
Abbasi and Houshmand [2009]	✓			
Aljuneidi and Bulgak [2016]	✓			
Ashraf and Hasan [2018]				✓
Benderbal et al. [2017a]		✓		✓
Benderbal et al. [2017b]				✓
Bensmaine et al. [2014]		✓		
Carlo et al. [2012]	✓			
Choi and Xirouchakis [2014]	✓	✓	✓	
Deif and ElMaraghy [2006a]	✓			
Dou et al. [2008]	✓			
Eguia et al. [2016]	✓			
Elmaraghy et al. [2008]	✓			
Liu et al. [2018]		✓	✓	
Maniraj et al. [2014]	✓	✓		
Musharavati and Hamouda [2012b]	✓			
Saxena and Jain [2012]	✓			
Shabaka and ElMaraghy [2008]	✓			
Spicer et al. [2005]	✓			
Spicer and Carlo [2007]	✓			
Touzout and Benyoucef [2018]	✓	✓	✓	
Wang and Koren [2012]	✓			
Xie et al. [2012]	✓			
Ye and Liang [2006]	✓			
Youssef and ElMaraghy [2008a]				✓

2.3 Aperçu des approches de résolution

Dans les sections précédentes, nous avons introduit et décrit les indicateurs utilisés dans la littérature pour évaluer la performance d'un RMS. Cette section donne un aperçu des approches de résolution développées pour résoudre les problèmes d'optimisation liés aux RMS. Une classification de ces problèmes est donnée dans la section suivante.

Afin de résoudre un problème d'optimisation, deux types de méthodes peuvent être utilisées : les méthodes exactes qui garantissent de trouver une solution optimale, et les méthodes

approchées, telles que les heuristiques et métaheuristiques, qui visent à fournir des solutions de bonne qualité en un temps raisonnable sans pour autant garantir l'optimalité.

2.3.1 Approches exactes

Les méthodes exactes visent généralement à trouver au moins une solution optimale à un problème d'optimisation donné. Ces méthodes se basent principalement sur la recherche arborescente et sur l'énumération partielle de l'espace de solutions. Les méthodes exactes les plus connues et les plus utilisées sont la méthode de séparation et évaluation (*Branch-and-Bound*, en anglais), le *Branch-and-Cut*, le *Branch-and-Price*, la programmation dynamique et la programmation par contraintes.

Le Branch-and-Bound est une méthodologie fondamentale et largement utilisée pour obtenir des solutions optimales aux problèmes d'optimisation. Cette technique a été proposée pour la première fois par Land and Doig [1960] et consiste à énumérer implicitement toutes les solutions possibles du problème considéré, en stockant les solutions partielles, appelées sous-problèmes, dans une structure arborescente. Les nœuds non explorés de l'arbre génèrent des enfants en partitionnant l'espace de solutions en régions plus petites qui peuvent être résolues récursivement (c'est-à-dire par branchement), et des règles sont utilisées pour éliminer les régions de l'espace de recherche qui sont manifestement dominés par d'autres (c'est-à-dire par bornage). Une fois que l'arbre entier a été exploré, la meilleure solution trouvée pendant la recherche est retournée. Le *Branch-and-Cut* quant à lui est une version améliorée du Branch-and-Bound [Wolsey, 1998]. Il utilise la méthode des plans sécants pour générer des coupes, sous forme de contraintes, qui permettent de rapprocher un programme linéaire vers des solutions intégrales. De nos jours, plusieurs solveurs commerciaux existent qui se basent principalement sur le *Branch-and-Cut* pour résoudre des programmes linéaires en nombres entiers (PLNE). Les plus courants sont IBM ILOG CPLEX, GUROBI, FICO Xpress Optimization ou LINGO.

La programmation dynamique ou DP (*Dynamic Programming*, en anglais) est une méthode de résolution exacte qui repose sur la décomposition d'un problème d'optimisation en une séquence de sous-problèmes de plus petite taille résolus de façon récursive. Ainsi, la solution optimale du problème global dépend de la solution optimale de ses sous-problèmes individuels.

D'autres méthodes de résolution exactes sont utilisées dans la littérature des RMS. Par exemple, Borisovsky et al. [2013a,b] ont utilisé la génération de contraintes (GC) pour résoudre un problème d'équilibrage de lignes d'usinage reconfigurables. La GC est un algorithme exact basé sur une formulation mathématique, dans lequel certaines contraintes difficiles du problème original sont relâchées. De façon itérative, de nouvelles contraintes sont générées sur la base des solutions obtenues du problème relaxé. Cependant, l'efficacité de ces approches dépend fortement du type et de la structure du problème.

2.3.2 Approches heuristiques

Les méthodes exactes sont généralement utilisées pour aborder des problèmes d'optimisation de petite ou moyenne taille. Dans ce cas, le nombre de combinaisons réalisables est suffisamment faible permettant ainsi l'exploration de l'espace des solutions en un temps raisonnable. Par ailleurs, lorsque la taille du problème optimisé est grande, il est difficile d'éviter le phénomène d'explosion combinatoire caractérisant les problèmes NP-difficles. Face à cette situation les méthodes exactes sont inefficaces. Par conséquent, des approches heuristiques sont utilisées pour obtenir des solutions de bonne qualité en un temps raisonnable. Par exemple, les algorithmes génétiques (AG) ou le (NSGA-II), qui est une version multicritère des AG, sont largement utilisés et ont prouvé leurs efficacité dans plusieurs problèmes d'optimisation. En effet, l'AG est un algorithme évolutif basé sur la notion de population. Son fonctionnement s'inspire du processus de sélection naturelle. L'idée consiste à générer un ensemble de solutions candidates où chacune est représentée à travers un génotype. Par la suite, des croisements et des mutations sont continuellement appliqués aux génotypes afin de produire de nouvelles solutions appelées progénitures. L'algorithme s'arrête lorsque la qualité de solution souhaitée est obtenue ou lorsqu'une condition d'arrêt est atteinte.

Par ailleurs, le recuit simulé ou SA (*Simulated Annealing*, en anglais) et sa variante multi-objective (MOSA) sont aussi largement utilisés pour résoudre les problèmes d'optimisation liés aux RMS. Ces métaheuristiques s'inspirent du processus expérimental de recuit en métallurgie, qui consiste à chauffer un métal puis à le refroidir lentement pour réduire les défauts. L'implémentation algorithmique du SA est basée sur un paramètre de température, qui contrôle la probabilité d'accepter des solutions plus mauvaises lors de l'exploration d'un espace de solutions. L'algorithme s'arrête lorsque la température ambiante est atteinte.

D'autres heuristiques ont également été utilisées dans la littérature pour résoudre différents problèmes d'optimisation RMS. De brèves descriptions de ces méthodes sont données ci-dessous.

Algorithme de colonies de fourmis ou ACO (*Ant Colony Optimization*, en anglais) est un algorithme à base de population inspiré du comportement des fourmis qui recherchent le chemin le plus court entre le nid et une source de nourriture. Dans l'ACO, des fourmis artificielles sont utilisées pour explorer un espace de solutions. La solution est construite sur la base d'un modèle de phéromone, qui est un composé organique. Une intensité plus élevée de phéromone sur un chemin reflète une meilleure solution.

Recherche locale itérée est une méthode qui vise à explorer le voisinage d'une solution initiale admissible donnée. Les perturbations sont appliquées de manière aléatoire pour visiter d'autres voisinages de la solution, évitant ainsi un optimum local.

Optimisation par essaims particules ou PSO (*Particle Swarm Optimization*, en anglais) s'inspire du comportement social des oiseaux ou des poissons. PSO utilise des agents (ou

particules) pour rechercher la meilleure solution. Le mouvement de chaque particule est affecté par sa meilleure position atteinte ainsi que par la meilleure position du groupe. L'optimisation par essaims de particules multi-objectifs (MOPSO) est une version multi-critère de PSO.

La recherche tabou ou TS (*Tabu Search*, en anglais) est un algorithme basé sur la recherche locale, qui vérifie les voisins immédiats d'une solution admissible donnée. Habituellement, ces méthodes mettent en œuvre une structure de mémoire pour éviter de manipuler des solutions déjà visitées.

Système immunitaire artificiel ou AIS (*Artificial Immune System*, en anglais) est une méta-heuristique à base de population. Elle s'inspire d'un système immunitaire biologique dont le but est de protéger l'organisme contre les antigènes.

2.4 Classification des problèmes d'optimisation des RMS

D'après notre analyse des articles sélectionnés, les problèmes d'optimisation des RMS peuvent être regroupés en quatre catégories : 1) la conception des RMS, 2) la planification et l'ordonnancement de la production, 3) la conception de l'agencement, et 4) l'équilibrage et le rééquilibrage des lignes.

2.4.1 Conception des RMS

Un RMS est composé de machines reconfigurables, de machines CNC et d'opérateurs reliés par un système de manutention afin de produire une famille de produits. Ces ressources sont généralement dotées d'une technologie avancée, ce qui se traduit par des coûts d'investissement et d'exploitation élevés. Par conséquent, la conception d'un RMS est une démarche très importante, qui nécessite la prise de décisions critiques. Ces décisions comprennent la conception des produits, la formation de la famille de produits, la planification du processus, la conception de la configuration de la ligne, l'agencement des ressources ainsi que l'équilibrage [Battaïa and Dolgui, 2013].

La conception d'un produit fournit les informations nécessaires sur les procédures de fabrication. La formation de familles de produits quant à elle consiste à identifier et à regrouper les produits en familles en fonction de leurs similitudes opérationnelles [Abdi and Labib, 2004, Galan et al., 2007] ou de la similitude du séquencement des opérations [Goyal et al., 2013, Gupta et al., 2013, Bortolini et al., 2018]. Ces deux étapes étant résolues à l'aide de méthodes autres que l'optimisation (telles que les techniques de regroupement et les méthodes de prise de décision multicritères), nous n'avons pas jugé nécessaire de les mentionner en détail dans ce document

(pour plus d'informations, voir Bortolini et al. [2018]). Nous nous intéresserons principalement aux autres étapes restantes, où d'importants problèmes d'optimisation sont soulevés.

a. Planification du processus

Comme défini dans Zhang et al. [1997], la planification du processus détermine les exigences d'usinage détaillées pour transformer une matière brute en une pièce finie. Une pièce est définie par un ensemble de caractéristiques, telles qu'une forme géométrique, des trous, des bossages, etc. L'objectif de la planification des processus est de sélectionner les tâches et de déterminer leur séquence sur les lignes de fabrication. En outre, il faut sélectionner les outils d'usinage et de coupe, les dispositifs de fixation et tous les paramètres nécessaires [Alting and Zhang, 1989].

Plusieurs auteurs ont abordé l'étape de la planification des processus en considérant des RMT dans un environnement reconfigurable [Shabaka and ElMaraghy, 2008, Musharavati and Hamouda, 2012a,b, Bensmaine et al., 2014]. Le tableau 2.3 présente les articles traitant de la planification des processus dans les RMS, les fonctions objectifs et les approches de résolution utilisées. La minimisation du coût total est l'objectif qui a été le plus fréquemment étudié [Shabaka and ElMaraghy, 2008, Chaube et al., 2010, Musharavati and Hamouda, 2012b, Xie et al., 2012, Maniraj et al., 2014, Touzout and Benyoucef, 2019]. On remarque aussi que de nombreux auteurs ont envisagé l'optimisation multi-objectif. Par exemple, Chaube et al. [2010], Bensmaine et al. [2013] et Benderbal et al. [2017b] ont minimisé à la fois le temps total et le temps d'achèvement. De plus, Benderbal et al. [2017a,b] ont considéré la maximisation de la modularité et de la flexibilité du système, respectivement. Touzout and Benyoucef [2018, 2019] visent à minimiser la consommation d'énergie et le temps d'exploitation des machines. A l'inverse, des auteurs comme Musharavati and Hamouda [2012b] ont considéré le débit comme un second objectif en plus du coût total. Xie et al. [2012] ont introduit et optimisé la précision des machines ainsi que l'indice de lissage, qui reflète la répartition de la charge de travail entre les machines.

Il convient de noter que la plupart des études susmentionnées ne prennent pas en compte une disposition particulière des RMT. Comme dans un job-shop, le seul aspect pris en compte est l'acheminement des pièces et les configurations des RMT. Cela signifie que deux pièces peuvent avoir un routage différent à travers le même ensemble de machines disponibles. Par conséquent, ces systèmes peuvent être considérés comme des job-shops reconfigurables, qui privilégient la flexibilité plutôt que la productivité. Dans les lignes reconfigurables, un ensemble de postes de travail est disposé de manière à ce que tous les produits suivent le même chemin, c'est-à-dire du premier poste de travail jusqu'au dernier. Ces systèmes offrent donc une productivité élevée ainsi qu'une grande aptitude à l'évolution.

Tableau 2.3 – Problèmes de planification du processus et leurs approches de résolution

Références	Objectif(s)	Approche(s)
Benderbal et al. [2017a]	↓ Temps de traitement ↑ Flexibilité du système	NSGA-II
Benderbal et al. [2017b]	↓ Temps de traitement ↑ Modularité du système	AMOSA
Bensmaine et al. [2013]	↓ Temps de traitement	NSGA-II
Bensmaine et al. [2014]	↓ Temps d'achèvement	Heuristique
Chaube et al. [2010]	↓ Coût total ↓ Temps de traitement	NSGA-II
Maniraj et al. [2014, 2015]	↓ Coût total	ACF
Musharavati and Hamouda [2012a]	↓ Coût total	SA
Musharavati and Hamouda [2012b]	↓ Coût total ↑ Productivité	SA
Shabaka and ElMaraghy [2008]	↓ Coût total	GA
Touzout and Benyoucef [2018]	↓ Coût total ↓ Temps de traitement ↓ Consommation énergétique	MOPLNE AMOSA NSGA-II
Touzout and Benyoucef [2019]	↓ Coût total ↓ Temps de traitement ↓ Temps d'exploitation des machines	ILS NSGA-II
Xie et al. [2012]	↓ Coût total ↑ Précision des machines ↑ Fluidité de la reconfiguration	GA

Ici, ↑ (resp. ↓) indique un objectif à maximiser (resp. minimiser)

b. Conception d'une configuration

La sélection d'une configuration pour une ligne de production reconfigurable est l'un des problèmes les plus étudiés dans la littérature. Il consiste à déterminer le nombre de stations, le nombre de machines par station et l'affectation des tâches de production. Chaque station possède un ou plusieurs RMT identiques parallèles pouvant avoir différentes configurations. Par exemple, dans Dou et al. [2008, 2009b,c], les auteurs visent à trouver une configuration en considérant qu'un seul type de produit (ligne mono-produit). L'objectif est de minimiser le coût d'investissement tout en garantissant que la demande sur une période donnée est respectée. Une extension de cette étude a été publiée par Dou et al. [2009a] où le même problème est traité en considérant plusieurs produits. Dans Dou et al. [2016], les auteurs ont intégré la sélection de la configuration et l'ordonnancement. En plus du coût total, le coût de reconfiguration et le retard total ont été minimisés. La conception d'une ligne mono-produit reconfigurable a été aussi envisagée par Goyal et al. [2012], où d'une part la reconfigurabilité et la capacité opérationnelle

des RMT sont maximisées, et où d'un autre part les coûts d'exploitation sont minimisés. En plus des objectifs précédents, Goyal and Jain [2015] ont considéré la maximisation du taux d'utilisation des machines et la convertibilité de la configuration.

Les articles précédents se sont concentrés sur les aspects liés à la reconfigurabilité des machines. Cependant, d'autres auteurs ont considéré le caractère évolutif où des fluctuations de la demande de produits, sur différentes périodes, sont prises en compte lors de la conception d'une ligne reconfigurable. L'objectif est d'identifier quand et comment reconfigurer le système afin de répondre au mieux aux besoins et fluctuations de la demande. Une reconfiguration dans ce cas là implique généralement l'ajout, le déplacement ou la suppression de RMT dans différentes stations. Dans ce contexte, Carlo et al. [2012] a proposé une approche pour la conception d'un RFL tout en considérant une politique de commande pour atteindre la scalabilité requise. Les objectifs consistent à minimiser les coûts d'investissement et ceux liés à l'inventaire. Par ailleurs, un outil d'aide à la décision basé sur l'AG a été développé par Deif and ElMaraghy [2006b] pour aider les concepteurs à décider quand et comment reconfigurer le système afin de satisfaire les exigences de la demande. L'un des objectifs consiste à minimiser les coûts liés à la reconfiguration. Le tableau 2.4 résume les articles susmentionnés, leurs fonctions objectifs et les approches de résolution respectives.

c. Conception de CMS reconfigurables

Selon Ghotboddini et al. [2011], un CMS est conçu en quatre étapes : (1) la formation des cellules, qui consiste à regrouper les pièces similaires en familles de pièces et à affecter les machines aux cellules ; (2) la sélection de la disposition des machines à l'intérieur des cellules ; (3) l'ordonnancement des pièces ; et (4) l'affectation des ressources.

Dans un problème de conception de CMS reconfigurable, RCMS (*Reconfigurable Cellular Manufacturing System*, en anglais), les machines RMT et CNC sont regroupées en cellules d'usinage en fonction des similitudes des tâches de fabrication des pièces. L'objectif d'un tel système est de fournir une indépendance aux cellules tout en assurant une flexibilité souhaitée pour s'adapter aux changements de la demande sur différentes périodes. Le tableau 2.5 présente les articles pertinents sur l'optimisation des RCMS, avec les objectifs et les approches de résolution correspondants. De ce fait, Renna and Ambrico [2014] ont proposé trois modèles mathématiques. Le premier modèle est utilisé pour concevoir un RCMS tout en considérant que la demande d'un produit, sur une période donnée, comme un paramètre stochastique. L'objectif ici est la minimisation des mouvements inter-cellules et des coûts d'investissement des machines. Le deuxième modèle multi-objectif est utilisé pour proposer une stratégie de reconfiguration entre les périodes de manière à maximiser le profit et à satisfaire les demandes. Finalement, le troisième modèle est utilisé pour ordonnancer les pièces dans chaque cellule de manière à maximiser le profit. Les

Tableau 2.4 – Problèmes de conception de lignes reconfigurables et leurs approches de résolution

Références	Objectif(s)	Approche(s)
Ashraf and Hasan [2018]	↓ Coût total ↑ Reconfiguration ↑ Capacité opérationnelle ↑ Fiabilité	NSGA-II
Bryan et al. [2013]	↓ Coûts du cycle de vie	DP & AG
Carlo et al. [2012]	↓ Coût d'investissement ↓ Coût des stocks	PNLNM AG
Deif and ElMaraghy [2006a,b]	↓ Coût de production ↓ Coût de reconfiguration	AG DP
Dou et al. [2008, 2009a,b,c]	↓ Coût d'investissement	AG
Dou et al. [2016]	↓ Coût d'investissement ↓ Coût de reconfiguration ↓ Retard total	NSGA-II
Goyal et al. [2012]	↑ Machine Reconfiguration ↑ Capacité opérationnelle ↓ Coût d'exploitation	NSGA-II
Goyal and Jain [2015]	↓ Coût d'investissement ↑ Capacité opérationnelle ↑ Utilisation des machines ↑ Convertibilité de la configuration	MOPSO
Kumar et al. [2018]	↓ Coût de reconfiguration ↓ Retards ↑ Équilibre de la charge de travail	MOSOMA
Moghaddam et al. [2017] Moghaddam et al. [2020]	↓ Coût d'investissement ↓ Coût de reconfiguration	PLNE PLNM
Saxena and Jain [2012]	↓ Coût d'investissement ↓ Coût de reconfiguration ↓ Coût d'exploitation ↓ Coût de la maintenance	AIS
Spicer and Carlo [2007]	↓ Coût d'investissement ↓ Coût de reconfiguration	DP PLNE
Youssef and ElMaraghy [2006]	↓ Coût d'investissement	AG
Youssef and ElMaraghy [2007, 2008b]	↓ Coût d'investissement ↑ Disponibilité du système	AG RT

Ici, ↑ (resp. ↓) indique un objectif à maximiser (resp. minimiser)

trois modèles sont résolus séparément et le résultat d'un modèle constitue une donnée d'entrée au modèle suivant. D'autres travaux ont été menés dans ce contexte [Pattanaik et al., 2006, Aljuneidi and Bulgak, 2016, Eguia et al., 2016].

Tableau 2.5 – Problèmes de conception de RCMS et leurs approches de résolution

Références	Objectif(s)	Approche(s)
Aljuneidi and Bulgak [2016, 2017]	↓ Coûts liés aux machines ↓ Coût de production	PLNM
Eguia et al. [2016]	↓ Mouvements inter-cellules ↓ Vide intra-cellulaire ↓ Coût total	PLNE PLNM
Pattanaik et al. [2006]	↓ Mouvements inter-cellules ↓ Changements des modules auxiliaires	AG
Yu et al. [2012]	↑ Équilibre de la charge de travail	AI & PLNE

Ici, ↑ (resp. ↓) indique un objectif à maximiser (resp. minimiser) et AI est pour Algorithme Itératif

Comparé aux RCMS, un CMS dynamique n'est pas composé de machines RMT et CNC. Cependant, la reconfigurabilité est assurée par des machines mobiles qui peuvent être facilement déplacées entre différentes cellules afin de faire face à la diversité de la gamme de produits et à la variation de la demande. L'objectif dans ce cas consiste à affecter ces machines aux cellules et de proposer une stratégie de reconfiguration entre les différentes périodes. Par exemple, dans Tavakkoli-Moghaddam et al. [2010], Ghotboddini et al. [2011] et Rabbani et al. [2014], les auteurs ont minimisé différents coûts tels que ceux engendrés par les déplacements inter/intra-cellulaires des produits, le déplacement des machines et les coûts d'achat des machines. En outre, Rabbani et al. [2014] a maximisé le taux d'utilisation des machines. Ghotboddini et al. [2011] a pris en compte la présence d'opérateurs et l'objectif est de trouver les meilleures affectations pour toutes les cellules dans chaque période.

d. Conception de tables d'usinage rotatives reconfigurables

Peu d'études ont été menées sur la conception de systèmes d'usinage à table rotative dans un environnement reconfigurable. Battaïa et al. [2016] et Liu et al. [2018] ont tenté de combler cette faille en considérant des positions de travail reconfigurables capables de produire plusieurs types de pièces. Comme le montre le tableau 2.6, ces deux articles proposent un outil d'aide à la décision (basé sur des méthodes d'optimisation) dans le but de concevoir de tels systèmes tout en minimisant le coût total (coûts liés aux tourelles, aux broches et aux modules d'usinage). Typiquement, ce problème consiste à affecter un ensemble de tâches aux modules d'usinage et à allouer les ressources appropriées (broches et tourelles) pour les réaliser. De plus, Liu et al. [2018] propose une approche durable en incluant le coût de la consommation d'énergie des machines pendant leur phase de fonctionnement.

Tableau 2.6 – Problèmes de conception de tables d'usinage rotatives reconfigurables et leurs approches de résolution

Références	Objectif(s)	Approche(s)
Battaïa et al. [2016]	↓ Coût total	PLNM
Liu et al. [2018]	↓ Coût total	PLNM
	↓ Temps de cycle	MOSA
Ici, ↑ (resp. ↓) indique un objectif à maximiser (resp. minimiser)		

2.4.2 La planification et l'ordonnancement dans un RMS

La Planification et l'Ordonnancement de la Production (POP) ont pour but de maximiser l'efficacité d'un système de production. Dans cette étape, les quantités de produits à fabriquer, les tailles des lots et l'ordre de traitement des produits sont définis.

Dans un RMS, l'ordonnancement est souvent associé au problème de la conception de la configuration et de la planification du processus. La raison en est que chaque produit nécessite une configuration particulière des machines. De ce fait, le résultat comprend la séquence des produits à lancer, la sélection des machines et leurs configurations appropriées, les affectations des produits et le séquençage des machines. Le tableau 2.7 présente les articles pertinents traitant de l'ordonnancement dans les RMS. Dans Bensmaine et al. [2014], les auteurs ont intégré le problème de planification du processus au problème d'ordonnancement afin de trouver la meilleure séquence de fabrication des pièces tout en considérant les différentes configurations des RMT et leur disponibilité. L'objectif consiste à minimiser le makespan, qui comprend le temps de traitement des tâches, le temps de transport entre les machines et le temps de reconfiguration de ces dernières. De la même façon, Dou et al. [2016] a considéré les deux problèmes de génération de configuration et d'ordonnancement dans une ligne reconfigurable. Les objectifs comprennent la minimisation du coût total et du temps de retard.

Pour la planification de la production, Abbasi and Houshmand [2009, 2010] ont respectivement proposé une formulation mathématique et un algorithme génétique. Dans les deux articles, le but consiste à définir, entre autres, la taille des lots et leurs séquences de passage. Choi and Xirouchakis [2014] se sont concentrés sur un problème de planification de la production, visant à trouver la quantité optimale de production de chaque produit tout en considérant différents plans de processus. Les auteurs ont étudié la consommation énergétique du système, les impacts environnementaux et le débit. Le tableau 2.8 résume les articles susmentionnés, en soulignant les objectifs abordés et les approches de résolution utilisées.

Tableau 2.7 – Problèmes d'ordonnancement et leurs approches de résolution

Références	Objectif(s)	Approche(s)
Bensmaine et al. [2014]	↓ Temps d'achèvement	Heuristique
Dou et al. [2016]	↓ Coût total ↓ Somme des retards	NSGA-II
Ye and Liang [2006]	↓ Coût total	AG
Yu et al. [2013]	↓ Temps d'achèvement ↓ Temps de séjour moyen ↓ Retard moyen	Heuristique

Ici, ↑ (resp. ↓) indique un objectif à maximiser (resp. minimiser)

Tableau 2.8 – Problèmes de planification et leurs approches de résolution

Références	Objectif(s)	Approche(s)
Abbasi and Houshmand [2009, 2010]	↑ Profit	AG & RT PNLNE
Choi and Xirouchakis [2014]	↓ Consommation énergétique ↑ Productivité	PL

Ici, ↑ (resp. ↓) indique un objectif à maximiser (resp. minimiser)

2.4.3 Problème d'agencement des machines

Un problème de conception d'agencement, généralement appelé problème d'agencement d'espace (PAE), consiste à trouver une disposition efficace des ressources dans un atelier de production en tenant compte de différentes contraintes. Les principaux objectifs sont de minimiser le coût et le temps associés à la manutention, qui ont un impact significatif sur la productivité. De plus, une telle décision est cruciale puisque le coût de la manutention représente 20 à 50% du coût total, et qu'une disposition adéquate des machines peut réduire ce dernier d'au moins 10 à 30% [Hosseini-Nasab et al., 2017]. En raison de l'importance de cette question, de nombreuses études ont été menées dans la littérature.

À partir de la revue de littérature de Hosseini-Nasab et al. [2017], on constate que la plupart des articles visent à trouver un agencement statique des ressources. Ce dernier tire sa définition du fait que le flux de produits ne change pas dans le temps. En d'autres termes, il peut être considéré dans un environnement de production dédié, ce qui n'est évidemment pas une solution efficace pour les conditions actuelles du marché. Par conséquent, on a assisté durant ces deux dernières décennies à de nombreux travaux sur la conception d'un agencement dynamique, où le flux peut changer au cours des horizons de planification [Hosseini-Nasab et al., 2017]. Même si ces travaux n'ont pas été effectués dans le contexte des RMS, ceci peut constituer une piste prometteuse à explorer. Dans la littérature lié aux RMS, seul Guan et al. [2012] a considéré

un agencement dynamique dans un environnement composé de postes de travail et de véhicules auto-guidés (VAG) pour déplacer les produits d'un poste vers un autres. Dans ce cas, une reconfiguration consiste à changer le routage des VAG lorsque le produit change. L'objectif est de minimiser les coûts de manutention et de reconfiguration. Le tableau 2.9 montre les objectifs et les approches de résolution utilisés dans les articles traitant des problèmes d'agencement dans un environnement reconfigurable.

Tableau 2.9 – Problèmes d'agencement des machines et leurs approches de résolution

Références	Objectif(s)	Approche(s)
Guan et al. [2012]	↓ Coût de manutention ↓ Coût de reconfiguration	REM
Haddou Benderbal and Benyoucef [2019]	↓ Pénalité	AMOSA

Ici, ↑ (resp. ↓) indique un objectif à maximiser (resp. minimiser)

2.4.4 Équilibrage et rééquilibrage de lignes reconfigurables

Le problème d'équilibrage se pose généralement sur les lignes d'assemblage, de désassemblage et de transfert. Ces lignes sont composées d'un ensemble de postes de travail disposés généralement de façon linéaire où chaque poste de travail peut exécuter un sous-ensemble d'opérations. Ces dernières sont caractérisées par leur temps de traitement, et une séquence qui est généralement exprimée par un graphe de précédence. Les produits passent par les postes de travail d'une manière synchronisée définie par le temps de cycle. Deux objectifs principaux sont considérés pour ce type de problèmes : i) minimiser le temps de cycle, c-à-d maximiser la productivité, et ii) minimiser le nombre de postes de travail. Une analyse plus détaillée du problème en question sera donné dans le chapitre suivant.

En ce qui concerne ce problème dans un environnement reconfigurable, Borisovsky et al. [2013a,b] ont étudié une ligne d'usinage où chaque poste de travail peut être équipé d'une ou plusieurs machines CNC disposées en parallèle. Ces machines peuvent être réaffectées, permettant la reconfiguration de la ligne afin de répondre aux nouvelles exigences de la demande. Cependant, les auteurs n'ont considéré que l'équilibrage initial de la ligne et non l'équilibrage de la ligne reconfigurée. L'objectif était de minimiser le nombre total de machines CNC tout en satisfaisant certaines contraintes telles que le temps de cycle, la relation de précédence entre les tâches, l'accessibilité et les contraintes d'inclusion et d'exclusion. Les approches de résolution qu'ils ont utilisées sont présentées dans le tableau 2.10.

Tableau 2.10 – Problèmes d'équilibrage de lignes reconfigurables et leurs approches de résolution

Références	Objectif(s)	Approche(s)
Borisovsky et al. [2013b,a]	↓ Nombre de machines CNC	CG & DP & AG
Ici, ↑ (resp. ↓) indique un objectif à maximiser (resp. minimiser)		

2.5 Conclusion

Une revue de la littérature sur les problèmes d'optimisation liés aux RMS a été présentée dans ce chapitre. Tout d'abord, différents indicateurs de performance d'un RMS ont été définis selon deux niveaux : le niveau machine et le niveau système. Par la suite, les problèmes d'optimisation ont été classés en se basant sur les articles de journaux pertinents. Nous avons montré comment les indicateurs de performance mentionnés précédemment étaient utilisés pour atteindre différents objectifs. En outre, les approches d'optimisation ont été brièvement décrites.

Sur la base de notre étude, nous observons que la conception des RMS a reçu plus d'attention comparée aux autres problématiques. En effet, la littérature sur la planification des processus et la conception de lignes reconfigurables est plus riche et plus diversifiée que celle sur l'agencement des ressources, la planification de la production, l'ordonnancement et l'équilibrage de lignes. En outre, plus de 70% des articles ont considéré ces problèmes en utilisant des méthodes approchées. En ce qui concerne la fonction objectif, la plupart est associée aux coûts et au temps. De plus, la reconfigurabilité et la scalabilité sont les caractéristiques qui ont été le plus considérées. Finalement, la majorité des problèmes abordés sont liés à l'usinage de pièces mécaniques, tandis que d'autres secteurs industriels ont reçu moins d'intérêt.

L'état de l'art présenté dans ce chapitre a été présenté à la conférence MIM 2019 [Brahimi et al., 2019] et a été publié à la revue « *International Journal of Production Research* » [Yelles-Chaouche et al., 2021d].

Chapitre 3

Équilibrage de lignes de production mono et multi-produits

Ce chapitre a pour but d'introduire le problème d'équilibrage de lignes d'assemblage. Pour ce faire, certaines notions importantes de ce problème seront d'abord introduites. Ensuite, les différents types de ce problème seront décrits et analysés.

3.1 Conception de lignes d'assemblage

Dans une ligne d'assemblage, le produit est fabriqué suivant un acheminement rigide. Plus particulièrement, ces lignes sont principalement composées d'un ensemble de postes de travail, aussi appelées stations. Celles-ci sont disposées le long d'un système de manutention tel qu'un convoyeur. Les produits sont lancés consécutivement sur le convoyeur qui les déplace, de façon régulière et cadencée, d'une station vers une autre. Dans chaque station, le produit subit une transformation à valeur ajoutée jusqu'à obtenir, à la sortie de la dernière station, le produit final. Typiquement, ces transformations sont principalement effectuées par des opérateurs et/ou machines présents au niveau des stations. En effet, chaque opérateur/machine doit exécuter une ou plusieurs tâches, préalablement définies, sur le produit. Le délai que nécessite chaque tâche est appelé « temps de traitement ». Dans une ligne d'assemblage cadencée, il est nécessaire que toutes les tâches affectées à une station soient effectuées avant de passer à la station suivante. Par conséquent, le temps que passe un produit au niveau de chaque station est limité. Cette limite, appelée « temps de cycle », dicte la cadence de la ligne. Ainsi, le taux de production (nombre de pièces produites par unité de temps) de ce type de lignes est fixe.

Concevoir une ligne d'assemblage est un processus très complexe qui nécessite de prendre un certain nombre de décisions importantes. Traditionnellement, et en absence d'une méthodologie bien définie, ce processus demandait du temps et manquait d'efficacité. Il consistait, selon Hopp and Spearman [2000], à proposer une conception préliminaire qui vise à :

1. Établir la taille et la forme de l'usine,
2. Déterminer l'emplacement des stations,

3. Définir le flux du produit.

L'objectif principal de ce processus est de minimiser les coûts d'investissement. Par la suite, la ligne conçue était évaluée et adaptée de façon répétitive jusqu'à atteindre des performances acceptables. En effet, on peut remarquer que ce processus donne peu de considération au flux du produit jusqu'à ce que la majeure partie de l'usine ait été conçue. Or, le principal objectif d'une ligne est de fournir un produit de qualité rapidement et de manière compétitive. Par conséquent, un processus de conception conforme à cet objectif peut être atteint en inversant l'approche traditionnelle [Hopp and Spearman, 2000], c'est-à-dire :

1. Le client détermine le produit,
2. Le produit détermine le processus,
3. Le processus détermine l'ensemble de stations,
4. Les stations définissent les installations nécessaires,
5. Les installations définissent la structure globale et la taille de l'usine.

Les deux premières étapes sont primordiales, elles fournissent les informations sur le travail qui doit être effectué sur la ligne d'assemblage tel que les gammes de produit, les volumes à produire, les temps de cycle ainsi que des éventuelles contraintes. Ces dernières peuvent être liées, par exemple, à des facteurs technologiques, économiques ou environnementaux. La troisième étape consiste à définir le schéma de la ligne (ligne droite, en U, avec transfert circulaire, etc.). Ceci permet de proposer un agencement approprié des stations et la direction du flux des produits à fabriquer. La quatrième étape est pertinente, car elle a un impact direct sur les performances de la ligne conçue. En effet, dans cette étape, les tâches de production sont affectées aux stations et aux ressources qui composent la ligne. Une fois la ligne conçue, la dernière étape est presque intuitive du fait que les décideurs disposent des dimensions et de la grandeur de la ligne. Dans la littérature, ces décisions sont considérées séparément. Ceci est principalement dû au fait que la quantité d'information et de données à traiter à chaque niveau rend le processus décisionnel très complexe. Pour cela, la prise de ces décisions se base sur des approches scientifiques très poussées. En effet, on peut constater à travers la littérature que la majorité de ces décisions sont formulées comme un problème d'optimisation. Par conséquent, l'utilisation des méthodes de la recherche opérationnelle et de la simulation sont nécessaires afin de résoudre efficacement ces problèmes.

Dans le cadre de cette thèse, nous nous intéressons particulièrement au problème d'optimisation qui se pose dans l'étape 4. Cette étape vise à affecter les opérations/tâches aux différentes stations. Dans la littérature, ce problème est appelé problème d'équilibrage de lignes d'assemblage ou ALBP (*Assembly Line Balancing Problem*, en anglais).

3.2 Problème d'équilibrage de lignes d'assemblage

En 1955, Salveson [1955] a proposé la première modélisation mathématique du ALBP. Comme mentionné précédemment, le ALBP consiste à trouver la meilleure répartition des tâches dans les stations, dans le but d'atteindre une productivité optimale. Plus particulièrement, étant donné un ensemble de tâches V, numérotées de 1 à $|V|$, nécessaires pour fabriquer un produit, et un ensemble de stations $W = \{1, 2, \ldots, |W|\}$, l'équilibrage consiste à affecter toutes les tâches à ces stations. Naturellement, ces tâches doivent être effectuées dans un ordre partiel défini au préalable. Par exemple, dans une ligne d'assemblage de voitures, le volant ne peut pas être monté si le tableau de bord n'est pas encore mis en place. De ce fait, l'utilisation d'une représentation sous forme de graphe est principalement utilisée pour décrire cet ordre. Ces graphes, communément appelés graphes de précédence, sont composés d'un ensemble de nœuds connectés par des arcs. Chaque nœud représente une tâche $i \in V$, et un arc symbolise une relation de précédence entre deux tâches. La figure 3.1 montre un exemple d'un graphe de précédence composé de 8 nœuds et 8 arcs. Dans ce graphe, on peut constater, par exemple, que la tâche numéro 2 ne peut

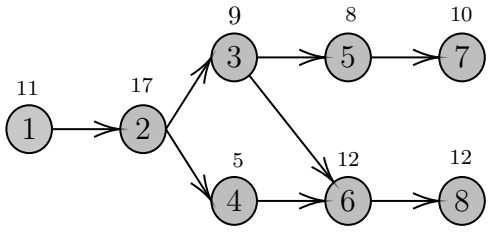


FIGURE 3.1 – Exemple d'un graphe de précedence

pas commencer avant la fin de l'exécution de la tâche numéro 1. De plus, il est important de rappeler que chaque tâche $i \in V$ nécessite un temps opératoire t_i pour être effectuée. Ce dernier est indiqué au dessus de chaque nœud dans la figure 3.1. Par conséquent, la charge de chaque station est calculée comme étant la somme du temps opératoire des tâches qui lui sont affectées. Cette charge ne doit pas dépasser le temps de cycle de la ligne. Par exemple, en considérant 3 stations, le graphe de précédence de la figure 3.1 et un temps de cycle de 40 unité de temps, un équilibrage admissible (c-à-d qui satisfait toutes les contraintes) est montré à la figure 3.2.

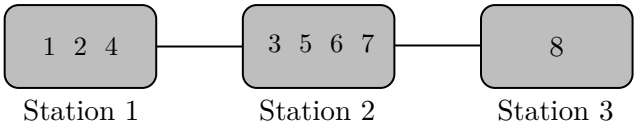


FIGURE 3.2 – Exemple d'une ligne d'assemblage

3.2.1 Classification du ALBP

Les lignes d'assemblage sont présentes dans plusieurs secteurs industriels en raison de leur capacité à fabriquer une large quantité de produits efficacement, rapidement et à moindre coût [Boysen et al., 2008]. En effet, les différents produits qui sont actuellement traités par ces lignes sont divers et variés. On peut citer comme exemple les lignes d'assemblage de voitures, moteurs, ordinateurs, téléphones portables, télévisions, etc. Ainsi, pour concevoir une ligne, il est important de prendre en considération les caractéristiques intrinsèques du produit en question. De ce fait, il existe plusieurs variantes du ALBP dont chacune est spécifique à un contexte industriel particulier. La figure 3.3, proposée par Scholl [1999], montre une classification des problèmes du ALBP sur la base de différentes caractéristiques comme par exemple le fonctionnement de la ligne (cadencée ou non), le temps de traitement des tâches (déterministe ou stochastique) et la forme de la ligne (ligne droite, en U ou transfert circulaire). Ainsi, sur cette figure, on peut distinguer trois types de ALBP :

1. **Le modèle simple** : un seul produit est fabriqué sur la ligne.
2. **Le modèle mixte** : Différents produits (plus ou moins similaires) sont produits simultanément sur la ligne.
3. **Le modèle multiple** : Différents produits sont fabriqués en lots séparés.

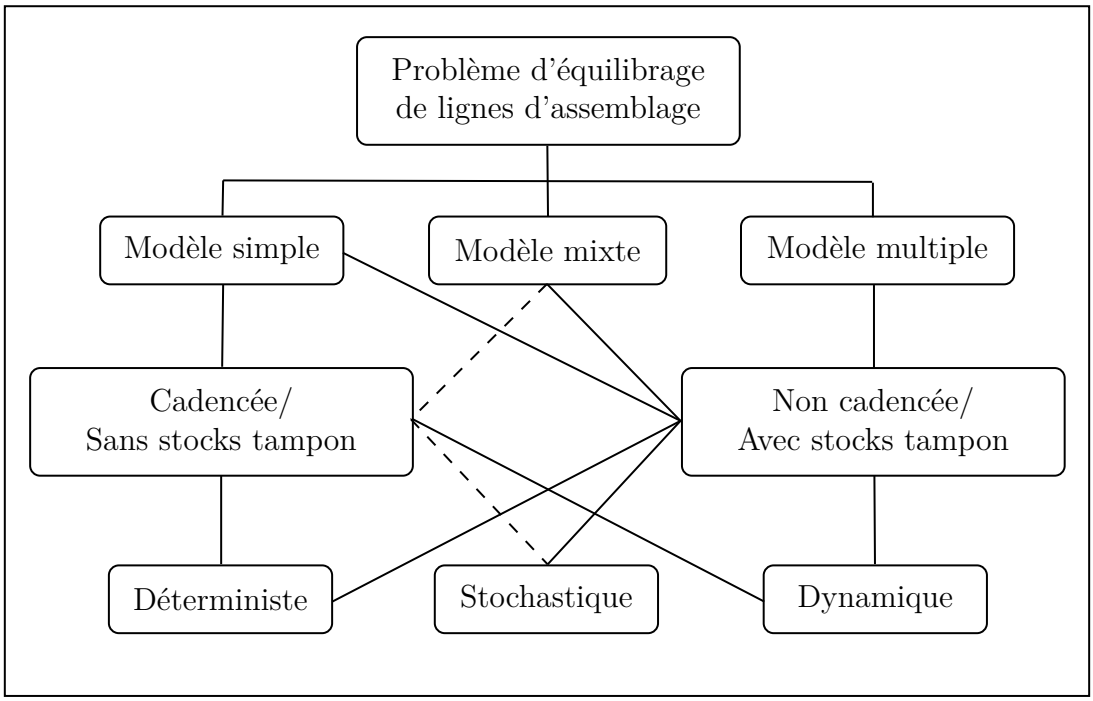


FIGURE 3.3 – Classification des problèmes d'équilibrage de lignes d'assemblage

La figure 3.4 illustre les trois différents types de lignes en fonction du flux de produits. Dans cette figure, une ligne à modèle unique, associée au modèle simple du ALBP, peut traiter un seul type de produit, représenté par une forme losange. Dans une ligne à modèle mixte, différents produits ayant une forme losange, ronde et triangulaire passent sur la ligne simultanément. Enfin, une ligne à modèle multiple est capable de traiter plusieurs produits en lots séparés. Ainsi, une reconfiguration, notée par « R », est nécessaire pour passer d'un produit vers un autre. Une description détaillée du problème d'équilibrage dans chacune de ces lignes sera donnée dans les sections suivantes.

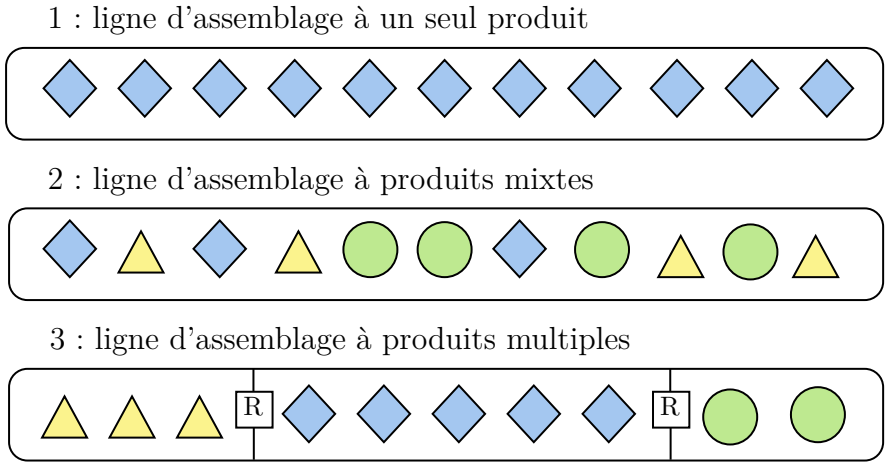


FIGURE 3.4 – Trois types de lignes d'assemblage pour un ou plusieurs produits

Le ALBP a été largement traité dans la littérature. Ceci est reflété à travers la multitude d'états de l'art tels que ceux de Boysen et al. [2008], Battaïa and Dolgui [2013] ou Sivasankaran and Shahabudeen [2014]. Du fait qu'il existe un grand nombre de modèles et de méthodologies, nous excluons de ce manuscrit les lignes non-cadencées et les modèles stochastiques. Nous nous focalisons seulement sur les approches considérant les lignes (droites) cadencées avec des temps de traitement déterministes. Par conséquent, nous allons considérer tout au long de ce manuscrit les propriétés suivantes :

— Toutes les données du problème sont connues,
— Une tâche doit être effectuée sur une seule station,
— Toutes les tâches doivent être effectuées sur la ligne,
— Le temps de cycle et les relations de précédence entre les tâches doivent être respectées.

3.3 Problème d'équilibrage de lignes d'assemblage simples

Le problème d'équilibrage de lignes d'assemblage simples ou SALBP (*Simple Assembly Line Balancing Problem*, en anglais), est le problème d'optimisation de base du ALBP. Il consiste à concevoir une ligne dédiée à un seul produit. Plusieurs types du SALBP existent, dont chacun vise à atteindre un objectif bien précis. Selon Scholl [1999], les principaux types du SALBP sont les suivants :

- **Problème de faisabilité (SALBP-F)** : Il s'agit d'un problème de décision qui consiste à déterminer s'il existe ou non une affectation des tâches qui soit réalisable pour un nombre de stations, un graphe de précédence et un temps de cycle donnés.
- **Minimisation du nombre de stations (SALBP-1)** : cette version du SALBP est la plus connue et la plus traitée dans la littérature. Elle consiste à minimiser le nombre de stations sur la base d'un temps de cycle prédéfini [Baybars, 1986].
- **Minimisation du temps de cycle (SALBP-2)** : contrairement au SABLP-1, le SALBP-2 vise à minimiser le temps de cycle, en affectant toutes les tâches sur un nombre de stations donné et fixe.
- **Maximisation de l'efficacité de la ligne (SALBP-E)** : cette version du SALBP est une combinaison du SALBP-1 et SALBP-2, c'est-à-dire qu'elle considère à la fois le temps de cycle et le nombre de stations utilisées comme des variables de décision. Par conséquent, résoudre ce problème consiste à trouver un équilibrage de la ligne qui maximisera son efficacité, exprimée comme le produit de ces deux derniers paramètres.

Tous ces problèmes sont connus pour être NP-difficiles [Scholl, 1999]. En effet, le SALBP-F se réduit au problème de Partition qui consiste à décider si oui ou non un ensemble d'entiers positifs peut être subdivisé en deux sous-ensembles ayant la même somme. De la même façon, le SALBP-1 est une variante du problème de Bin-Packing [Wee and Magazine, 1982], et le SALBP-2 peut être réduit au problème d'ordonnancement de machines parallèles. Par conséquent, la résolution de ces problèmes est très difficile et nécessite des approches d'optimisation complexes. Dans la section suivante, nous décrivons brièvement les différentes approches utilisées dans la littérature pour résoudre le SALBP.

3.3.1 Approches de résolution pour le SALBP

Dans la littérature, deux types de méthodes sont utilisés pour résoudre le SALBP, à savoir les méthodes exactes et les méthodes approchées. Les méthodes exactes visent à trouver une solution optimale au problème alors que les secondes consistent souvent à trouver une solution réalisable de bonne qualité, c'est-à-dire la plus proche possible de l'optimum. Les deux méthodes sont complémentaires. En effet, SALBP est un problème NP-difficile, ce qui implique que le temps

de calcul nécessaire pour obtenir une solution optimale augmente de manière exponentielle avec la taille du problème. Dans ce cas, l'utilisation des méthodes approchées s'avère nécessaire, car elle permet d'obtenir une solution de bonne qualité en un temps relativement court. Cela dit, le développement de méthodes exactes est une étape très importante qui permet de mieux évaluer les performances des méthodes approchées.

Méthodes exactes

Le SALBP est largement traité dans la littérature donnant ainsi lieu à une multitude d'approches de résolution exactes. Salveson [1955] a développé le premier PLNE pour résoudre le SALBP-1 classique. Sur cette base, d'autres PLNE ont pu être proposés en considérant de nouvelles exigences opérationnelles liées à différents secteurs industriels [Gao et al., 2009, Bautista and Pereira, 2011]. Ces modèles mathématiques sont donc principalement résolus à l'aide de solveurs commerciaux qui utilisent la méthode de *Branch-and-cut*. Par ailleurs, des approches basées sur l'algorithme du *Branch-and-cut* ont été proposées pour résoudre le SALBP telles que FABLE [Johnson, 1988], EUREKA [Hoffmann, 1992] ou SALOME [Scholl and Klein, 1997]. Des auteurs comme Bard [1989] et Topaloglu et al. [2012] ont proposé des méthodes de résolution basées sur la programmation dynamique et sur la programmation par contraintes.

Méthodes approchées

Comme on a pu le voir dans la section précédente, de nombreuses méthodes exactes ont été développées dans la littérature du SALBP. Cependant, elles atteignent leurs limites face à des instances de grande taille. Des méthodes de résolution approchées sont alors plus adéquates. En effet, plusieurs heuristiques et métaheuristiques ont été développées dans la littératures. Ces méthodes ne garantissent pas une solution optimale, mais ont tendance à obtenir des solutions satisfaisantes en un temps relativement court comparé aux méthodes exactes. Cette section vise donc à présenter brièvement les plus connues.

Plusieurs heuristiques ont été adaptées au SALBP. Celles-ci se basent sur des règles simples qui, au fur et à mesure, vont permettre de construire des solutions réalisables. La majorité de ces méthodes ont été proposées pour le SALBP-1 et sont basées sur des règles de priorité. D'autres sont des procédures énumératives restreintes. Une des plus récentes études de ces approches est donnée par Scholl and Becker [2006].

Les procédures basées sur des règles de priorité utilisent des valeurs calculées pour les différentes tâches sur la base de leur temps opératoire, successeurs ou prédécesseurs. Par exemple, l'heuristique proposée par Helgeson and Birnie [1961] consiste à calculer pour chaque tâche une valeur qui se base sur la somme de son temps opératoire et celui de ses successeurs. Ensuite, toutes ces valeurs sont triées par ordre décroissant et, en suivant cet ordre, chaque tâche est

affectée à une station si elle n'a pas de prédécesseurs ou si tous ses prédécesseurs sont déjà affectés. Lorsqu'une station est pleine, c'est-à-dire qu'aucune tâche ne peut être affectée sans dépasser le temps de cycle, une nouvelle station est ouverte. L'algorithme s'arrête quand toutes les tâches sont affectées. Naturellement, l'objectif est de minimiser le nombre de stations. De la même façon, d'autres heuristiques ont été développées pour résoudre les différentes variantes du SALBP (voir Scholl and Becker [2006] et Battaïa and Dolgui [2013] pour plus de détails).

Quant aux métaheuristiques, les plus utilisées pour ce problème, selon Battaïa and Dolgui [2013], sont l'algorithme génétique et le recuit simulé.

3.4 Problème de rééquilibrage de lignes d'assemblage

Compte tenu du progrès technologique et de la forte compétitivité qui caractérisent le marché actuel, il arrive souvent que le produit et/ou les volumes de production changent et que la main-d'œuvre disponible, les ressources ou les objectifs de production évoluent. Par conséquent, il est nécessaire qu'une ligne d'assemblage existante s'adapte afin de répondre au mieux à ces changements. Dans ce cas-là, la ligne d'assemblage en question doit être rééquilibrée. Dans la pratique, il n'est pas rare que cela se produise. En effet, selon [Falkenauer, 2005], une ligne d'assemblage n'est généralement pas conçue à partir de zéro, mais représente une reconfiguration d'une ligne déjà existante. Néanmoins, ce changement s'effectue de façon occasionnelle rendant ainsi la nouvelle configuration valable sur un horizon temporel relativement long.

De ce fait, le problème de rééquilibrage de lignes d'assemblage ou ALRBP (*Assembly Line Re-Balancing Problem*, en anglais) consiste à réaffecter certaines tâches entre les stations existantes. Ceci peut entraîner une relocalisation des ressources disponibles telles que les opérateurs et les machines. Naturellement, un des principaux objectifs du ALRBP est de minimiser le nombre de tâches réaffectées [Grangeon et al., 2011, Makssoud et al., 2015, Sancı and Azizoğlu, 2017], ou le coût associé au rééquilibrage [Yang et al., 2013, Makssoud et al., 2014]. La figure 3.5 montre un exemple de rééquilibrage d'une ligne d'assemblage, où les tâches 4, 5 et 7 ont dû être réaffectées.

Malgré l'importance de ce problème dans l'industrie, peu de recherches ont été effectuées sur le ALRBP. Grangeon et al. [2011] a traité l'ALRBP en considérant la minimisation du nombre de tâches réaffectées et du nombre de stations. Pour le même problème, Makssoud et al. [2014] a proposé un programme linéaire en nombres mixtes (PLNM) pour minimiser le coût d'investissement de nouveaux équipements en tenant compte de la possibilité de réutiliser ceux déjà disponibles. Sancı and Azizoğlu [2017] ont considéré un problème de rééquilibrage dans lequel une panne ou un arrêt de la station se produit. Dans ce cas, les tâches effectuées par celui-ci doivent être réaffectées aux stations disponibles. Les auteurs ont cherché à obtenir un

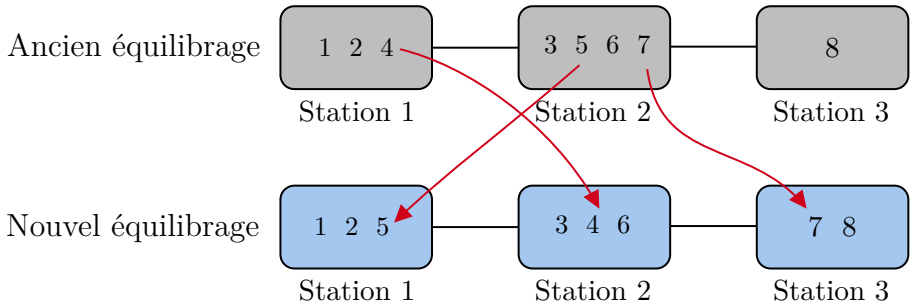


FIGURE 3.5 – Exemple d'un rééquilibrage d'une ligne d'assemblage

compromis entre le temps de cycle et le nombre de tâches réaffectées. Yang et al. [2013] a étudié une ligne d'assemblage à modèle mixte composée d'opérateurs tout en considérant les demandes saisonnières. Une telle ligne peut être rééquilibrée (reconfigurée) pour s'adapter aux variations de la demande. La fonction objectif minimise le nombre de stations et leurs variations de charge ainsi que le coût de rééquilibrage.

3.5 Problème d'équilibrage de lignes d'assemblage à modèles mixtes

L'apparition des systèmes flexibles et reconfigurables a permis de concevoir des lignes capables de fabriquer plusieurs produits. En effet, chaque modèle de produit présente des caractéristiques différentes comme par exemple la couleur, la taille, le matériau ou les équipements utilisés. Néanmoins, malgré les efforts considérables déployés pour rendre les systèmes de production plus polyvalents, il n'est toujours pas évident de considérer des produits qui présentent des différences significatives entre eux. Par conséquent, on suppose généralement que tous les modèles sont des variations d'un même produit de base et ne diffèrent que par des attributs spécifiques, également appelés options. Ceci se traduit par le fait qu'une grande partie des tâches à effectuer est commune entre les différents modèles, et que d'autres tâches supplémentaires sont par ailleurs spécifiques aux options correspondantes à un modèle en particulier. Par conséquent, des temps opératoires de certaines tâches peuvent varier d'un modèle à un autre. Dans l'industrie automobile, par exemple, l'installation d'un toit ouvrant électrique nécessite un temps différent de celui d'un toit ouvrant manuel.

Une ligne de production est dite mixte, si tous les modèles du produit sont fabriqués simultanément sur la ligne. Selon Scholl [1999], cette structure présente beaucoup d'avantages. Elle permet à la fois d'avoir un flux continu de produits tout en disposant d'un certain niveau de flexibilité qui lui permet de s'adapter facilement aux changements. De plus, dans ce type

de lignes, les temps de réglage entre les modèles sont suffisamment réduits pour être ignorés. Cependant, ces lignes nécessitent un coût d'investissement élevé, car elles sont principalement composées de machines flexibles.

De même que les lignes mono-produits décrites dans la section précédente, la conception de lignes de production mixtes nécessite de prendre les mêmes décisions. Ceci dit, dans la phase d'équilibrage d'une ligne mixte, il est nécessaire de prendre en considération les caractéristiques de chaque produit. Dans la littérature, ce problème est appelé problème d'équilibrage de lignes d'assemblage mixtes ou MiMALBP (*Mixed-Model Assembly Line Balancing Problem*, en anglais).

Le MiMALBP est basé sur les mêmes hypothèses que le SALBP, à savoir des temps opératoires déterministes, aucune restriction d'affectation, disposition de la ligne en série. Par ailleurs, les hypothèses suivantes doivent aussi être prises en compte pour prendre en considération les différents modèles à produire :

— L'assemblage de chaque modèle nécessite l'exécution d'un ensemble de tâches liées par des relations de précédence (un graphe de précedence pour chaque modèle).
— Un sous-ensemble de tâches est commun à plusieurs modèles.
— Le sous ensemble de tâches en commun doit être exécuté sur la même station, mais peut avoir des temps opératoires différents.
— Le temps total disponible pour la production pendant la période de planification est donné et fixe.
— Les demandes prévues pour tous les modèles pendant la période de planification sont connues.

Dans la littérature, le MiMALBP est souvent associé au problème de séquencement. Alors que l'équilibrage de la ligne décide de l'affectation des tâches aux stations et détermine ainsi leurs charges, le séquencement défini l'ordre de passage des différents modèles sur la ligne pour un horizon de temps court [Boysen et al., 2009a]. En effet, si plusieurs modèles qui nécessitent une forte intensité de travail se succèdent dans la même station, des surcharges de travail peuvent se produire. Celles-ci doivent donc être compensées par des ressources supplémentaires. Ainsi, un séquencement optimal des modèles permet, entre autres, d'éviter cette situation. Ceci dit, selon Boysen et al. [2008] cette approche simultanée n'est pas toujours viable car les deux problèmes de décision sont dans des horizons de temps différents.

Tout comme le SALBP, plusieurs objectifs sont considérés dans le MiMALBP comme par exemple la minimisation du nombre de stations (**MiMALBP-1**), du temps de cycle (**MiMALBP-2**) ou la maximisation de l'efficacité de la ligne (**MiMALBP-E**). Ces objectifs visent à fournir un équilibrage optimal de la ligne dans des conditions idéales et sans aléas. Or, lorsqu'il s'agit de plusieurs modèles de produit, ces hypothèses conduisent, selon Scholl [1999], à des solutions qui ne sont pas efficaces en termes de performances de la ligne. Pour cause, les variations des temps

opératoire entre les différents modèles peuvent engendrer des surcharges ou des temps morts au sein des stations. La première se produit lorsqu'une tâche affectée à une station n'est pas achevée avant l'écoulement du temps de cycle, tandis que la seconde survient quand la station reste inactive. Par conséquent, plusieurs auteurs ont considéré le lissage des temps associés aux stations [Decker, 1993]. Le terme lissage fait référence au fait que toutes les stations doivent avoir plus ou moins la même charge. Celui-ci permet à la fois de réduire le temps mort et la surcharge des stations [Boysen et al., 2009b].

En ce qui concerne les approches de résolution, Hashemi-Petroodi et al. [2021] ont fourni une revue de la littérature approfondie sur les stratégies de reconfiguration des opérateurs dans des lignes d'assemblage mixtes. Les auteurs ont remarqué, entre autres, que les méthodes approchées, telles que les heuristiques et méta-heuristiques, sont principalement utilisées pour résoudre le MiMALBP. Cela reflète la complexité de ces problèmes, qui se justifie par le fait que plusieurs produits doivent être considérés simultanément lors de l'équilibrage de la ligne.

3.6 Problème d'équilibrage de lignes d'assemblage à modèles multiples

En présence de plusieurs produits, il n'est pas toujours évident de concevoir une ligne mixte où les produits passent simultanément sur la ligne. En effet, il existe des situations où des différences significatives dans le processus de production font qu'il est impossible de concevoir une configuration commune à tous les produits. Par conséquent, des modifications de l'équipement de la ligne sont nécessaires en cas de changement de produit. Ceci implique que les produits doivent être fabriqués par lot afin de pouvoir minimiser les coûts et les temps associés aux changements. Ces lignes sont appelées lignes multi-produits et le problème d'équilibrage associé est connu comme le problème d'équilibrage de lignes d'assemblage multi-modèles ou MuMABLP (*Multi-Model Assembly Line Balancing Problem*, en anglais). À notre connaissance, le MuMALBP a été peu considéré dans la littérature, comparé aux SALBP et MiMALBP, malgré son importance actuelle dans l'industrie. Ceci peut être vu à travers la classification fournie par Boysen et al. [2007], où le nombre d'études qui traitent du MiMALBP est trois fois plus élevé que celles relatives au MuMALBP.

Dans la littérature, le MuMALB est souvent associé aux problèmes de dimensionnement des lots de produits et de séquencement. Le premier vise à trouver la quantité de produit, du même modèle, à produire, alors que le second consiste à identifier la meilleure séquence de passage des lots dans la ligne. Même si ces deux problèmes sont généralement traités séparément, ils sont complémentaires et ont pour objectif de minimiser le temps de réglage (*setup time*, en anglais) de la ligne lors du passage de la configuration d'un produit vers une autre. Ceci étant dit, ces deux

problèmes d'optimisation considèrent que la ligne est déjà conçue et que les temps de réglage sont donnés. Par ailleurs, lorsqu'il s'agit de concevoir une ligne d'assemblage multi-produits, les quelques études présentes dans la littérature le simplifient généralement pour le rendre plus facile à résoudre. Par exemple, certains articles traitant du MuMALBP le considèrent comme une version dérivée de MiMALBP ou tentent de le convertir en SALBP (voir par exemple Battaïa and Dolgui [2013]). Pour ce faire, une méthode couramment utilisée consiste à combiner les graphes de précédence et à adapter les temps opératoires des tâches de chaque produit [van Zante-de Fokkert and de Kok, 1997]. L'avantage de cette approche est qu'il n'est pas nécessaire de reconfigurer la ligne conçue. Il en résulte donc une seule configuration avec un compromis de performance entre les différents produits, ce qui peut diminuer son efficacité [Boysen et al., 2008]. En outre, la technique de combinaison des graphes de précédence ne fonctionne efficacement que lorsque les graphes sont similaires [Boysen et al., 2009c]. Une autre approche pour résoudre le MuMALBP consiste à le considérer soit *i)* comme un MiMALBP lorsque la taille du lot de produits est petite et que la reconfiguration peut être négligée ou *ii)* comme plusieurs SALBP lorsque la taille du lot de produits est plus grande [Becker and Scholl, 2006]. Cependant, lorsque l'objectif est de minimiser le nombre de réaffectations de tâches (ou les temps et coûts associés), ces approches peuvent conduire à des solutions de mauvaise qualité.

Le tableau 3.1 présente les publications les plus pertinentes sur le MuMALBP. La deuxième colonne du tableau indique le ou les objectifs à minimiser ou à maximiser ; la troisième colonne présente les différentes approches de résolution ; et enfin, la dernière indique si l'étude est liée à une application industrielle.

Tableau 3.1 – Résumé des publications portant sur le MuMALBP

Référence	Objectif	Approche de résolution	Application industrielle
Buxey et al. [1973]	Min. Coût de production	Simulation	×
Chakravarty and Shtub [1985]	Min. Coût total	Heuristique	×
Kabir and Tabucanon [1995]	Min. Nombre de stations	Simulation	calculatrices d'impression
van Zante-de Fokkert and de Kok [1997]	Min. Nombre de stations	Heuristique	×
Kimms [2000]	Min. Nombre de stations	Heuristique + génération de colonne	×
Lapierre et al. [2000]	Min. Temps de cycle	Relaxation langrangienne	carte de circuit imprimé
Pastor et al. [2002]	Max. Rendement de la ligne	Recherche tabou	appareils électroménagers
Yu and Shi [2013]	Min. Nombre de stations	Algorithme génétique	×
Liao [2014]	Min. Temps de cycle	PLNE	×
Qu and Jiang [2014]	Max. Efficacité de la ligne	Algorithme mémétique	construction navale
Christensen et al. [2017]	Min. Temps de cycle	Heuristique	×
Pereira [2018]	Min. Coût des stations	Heuristique	vêtements
Asl et al. [2019]	Min. Temps de cycle + autres	PLNE	fabrication de moteurs
Chen et al. [2019]	Min. Nombre de stations + ressources	Algorithme génétique + PLNE	écrans LCD
Koskinen et al. [2020]	Min. Retard	Heuristique	carte de circuit imprimé

La plupart des applications industrielles se situent dans le secteur de l'électronique. Kabir and Tabucanon [1995] considèrent l'assemblage de calculatrices imprimantes, qui sont encore couramment utilisées dans certaines applications. Lapierre et al. [2000] et Koskinen et al. [2020] étudient le problème du placement de composants sur des circuits imprimés (CI). Un tel problème est caractérisé par une variété de types de CI et de multiples machines de placement

pour les composants. Pastor et al. [2002] explorent un problème dans lequel 4 types de produits électroménagers sont produits. Plus particulièrement, ce problème traite de la fabrication d'écrans à cristaux liquides (écrans LCD). Chen et al. [2019] considèrent le même problème tout en intégrant des spécificités fonctionnelles telles que l'efficacité des travailleurs polyvalents et des opérateurs. Parmi les autres applications de MuMALBP, citons la construction navale [Qu and Jiang, 2014], l'industrie du textile [Pereira, 2018] et la fabrication des moteurs à combustion [Asl et al., 2019].

Comme l'indique le tableau 3.1, les études sur MuMALBP considèrent différents objectifs, mais la plupart d'entre eux tentent soit de minimiser le nombre de stations, soit le temps de cycle. La raison pour laquelle la plupart des chercheurs ont considéré ces objectifs est que ce sont les objectifs traditionnels dans la plupart des problèmes d'équilibrage de lignes. Les premières études de Buxey et al. [1973] et Chakravarty and Shtub [1985] ont pour objectif la minimisation du coût qui est composé, entre autres, de coûts de main-d'œuvre, de stockage et de réglage. Pastor et al. [2002] et Qu and Jiang [2014] tentent de maximiser le rendement et l'efficacité de la ligne de production, respectivement. Plus particulièrement, Pastor et al. [2002] considère la maximisation du rendement dans un modèle multi-objectif qui inclut également l'équilibrage de la charge de travail et la minimisation de la dispersion des tâches des travailleurs. Parmi les études ci-dessus, Asl et al. [2019] traitent un MuMALBP multi-objectif, où l'un des objectifs consiste à maximiser le nombre de tâches communes entre les stations, ce qui est similaire à l'un des objectifs principaux qui sera étudié dans ce mémoire.

Les problèmes d'équilibrage de lignes d'assemblage sont connus pour être NP-difficiles [Scholl, 1999]. Par conséquent, la plupart des méthodes de résolution proposées pour résoudre les différentes extensions de ces problèmes sont approchées. Parmi ces méthodes, on trouve des méta-heuristiques dont les algorithmes génétiques [Yu and Shi, 2013, Chen et al., 2019], la recherche tabou [Pastor et al., 2002], et les algorithmes mémétiques [Qu and Jiang, 2014]. Les heuristiques de construction simples ont également pris de l'importance dans la résolution de MuMALBP. Par exemple, Chakravarty and Shtub [1985] ont proposé une heuristique, basée sur la procédure de Helgeson and Birnie [1961], qui permet d'obtenir rapidement des solutions de bonne qualité. Lapierre et al. [2000] a appliqué la relaxation lagrangienne pour minimiser le temps de cycle des produits en fonction de leurs proportions dans la production. Koskinen et al. [2020] a proposé un modèle mathématique et une heuristique à deux phases pour minimiser la somme des temps de retard de chaque produit.

3.7 Conclusion

Dans ce chapitre, nous avons abordé les différents problèmes d'équilibrage de lignes d'assemblage. Pour ce faire, nous avons initialement défini et décrit le SALBP. Ensuite, nous nous sommes intéressés aux lignes multi-produits à travers les deux problèmes d'optimisation MiMALBP et MuMALBP. En analysant la littérature, nous avons pu remarqué que peu de travaux s'intéressent aux MuMALBP malgré son importance dans l'industrie. Ceci peut se traduire par le fait que la plupart des lignes conçues en industrie sont peu flexibles surtout lorsqu'il s'agit de production de masse. Ainsi, les industriels favorisent une seule configuration commune à tous les produits, plutôt que plusieurs configurations pour chaque produit, et ce, afin d'éviter des coûts et arrêts liés à la reconfiguration. Néanmoins, nous avons pu voir dans le premier chapitre que les RMS sont conçus pour être rapidement reconfigurés et à moindre coût.

Sur la base des observations décrites ci-dessus et des lacunes identifiées, nous considérons dans ce mémoire un MuMALBP générique dans un environnement reconfigurable. Deux objectifs principaux seront donc étudiés dans ce qui suit. Le premier vise à optimiser la réaffectation des tâches entre les stations lors du passage de la configuration d'un lot de produits à un autre. Le second traite des lignes modulaires multi-produits et consiste à minimiser le nombre total de modules utilisés pour fabriquer différents produits.

Chapitre 4

Minimisation du nombre de réaffectations de tâches dans la conception de lignes reconfigurables multi-produits

Dans ce chapitre, nous introduisons un nouveau problème de conception de lignes de production reconfigurables multi-produits. Une description détaillée du problème sera initialement donnée. Par la suite, deux objectifs seront traités. Pour chacun d'eux, une formulation mathématique sera proposée et une approche heuristique efficace sera décrite. Finalement, les résultats expérimentaux seront présentés et une comparaison entre les deux approches sera donnée.

4.1 Introduction

L'un des défis auxquels les entreprises sont confrontées lorsqu'elles mettent en œuvre la reconfigurabilité est l'équilibrage de charges entre les différentes stations. Contrairement aux systèmes traditionnels, la conception d'une ligne reconfigurable doit prendre en considération différents produits, chacun d'entre eux étant associé à une configuration de ligne appropriée. Ce problème est similaire au problème d'équilibrage de lignes d'assemblage multi-produits (MuMALBP), introduit précédemment. Dans ce chapitre, nous étudions un nouveau problème d'optimisation qui vise à générer des configurations admissibles pour chaque produit tout en minimisant le nombre de tâches réaffectées lors d'une reconfiguration.

4.1.1 Description du problème

Les lignes de production reconfigurables étudiées sont capables de fabriquer différents produits de la même famille. Chaque produit doit être traité suivant une cadence de production, exprimée par le temps de cycle. Pour qu'un produit soit fabriqué, il faut qu'un nombre donné

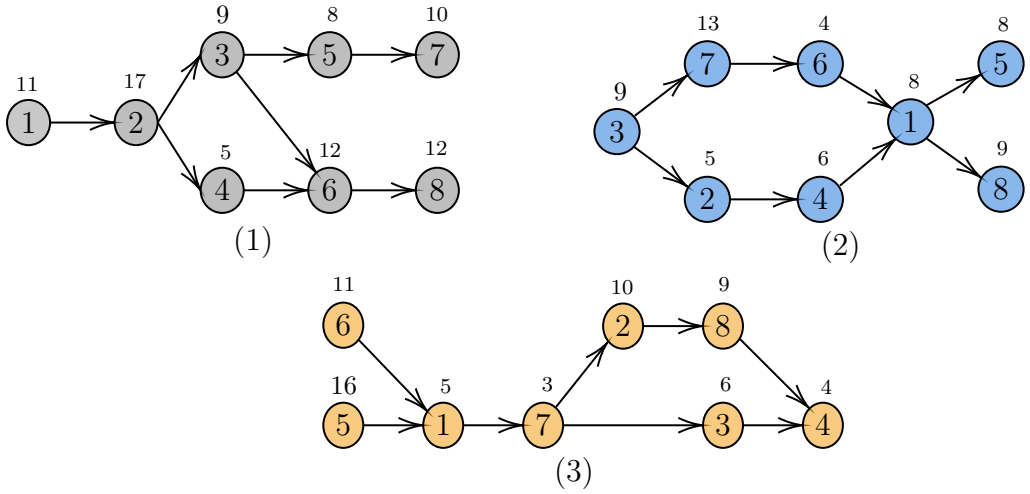


FIGURE 4.1 – (1), (2) et (3) sont les graphes de précédence qui correspondent aux produits 1, 2 et 3.

de tâches élémentaires soient réalisées, chacune nécessitant un temps opératoire bien défini. De plus, les tâches doivent être exécutées suivant un ordre partiel, généralement représenté par un graphe de précédence acyclique orienté.

Comme les produits sont de la même famille, on peut considérer qu'ils partagent le même ensemble de tâches à accomplir. Cette condition n'affecte pas le fait que certaines tâches peuvent être spécifiques à un produit seulement. Cependant, le graphe de précédence de chaque produit et le temps opératoire des tâches peuvent être différents en raison de leur niveau de complexité et des plans de processus sélectionnés pour chacun d'eux. Un exemple illustratif est donné à la figure 4.1. Cette dernière montre des graphes de précédence correspondant à trois produits nécessitant huit tâches. Le temps opératoire de chaque tâche est indiqué au dessus du sommet correspondant. L'objectif est de trouver une configuration de ligne appropriée pour chaque produit. Pour ce faire, on affecte toutes les tâches correspondantes à un nombre fixe de stations. Dans chaque configuration, les contraintes de temps de cycle et de précédence, pour chaque produit, doivent être respectées. La ligne étant reconfigurable, il est possible de passer d'une configuration à une autre en réaffectant certaines tâches. Ainsi, l'objectif est de générer les configurations qui optimisent le nombre de réaffectations.

Ainsi, deux variantes de ce problème seront étudiées :

1. Dans la première, nous considérons que l'ensemble de produits à fabriquer est donné et que la séquence de leur arrivée est connu à l'avance.

2. Nous supposons dans la deuxième variante que l'ensemble de produits est donné mais aucune information sur l'ordre d'arrivée des produits n'est fournie.

Ci-dessous les notations utilisées pour ces deux problèmes d'optimisation :

- V est l'ensemble de tâches ;
- W est l'ensemble de stations ;
- P est l'ensemble de produits ;
- $C^{(p)}$ est le temps de cycle pour le produit $p \in P$;
- $t_i^{(p)}$ est le temps opératoire de la tâche $i \in V$ du produit $p \in P$;
- $G^{(p)} = (V, A^{(p)})$ est un graphe acyclique orienté représentant les contraintes de précédence du produit $p \in P$. Dans cette notation, $A^{(p)}$ représente l'ensemble d'arcs pour $G^{(p)}$, où un arc $(i, j) \in A^{(p)}$ signifie que la tâche j doit être exécutée après la tâche i. Par conséquent, j peut être affectée soit à la même station que i, ou bien à celles qui lui succèdent.

D'autres notations spécifiques à chaque variante seront introduites par la suite.

4.1.2 Les intervalles d'affectation

Afin de renforcer les performances des modèles PLNE présentés ultérieurement, des techniques de prétraitement sont proposées dans cette section. Ces dernières tirent profit de la structure du problème et des données d'entrée afin de réduire l'espace de recherche. Elles sont principalement basées sur les graphes de précédence et le temps opératoire des tâches. L'utilisation de ces informations permet de réduire les intervalles d'affectation des tâches, notées par $Q_i^{(p)} = \left[l_i^{(p)}, u_i^{(p)}\right]$, où $l_i^{(p)}$ (resp. $u_i^{(p)}$) correspond à l'indice de la station au plus tôt (resp. au plus tard) où la tâche i du produit p pourrait être affectée. Initialement, l'intervalle d'affectation de chaque tâche est fixé à $[1, |W|]$.

Nous utilisons deux techniques distinctes pour calculer les intervalles d'affectation et les combinons par la suite pour obtenir des intervalles d'affectation aussi serrés que possibles. L'objectif de cette approche vise à réduire autant que possible l'intervalle d'affectation de chaque tâche. Ainsi, le premier intervalle, noté $Q1_i^{(p)}$, est calculé en se basant sur une technique bien connue, développée par Patterson and Albracht [1975] :

$$Q1_i^{(p)} = \left[\left\lceil \frac{t_i^{(p)} + \sum\limits_{j \in \mathcal{P}_i^{(p)}} t_j^{(p)}}{C^{(p)}} \right\rceil, |W| + 1 - \left\lceil \frac{t_i^{(p)} + \sum\limits_{j \in \mathcal{S}_i^{(p)}} t_j^{(p)}}{C^{(p)}} \right\rceil \right].$$

Ici, $\mathcal{P}_i^{(p)}$ (resp. $\mathcal{S}_i^{(p)}$) représente l'ensemble de tous les prédécesseurs (resp. tous les successeurs) de la tâche i par rapport au graphe de précédence $G^{(p)}$ du produit p. Dans $Q1_i^{(p)}$, l'indice de la station au plus tôt, où la tâche i pourrait être affectée, est donné par le rapport entre la somme de tous les temps opératoires correspondant à l'ensemble de tâches $\mathcal{P}_i^{(p)} \bigcup \{i\}$ et le temps de cycle du produit p. Ce rapport représente donc une borne inférieure sur le nombre de stations

nécessaires pour affecter la tâche i et tous ses prédécesseurs. De la même façon, l'indice de la dernière station où la tâche i pourrait être affectée est calculé en fonction de tous ses successeurs.

La deuxième méthode pour calculer les intervalles d'affectation est une adaptation de l'approche initialement proposée par Pirogov et al. [2021] pour le cas de l'équilibrage des lignes de transfert. Elle est basée sur la représentation du problème étudié comme un problème d'ordonnancement à une machine. En effet, il est possible d'associer à chaque tâche i du produit p son temps d'achèvement au plus tôt $\theta_i^{(p)}$ et son temps de démarrage au plus tard $\zeta_i^{(p)}$. Sur la base du temps de cycle de chaque produit p, il n'est pas difficile de voir que ces paramètres peuvent être calculés de la manière récursive suivante :

$$\theta_i^{(p)} = t_i^{(p)} + \max_{j \in \overline{\mathcal{P}}_i^{(p)}} \max \left\{ \theta_j^{(p)}, \left(\left\lceil \frac{\theta_j^{(p)} + t_i^{(p)}}{C^{(p)}} \right\rceil - 1 \right) \cdot C^{(p)} \right\},$$

$$\zeta_i^{(p)} = \min_{j \in \overline{\mathcal{S}}_i^{(p)}} \min \left\{ \zeta_j^{(p)}, \left(\left\lfloor \frac{\zeta_j^{(p)} - t_i^{(p)}}{C^{(p)}} \right\rfloor + 1 \right) \cdot C^{(p)} \right\} - t_i^{(p)},$$

où $\overline{\mathcal{P}}_i^{(p)}$ (resp. $\overline{\mathcal{S}}_i^{(p)}$) est l'ensemble des prédécesseurs directs (resp. successeurs directs) de la tâche i correspondant au graphe de précédence $G^{(p)}$ du produit p. Ici, $\theta_0^{(p)} = 0$ et $\zeta_{|V|+1}^{(p)} = C^{(p)} \times |W|$ sont respectivement le début et la fin fictifs de la séquence. Ainsi, le deuxième intervalle d'affectation, notée $Q2_i^{(p)}$, peut être obtenu de la façon suivante :

$$Q2_i^{(p)} = \left[\left\lceil \frac{\theta_i^{(p)}}{C^{(p)}} \right\rceil, 1 + \left\lfloor \frac{\zeta_i^{(p)}}{C^{(p)}} \right\rfloor \right].$$

Par conséquent, et afin de fournir les intervalles d'affectation les plus serrés pour chaque tâche, $Q_i^{(p)}$ est défini comme $Q1_i^{(p)} \bigcap Q2_i^{(p)}$ pour chaque tâche i et produit p.

4.2 L'ordre d'arrivée des produits est connu

Dans ce problème, nous considérons que la demande pour chaque produit est connue dans un horizon temporel donné, ce qui permet de déterminer à l'avance l'ordre d'arrivée des produits. En conséquence, une formulation PLNE est proposée et vise donc à minimiser le nombre total de réaffectation de tâches.

Par exemple, supposons que les trois produits de la figure 4.1 arrivent dans l'ordre suivant $1 \Rightarrow 2 \Rightarrow 3$, le temps de cycle est fixé à 40 unités de temps et 3 stations sont disponibles. La résolution de ce problème peut nous conduire à la solution optimale présentée dans la figure 4.2. Dans cette solution, les configurations 1, 2 et 3 sont associées aux graphes de précédence correspondant aux produits (1), (2) et (3) respectivement. Le passage de la configuration 1 à 2

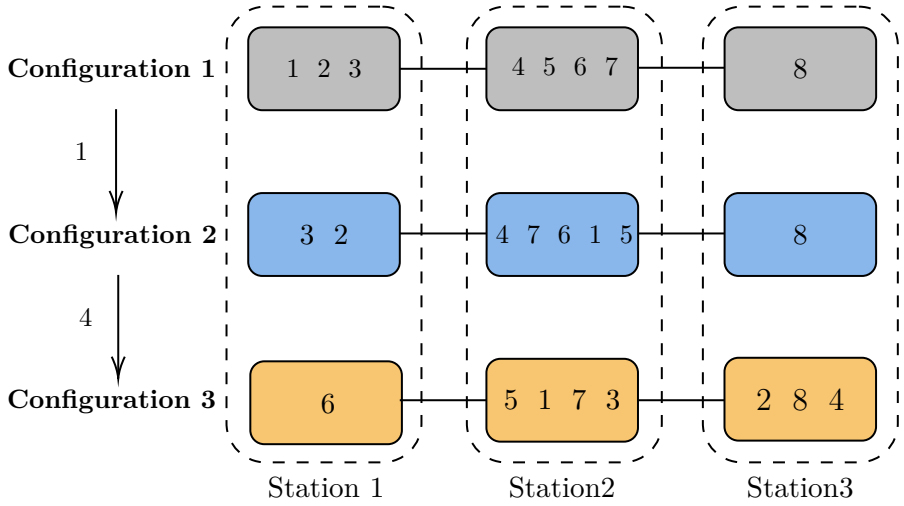


FIGURE 4.2 – Configurations optimales de chaque produit en connaissant leur ordre d'arrivée

(resp. 2 à 3) nécessite 1 réaffectation (resp. 4 réaffectations) qui consiste à déplacer la tâche 1 de la station 1 vers la station 2. Par conséquent, le nombre total de réaffectations est de 5.

Il n'est pas difficile de voir que le problème étudié est difficile à résoudre, puisque la recherche d'une configuration admissible pour chaque produit correspond à la résolution du problème de faisabilité d'équilibrage de lignes d'assemblage simples (SALBP-F), connu pour être NP-complet au sens fort [Scholl, 1999].

4.2.1 Formulation du problème

En plus des notations précédemment introduites, une nouvelle notation R est nécessaire pour représenter la séquence d'arrivée des produits, où $R = \{(1,2),(2,3),\ldots,(|P|-1,|P|)\}$ est, sans perte de généralités, l'ensemble de toutes les paires de produits successifs. Il convient de mentionner que cette séquence peut se faire dans n'importe quel ordre et pas forcément dans l'ordre croissant de l'indice des produits. Les variables de décisions utilisées sont donc :

- $x_{ik}^{(p)}$ est égale à 1 si la tâche $i \in V$ est affectée à la station $k \in W$ dans la configuration correspondante au produit $p \in P$, 0 sinon ;
- $z_{ik}^{(1,2)}$ (resp. $z_{ik}^{(2,3)}$, …, $z_{ik}^{(|P|-1,|P|)}$) est égale à 1 si la tâche $i \in V$ est affectée à la station $k \in W$ dans l'une des configuration 1 ou 2 (resp. 2 ou 3, …, $|P|-1$ ou $|P|$) mais pas dans l'autre, 0 sinon.

Ainsi, le modèle PLNE peut être formulé comme suit :

$$\min \quad \frac{1}{2} \sum_{i \in V} \sum_{k \in W} \sum_{(r_1,r_2) \in R} z_{ik}^{(r_1,r_2)} \tag{4.1}$$

sous contraintes :

$$-z_{ik}^{(r_1,r_2)} \leq x_{ik}^{(r_1)} - x_{ik}^{(r_2)} \leq z_{ik}^{(r_1,r_2)}, \quad \forall i \in V, \quad \forall k \in W, \quad \forall (r_1, r_2) \in R \tag{4.2}$$

$$\sum_{k \in W} x_{ik}^{(p)} = 1, \quad \forall i \in V, \quad \forall p \in P \tag{4.3}$$

$$\sum_{q=k}^{|W|} x_{iq}^{(p)} \leq \sum_{q=k}^{|W|} x_{jq}^{(p)}, \quad \forall (i,j) \in A^{(p)}, \ \forall k \in Q_i^{(p)} \bigcap Q_j^{(p)}, \ \forall p \in P \tag{4.4}$$

$$\sum_{i \in V} t_i^{(p)} \cdot x_{ik}^{(p)} \leq C^{(p)}, \quad \forall k \in W, \quad \forall p \in P \tag{4.5}$$

$$x_{ik}^{(p)} = 0, \quad \forall i \in V, \ \forall k \notin Q_i^{(p)}, \quad \forall p \in P \tag{4.6}$$

$$x_{ik}^{(p)} \in \{0,1\}, \quad \forall i \in V, \quad \forall k \in W, \quad \forall p \in P$$

$$z_{ik}^{(r_1,r_2)} \in [0,1], \quad \forall i \in V, \quad \forall k \in W, \quad \forall (r_1, r_2) \in R$$

La fonction objectif (4.1) minimise le nombre total de réaffectations de tâches nécessaires pour passer de la configuration 1 vers 2, de 2 vers 3, jusqu'à la configuration $|P| - 1$ vers $|P|$. Les contraintes (4.2) permettent de vérifier si une tâche est affectée ou non à la même station dans deux configurations successives. Par exemple, $z_{ik}^{(1,2)} = 1$ signifie que la tâche i est affectée à la station k dans une seule des deux configurations 1 et 2. Ce qui implique que la tâche i doit être réaffectée. Sinon, $z_{ik}^{(1,2)} = 0$. Les contraintes (4.3) s'assurent que pour chaque produit p, la tâche i doit être affectée à une seule station seulement. Afin de satisfaire le graphe de précédence de chaque produit, les contraintes (4.4) sont utilisées. Celles-ci sont renforcées par les intervalles d'affectation. Les inégalités (4.5) veillent à ce que la somme du temps opératoire des tâches affectées à une même station ne dépasse pas le temps de cycle correspondant au produit p. Les égalités (4.6) utilisent l'intervalle d'affectation des tâches pour fixer certaines variables de décision $x_{ik}^{(p)}$ à 0 dés le début.

4.2.2 L'heuristique Halt-and-Fix

Afin de résoudre efficacement les instances de grande taille du problème étudié en un temps raisonnable, un algorithme heuristique itératif, basé sur le PLNE, est proposé. L'idée principale de cette heuristique consiste à analyser une solution réalisable trouvée par le solveur après un temps de résolution relativement court. Cela permet d'identifier un certain nombre de caractéristiques liées au problème et de générer de nouvelles contraintes afin de réduire l'espace

de recherche. L'approche développée n'est pas exacte car elle est basée sur des règles qui ne garantissent pas l'obtention de l'optimum. Avant de présenter cet algorithme, il est nécessaire d'introduire les nouvelles notations suivantes :

— $K(r_1, r_2)$ est l'ensemble de tâches affectées à la même station pour la paire de produits r_1 et r_2 ;

— T est le temps de résolution maximal pour chaque itération de l'heuristique.

Les contraintes générées, quant à elles, sont exprimées sous la forme

$$z_{ik}^{(r_1,r_2)} = 0, \quad \forall (r_1, r_2) \in R, \quad \forall i \in K(r_1, r_2), \ \forall k \in W. \tag{4.7}$$

Au début de l'approche proposée, nommée Halt-and-Fix, aucune solution réalisable n'est connue et l'ensemble de contraintes (4.7) est vide, puisque $K(r_1, r_2) = \emptyset$ pour chaque $(r_1, r_2) \in R$. Chaque itération de cette heuristique consiste donc à résoudre, dans la limite du temps T, le modèle PLNE (4.1)-(4.7) en utilisant un solveur commercial. Le but est d'identifier une solution admissible qui est meilleure (en termes de valeur de la fonction objectif) que la meilleure solution trouvée lors de l'itération précédente. Si c'est le cas, la série d'étapes suivante est exécutée :

1. Le solveur est interrompu,
2. Sur la base de cette nouvelle meilleure solution, l'ensemble $K(r_1, r_2)$ est mis à jour pour chaque $(r_1, r_2) \in R$,
3. Une nouvelle itération pourra ainsi être lancée sur le modèle PLNE (4.1)-(4.7), renforcée par de nouvelles contraintes (4.7) et en utilisant la meilleure solution actuelle comme point de départ.

Deux autres cas spécifiques sont également possibles durant la période de résolution T de chaque itération. Le premier se produit lorsqu'une solution optimale est trouvée avant de dépasser la limite de temps T. Cette solution est alors retournée comme solution finale et l'algorithme s'arrête. Le deuxième cas, quant à lui, survient lorsque le but de l'itération en cours n'est pas atteint (*c.-à-d.*, aucune meilleure solution admissible n'est trouvée). Alors le solveur n'est pas interrompu et continue à résoudre jusqu'à ce que la valeur de la fonction objectif soit améliorée ou que la limite de temps globale, fixée pour l'heuristique Halt-and-Fix, soit dépassée. Dans cette dernière situation, l'algorithme s'arrête et la meilleure solution admissible trouvée est retournée comme solution finale. Les étapes 1 et 2 ci-dessus peuvent être désignées respectivement comme les étapes Halt (interrompre) et Fix (fixer).

Pour deux configurations données correspondant à une paire de produits r_1 et r_2, les contraintes (4.7) réduisent l'espace de recherche en forçant la tâche i de $K(r_1, r_2)$ à être affectées à une même station. Cependant, ces contraintes ne déterminent pas exactement la station où la tâche i doit être affectée et offrent ainsi une certaine liberté de choix sachant que dans la plupart des cas

$|Q_i^{(r_1)} \bigcap Q_i^{(r_2)}| > 1$.

Une description plus formelle de l'approche proposée est donnée ci-dessous dans Algorithme 1.

Algorithme 1 : Heuristique Halt-and-Fix

Étape 1 : **Initialisation.**
- Définir la limite de temps d'itération T.
- Initialiser $K(r_1, r_2) = \emptyset$ pour chaque $(r_1, r_2) \in R$.
- Initialiser la meilleure solution réalisable actuelle $s^{(M)}$ comme une solution vide.

Étape 2 : **Résoudre le PLNE.**
- Commencer à résoudre le problème (4.1)-(4.7) avec la solution $s^{(M)}$ comme point de départ.
- Si une solution optimale est trouvée dans un délai de T, alors aller à l'Étape 4.
- Si T est atteint et qu'une solution s meilleure que $s^{(M)}$ est trouvée, alors passer à l'Étape 3.
- Sinon, continuer à résoudre jusqu'à ce qu'une solution meilleure que $s^{(M)}$ soit trouvée. Ensuite, passer à l'Étape 3.

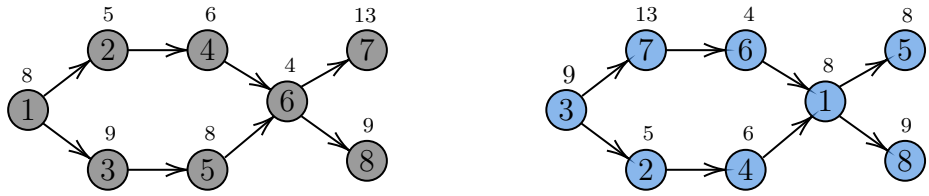
Étape 3 : **Analyser une nouvelle meilleure solution.**
- Interrompre la résolution du problème (4.1)-(4.7).
- Mettre à jour l'ensemble $K(r_1, r_2)$ pour chaque $(r_1, r_2) \in R$ en fonction de la nouvelle solution s trouvée.
- Réinitialiser $s^{(M)} := s$ et répéter l'Étape 2.

Étape 4 : **Stop, la solution finale approchée est trouvée.**

Dans la section suivante, une comparaison détaillée des performances entre le modèle PLNE seul et l'heuristique Halt-and-Fix est effectuée par le biais d'expériences numériques sur des instances de référence bien connues.

4.2.3 Résultats expérimentaux

L'objectif de cette section est de présenter les résultats expérimentaux de l'approche PLNE avec un solveur commercial, désignée dorénavant comme l'approche exacte, et d'évaluer les performances de l'heuristique Halt-and-Fix proposée. Pour ce faire, les instances de référence utilisées sont d'abord présentées, conçues et décrites. Ensuite, une analyse des résultats pour l'approche exacte est fournie et enfin comparée à ceux obtenus par l'heuristique. Les expériences numériques ci-dessous ont été menées sur un ordinateur Intel(R) Core(TM) i7-8650U à 1.90 GHz avec 32 Go de RAM.



Avant Après

FIGURE 4.3 – Exemple de modification d'un graphe de précédence.

a. Données d'entrée et modification des instances

Pour les expériences numériques, nous avons utilisé les instances de référence de Otto et al. [2013], qui sont principalement utilisées pour le SALBP. Dans ces expérimentations, seules les instances où les contraintes de précédence sont représentées par un graphe orienté acyclique faiblement connexe ont été prises en compte. De plus, nous avons remarqué que les graphes de précédence de chacune des instances sont générés de façon à ce que l'énumération des sommets suive un ordre topologique. Par conséquent et en vue d'éviter des instances triviales, chaque graphe de précédence utilisé a été modifié par rapport à une permutation de sommets générée de manière aléatoire. Cela peut conduire à des instances qui ne reflètent pas nécessairement la réalité, mais les rend plus difficiles à résoudre et, ainsi, meilleures pour évaluer les expérimentations numériques. La figure 4.3 montre un exemple d'une telle modification pour la permutation $(3, 7, 2, 6, 4, 1, 5, 8)$, où le graphe original se trouve à gauche et le graphe modifié à droite.

Deux ensembles d'instances ont été utilisées : celles de taille moyenne ($|V| = 50$) et celles de grande taille ($|V| = 100$). Chaque ensemble est divisé en trois séries d'instances selon la densité de leurs graphes de précédence ou OS (*Order Strentgh*, en anglais) dont la formule est :

$$OS = \frac{2 \cdot |E^{(p)}|}{|V|(|V| - 1)},$$

où $E^{(p)} = \left\{(i, j) \mid i \in V, j \in \mathcal{S}_i^{(p)}\right\}$ est l'ensemble d'arcs de la fermeture transitive de $G^{(p)}$. La première série (voir tableau 4.1) correspond aux instances ayant OS autour de 0.2, noté $OS \approx 0.2$. La deuxième (resp. la troisième) concerne les instances ayant $OS \approx 0.6$ (resp. $OS \approx 0.9$). Pour chaque série, les cas de $|P| = 2$ et $|P| = 3$ sont considérés. Par exemple, pour construire une instance pour deux (resp. trois) produits, il suffit de considérer deux (resp. trois) instances successives de Otto et al. [2013] de la même série. Au total, 640 (resp. 634) instances ont été générées pour $|P| = 2$ (resp. $|P| = 3$).

De plus, pour chaque instance, le temps de cycle $C^{(p)}$ de chaque produit $p \in P$ est fixé

Tableau 4.1 – Énumération des trois séries

Série	1	2	3
OS	≈ 0.2	≈ 0.6	≈ 0.9

comme suit :

$$C^{(p)} = \left\lceil 1.5 \cdot \max_{i \in V} t_i^{(p)} \right\rceil,$$

et le nombre de stations $|W|$ est calculée de la manière suivante :

$$|W| = \max_{p \in P} \left\{ \left\lceil 1.2 \cdot \sum_{i \in V} t_i^{(p)} / C^{(p)} \right\rceil \right\}.$$

Ce choix a été fait après des expériences préliminaires afin d'obtenir des instances non triviales.

b. Résultats expérimentaux pour l'approche exacte

Le solveur CPLEX 12.10 a été utilisé pour résoudre le PLNE décrit précédemment. Les résultats de calcul sont présentés dans le tableau 4.2. Une limite de temps globale de 600 secondes a été fixée pour la résolution de chaque instance. Les deux premières colonnes du tableau 4.2 représentent le nombre de produits et le nombre de tâches, respectivement. La troisième indique la série de l'ensemble d'instances correspondant, tandis que la quatrième montre le nombre total d'instances dans chaque série. Le nombre d'instances résolues de manière optimale dans le délai global et leur temps moyen de résolution sont indiqués dans la cinquième et la sixième colonne, respectivement. Enfin, la dernière colonne représente le `GAP` moyen pour les instances restantes qui n'ont pas été résolues de manière optimale. Pour chaque instance, le `GAP` est calculé par rapport à la valeur de la fonction objectif de la meilleure solution trouvée ou UB (*Upper Bound*, en anglais) et à la valeur de la borne inférieure ou LB (*Lower Bound*, en anglais), déterminée par CPLEX en utilisant l'expression suivante :

$$\texttt{GAP} = 100\% \cdot (UB - LB)/UB$$

Comme on peut le voir dans le tableau 4.2, CPLEX ne parvient pas à trouver des solutions optimales dans la plupart des cas. En effet, on peut remarquer que l'approche exacte a de plus en plus de difficultés à résoudre les instances lorsque le nombre de produits et/ou le nombre de tâches augmentent. Cela peut être constaté par le fait que 71.1% des instances correspondant à $|V| = 50$ et $|P| = 2$ ont été résolues de manière optimale, contre seulement 24.26% dans le cas où $|V| = 50$ et $|P| = 3$. De plus, pour chaque ensemble, les instances avec un OS plus

grand semblent être plus faciles à résoudre que celles avec un OS plus petit, où l'on remarque une diminution du temps moyen de résolution et du `GAP`. En ce qui concerne les instances de $|V| = 100$, le solveur est incapable de trouver une solution optimale dans la plupart des cas pour $|P| = 2$ et $|P| = 3$. Cela est dû au fait que CPLEX améliore à peine la borne inférieure pendant le processus de résolution.

Tableau 4.2 – Résultats numériques pour l'approche exacte sur des instances modifiées

$\|P\|$	$\|V\|$	OS	#`Instances`	#`OPT`	Moy. `CPU`, (s.)	Moy. `GAP`, (%)
2	50	1	15	3	291.92	45.03
		2	219	142	168.76	13.10
		3	74	74	14.29	⊕
	100	1	39	0	⊖	87.27
		2	219	0	⊖	46.06
		3	74	2	239.65	16.42
Total			640	221 (34.53%)	119.35	37.43
3	50	1	14	0	⊖	81.28
		2	218	9	346.80	26.29
		3	73	65	161.33	6.68
	100	1	38	0	⊖	90.03
		2	218	0	⊖	56.46
		3	73	0	⊖	29.05
Total			634	74 (11.67%)	183.89	43.81

(⊕) Toutes les solutions optimales ont été trouvées dans le temps imparti de 600 secondes.
(⊖) Aucune solution optimale n'a été trouvée dans le temps imparti de 600 secondes.

Le tableau 4.3 montre les résultats de calcul obtenus pour les mêmes instances utilisées pour le tableau 4.2, mais sans modifier leurs graphes de précédence. À partir de ce tableau, il est clair que les instances de taille moyenne ($|P| = 2$ ou $|V| = 50$) sont faciles à résoudre pour la majorité des tests. En revanche, celles de grande taille ($|P| = 3$ et $|V| = 100$) sont relativement difficiles, mais toujours plus faciles à résoudre par rapport aux instances modifiées présentées dans le tableau 4.2. On peut aussi remarquer que les valeurs moyennes du `GAP` sont aberrantes ($\approx 90\%$). Ce comportement est principalement dû au fait que la borne inférieure trouvée par CPLEX est le plus souvent égale à 0 en raison des graphes de précédence non modifiés. Par conséquent, cela ne donne aucun aperçu de la complexité des instances non résolues. Ce qui n'est pas le cas pour les valeurs moyennes du `GAP` fournies dans le tableau 4.2.

Tableau 4.3 – Résultats numériques pour l'approche exacte sur des instances non-modifiées

$\|P\|$	$\|V\|$	OS	#Instances	#OPT	Moy. CPU, (s.)	Moy. GAP, (%)
2	50	1	15	15	1.28	$\oplus$
		2	219	215	1.23	81.25
		3	74	74	5.63	$\oplus$
	100	1	39	34	119.43	99.99
		2	219	216	115.57	66.17
		3	74	55	74.14	83.90
Total			640	609 (95.16%)	55.50	84.44
3	50	1	14	14	7.79	$\oplus$
		2	218	183	37.43	75.24
		3	73	66	57.21	6.27
	100	1	38	10	343.90	97.58
		2	218	56	354.51	96.54
		3	73	1	14.59	96.60
Total			634	330 (52.05%)	103.14	92.12

($\oplus$) Toutes les solutions optimales ont été trouvées dans le temps imparti de 600 secondes.

c. L'approche exacte vs Halt-and-Fix

Dans cette partie, les performances de l'heuristique Halt-and-Fix sont analysées. L'heuristique a été codée en Python avec la bibliothèque DOcplex, qui est un API permettant d'utiliser le solveur CPLEX. Elle a été testée sur les deux ensembles d'instances décrits ci-dessus correspondant à $|V| = 50$ et $|V| = 100$ avec 2 et 3 produits. Les tableaux 4.5 et 4.4 montrent les résultats comparatifs entre l'approche exacte et l'heuristique pour les instances modifiées et non modifiées, respectivement. Pour les deux tableaux, la quatrième (resp. cinquième) colonne représente le GAP moyen entre la meilleure borne supérieure trouvée et la borne supérieure trouvée par l'approche exacte (resp. heuristique). Ce gap est calculé comme suit :

$$\mathtt{GAP}_{\mathtt{E}}^{\mathtt{B}} = 100\% \cdot \frac{(\mathtt{UB}_{\mathtt{E}} - \mathtt{UB}_{\mathtt{B}})}{\mathtt{UB}_{\mathtt{B}}}$$

pour l'approche exacte et,

$$\mathtt{GAP}_{\mathtt{H}}^{\mathtt{B}} = 100\% \cdot \frac{(\mathtt{UB}_{\mathtt{H}} - \mathtt{UB}_{\mathtt{B}})}{\mathtt{UB}_{\mathtt{B}}}$$

pour l'heuristique, dans lesquelles $\mathtt{UB}_{\mathtt{E}}$ (resp. $\mathtt{UB}_{\mathtt{H}}$) est la borne supérieure trouvée par l'approche exacte (resp. heuristique) et $\mathtt{UB}_{\mathtt{B}} = \min\{\mathtt{UB}_{\mathtt{E}}, \mathtt{UB}_{\mathtt{H}}\}$. Le temps moyen de résolution (calculé sur toutes les instances) de l'approche exacte ($\mathtt{CPU}_{\mathtt{E}}$) et de l'heuristique ($\mathtt{CPU}_{\mathtt{H}}$) est indiqué dans la

sixième et la septième colonne, respectivement. La limite de temps d'itération T a été fixée à 10 secondes.

Le tableau 4.5, correspondant aux instances non modifiées, montre que même si l'approche exacte peut résoudre de nombreuses instances de manière optimale, l'heuristique parvient également à fournir des solutions optimales ou proches de l'optimum, ce qui peut être particulièrement observé pour $|P| = 2$ et $|V| = 50$, où dans 98% des cas, l'heuristique a pu atteindre les mêmes solutions optimales que celles trouvées par l'approche exacte. De plus, il est clair que lorsque la taille de l'instance augmente ($|P| = 2$ et $|V| = 100$ ou $|P| = 3$), l'heuristique trouve de meilleures solutions que l'approche exacte en un temps moyen considérablement plus court. Par exemple, pour $|P| = 3$ et $|V| = 50$, l'heuristique Halt-and-Fix a trouvé les meilleures solutions dans 77% des cas. De même, en traitant des instances correspondant à $|V| = 100$, la même heuristique a été capable de trouver les meilleures solutions dans 90% des cas en un temps de résolution nettement inférieur à celui de l'approche exacte. Une exception où l'heuristique n'est pas efficace peut être remarquée pour les instances correspondant à la troisième série ($OS \approx 0.9$). Ceci est principalement dû au fait que ces instances ont des graphes de précédence denses qui diminuent le nombre de solutions réalisables et conduisent l'heuristique vers des optima locaux de mauvaise qualité.

Tableau 4.4 – Approche exacte vs Heuristique pour les instances modifiées

$\|P\|$	$\|V\|$	OS	Moy. $\texttt{GAP}^{\texttt{B}}_{\texttt{E}}$, (%)	Moy. $\texttt{GAP}^{\texttt{B}}_{\texttt{H}}$, (%)	Moy. $\texttt{CPU}_{\texttt{E}}$, (s.)	Moy. $\texttt{CPU}_{\texttt{H}}$, (s.)
2	50	1	2.30	5.38	538.40	27.01
		2	0.23	1.94	318.42	14.42
		3	0.00	0.42	14.29	3.68
	100	1	20.49	3.60	600	587.31
		2	6.40	1.23	600	545
		3	0.82	1.35	590.36	342.47
3	50	1	17.22	1.46	600	286.55
		2	3.61	1.74	589.57	141.48
		3	0.00	1.58	209.42	21.54
	100	1	0.70	3.34	600	600
		2	4.02	2.45	600	600
		3	3.88	0.51	600	537.06

Le tableau 4.4 montre que, pour les instances modifiées, l'approche heuristique reste plus performante que l'approche exacte dans la plupart des cas. Pour $|P| = 2$ et $|V| = 50$, où l'approche exacte a pu résoudre de manière optimale 71.1% des instances, l'heuristique a montré des résultats prometteurs. En effet, dans 72.39% des cas, l'heuristique a pu trouver la même

solution optimale en moins de 30 secondes seulement. Pour les autres instances, l'heuristique a fourni des solutions avec un $\texttt{GAP}^{\texttt{B}}_{\texttt{H}}$ moyen de 1.80%. Pour des valeurs de $|P| = 3$ et $|V| = 50$, l'approche heuristique a fourni des solutions jusqu'à 17 fois meilleures (en termes de valeur de la fonction objectif) que l'approche exacte, en particulier pour les instances les plus complexes correspondant à $OS \approx 0.2$ et $OS \approx 0.6$, où le $\texttt{GAP}^{\texttt{B}}_{\texttt{E}}$ est égal à 17.22% et 3.61%, respectivement. Pour les instances de grande taille correspondant à $|V| = 100$, l'heuristique donne de meilleurs résultats que l'approche exacte. Nous pouvons clairement remarquer que presque toutes les meilleures solutions ont été principalement trouvées par l'heuristique.

Tableau 4.5 – Approche exacte vs Heuristique pour les instances non-modifiées

$\|P\|$	$\|V\|$	OS	Moy. $\texttt{GAP}^{\texttt{B}}_{\texttt{E}}$, (%)	Moy. $\texttt{GAP}^{\texttt{B}}_{\texttt{H}}$, (%)	Moy. $\texttt{CPU}_{\texttt{E}}$, (s.)	Moy. $\texttt{CPU}_{\texttt{H}}$, (s.)
2	50	1	0.00	0.00	1.28	1.27
		2	0.00	0.57	12.17	3.56
		3	0.00	1.17	5.63	3.39
	100	1	12.82	0.00	193	24.16
		2	0.91	0.00	122.21	34.18
		3	1.35	14.27	216.74	60.51
3	50	1	0.00	0.00	7.79	4.33
		2	1.21	7.87	127.77	12.58
		3	0.00	21.08	72.09	9.88
	100	1	71.05	2.63	546.22	195.19
		2	70.61	1.94	541.53	229.82
		3	27.43	34.00	592.39	168.04

Pour une meilleure compréhension, les figures 4.4 et 4.5 montrent respectivement comment le GAP moyen par rapport à la meilleure borne supérieure trouvée, noté $\texttt{GAP}^{\texttt{B}}_{(.)}$, et le temps de résolution moyen varient en fonction de l'OS pour l'approche exacte (marqueurs ronds de couleur bleue) et l'approche heuristique (marqueurs carrés de couleur rouge). Dans chaque figure, le graphique de gauche correspond aux instances de 50 tâches et celui de droite à celles de 100 tâches. D'après la figure 4.4, on peut confirmer que l'heuristique a montré une meilleure performance globale en termes de $\texttt{GAP}^{\texttt{B}}_{(.)}$ moyen. La différence devient plus perceptible lorsque le nombre de tâches passe de 50 à 100. Enfin, la figure 4.5 met clairement en évidence l'efficacité de l'heuristique à fournir de meilleures solutions en un temps de résolution beaucoup plus court par rapport à l'approche exacte.

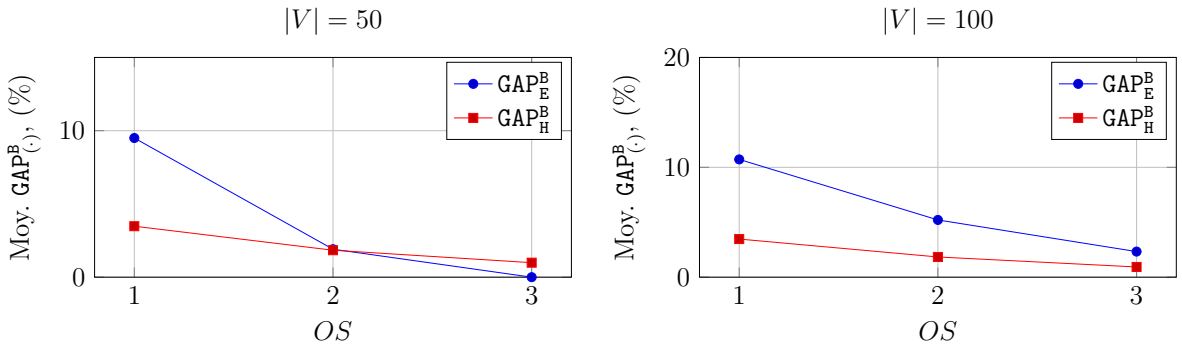


FIGURE 4.4 – Comparaison des résultats entre $\mathtt{GAP}^{\mathtt{B}}_{\mathtt{E}}$ et $\mathtt{GAP}^{\mathtt{B}}_{\mathtt{H}}$ pour les instances modifiées

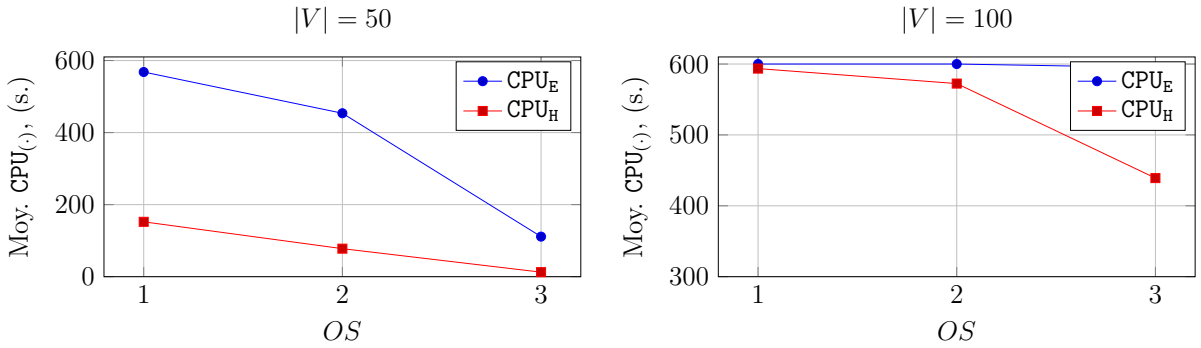


FIGURE 4.5 – Comparaison des Moy. CPU entre l'approche exacte et l'heuristique HALT-AND-FIX pour les instances modifiées

4.3 L'ordre d'arrivée des produits est inconnu

Lorsque l'ordre d'arrivée des produits n'est pas connu, il devient impossible de minimiser le nombre total de réaffectations. Dans ce cas là, il est important de concevoir pour chaque produit une configuration de telle sorte à optimiser le nombre de tâches réaffectées pour n'importe quelle séquence d'arrivée des produits. Plus particulièrement, ceci consiste à considérer toutes les combinaisons possibles de la séquence d'arrivée des produits, et de générer en conséquence des configurations robustes qui minimisent le nombre de réaffectation dans le pire des cas. Ainsi, l'objectif dans cette section est concevoir des configurations de manière à minimiser le nombre maximal de réaffectations de tâches entre deux configurations quelconques de la ligne.

Par exemple, reprenons les graphes de précédence illustrés dans la figure 4.1. De la même façon que dans la section précédente, on considère ici que le temps de cycle pour les trois produits est égal à 40 unités de temps, et 3 stations sont disponibles. La seule différence est que l'ordre d'arrivée des produits n'est pas connu, et donc toute combinaison possible entre les produits doit être prise en considération. La figure 4.6 montre les configurations optimales de chaque produit.

À partir de cette solution, on constate que 2 réaffectations seraient nécessaires pour passer de la configuration 1 à la configuration 2. De même, 4 tâches devraient être réaffectées pour passer de la configuration 2 à 3 et de la configuration 1 à 3. Ici, le 4 qui est placé tout à gauche de la figure, représente la valeur de la fonction objectif pour cette solution. Celle-ci représente donc le nombre maximum de réaffectations de tâches entre deux configurations quelconques de la ligne.

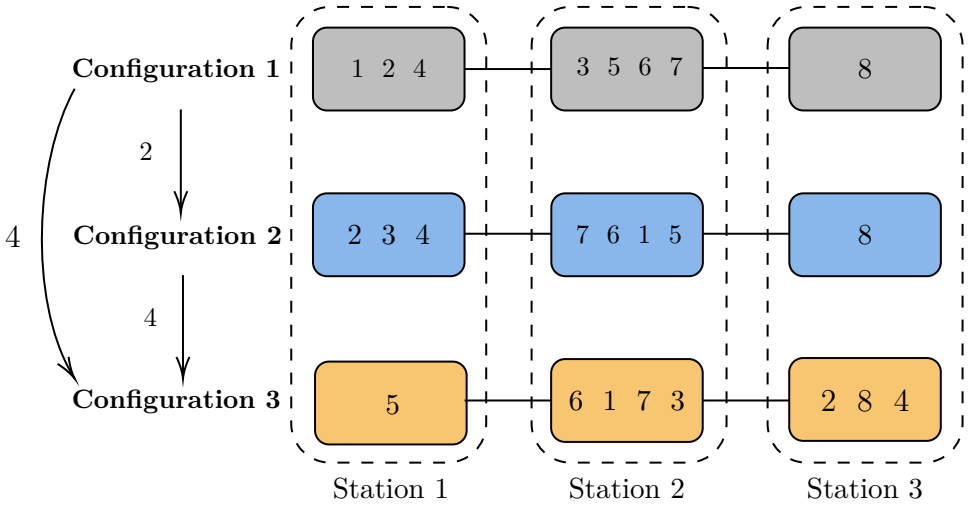


FIGURE 4.6 – Configurations optimales de chaque produit quelque soit leur ordre d'arrivée

4.3.1 Formulation du problème

Comme pour le problème précédent, une nouvelle notation doit être introduite en plus de celle présentée au début du chapitre. Du fait que l'ordre d'arrivée des produits n'est pas connu, il est nécessaire de considérer toutes les paires possibles représentées par l'ensemble $U = \{(p, q) \in P \times P : p < q\}$. Ainsi, les variables de décisions peuvent être représentées comme suite :

- $x_{ik}^{(p)}$ est égale à 1 si la tâche $i \in V$ est affectée à la station $k \in W$ dans la configuration qui correspond au produit $p \in P$, 0 sinon.
- $y_{ik}^{(p,q)}$ est égale à 1 si la tâche $i \in V$ est affectée à la même station $k \in W$ dans les deux configurations des produits p et q, 0 sinon.
- $r^{(p,q)}$ est le nombre de tâches réaffectées en passant de la configuration du produit p vers celle du produit q.
- $r_{\max}$ est le nombre maximum de réaffectations parmi toutes les paires possibles.

En utilisant les variables introduites, la formulation PLNE du problème étudié est la suivante.

$$\min \quad r_{\max} + \frac{1}{|V| \cdot |U|} \cdot \sum_{(p,q) \in U} r^{(p,q)} \tag{4.8}$$

sous contraintes :

$$|V| - \sum_{i \in V} \sum_{k \in W} y_{ik}^{(p,q)} \leq r^{(p,q)} \leq r_{\max}, \quad \forall (p,q) \in U \tag{4.9}$$

$$y_{ik}^{(p,q)} \leq \frac{1}{2} \cdot \left(x_{ik}^{(p)} + x_{ik}^{(q)} \right), \quad \forall i \in V, \quad \forall k \in W, \quad \forall (p,q) \in U \tag{4.10}$$

$$\sum_{k \in W} x_{ik}^{(p)} = 1, \quad \forall i \in V, \quad \forall p \in P \tag{4.11}$$

$$\sum_{q=k}^{|W|} x_{iq}^{(p)} \leq \sum_{q=k}^{|W|} x_{jq}^{(p)}, \quad \forall (i,j) \in A^{(p)}, \ \forall k \in Q_i^{(p)} \bigcap Q_j^{(p)}, \ \forall p \in P \tag{4.12}$$

$$\sum_{i \in V} t_i^{(p)} \cdot x_{ik}^{(p)} \leq C^{(p)}, \quad \forall k \in W, \quad \forall p \in P \tag{4.13}$$

$$x_{ik}^{(p)} = 0, \quad \forall i \in V, \ \forall k \notin Q_i^{(p)}, \quad \forall p \in P \tag{4.14}$$

$$x_{ik}^{(p)} \in \{0,1\}, \quad \forall i \in V, \quad \forall k \in W, \quad \forall p \in P$$

$$y_{ik}^{(p,q)} \in \{0,1\}, \quad \forall i \in V, \quad \forall k \in W, \quad \forall (p,q) \in U$$

$$r^{(p,q)} \geq 0, \quad \forall (p,q) \in U$$

$$r_{\max} \geq 0$$

Le but principal de la fonction objectif (4.8) est de trouver une solution minimisant $r_{\max}$. De plus, parmi les solutions ayant la même valeur de $r_{\max}$, l'expression composée de (4.8) permet de favoriser celles minimisant le nombre total de réaffectations de tâches. Cela est démontré ci-dessous par le Lemme 1. Le nombre de réaffectations de tâches pour deux configurations quelconques ainsi que sa valeur maximale sont déterminés par les contraintes (4.9). Les contraintes (4.10) permettent à chaque variable $y_{ik}^{(p,q)}$ d'être égale à 1 si, et seulement si, la tâche i est affectée à la même station k pour les deux configurations correspondantes aux produits p et q. Les inégalités (4.11) sont utilisées pour garantir que chaque tâche $i \in V$ est affectée à exactement une station pour chaque produit $p \in P$. Le respect des contraintes de précédence pour chaque produit est assuré par les expressions (4.12). De la même façon, la contrainte de temps de cycle

pour chaque produit est exprimée par (4.13). Enfin, les contraintes (4.14) visent à réduire l'espace de recherche en fixant toutes les variables $x_{ik}^{(p)}$ à 0, si la tâche i ne peut pas être affectée à la station k dans la configuration correspondant au produit p par rapport à l'intervalle d'affectation $Q_i^{(p)}$.

Avant de présenter les résultats expérimentaux pour le modèle décrit ci-dessus, il est important de mieux expliquer la fonction objectif. En effet, celle-ci est une somme de deux expressions. La première vise à minimiser $r_{\max}$. La seconde quant à elle est utilisée pour trouver les meilleures configurations possibles. En reprenant l'exemple de la figure 4.6, la deuxième expression est nécessaire et a permis de réduire à 2 réaffectations seulement en passant de la configuration 1 vers la configuration 2. Sans cette deuxième expression, toutes les réaffectations auraient conduit à 4 changements d'affectation.

Lemme 1. *La fonction mono-objectif* (4.8) *a une représentation lexicographique bi-objectif équivalente, où le premier objectif est* $r_{\max}$ *et le second est* $\sum_{(p,q)\in U} r^{(p,q)}$ *(les deux devant être minimisés).*

Démonstration. ($\Rightarrow$) Soit r, représenté par la paire $\left(r_{\max}, \sum_{(p,q)\in U} r^{(p,q)}\right)$, une solution optimale pour le problème étudié avec la fonction mono-objectif (4.8), notée F_1, et prouvons que r est également optimale pour la représentation lexicographique bi-objective de (4.8), notée F_2.

Supposons que ce ne soit pas le cas et qu'il existe une solution s, exprimée par la paire $\left(s_{\max}, \sum_{(p,q)\in U} s^{(p,q)}\right)$, qui domine r au sens lexicographique. Ceci signifie que l'un des deux cas suivants se produirait : i) $s_{\max} = r_{\max}$ et $\sum_{(p,q)\in U} s^{(p,q)} < \sum_{(p,q)\in U} r^{(p,q)}$ ou ii) $s_{\max} < r_{\max}$.

Pour le premier cas, nous avons évidemment ce qui suit :

$$F_1(s) = s_{\max} + \frac{1}{|V|\cdot|U|} \cdot \sum_{(p,q)\in U} s^{(p,q)} < r_{\max} + \frac{1}{|V|\cdot|U|} \cdot \sum_{(p,q)\in U} r^{(p,q)} = F_1(r).$$

Pour le deuxième cas, du fait que $0 \leq s_{\max} < r_{\max} \leq |V|$ et en raison de la définition de $s^{(p,q)}$ and $r^{(p,q)}$, nous pouvons en déduire les deux inégalités suivantes $\sum_{(p,q)\in U} s^{(p,q)} < |V|\cdot|U|$ et $0 < \sum_{(p,q)\in U} r^{(p,q)}$. Sur la base de ces observations, nous obtenons

$$F_1(s) = s_{\max} + \frac{1}{|V|\cdot|U|} \cdot \sum_{(p,q)\in U} s^{(p,q)} < s_{\max} + 1 \leq r_{\max} < r_{\max} + \frac{1}{|V|\cdot|U|} \cdot \sum_{(p,q)\in U} r^{(p,q)} = F_1(r).$$

Pour les deux cas, on déduit que $F_1(s) < F_1(r)$, ce qui est en contradiction avec l'optimalité de r pour F_1. En conséquence, r est également optimale pour F_2.

($\Leftarrow$) Maintenant, soit s une solution lexicographiquement optimale pour F_2. Et supposons par contradiction que s n'est pas optimale pour F_1. Par conséquent, il existe une solution optimale r pour F_1 telle que $F_1(r) < F_1(s)$. De plus, puisque r est optimale pour F_1, alors elle est

nécessairement optimale pour F_2 (voir la preuve de $\Rightarrow$). Cette dernière nous amène à conclure que $s_{\max} = r_{\max}$ et $\sum_{(p,q)\in U} s^{(p,q)} = \sum_{(p,q)\in U} r^{(p,q)}$, ce qui est en contradiction avec $F_1(r) < F_1(s)$. Ainsi, s est également optimale pour F_1. □

4.3.2 L'heuristique Halt-and-Fix

Du fait que ce problème soit très similaire à celui où l'ordre d'arrivée des produits est connu, nous avons utilisé la même heuristique Halt-and-Fix décrite précédemment. Rappelons que cette heuristique se base sur l'ensemble $K(r_1, r_2)$ et sur un temps de résolution relativement court, noté T. Le premier contient les tâches affectées ensemble dans une même station entre les deux configurations r_1 et r_2, le second quant à lui est le temps de résolution maximale pour chaque itération de l'heuristique. Ainsi, dans ce problème, à la fin de chaque itération l'ensemble $K(r_1, r_2)$ est mis à jour et les contraintes suivantes sont ajoutées.

$$\sum_{k\in W} y_{ik}^{(p,q)} = 1, \quad \forall (p,q) \in U, \quad \forall i \in K(p,q) \tag{4.15}$$

4.3.3 Résultats expérimentaux

Les expériences numériques ont été faites sur les mêmes ensembles d'instances (modifiées et non-modifiées) de $|V| = 50$ et $|V| = 100$ utilisées dans la section précédente. En effet, chaque ensemble est divisé en trois séries selon l'OS. Le PLNE a été résolu en utilisant le solveur CPLEX 20.1.0. Quant à l'heuristique Halt-and-Fix, elle a été codée en Julia 1.5.4 en utilisant le langage de modélisation JuMP avec le solveur CPLEX 20.1.0. Finalement, la limite de temps de résolution globale a été fixée à 600 secondes et la limite de temps de chaque itération de l'heuristique a été fixée à 10 secondes pour chaque instance.

a. Résultats expérimentaux pour l'approche exacte

Les résultats de calcul de la formulation PLNE sont donnés dans les tableaux 4.6 et 4.7. Le premier montre les résultats obtenus à partir des instances modifiées, tandis que le second affiche ceux obtenus à partir des instances originales. Dans les deux tableaux, les trois premières colonnes représentent, respectivement, le nombre de produits, le nombre de tâches et l'OS. La quatrième colonne indique le nombre total d'instances dans chaque série. Le nombre d'instances résolues de manière optimale et leur temps de résolution moyen sont indiqués dans la cinquième et la sixième colonne. Quant aux instances qui n'ont pas été résolues à l'optimalité en 600 secondes, leur `GAP` moyen est affiché dans la dernière colonne.

Le tableau 4.6 montre que la plupart des instances ne sont pas résolues dans le temps imparti. Plus précisément, seulement 20% de toutes les instances ont été résolues de manière optimale.

De plus, on peut voir que les instances avec $OS \approx 0.2$ sont plus difficiles à résoudre que les autres. Ceci peut être remarqué à travers deux facteurs : (1) le nombre total d'instances résolues de manière optimale et (2) le `GAP` moyen des instances, qui est relativement élevé. Cependant, nous pouvons remarquer que pour les autres séries d'OS (2 et 3), davantage d'instances ont été résolues. De plus, nous pouvons voir à travers le `GAP` moyen, que plus l'OS est élevé, moins les instances sont difficiles à résoudre. En effet, le `GAP` et le `CPU` moyens diminuent significativement lorsque l'OS augmente. Néanmoins, malgré cette tendance, il est clair que le nombre d'instances non résolues et leur `GAP` moyen correspondant restent relativement élevés, en particulier pour les instances de grande taille correspondant à $|V| = 100$, où peu d'instances ont été résolues de manière optimale. De plus, pour une instance, notée dans le tableau par (+1), CPLEX n'a pas été capable de trouver une première solution initiale en 600 secondes.

Tableau 4.6 – Résultats numériques pour l'approche exacte sur des instances modifiées

$\|P\|$	$\|V\|$	OS	#`Instances`	#`OPT`	Moy. `CPU`, (s.)	Moy. `GAP` (%)
2	50	1	15	2	181.3	53.46
		2	219	136	148.4	18.60
		3	74	74	16.4	⊕
	100	1	39	0	⊖	94.73
		2	219	0	⊖	52.94
		3	73(+1)	1	569.7	22.84
Total			640	212 (33.17%)	105.3	44.84
3	50	1	14	0	⊖	85.21
		2	218	0	⊖	42.64
		3	73	31	276.1	7.34
	100	1	38	0	⊖	97.11
		2	218	0	⊖	60.11
		3	72(+1)	0	⊖	29.30
Total			634	31 (4.88%)	276.1	49.30

(⊕) Toutes les solutions optimales ont été trouvées dans le temps imparti de 600 secondes.
(⊖) Aucune solution optimale n'a été trouvée dans le temps imparti de 600 secondes.

En plus de ces expériences numériques, le tableau 4.7 montre les résultats pour les mêmes instances mais sans appliquer de permutations aléatoires à leurs graphes de précédence. Dans ce tableau, il est clair que les instances de taille moyenne (avec $|P| = 2$ ou $|V| = 50$) sont plus faciles à résoudre que les instances modifiées. Cependant, dans le cas correspondant à $|P| = 3$ et $|V| = 100$, aucune des instances n'a été résolue de manière optimale. De plus, le `GAP` moyen fourni pour ces instances ($\approx 100\%$) n'est pas une mesure fiable et ne fournit donc aucune information sur leur complexité. Ceci est principalement dû au fait que la borne inférieure est dans la plupart des cas égale à 0 car tous les numéros de tâches se trouvent dans l'ordre topologique

dans leur graphe de précédence correspondant. Néanmoins, ce tableau vient confirmer le fait que ce problème reste compliqué même sur des instances originales.

Tableau 4.7 – Résultats numériques pour l'approche exacte sur des instances non-modifiées

$\|P\|$	$\|V\|$	OS	#Instances	#OPT	Moy. CPU, (s.)	Moy. GAP (%)
2	50	1	15	15	4.0	$\oplus$
		2	219	215	4.0	83.33
		3	74	74	8.6	$\oplus$
	100	1	39	28	154.8	100
		2	218	186	142.6	100
		3	74	55	130.2	93.70
Total			640	573 (89.53%)	69.0	95.72
3	50	1	14	13	209.2	$\oplus$
		2	218	158	65.1	83.79
		3	73	58	136.2	26.46
	100	1	38	0	$\ominus$	100
		2	217	0	$\ominus$	100
		3	73	0	$\ominus$	99.65
Total			634	229 (36.11%)	91.28	94.31

($\oplus$) Toutes les solutions optimales ont été trouvées dans le temps imparti de 600 secondes.
($\ominus$) Aucune solution optimale n'a été trouvée dans le temps imparti de 600 secondes.

En résumé, la méthode exacte est relativement efficace pour trouver des solutions initiales prometteuses, mais peine à les améliorer ou à prouver qu'elles sont optimales.

b. Approche exacte vs Halt-and-Fix

Dans cette sous-section, les résultats de l'heuristique Halt-and-Fix sont analysés et comparés à ceux obtenus avec l'approche exacte. Les tableaux 4.8 et 4.9 montrent cette comparaison sur les mêmes instances modifiées et non modifiées que celles utilisées dans la sous-section précédente. Dans ces tableaux, la quatrième (resp. cinquième) colonne représente l'écart entre la borne supérieure obtenue avec la méthode exacte (resp. l'heuristique) et la meilleure borne supérieure trouvée entre les deux méthodes. Ces écarts sont calculés comme suit :

$$\mathtt{GAP}_{\mathtt{E}}^{\mathtt{B}} = 100\% \cdot \frac{(\mathtt{UB}_{\mathtt{E}} - \mathtt{UB}_{\mathtt{B}})}{\mathtt{UB}_{\mathtt{B}}}$$

pour l'approche exacte et

$$\mathtt{GAP}_{\mathtt{H}}^{\mathtt{B}} = 100\% \cdot \frac{(\mathtt{UB}_{\mathtt{H}} - \mathtt{UB}_{\mathtt{B}})}{\mathtt{UB}_{\mathtt{B}}}$$

pour l'approche heuristique, où $\texttt{UB}_\texttt{E}$ (resp. $\texttt{UB}_\texttt{H}$) est la borne supérieure trouvée par l'approche exacte (resp. heuristique) et $\texttt{UB}_\texttt{B} = \min\{\texttt{UB}_\texttt{E}, \texttt{UB}_\texttt{H}\}$. Enfin, les deux dernières colonnes $\texttt{CPU}_\texttt{E}$ et $\texttt{CPU}_\texttt{H}$ représentent le temps moyen de CPU sur toutes les instances pour respectivement l'approche exacte et l'approche heuristique.

D'après le tableau 4.8, nous pouvons voir que l'heuristique fournit une meilleure performance globale que la méthode exacte. En effet, le GAP moyen et le CPU moyen de l'heuristique sont significativement plus petits que ceux de la méthode exacte. De plus, pour les instances correspondant à $|P| = 2$ et $|V| = 50$, pour lesquelles les solutions optimales étaient principalement trouvées par l'approche exacte, l'heuristique parvient également à trouver des solutions de bonne qualité, en quelques secondes seulement. Pour les grandes instances de $|V| = 100$, la plupart des meilleures solutions ont été trouvées par l'heuristique en un temps de résolution significativement plus court. L'heuristique parvient à résoudre la plupart des instances avant d'atteindre la limite de temps de résolution de 600 secondes.

Tableau 4.8 – Approche exacte vs Heuristique pour les instances modifiées

$\|P\|$	$\|V\|$	OS	Moy. $\texttt{CPU}_\texttt{E}$, (s.)	Moy. $\texttt{CPU}_\texttt{H}$, (s.)	Moy. $\texttt{GAP}^\texttt{B}_\texttt{E}$, (%)	Moy. $\texttt{GAP}^\texttt{B}_\texttt{H}$ (%)
2	50	1	600	25.7	5.00	3.43
		2	319.5	23.6	0.53	2.89
		3	15.2	8.44	0.00	0.73
	100	1	600	366.6	22.52	5.47
		2	600	221.7	21.77	0.08
		3	599.6	244.8	4.04	0.43
Total			436.3	136.16	9.58	1.56
3	50	1	600	77.3	17.07	0.00
		2	600	70.2	6.31	0.83
		3	462.4	16.4	0.24	3.56
	100	1	600	555.8	34.24	0.00
		2	600	431.9	27.12	0.03
		3	600	337.0	6.77	0.30
Total			584.1	248.35	14.73	0.74

Le tableau 4.9 fournit la même comparaison pour les instances non modifiées. Contrairement aux instances modifiées, les valeurs de l'objectif de ces instances sont plus susceptibles d'être égales à 0, ce qui entraîne des valeurs de GAP très grandes et aberrantes. Pour éviter cela, ces dernières sont calculées comme suit :

$$\texttt{GAP}^\texttt{B}_\texttt{E} = 100\% \cdot (\texttt{UB}_\texttt{E} - \texttt{UB}_\texttt{B})/(\max(1, \texttt{UB}_\texttt{B}))$$

pour l'approche exacte et

$$\texttt{GAP}^{\texttt{B}}_{\texttt{H}} = 100\% \cdot (\texttt{UB}_{\texttt{H}} - \texttt{UB}_{\texttt{B}})/(\max(1, \texttt{UB}_{\texttt{B}}))$$

pour l'heuristique. Comme on peut le voir dans ce tableau, l'heuristique est capable de fournir des solutions de bonne qualité, proches des solutions optimales. Par exemple, l'heuristique a pu trouver toutes les solutions optimales pour les instances correspondant à $|P| = 2$ et $|V| = 50$. Pour les instances plus importantes de $|P| = 3$ et $|V| = 50$, l'heuristique fournit une meilleure solution que l'approche exacte dans 75% des cas. Quant aux instances avec $|V| = 100$, 96% et 74% des meilleures solutions ont été trouvées par l'heuristique Halt-and-Fix pour $|P| = 2$ et $|P| = 3$, respectivement. Une exception est notable dans le tableau concernant les instances correspondant à $|P| = 3$, $|V| = 50$ et $OS = 3$, où la méthode exacte a trouvé les meilleures solutions et a surpassé l'approche heuristique. Cela peut s'expliquer par le fait que ces instances sont caractérisées par des graphes de précédence denses. Cela réduit le nombre de solutions réalisables possibles et conduit l'heuristique à des optima locaux.

Tableau 4.9 – Approche exacte vs Heuristique pour les instances non-modifiées

$\|P\|$	$\|V\|$	OS	Moy. $\texttt{GAP}^{\texttt{B}}_{\texttt{E}}$, (%)	Moy. $\texttt{GAP}^{\texttt{B}}_{\texttt{H}}$, (%)	Moy. $\texttt{CPU}_{\texttt{E}}$, (s.)	Moy. $\texttt{CPU}_{\texttt{H}}$ (s.)
2	50	1	0	0	4.0	3.7
		2	0	0	14.9	4.7
		3	0	0	8.6	3.8
	100	1	892.74	9.97	280.4	180.1
		2	354.19	3.54	211.5	95.8
		3	7.6	1.29	250.9	59.2
Total			175.92	1.96	124.3	52.5
3	50	1	29.04	0	237.1	57.5
		2	3.60	7.1	212.4	22.0
		3	0.00	34.7	231.5	11.4
	100	1	463.39	5.91	600.0	598.2
		2	2758.40	2.71	600.0	431.6
		3	383.13	40.15	600.0	222.9
Total			1017.88	12.34	415.3	219.3

Pour une meilleure vue d'ensemble, la figure 4.7 montre les résultats obtenus avec des instances modifiées pour la méthode exacte (marqueurs ronds de couleur bleue) et l'heuristique Halt-and-Fix (marqueurs carrés de couleur rouge). Dans cette figure, les deux graphiques du haut représentent l'écart moyen par rapport à la meilleure solution trouvée, désigné par Moy. $\texttt{GAP}^{\texttt{B}}_{(\cdot)}$, obtenu pour des instances avec $|V| = 50$ et $|V| = 100$ pour chaque série d'OS. Les deux graphiques dessous montrent quant à eux la comparaison du temps CPU moyen entre les deux

méthodes.

Comme on peut le voir sur ces graphiques, l'heuristique est plus performante que la méthode exacte, fournissant des solutions avec un GAP moyen jusqu'à 10% meilleur pour $|V| = 50$ et jusqu'à 25% meilleur pour $|V| = 100$. Quant au temps moyen de résolution, il est clairement visible que l'heuristique proposée est plus performante que la méthode exacte pour les deux tailles d'instances considérées. En effet, on peut remarquer que le temps de résolution de l'heuristique est plutôt stable comparée à celui de l'approche exacte qui varie considérablement avec l'OS.

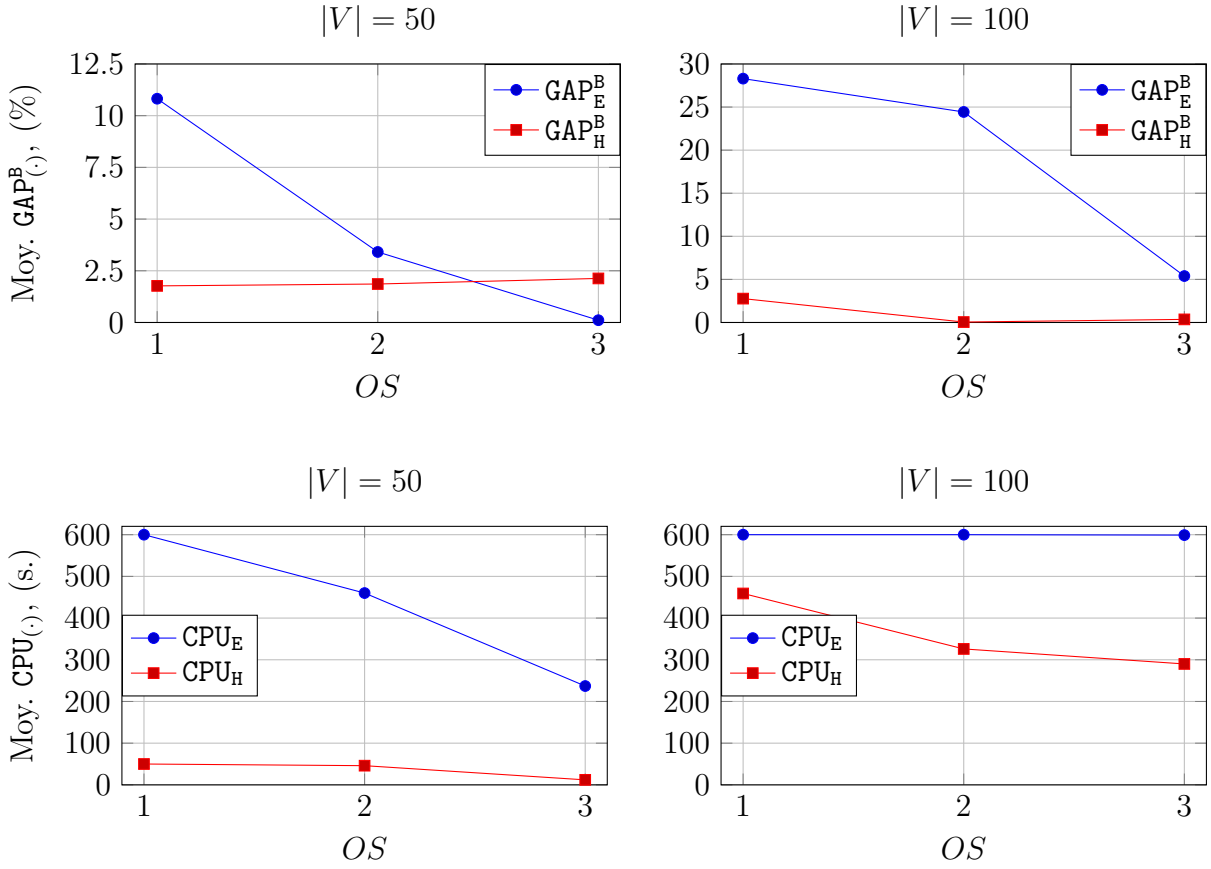


FIGURE 4.7 – Comparaison des Moy. $\texttt{GAP}^{\texttt{B}}_{(\cdot)}$ et Moy. CPU entre l'approche exacte et l'heuristique HALT-AND-FIX pour les instances modifiées

4.4 Conclusion

Nous avons étudié dans ce chapitre un problème d'équilibrage de lignes de production multi-produits dans un environnement reconfigurable avec un nombre fixe de stations. L'objectif est de trouver pour chaque produit la meilleure configuration de ligne de manière à optimiser le

nombre de réaffectations de tâches entre les stations lors du passage d'une configuration vers une autre. Chaque configuration est soumise à un temps de cycle et à des contraintes de précédences différents en fonction du produit à fabriquer.

Deux problèmes d'optimisation ont donc été étudiés. Dans le premier, nous considérons que l'ordre d'arrivée des produits est connu à l'avance. L'objectif, dans ce cas là, consiste à minimiser le nombre total de réaffectations de tâches suivant la séquence de passage de ces produits. Dans le deuxième problème, nous supposons que l'ensemble de produits à fabriquer est donné, mais leur ordre d'arrivée, quant à lui, n'est pas pas connu d'avance. Le but était donc axé sur la recherche d'une configuration de ligne appropriée pour chaque produit de manière à minimiser le nombre maximal de réaffectations de tâches entre deux configurations de ligne quelconques.

Pour chacun de ces deux problèmes, nous avons proposé une formulation PLNE, renforcée par des techniques de pré-traitement qui visaient à réduire l'espace de recherche en calculant pour chaque tâche de chaque produit son intervalle d'affectation. Ensuite, nous avons développé une heuristique efficace, nommée Halt-and-Fix. Basée sur les solutions réalisables de bonne qualité trouvées, la particularité de cette heuristique consistait à générer de nouvelles contraintes pour la formulation PLNE originale. Une comparaison numérique entre ces deux approches a montré que l'heuristique a été plus efficace et a réussi à trouver des solutions de meilleure qualité en un temps `CPU` plus court que l'approche exacte.

Plusieurs extensions possibles peuvent être explorées pour les travaux de recherche futurs. L'une des plus prometteuses est basée sur la prise en compte des données historiques disponibles concernant la fréquence d'arrivée des produits et le passage d'un produit vers un autre. Ceci peut être réalisé par un modèle PLNE stochastique plus fin, qui reflète mieux la réalité que le modèle déterministe. De plus, nous espérons également développer une heuristique ou méta-heuristique appropriée et efficace fournissant une solution admissible de démarrage à chaud de bonne qualité pour la méthode Halt-and-Fix afin d'améliorer ses performances.

Les résultats présentés dans ce chapitre ont donné lieu à quatre communications aux conférences ROADEF 2020 [Yelles-Chaouche et al., 2020], IFAC World Congress 2021 [Yelles-Chaouche et al., 2021a] , ROADEF 2021 [Yelles-Chaouche et al., 2021c] et EURO 2021 [Tremblet et al., 2021], ainsi qu'à deux articles soumis aux revues « *European Journal of Operational Research* » [Yelles-Chaouche et al., xxxxa] et « *Journal of the Operational Research Society* » [Tremblet et al., xxxxb]. De plus, une partie de ce travail a été réalisée par un étudiant que j'ai co-encadré durant son stage de fin d'études de Master.

Chapitre 5

Minimisation du nombre de modules dans la conception de lignes reconfigurables multi-produits en présence de machines modulaires

---

Dans ce chapitre, un nouveau problème d'optimisation des lignes de production modulaires multi-produits est introduit et décrit. Plus particulièrement, ce problème consiste à minimiser le nombre de modules nécessaire pour fabriquer un certain nombre de produits. Ainsi, une formulation mathématique sera introduite dans un premier temps. Ensuite, une approche de décomposition est proposée pour résoudre efficacement des instances du problème de tailles moyennes. Les résultats expérimentaux seront finalement présentés.

5.1 Introduction

La modularité est l'une des caractéristiques les plus importantes des RMS. Celle-ci fournit aux RMS la capacité à s'adapter rapidement, efficacement et à moindre coût aux fluctuations de la demande et aux changements de produits. En effet, la modularité rend gérable la nature complexe d'un système en fournissant une division efficace des différentes tâches [Baldwin and Clark, 2006]. De plus, chaque module peut être changé, amélioré ou remplacé sans pour autant impacter le reste du système, offrant ainsi une certaine flexibilité qui lui permet de s'adapter dans un environnement dynamique. Une architecture modulaire d'un système de production permet aussi de mieux faire face aux aléas (tels que les pannes) puisque la perturbation d'un module n'aura pas d'impact négatifs sur les autres. Enfin, les systèmes modulaires sont moins coûteux à développer et à modifier. Cela signifie qu'au lieu de développer ou de modifier un système complexe entier, il suffit de développer ou de modifier des modules individuels.

Malgré tous les avantages de la modularité, la conception d'un système modulaire reste un problème très complexe. Dans le contexte des systèmes de production, ce problème consiste à

identifier et à concevoir les modules nécessaires pour fabriquer les produits. Chaque module pourra alors effectuer un certains nombre de tâches. Ces modules doivent ensuite être affectés aux machines qui sont compatibles d'un point de vu logiciel et mécanique (intégrabilité). Par ailleurs, lors de la conception de n'importe quel système modulaire, il est important de réduire le plus possible le nombre de modules. En effet, moins il y a de modules, moins il y a d'interfaces et, dans le cadre des RMS, moins il y a de reconfigurations et de changements à faire.

Dans ce contexte, nous considérons dans ce chapitre un problème de minimisation du nombre de modules dans une ligne de production reconfigurable multi-produit. Il est à noter que peu de travaux ont été menés dans ce sens, et que la majorité des articles sur la modularité traitent de la conception de produits modulaires en utilisant des outils d'analyse et de gestion de systèmes complexes, tels que la matrice de structure de conception. Un tel outil permet, entre autres, d'identifier les éventuels modules et non pas de les optimiser.

5.2 Description du problème

Dans cette section, un problème d'optimisation qui vise à concevoir une ligne multi-produit composée de machines modulaires est étudié. Un module est considéré comme un dispositif physique capable d'effectuer un nombre limité de tâches de fabrication. Ainsi, la reconfiguration de cette ligne consiste à réarranger les différents modules.

Plus spécifiquement, la ligne étudiée est composée d'un nombre fixe de machines disposées de façon linéaire et reliées par un système de manutention tel qu'un convoyeur. Chaque machine dispose d'un nombre limité de positions, appelées emplacements, dans lesquelles les modules peuvent être placés. Les emplacements de toutes les machines sont considérés comme compatibles avec tous les modules. Les produits se déplacent d'une machine à l'autre. Dans chaque machine, les modules en place sont activés de manière successive. Ces modules peuvent effectuer un nombre limité de tâches en séquence. La ligne étant cadencée, le temps d'exécution de chaque machine ne doit pas dépasser le temps de cycle de chaque produit. De plus, pour chaque produit, la répartition des tâches sur la ligne doit respecter un ordre partiel spécifié par un graphe de précédence acyclique orienté. Ainsi, une configuration de ligne appropriée est nécessaire pour produire chaque type de produit. Les machines étant modulaires, le passage d'une configuration à une autre nécessite une reconfiguration, qui consiste à ajouter, déplacer ou retirer des modules entre les emplacements disponibles.

La description ci-dessus met en évidence un important problème d'optimisation, qui consiste à trouver une configuration de ligne admissible pour chaque type de produit tout en minimisant le nombre total de modules utilisés. Pour ce faire, une nouvelle formulation PLNE compacte est proposée. Par la suite, une formulation PLNE étendue plus efficace est développée.

Pour mieux illustrer le problème étudié, la figure 5.1 montre un exemple de deux graphes de précédence correspondant à deux produits. On peut remarquer que les produits partagent le même ensemble de tâches à exécuter. Cependant, le graphe de précédence et les temps opératoires des tâches diffèrent d'un produit à un autre. Dans cet exemple, le temps de cycle des produits 1 et 2 est 95 et 75, respectivement.

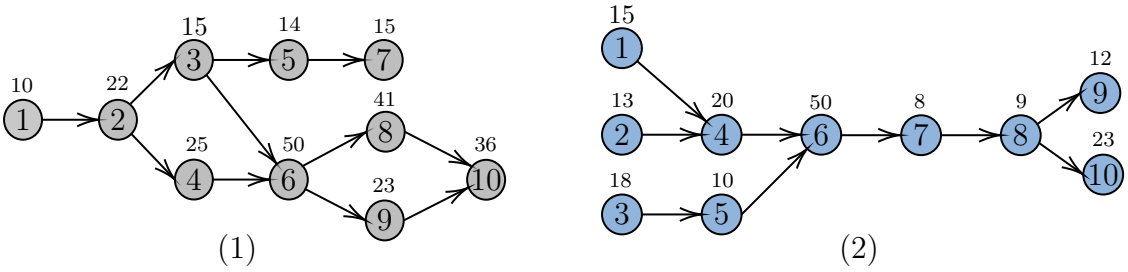


FIGURE 5.1 – (1) et (2) sont les graphes de précédence des produits 1 et 2.

La solution optimale, représentée dans la figure 5.2, montre que 6 modules sont nécessaires pour produire les deux produits. Dans cette figure, Mi fait référence au i-ième module, chacun d'entre eux pouvant effectuer un nombre limité de tâches. En utilisant ces modules, deux configurations peuvent être construites comme le montre la figure 5.3, où la configuration 1 (resp. 2) correspond à l'arrangement de ces modules dans les machines disponibles de manière à produire le produit 1 (resp. 2). Le passage de la configuration 1 à 2 s'effectue en déplaçant le module M4 de la machine 2 à la machine 3 et le module M5 de la machine 3 à la machine 2.

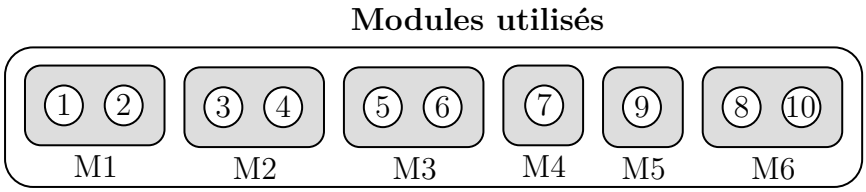


FIGURE 5.2 – Les modules optimaux générés.

Des études antérieures ont abordé le problème de l'optimisation des équipements dans les lignes d'assemblage/transfert, où un seul produit était considéré. Le présente étude est, à notre connaissance, une première tentative d'aborder ce type de problème en considérant plusieurs produits.

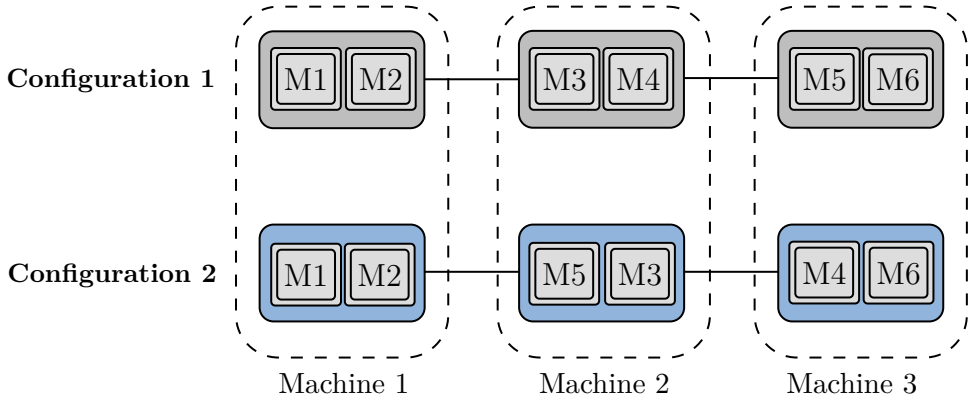


FIGURE 5.3 – Configurations optimales des produits qui minimisent le nombre total de modules.

5.3 Formulation compacte

Dans cette section, le problème d'optimisation étudié est défini et une première formulation mathématique compacte est proposée. Étant donné un nombre fixe de machines, un nombre limité d'emplacements de modules par machine, un graphe de précédence et un temps de cycle pour chaque produit, le problème consiste à générer un nombre minimal de modules qui peuvent être utilisés pour fabriquer tous les produits. Le nombre de tâches qu'un module peut effectuer est limité et donné. Pour des raisons de simplicité, et sans perte de généralités, les contraintes d'inclusion et d'exclusion « tâche-module » et « module-machine » ne sont pas considérées dans cette étude. Ainsi, toutes les tâches peuvent être exécutées par tous les modules, et tous les modules peuvent être placés dans toutes les machines. Ci-dessous, les notations, variables de décision utilisées ainsi que la formulation PLNE du problème sont présentées.

Notations :

— V est l'ensemble de toutes les tâches ;
— W est l'ensemble des machines disponibles ;
— P est l'ensemble des produits ;
— M est l'ensemble de tous les modules qui peuvent être générés ;
— $m_{\max}$ est le nombre maximum de modules par machine ;
— $r_{\max}$ est le nombre maximum de tâches par module ;
— $E = \{1, \ldots, |W| \cdot m_{\max}\}$ est l'ensemble de tous les emplacements des modules (voir figure 5.4) ;
— $E(k) = \{(k-1)m_{\max}+1, \ldots, km_{\max}\}$ est l'ensemble des emplacements des modules dans la machine $k \in W$;
— $C^{(p)}$ est le temps de cycle correspondant au produit $p \in P$;

— $t_i^{(p)}$ est le temps opératoire de la tâche $i \in V$ du produit $p \in P$;
— $G^{(p)} = (V, A^{(p)})$ représente le graphe de précédence du produit $p \in P$.

Variables de décision :

— x_{im} est égale à 1 si la tâche $i \in V$ est affectée au module $m \in M$, 0 sinon.
— $y_{me}^{(p)}$ est égale à 1 si le module $m \in M$ est placé à l'emplacement $e \in E$ dans la configuration du produit $p \in P$, 0 sinon.
— $z_{ime}^{(p)}$ est égale à 1 si la tâche $i \in V$ est affectée au module $m \in M$ qui est placé à l'emplacement $e \in E$ dans la configuration du produit $p \in P$, 0 sinon.
— s_m est égale à 1 si le module $m \in M$ contient au moins une tâche, 0 sinon.

Avant de présenter la formulation PLNE, il est important de décrire son fonctionnement global. Ainsi, comme le montre la figure 5.4, étant donné un ensemble d'emplacements dans chaque machine et un ensemble de modules, le PLNE ci-dessous vise d'abord à construire les différents modules en leur attribuant des tâches, pour ensuite les répartir dans les emplacements disponibles. Ainsi, la formulation PLNE procède de façon simultanée à l'affectation des tâches aux modules et à l'allocation des modules aux emplacements des machines.

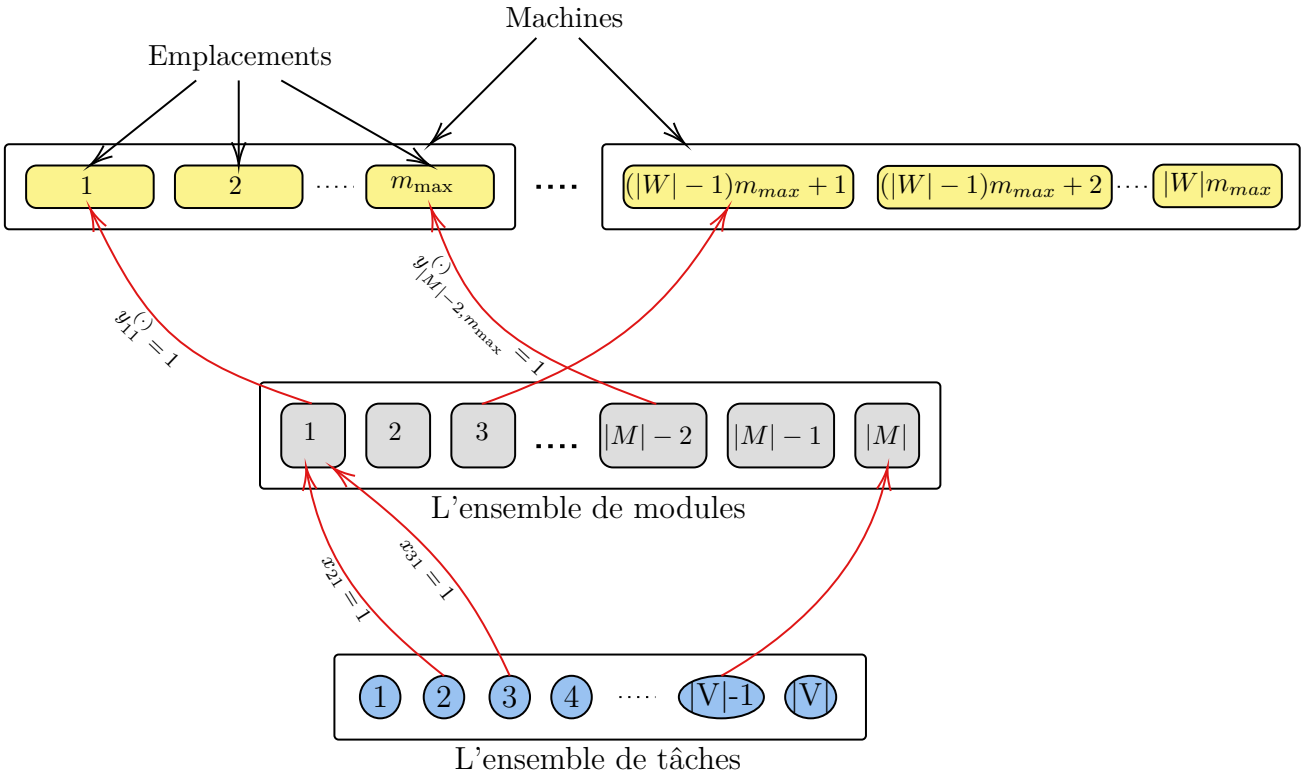


FIGURE 5.4 – Principe de fonctionnement de la formulation PLNE.

$$\min \sum_{m \in M} s_m \tag{5.1}$$

sous contraintes :

$$1 \leq \sum_{m \in M} x_{im} \leq |P|, \quad \forall i \in V \tag{5.2}$$

$$\sum_{e \in E} y_{me}^{(p)} \leq 1, \quad \forall m \in M, \quad \forall p \in P \tag{5.3}$$

$$\sum_{m \in M} y_{me}^{(p)} \leq 1, \quad \forall e \in E, \quad \forall p \in P \tag{5.4}$$

$$\sum_{m \in M} \sum_{e \in E} z_{ime}^{(p)} = 1, \quad \forall i \in V, \quad \forall p \in P \tag{5.5}$$

$$x_{im} \leq s_m, \quad \forall i \in V, \quad \forall m \in M \tag{5.6}$$

$$x_{im} + y_{me}^{(p)} \leq z_{ime}^{(p)} + 1, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P \tag{5.7}$$

$$z_{ime}^{(p)} \leq x_{im}, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P \tag{5.8}$$

$$z_{ime}^{(p)} \leq y_{me}^{(p)}, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P \tag{5.9}$$

$$\sum_{m \in M} \sum_{e \in E} e \cdot z_{ime}^{(p)} \leq \sum_{m \in M} \sum_{e \in E} e \cdot z_{jme}^{(p)}, \quad \forall (i,j) \in A^{(p)}, \quad \forall p \in P \tag{5.10}$$

$$\sum_{e \in E(k)} \sum_{m \in M} \sum_{i \in V} t_i^{(p)} \cdot z_{ime}^{(p)} \leq C^{(p)}, \quad \forall k \in W, \quad \forall p \in P \tag{5.11}$$

$$\sum_{i \in V} t_i^{(p)} \cdot x_{im} \leq C^{(p)}, \quad \forall m \in M, \quad \forall p \in P \tag{5.12}$$

$$\sum_{i \in V} x_{im} \leq r_{\max}, \quad \forall m \in M \tag{5.13}$$

$$\left\lceil \frac{|V|}{r_{\max}} \right\rceil \leq \sum_{m \in M} s_m \leq |V| \tag{5.14}$$

$$s_{m+1} \leq s_m, \quad \forall m \in M \setminus \{|M|\} \tag{5.15}$$

$$z_{ime}^{(p)} = 0, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \notin \bigcup_{k \in Q_i^{(p)}} E(k), \quad \forall p \in P \tag{5.16}$$

$$x_{im} \in \{0,1\}, \quad \forall i \in V, \quad \forall e \in E$$

$$y_{me}^{(p)} \in \{0,1\}, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P$$

$$z_{ime}^{(p)} \in \{0,1\}, \quad \forall i \in V, \quad \forall m \in M, \quad \forall e \in E, \quad \forall p \in P$$

$$s_m \in \{0,1\}, \quad \forall m \in M$$

La fonction objectif (5.1) minimise le nombre total de modules utilisés. Les contraintes (5.2) garantissent que chaque tâche est affectée à au moins un et au plus $|P|$ modules. Les contraintes (5.3) et (5.4) indiquent que pour chaque configuration, un module doit être affecté à un seul emplacement et un emplacement ne peut pas accueillir plus d'un module. Les contraintes (5.5) expriment que chaque tâche doit être présente dans une configuration. Les contraintes (5.6) indiquent qu'un module n'est pas vide si une tâche lui est affectée. Les contraintes (5.7), (5.8) et (5.9) obligent un module non vide généré à être affecté à un emplacement dans au moins une configuration. Les relations de précédence des tâches pour chaque configuration sont exprimées par les inégalités (5.10). Les contraintes (5.11) garantissent que dans chaque configuration, la somme des temps de traitement des modules affectés à l'emplacement d'une machine ne dépasse pas le temps de cycle. Les contraintes (5.12) empêchent de générer des modules dont le temps de traitement dépasse le temps de cycle correspondant à une configuration. Le nombre maximal de tâches à affecter dans un module est limité par les inégalités (5.13). Les contraintes (5.14) et (5.15) sont des inégalités valides. A savoir, (5.14) définissent les bornes inférieures et supérieures sur le nombre de modules à utiliser. Tandis que (5.15) garantissent qu'un module ne peut être généré que si le module précédent est non vide. Les contraintes (5.16) imposent que la tâche i soit affectée à un nombre restreint d'emplacements donné par l'intervalle d'affectation $Q_i^{(p)}$. Celui-ci est calculé en utilisant la même méthode introduite dans la chapitre précédent.

5.4 Formulation étendue

Nous avons présenté dans la section précédente une formulation dite compacte qui, de façon simultanée, génère les modules et les affectent aux machines. Bien que cette approche soit originale, celle-ci reste peu performante en raison du fait que le nombre de solutions possibles et, par conséquent, l'espace de recherche, sont très larges. Pour éviter cela, nous présentons dans

cette section une approche de décomposition du problème. A savoir, nous avons développé un nouvel algorithme qui consiste à générer tous les modules possibles à partir d'un graphe de précédence. Ensuite, une nouvelle formulation PLNE, dite étendue, se charge uniquement d'affecter ces modules aux machines disponibles. Des notations supplémentaires et de nouvelles variables de décision sont introduites ci-dessous.

Notations additionnelles :

— $M^{(k,p)}$ est l'ensemble des modules qui peuvent être affectés à la machine $k \in W$ dans la configuration du produit $p \in P$;

— α_{im} est égale à 1 si la tâche $i \in V$ est présente dans le module $m \in M$;

— $c_m^{(p)}$ est la charge du module $m \in M$ dans la configuration du produit $p \in P$.

Variables de décision :

— y_m est égale à 1 si le module $m \in M$ est utilisé dans au moins une configuration, 0 sinon.

— $\lambda_m^{(k,p)}$ est égale à 1 si le module $m \in M$ est affecté à la machine $k \in W$ dans la configuration du produit $p \in P$, 0 sinon.

$$\min \sum_{m \in M} y_m \tag{5.17}$$

$$\sum_{k \in W} \sum_{m \in M} \alpha_{im} \cdot \lambda_m^{(k,p)} = 1, \quad \forall i \in V, \quad \forall p \in P \tag{5.18}$$

$$\sum_{m \in M} c_m^{(p)} \cdot \lambda_m^{(k,p)} \leq C^{(p)}, \quad \forall k \in W, \quad \forall p \in P \tag{5.19}$$

$$\lambda_m^{(k,p)} \leq y_m, \quad \forall m \in M, \quad \forall k \in W, \quad \forall p \in P \tag{5.20}$$

$$\sum_{k \in W} \lambda_m^{(k,p)} \geq y_m, \quad \forall m \in M, \quad \forall p \in P \tag{5.21}$$

$$\sum_{q=k}^{|W|} \sum_{m \in M} \alpha_{im} \cdot \lambda_m^{(q,p)} \leq \sum_{q=k}^{|W|} \sum_{m \in M} \alpha_{jm} \cdot \lambda_m^{(q,p)}, \quad \forall (i,j) \in A^{(p)}, \ \forall k \in W, \ \forall p \in P \tag{5.22}$$

$$\lambda_m^{(k,p)} = 0, \ \forall m \notin M^{(k,p)}, \ \forall k \in W, \ \forall p \in P \tag{5.23}$$

$$y_m \in \{0,1\}, \quad \forall m \in M$$

$$\lambda_m^{(k,p)} \in \{0,1\}, \quad \forall m \in M, \quad \forall k \in W, \quad \forall p \in P$$

La fonction objectif (5.17) minimise le nombre total de modules sélectionnés. Les contraintes (5.18) assurent qu'une tâche i est affectée dans chaque configuration p. Les contraintes (5.19) stipulent que la somme des temps opératoire des tâches effectuées par les modules affectés à une machine ne doit pas dépasser le temps de cycle correspondant au produit p. Les contraintes (5.20) et (5.21) expriment qu'un module est considéré comme utilisé s'il est affecté à au moins une configuration. Les contraintes (5.22) assurent que la relation de précédence entre les tâches est respectée pour chaque configuration de produit.

5.4.1 Algorithme de génération de modules

La construction de l'ensemble M passe par plusieurs étapes. En utilisant l'intervalle d'affectation $Q_i^{(p)}$ (présenté dans le chapitre 4), la première étape consiste à identifier le sous-ensemble de tâches qui peut être affecté à la machine $k \in W$ pour chaque produit $p \in P$, noté $V_k^{(p)}$. Ce dernier permet d'obtenir un graphe de précédence induit, noté $G_k^{(p)} = (V_k^{(p)}, A_k^{(p)})$, où $V_k^{(p)} \subseteq V$ et $A_k^{(p)} \subseteq A^{(p)}$ est l'ensemble d'arcs entre les tâches $V_k^{(p)}$.

Dans la deuxième étape, un algorithme de génération de patterns est appliqué afin de générer $M^{(k,p)}$, pour chaque $k \in W$ de chaque produit $p \in P$, en se basant sur $G_k^{(p)}$, le temps de cycle $C^{(p)}$ et le $r_{\max}$. L'idée principale de cet algorithme consiste à identifier toutes les composantes faiblement connexes de longueur $r_{\max}$ dans $G_k^{(p)}$. A noter que ces dernières représentent des modules potentiels à utiliser. Pour ce faire, l'algorithme de Kahn [1962] est d'abord appliqué afin de définir pour chaque nœud son niveau topologique au sein de $G_k^{(p)}$. Ensuite, pour chaque tâche de chaque niveau topologique, les modules possibles sont identifiés en parcourant les tâches du même niveau ainsi que celles des niveaux suivants, tout en s'assurant de ne générer que les modules qui satisfont les contraintes de temps de cycle et de précédence. Une description plus détaillée de l'algorithme décrit ci-dessus est donnée par l'Algorithme 2.

Ainsi, $M^{(p)} = \bigcup_{k \in W} M^{(k,p)}$ est obtenu. Ce dernier représente l'ensemble de tous les modules possibles qui peuvent être utilisés dans la configuration du produit $p \in P$. Enfin, $M = \bigcup_{p \in P} M^{(p)}$ est l'ensemble des modules possibles tous produits confondus.

La figure 5.5 montre un exemple du fonctionnement de l'algorithme. À gauche de la figure, on peut voir un graphe induit à partir du graphe de précédence (1) de la figure 5.1. Étant donné $C = 100$ et $r_{\max} = 3$, les modules qui sont générés et ceux non-générés à partir du graphe induit, en utilisant l'algorithme de génération de patterns, sont représentés dans la partie droite. M1, M2, M3 et M4 sont, entre autres, des modules réalisables qui respectent les contraintes de précédence, et dont la charge ne dépasse pas le temps de cycle. Inversement, M1′, M2′ et M3′ sont le type de modules non générés par l'algorithme, car soit ils violent les contraintes de précédence (modules M1′ et M3′), soit leur charge dépasse le temps de cycle (module M2′).

Algorithme 2 : GÉNÉRATION DE PATTERNS

Entrée : Un graphe acyclique orienté G, le temps de cycle C et $r_{\max}$.
Sortie : L'ensemble $\mathcal{L}$ de modules générés.
Étape 1 : **Initialisation.**

1.1 Considérons une pile vide $\mathcal{S}$ de modules et fixons $\mathcal{L} = \emptyset$ et $k = 1$.
1.2 Définir pour chaque nœud (ou tâche) de G son niveau en utilisant l'algorithme de tri topologique de Kahn [1962]. Et admettons que $l_{\max}$ soit le plus grand niveau topologique de G.

Étape 2 : **Génération de modules.**

2.1. Soit V_k l'ensemble des tâches appartenant au même niveau topologique k. Pour chaque tâche de V_k, générer un module composé uniquement de cette tâche et l'empiler dans la pile $\mathcal{S}$.
2.2. Si $\mathcal{S}$ n'est pas vide, alors on dépile un nouveau module R de $\mathcal{S}$. Sinon, passer à l'étape 3.
2.3. Ajouter R à $\mathcal{L}$. Si $|R| < r_{\max}$, passer à l'étape 2.4. Sinon, si $|R| = r_{\max}$, aller à l'étape 2.2.
2.4. Soit j la dernière tâche ajoutée à R et l son niveau topologique respectif.
2.5. Pour chaque $i \in V_l$, générer un nouveau module $R \cup \{i\}$ et l'empiler dans $\mathcal{S}$, si $i > j$ et $\sum_{q \in R} t_q \leq C - t_i$.
2.6. Si $l + 1 \leq l_{\max}$, alors pour chaque $i \in V_{l+1}$, générer un nouveau module $R \cup \{i\}$ et le pousser vers $\mathcal{S}$, si $\sum_{q \in R} t_q \leq C - t_i$ et l'intersection entre tous les successeurs de R et tous les prédécesseurs de i sans tenir compte de $R \cup \{i\}$ est vide, *c.-à-d.*, $\Big(\Big(\bigcup_{q \in R} \mathcal{S}_q\Big) \bigcap \mathcal{P}_i\Big) \setminus \Big(R \cup \{i\}\Big) = \emptyset$.
2.7. Aller à l'étape 2.2.

Étape 3 : **Analyse.**

3.1 Si $k < l_{\max}$, alors incrémenter $k := k + 1$ et aller à l'étape 2. Sinon, stop. L'ensemble $\mathcal{L}$ des modules est généré.

5.4.2 Filtrage de modules

En analysant une solution optimale, on peut remarquer la présence d'un grand nombre de modules générés qui ne sont jamais utilisés dans aucune configuration. Par conséquent, l'identification de ces modules et leur suppression de M et $M^{(k,p)}$ pourrait réduire considérablement l'espace de recherche. En effet, soit m un module généré pour un produit donné $p \in P$. Il est facile de remarquer que si $m \notin \bigcap_{p \in P} M^{(p)}$, alors m peut être retiré de M. Pour mieux illustrer cela, la figure 5.6 montre un exemple de deux solutions réalisables 1 et 2 d'un problème avec trois produits et $r_{\max} = 2$. Le nombre total de modules utilisés dans la solution réalisable 1 est de 4, alors que 3 modules seulement sont nécessaires comme indiqué dans la solution réalisable 2. Ainsi, des modules supplémentaires sont à prévoir pour la dernière configuration. Cette situation

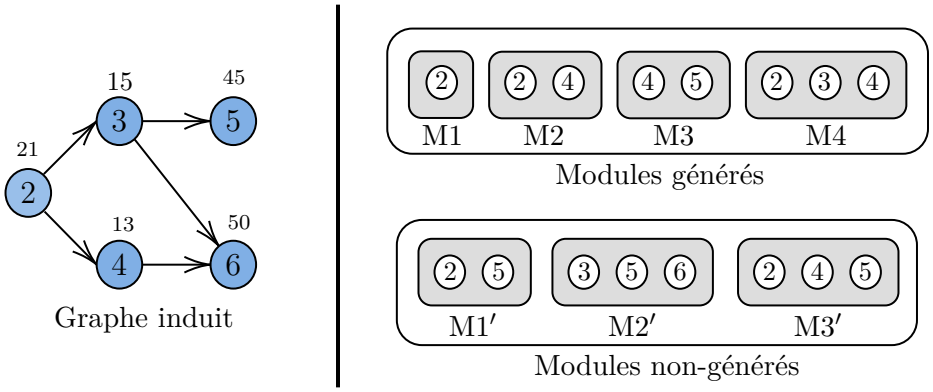


FIGURE 5.5 – Un exemple de modules (non-)générés à partir d'un graphe induit avec $C = 100$ et $r_{\max} = 3$.

peut être évitée en ne considérant, dès le début, que les modules qui sont en communs à toutes les configurations. Cela nous permet de réduire l'espace de recherche et donc d'atteindre plus rapidement la deuxième solution réalisable. Par conséquent, $M = \bigcap_{p \in P} M^{(p)}$.

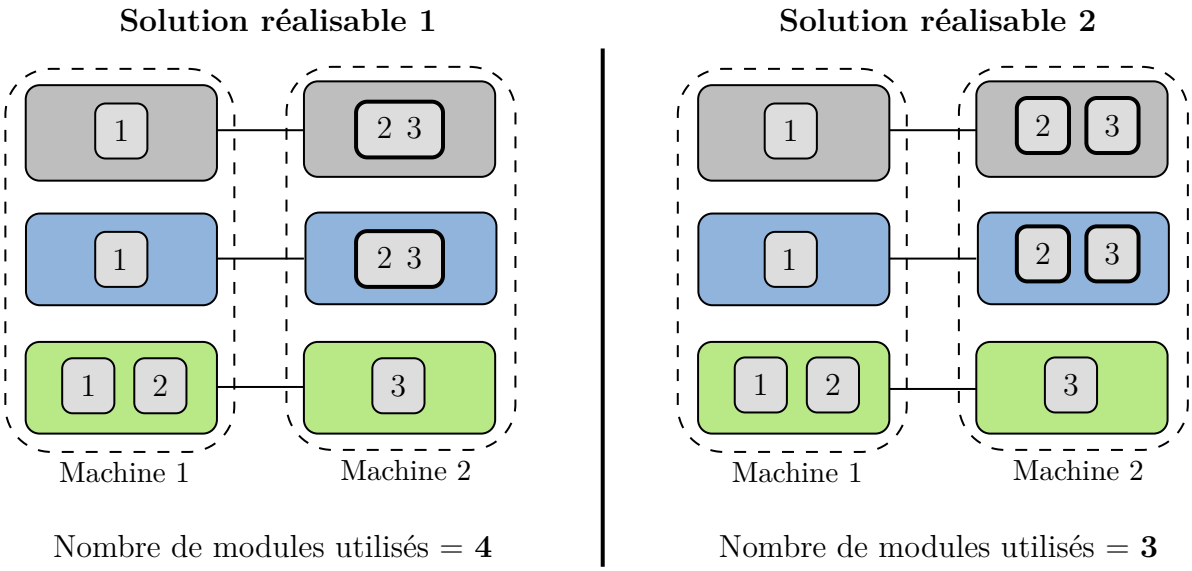


FIGURE 5.6 – Deux solutions réalisables (gauche et droite) ayant un nombre différent de modules utilisés au total

5.5 Résultats expérimentaux

Dans cette section, les résultats expérimentaux des formulations PLNE compacte et étendue sont présentés et comparés.

Ces expérimentations ont été menées en utilisant les instances de référence bien connues de Otto et al. [2013]. Pour la formulation compacte, seul un ensemble d'instances de petite taille ($|V| = 20$) a été utilisé. Alors que deux ensembles d'instances de taille petite et moyenne ($|V| = 20$ et $|V| = 50$) ont été considérés pour la formulation étendue. Ceci est justifié par le fait que la formulation compacte a montré de mauvaises performances pour les instances de petite taille et donc nous n'avons pas jugé nécessaire d'aller plus loin. De la même façon que dans le chapitre précédent, chaque ensemble d'instances est regroupé en trois séries distinctes sur la base de la densité de leurs graphes de précédence (OS). Ainsi, la première série correspond aux instances ayant un $OS \approx 0.2$, la seconde à celles ayant un $OS \approx 0.6$, et la dernière à celles ayant un $OS \approx 0.8$. Ainsi, pour construire une instance avec trois produits, il suffit de prendre trois instances successives appartenant à la même série.

Pour chaque instance, le temps de cycle $C^{(p)}$ de chaque produit $p \in P$ est fixé à $\left\lceil 1.5 \cdot \max_{i \in V} t_i^{(p)} \right\rceil$, le nombre de machines est fixé à

$$|W| = \max_{p \in P} \left\{ \left\lceil 1.2 \cdot \sum_{i \in V} t_i^{(p)} / C^{(p)} \right\rceil \right\},$$

et le $m_{\max}$ est calculé comme suit

$$m_{\max} = \max_{p \in P} \max \left\{ k \mid \sum_{i=1}^{k} t_{\pi_i}^{(p)} \leq C^{(p)} \right\},$$

où $(\pi_1, \pi_2, \ldots, \pi_{|V|})$ est une permutation de V selon l'ordre croissant de leurs temps opératoires correspondants au produit $p \in P$.

Les deux formulations proposées ont été résolues en utilisant CPLEX 12.10.0 installé sur un ordinateur Intel(R) Core(TM) i7-8650U à 1.90 GHz avec 32 Go de RAM. Les résultats obtenus pour les formulations compacte et étendue sont présentés dans les tableaux 5.1 et 5.2, respectivement. Dans les deux tableaux, la première colonne indique le nombre de tâches, la deuxième exprime le nombre maximal de tâches par module, la troisième colonne correspond au nombre de produits et la quatrième à la série d'OS. Le nombre total d'instances dans chaque série est indiqué dans la cinquième colonne. Le nombre d'instances résolues à l'optimalité et leur temps `CPU` moyen sont donnés dans les sixième et septième colonnes, respectivement. Enfin, la dernière colonne indique le `GAP` moyen pour les instances restantes qui n'ont pas été résolues de manière optimale dans le temps de résolution limité à 600 secondes. Comme le montre l'équation ci-dessous, le calcul du `GAP` se base sur la borne inférieure (LB) et la borne supérieure (UB) qui représente la meilleure solution trouvée.

$$\mathtt{GAP}_{\mathtt{E}}^{\mathtt{B}} = 100\% \cdot \frac{(\mathtt{UB}_{\mathtt{E}} - \mathtt{UB}_{\mathtt{B}})}{\mathtt{UB}_{\mathtt{B}}}$$

Tableau 5.1 – Résultats numériques pour la formulation PLNE compacte.

$\|V\|$	$r_{\max}$	$\|P\|$	OS	#`Instances`	#`OPT`	Moy. `CPU`, (s.)	Moy. `GAP`, (%)
20	2	2	1	3	1	165.30	21.20
			2	199	6	298.73	31.89
			3	68	0	⊖	39.52
		3	1	2	0	⊖	48.70
			2	193	0	⊖	46.39
			3	65	0	⊖	74.87
	3	2	1	3	1	162.00	38.55
			2	199	5	261.57	40.54
			3	68	0	⊖	49.51
		3	1	2	0	⊖	46.05
			2	193	0	⊖	59.03
			3	68	0	⊖	60.07

(⊖) Aucune solution optimale n'a été trouvée dans le temps imparti de 600 secondes.

En analysant le tableau 5.1, il est évident que la formulation PLNE compacte n'est pas efficace. En effet, 99% des instances n'ont pas été résolues de manière optimale. De plus, le `GAP` moyen obtenu est relativement élevé par rapport à la taille des instances. Il est également intéressant d'observer que plus l'OS des instances est élevé, plus le `GAP` moyen est élevé. Ceci s'explique par le fait que le nombre de modules générés est plus important lorsque la densité d'un graphe de précédence augmente, ce qui implique que l'espace de recherche devient plus grand. Au vu de ces résultats, il n'a pas été nécessaire de tester le modèle compact sur des instances plus grandes.

Contrairement au modèle compact, le modèle étendu montre de meilleures performances sur les mêmes instances. Ceci est visible dans le tableau 5.2, où toutes les instances correspondantes à $|V| = 20$ ont été résolues de manière optimale en moins d'une seconde en moyenne. Quant aux instances de taille moyenne correspondant à $|V| = 50$, la formulation étendue reste efficace dans le sens où elle peut résoudre un certain nombre d'instances à l'optimalité. Par exemple, 60% des instances correspondantes à $r_{\max} = 2$ ont été résolues en moins de 200 secondes en moyenne. Les 40% d'instances non résolues ont fourni un `GAP` moyen raisonnable. On peut également remarquer que le nombre de produits affecte le temps de résolution ainsi que la qualité du `GAP` moyen. Ce comportement est naturel puisque l'augmentation du nombre de produits implique la génération de plus de modules. Enfin, la formulation étendue semble atteindre ses limites lors du traitement des instances correspondantes à $r_{\max} = 3$, où seulement 37% des instances ont été résolues de manière optimale. Le `GAP` moyen fourni pour les instances non résolues est élevé, notamment avec $|P| = 2$ où il atteint 59,30%.

Tableau 5.2 – Résultats numériques pour la formulation PLNE étendue.

$\|V\|$	$r_{\max}$	$\|P\|$	OS	#Instances	#OPT	Moy. CPU, (s.)	Moy. GAP, (%)
20	2	2	1	3	3	0.46	⊕
			2	199	199	0.35	⊕
			3	68	68	0.13	⊕
		3	1	2	2	1.03	⊕
			2	193	193	0.90	⊕
			3	65	65	0.23	⊕
	3	2	1	3	3	0.58	⊕
			2	199	199	0.86	⊕
			3	68	68	0.20	⊕
		3	1	2	2	0.79	⊕
			2	193	193	1.24	⊕
			3	65	65	0.23	⊕
50	2	2	1	15	8	146.80	8.06
			2	219	182	198.99	7.95
			3	74	73	54.09	3.80
		3	1	14	0	⊖	26.44
			2	218	42	224.43	17.37
			3	73	65	165.72	6.84
	3	2	1	11	4	179.38	59.30
			2	219	67	186.14	56.30
			3	74	71	89.11	6.90
		3	1	14	0	⊖	38.93
			2	218	28	215.12	38.44
			3	73	53	225.15	13.38

(⊕) Toutes les solutions optimales ont été trouvées dans le temps imparti de 600 secondes.
(⊖) Aucune solution optimale n'a été trouvée dans le temps imparti de 600 secondes.

5.6 Conclusion

Nous avons présenté et étudié dans ce chapitre un nouveau problème d'optimisation qui vise à concevoir une ligne de production multi-produit modulaire. Celle-ci est composée d'un ensemble de machines où des modules peuvent être placés. Ainsi, la ligne peut être facilement reconfigurée en ajoutant, déplaçant ou enlevant des modules pour passer de la production d'un produit vers un autre. L'objectif consiste donc à trouver le nombre minimum de modules pour fabriquer un ensemble de produits donné. Ceci revient à trouver le nombre maximum de modules

en commun entre les différentes configurations.

Deux formulation mathématiques sous forme de PLNE sont proposées pour résoudre ce problème. La première, dite formulation compacte, consiste à générer des modules au sein du modèle PLNE et à les affecter aux machines simultanément. La seconde, appelée formulation étendue, se base sur la décomposition du problème en générant les modules ad hoc et le PLNE se charge ensuite de les affecter aux machines. Pour ce faire, nous avons développé un nouvel algorithme qui se base sur le graphe de précédence pour générer les modules en identifiant toutes les composantes faiblement connexes du graphe.

Les résultats expérimentaux ont montré que l'approche compacte n'est pas très performante et n'a pu résoudre que très peu d'instances de petite taille à l'optimalité. En revanche, la formulation étendue est parvenue à résoudre en quelques secondes seulement toutes ces instances. Ceci étant dit, elle atteint ses limites lorsqu'il s'agit de problèmes de grande taille.

L'une des perspectives les plus intéressantes pour ce problème vise à développer une heuristique qui consiste à résoudre le problème, de façon itérative, pour un sous-ensemble de machine seulement. A la fin de chaque itération, certains modules sont fixés sur une machine et une contrainte correspondante sera ajoutée pour l'itération suivante. Ainsi, un grand nombre de modules sera écarté au fur et à mesure que l'heuristique avance permettant de réduire, après chaque itération, l'espace de recherche. Une autre extension possible vise à ajouter des contraintes technologiques telles que les contraintes d'inclusion et d'exclusion entre les tâches et les modules et les contraintes de compatibilité des modules sur les machines. Ceci permettrait de mieux refléter la réalité du terrain.

Les résultats obtenus dans cette section ont été présentés dans la conférence PMS 2021 [Yelles-Chaouche et al., 2021b]. De plus, un article de journal est en cours de rédaction pour le soumettre à la revue « *Computers & Operations Research* ».

CONCLUSION GÉNÉRALE ET PERSPECTIVES DE RECHERCHE

Cette thèse a été menée dans le cadre du concept des systèmes de production reconfigurables (RMS). Ainsi, dans le présent manuscrit nous avons étudié le problème de conception de lignes de production reconfigurables multi-produits. Contrairement aux systèmes de production dédiés (DML) et flexibles (FMS), les RMS sont conçus avec une flexibilité personnalisée leur permettant de traiter plusieurs produits, avec une capacité de production relativement élevée. Une description détaillée des RMS et de ses composants, ainsi qu'une comparaison entre DML, FMS et RMS ont été données dans le chapitre 1.

Dans le chapitre 2, nous avons fourni un état de l'art des RMS. Une attention particulière a été portée sur les problèmes d'optimisation qui sont soulevés dans la littérature. De ce fait, les mesures les plus utilisées pour évaluer les performances des RMS ont été introduites dans un premier temps. Ensuite, les approches de résolution développées pour résoudre les différents problèmes d'optimisation ont brièvement été présentées. Enfin, une classification de ces problèmes sur la base des objectifs considérés a été proposée. De cette dernière, nous avons observé que le problème de conception des RMS a reçu plus d'attention comparé aux autres. Par ailleurs, peu de travaux ont été menés sur le problème d'équilibrage de lignes reconfigurables. Ce dernier est crucial lors de la conception de n'importe quel type de systèmes de production.

Ainsi, dans le chapitre 3, le problème d'équilibrage de lignes d'assemblage (ALBP) a été présenté en détail. Les différentes hypothèses et certaines notions importantes de ce problème ont été introduites. Ceci nous a permis de définir les différents types du ALBP, à savoir, le problème d'équilibrage de lignes d'assemblage simples (SALBP), le problème d'équilibrage de lignes d'assemblage à modèles mixtes (MiMALBP), et finalement le problème d'équilibrage de lignes d'assemblage à modèles multiples (MuMALBP). Pour chacun de ces types de problèmes, nous avons fourni un aperçu de la littérature en présentant les différentes fonctions objectifs considérées et les approches de résolution les plus utilisées. Une attention particulière est donnée pour le MuMALBP où nous avons remarqué que peu de travaux ont été effectués dans la littérature.

Dans ce contexte, nous avons introduit le premier problème étudié qui vise à concevoir une ligne de production reconfigurable multi-produit. En effet, nous avons expliqué que les RMS permettent une reconfiguration rapide et efficace et que dans une ligne à modèle multiple il

est nécessaire d'avoir ces caractéristiques lorsque la configuration change. Ainsi, nous avons présenté dans le chapitre 4 un nouveau problème d'optimisation qui consiste à minimiser le nombre total de réaffectations de tâches lors du passage d'une configuration vers une autre. Nous avons ainsi étudié deux cas possibles, dans le premier, nous avons supposé que l'ordre d'arrivée des produits est connu à l'avance. Dans le second, par contre, nous avons considéré qu'aucune information n'est donnée sur l'ordre d'arrivée des produits. Pour ces deux cas, nous avons donc développé, dans un premier temps, une formulation PLNE renforcée par deux méthodes pour de calcul des intervalles d'affectation des tâches. De plus, nous avons proposé une nouvelle heuristique, appelée Halt-and-Fix. Celle-ci se base sur le modèle PLNE et consiste à identifier une solution réalisable trouvée par le solveur en un temps relativement réduit, et à générer de nouvelles contraintes forçant certaines tâches à être affectées dans une même station. Les résultats expérimentaux ont montré que l'approche heuristique développée fournit des résultats très compétitifs et surpasse l'approche PLNE seule, en termes de `GAP` et de temps `CPU`, pour les instances de grandes tailles.

Dans le chapitre 5, nous nous sommes intéressés aux aspects de modularité dans la conception de lignes de production multi-produits. Ceci peut être vu comme une extension du problème étudié dans le chapitre 4. À savoir, au lieu de considérer que chaque tâche est modulaire individuellement et peut être réaffectée entre les stations, nous supposons que plusieurs tâches peuvent être regroupées en un seul module. Ainsi, en déplaçant ce module entre les stations, toutes les tâches qui le composent seront déplacées en conséquence. Cette vision soulève un important problème d'optimisation, qui vise à trouver le nombre minimum de modules nécessaires pour fabriquer plusieurs produits. Nous avons donc proposé une première formulation PLNE, dite compacte, qui consiste à trouver pour chaque produit une configuration admissible qui minimise le nombre total de modules utilisés. Ce PLNE vise donc à affecter simultanément les tâches aux modules et les modules aux stations. Ce problème est fortement combinatoire ce qui fait que sa formulation compacte a montré de faibles performances même pour résoudre des instances de petites tailles. Pour cela, nous avons proposé une approche de décomposition qui se base sur un algorithme de génération de modules et d'une formulation PLNE, dite étendue. Les résultats expérimentaux ont montré que la deuxième approche est nettement plus efficace que la première. Néanmoins, celle-ci atteint ses limites lorsque la taille des instances augmente.

Les travaux de recherche et l'analyse de l'état de l'art effectués dans cette thèse nous permettent de constater que les RMS fournissent une nouvelle vision plus adaptée pour répondre aux exigences et fluctuations du marché actuel. Toutefois, atteindre cette vision nécessite des restructurations importantes des lignes de production aux niveaux stratégique, tactique et opérationnel. Même si d'importants travaux de recherche ont été effectués dans ce sens, un certain nombre de questions et de pistes restent cependant inexplorées. Ainsi, sur la base des problèmes

étudiées dans cette thèse, nous proposons des pistes de recherche intéressantes, parmi lesquelles :

- Développer de nouvelles approches d'optimisation pour le MuMALBP qui reste peu étudié malgré son importance. En effet, nous avons remarqué que les études réalisées sur ce problème se basent sur des cas industriels, et qu'il n'existe que peu d'études théoriques qui tentent de fournir des approches de résolution efficaces.
- Aborder le problème de minimisation du nombre de réaffectations de tâches dans un contexte de maintenance prédictive. Ce type de problème a été traité par Sancı and Azizoğlu [2017] où les auteurs considèrent que la configuration initiale de la ligne est donnée et qu'une nouvelle configuration doit être trouvée après qu'une station tombe en panne ou est indisponible. Il s'agit ici d'une maintenance curative. Or, il est possible d'envisager la conception d'une configuration de ligne résiliente qui pourrait faire face à la défaillance d'une station.
- Étudier le problème de conception de lignes reconfigurables multi-produits en considérant que la réaffectation de tâches est effectuée par des véhicules-robots autonomes ou par des opérateurs. Cette problématique est une extension directe du problème étudié dans le chapitre 4. Elle consiste à trouver la meilleure séquence de réaffectations de tâches qui minimise la distance totale parcourue.
- Explorer la possibilité d'introduire la modularité dans les autres types de lignes de production. Cette caractéristique étant importante, elle peut être considérée dans le SALBP et le MiMALBP, par exemple.

BIBLIOGRAPHIE

M. Abbasi and M. Houshmand. Production planning of reconfigurable manufacturing systems with stochastic demands using tabu search. *International Journal of Manufacturing Technology and Management*, 17(1/2) :125–148, 2009.

M. Abbasi and M. Houshmand. Production planning and performance optimization of reconfigurable manufacturing systems using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 54(1-4) :373–392, 2010.

M.R. Abdi and A.W. Labib. Feasibility study of the tactical design justification for reconfigurable manufacturing systems using the fuzzy analytical hierarchical process. *International Journal of Production Research*, 42(15) :3055–3076, 2004.

T. Aljuneidi and A.A. Bulgak. A mathematical model for designing reconfigurable cellular hybrid manufacturing-remanufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 87(5-8) :1585–1596, 2016.

T. Aljuneidi and A.A. Bulgak. Designing a cellular manufacturing system featuring remanufacturing, recycling, and disposal options : A mathematical modeling approach. *CIRP Journal of Manufacturing Science and Technology*, 19 :25–35, 2017.

L. Alting and H. Zhang. Computer aided process planning : the state-of-the-art survey. *International Journal of Production Research*, 27(4) :553–585, 1989.

A.-L. Andersen, T.D. Brunoe, and K. Nielsen. Reconfigurable manufacturing on multiple levels : Literature review and research directions. In S. Umeda, M. Nakano, H. Mizuyama, N. Hibino, D. Kiritsis, and G. von Cieminski, editors, *APMS 2015 : Advances in Production Management Systems : Innovative Production Management Towards Sustainable Growth*, volume 459 of *IFIP Advances in Information and Communication Technology*, pages 266–273. Springer, Cham, 2015.

M. Ashraf and F. Hasan. Configuration selection for a reconfigurable manufacturing flow line involving part production with operation constraints. *The International Journal of Advanced Manufacturing Technology*, 98(5-8) :2137–2156, 2018.

J.A. Asl, M. Solimanpur, and R. Shankar. Multi-objective multi-model assembly line balancing problem : a quantitative study in engine manufacturing industry. *OPSEARCH*, 56(3) :603–627, 2019.

C.Y. Baldwin and K.B. Clark. Modularity in the design of complex engineering systems. In *Understanding Complex Systems*, pages 175–205. Springer Berlin Heidelberg, 2006.

J.F. Bard. Assembly line balancing with parallel workstations and dead time. *International Journal of Production Research*, 1989.

O. Battaïa and A. Dolgui. A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, 142(2) :259–277, 2013.

O. Battaïa, A. Dolgui, and N. Guschinsky. Decision support for design of reconfigurable rotary machining systems for family part production. *International Journal of Production Research*, 55(5) :1368–1385, 2016.

O. Battaïa and A. Dolgui. A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, 142(2) :259–277, 2013.

J. Bautista and J. Pereira. Procedures for the time and space constrained assembly line balancing problem. *European Journal of Operational Research*, 212(3) :473–481, 2011.

İ. Baybars. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, 32(8) :909–932, 1986.

C. Becker and A. Scholl. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3) :694–715, 2006.

H.H. Benderbal, M. Dahane, and L. Benyoucef. Flexibility-based multi-objective approach for machines selection in reconfigurable manufacturing system (RMS) design under unavailability constraints. *International Journal of Production Research*, 55(20) :6033–6051, 2017a.

H.H. Benderbal, M. Dahane, and L. Benyoucef. Modularity assessment in reconfigurable manufacturing system (RMS) design : An archived multi-objective simulated annealing-based approach. *The International Journal of Advanced Manufacturing Technology*, 94(1-4) :729–749, 2017b.

A. Bensmaine, L. Benyoucef, and D. Dahane. A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers & Industrial Engineering*, 66(3) :519–524, 2013.

A. Bensmaine, L. Benyoucef, and D. Dahane. A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems. *International Journal of Production Research*, 52(12) :3583–3594, 2014.

P.A. Borisovsky, X. Delorme, and A. Dolgui. Genetic algorithm for balancing reconfigurable machining lines. *Computers & Industrial Engineering*, 66(3) :541–547, 2013a.

P.A. Borisovsky, X. Delorme, and A. Dolgui. Balancing reconfigurable machining lines via a set partitioning model. *International Journal of Production Research*, 52(13) :4026–4036, 2013b.

M. Bortolini, F. G. Galizia, and C. Mora. Reconfigurable manufacturing systems : Literature review and research trend. *Journal of Manufacturing Systems*, 49 :93–106, 2018.

N. Boysen, M. Fliedner, and A. Scholl. A classification of assembly line balancing problems. *European Journal of Operational Research*, 183(2) :674–693, 2007.

N. Boysen, M. Fliedner, and A. Scholl. Assembly line balancing : Which model to use when ? *International Journal of Production Economics*, 111(2) :509–528, 2008.

N. Boysen, M. Fliedner, and A. Scholl. Sequencing mixed-model assembly lines : Survey, classification and model critique. *European Journal of Operational Research*, 192(2) :349–373, 2009a.

N. Boysen, M. Fliedner, and A. Scholl. Production planning of mixed-model assembly lines : overview and extensions. *Production Planning & Control*, 20(5) :455–471, 2009b.

N. Boysen, M. Fliedner, and A. Scholl. Assembly line balancing : Joint precedence graphs under high product variety. *IIE Transactions*, 41(3) :183–193, 2009c.

N. Brahimi, A. Dolgui, E. Gurevsky, and A.R. Yelles-Chaouche. A literature review of optimization problems for reconfigurable manufacturing systems. *IFAC-PapersOnLine*, 52(13) : 433–438, 2019.

A. Bryan, S. Jack Hu, and Y. Koren. Assembly system reconfiguration planning. *Journal of Manufacturing Science and Engineering*, 135(4) :13 pages, 2013.

G.M. Buxey, N.D. Slack, and R. Wild. Production flow line system design - a review. *AIIE Transactions*, 5(1) :37–48, 1973.

H.J. Carlo, J.P. Spicer, and A. Rivera-Silva. Simultaneous consideration of scalable-reconfigurable manufacturing system investment and operating costs. *Journal of Manufacturing Science and Engineering*, 134(1) :8 pages, 2012.

A.K. Chakravarty and A. Shtub. Balancing mixed model lines with in-process inventories. *Management Science*, 31(9) :1161–1174, 1985.

A. Chaube, L. Benyoucef, and M.K. Tiwari. An adapted NSGA-2 algorithm based dynamic process plan generation for a reconfigurable manufacturing system. *Journal of Intelligent Manufacturing*, 23(4) :1141–1155, 2010.

J.C. Chen, Y.-Y. Chen, T.-L. Chen, and Y.-H. Kuo. Applying two-phase adaptive genetic algorithm to solve multi-model assembly line balancing problems in TFT–LCD module process. *Journal of Manufacturing Systems*, 52 :86–99, 2019.

Y.-C. Choi and P. Xirouchakis. A holistic production planning approach in a reconfigurable manufacturing system with energy consumption and environmental effects. *International Journal of Computer Integrated Manufacturing*, 28(4) :379–394, 2014.

M.K. Christensen, M.N. Janardhanan, and P. Nielsen. Heuristics for solving a multi-model robotic assembly line balancing problem. *Production & Manufacturing Research*, 5(1) :410–424, 2017.

M. Decker. Capacity smoothing and sequencing for mixed-model lines. *International Journal of Production Economics*, 30-31 :31–42, 1993.

A.M. Deif and W. ElMaraghy. Effect of reconfiguration costs on planning for capacity scalability in reconfigurable manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 18(3) :225–238, 2006a.

A.M. Deif and W. ElMaraghy. Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system. *The International Journal of Advanced Manufacturing Technology*, 32(5-6) :557–562, 2006b.

J. Dou, X. Dai, and Z. Meng. Graph theory-based approach to optimize single-product flow-line configurations of RMS. *The International Journal of Advanced Manufacturing Technology*, 41(9-10) :916–931, 2008.

J. Dou, X. Dai, and Z. Meng. Optimisation for multi-part flow-line configuration of reconfigurable manufacturing system using GA. *International Journal of Production Research*, 48(14) :4071–4100, 2009a.

J. Dou, X. Dai, and Z. Meng. A GA-based approach for optimizing single-part flow-line configurations of RMS. *Journal of Intelligent Manufacturing*, 22(2) :301–317, 2009b.

J. Dou, J. Li, and C. Su. Bi-objective optimization of integrating configuration generation and scheduling for reconfigurable flow lines using NSGA-II. *The International Journal of Advanced Manufacturing Technology*, 86(5-8) :1945–1962, 2016.

J.P. Dou, X. Dai, and Z. Meng. Precedence graph-oriented approach to optimise single-product flow-line configurations of reconfigurable manufacturing system. *International Journal of Computer Integrated Manufacturing*, 22(10) :923–940, 2009c.

I. Eguia, J.C. Molina, S. Lozano, and J. Racero. Cell design and multi-period machine loading in cellular reconfigurable manufacturing systems with alternative routing. *International Journal of Production Research*, 55(10) :2775–2790, 2016.

W.H. Elmaraghy, O.A. Nada, and H.A. Elmaraghy. Quality prediction for reconfigurable manufacturing systems via human error modelling. *International Journal of Computer Integrated Manufacturing*, 21(5) :584–598, 2008.

E. Falkenauer. Line balancing in the real world. In *Product lifecycle management : emerging solutions and challenges for global networked enterprise*. Inderscience Enterprises, 2005.

M. Gadalla and D. Xue. Recent advances in research on reconfigurable machine tools : a literature review. *International Journal of Production Research*, 55(5) :1440–1454, 2016.

R. Galan, J. Racero, I. Eguia, and J.M. Garcia. A systematic approach for product families formation in reconfigurable manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 23(5) :489–502, 2007.

J. Gao, L. Sun, L. Wang, and M. Gen. An efficient approach for type II robotic assembly line balancing problems. *Computers & Industrial Engineering*, 56(3) :1065–1080, 2009.

M.M. Ghotboddini, M. Rabbani, and H. Rahimian. A comprehensive dynamic cell formation design : Benders' decomposition approach. *Expert Systems with Applications*, 38(3) :2478–2488, 2011.

K.K. Goyal and P.K. Jain. Design of reconfigurable flow lines using MOPSO and maximum deviation theory. *The International Journal of Advanced Manufacturing Technology*, 84 : 1587–1600, 2015.

K.K. Goyal, P.K. Jain, and M. Jain. Optimal configuration selection for reconfigurable manufacturing system using NSGA II and TOPSIS. *International Journal of Production Research*, 50(15) :4175–4191, 2012.

K.K. Goyal, P.K. Jain, and M. Jain. A comprehensive approach to operation sequence similarity based part family formation in the reconfigurable manufacturing system. *International Journal of Production Research*, 51(6) :1762–1776, 2013.

N. Grangeon, P. Leclaire, and S. Norre. Heuristics for the re-balancing of a vehicle assembly line. *International Journal of Production Research*, 49(22) :6609–6628, 2011.

T.J. Greene and R.P. Sadowski. A review of cellular manufacturing assumptions, advantages and design techniques. *Journal of Operations Management*, 4(2) :85–97, 1984.

X. Guan, X. Dai, B. Qiu, and J. Li. A revised electromagnetism-like mechanism for layout design of reconfigurable manufacturing system. *Computers & Industrial Engineering*, 63(1) : 98–108, 2012.

A. Gupta, P.K. Jain, and D. Kumar. A novel approach for part family formation for reconfiguration manufacturing system. *OPSEARCH*, 51(1) :76–97, 2013.

H. Haddou Benderbal and L. Benyoucef. Machine layout design problem under product family evolution in reconfigurable manufacturing environment : a two-phase-based AMOSA approach. *The International Journal of Advanced Manufacturing Technology*, 104(1-4) :375–389, 2019.

R.B. Handfield and M.D. Pagell. An analysis of the diffusion of flexible manufacturing systems. *International Journal of Production Economics*, 39(3) :243–253, 1995.

S.E. Hashemi-Petroodi, A. Dolgui, S. Kovalev, M.Y. Kovalyov, and S. Thevenin. Workforce reconfiguration strategies in manufacturing systems : a state of the art. *International Journal of Production Research*, 59(22) :6721–6744, 2021.

W.B. Helgeson and D.P. Birnie. Assembly line balancing using the ranked positional weight technique. *Journal of Industrial Engineering*, 12(6) :394–398, 1961.

T.R. Hoffmann. Eureka : A hybrid system for assembly line balancing. *Management Science*, 38(1) :39–47, 1992.

W.J. Hopp and M.L. Spearman. *Factory Physics*. McGraw-Hill/Irwin, 2 edition, 2000.

H. Hosseini-Nasab, S. Fereidouni, S.M.T. Fatemi Ghomi, and M.B. Fakhrzad. Classification of facility layout problems : a review study. *The International Journal of Advanced Manufacturing Technology*, 94(1-4) :957–977, 2017.

R. V. Johnson. Optimally balancing large assembly lines with “fable”. *Management Science*, 34 (2) :240–253, 1988.

M.A. Kabir and M.T. Tabucanon. Batch-model assembly line balancing : A multiattribute decision making approach. *International Journal of Production Economics*, 41(1-3) :193–201, 1995.

A.B. Kahn. Topological sorting of large networks. *Communications of the ACM*, 5(11) :558–562, 1962.

R. Katz. Design principles of reconfigurable machines. *The International Journal of Advanced Manufacturing Technology*, 34(5-6) :430–439, 2006.

A. Kimms. Minimal investment budgets for flow line configuration. *IIE Transactions*, 32(4) : 287–298, 2000.

Y. Koren. *The Global Manufacturing Revolution.* John Wiley & Sons, Inc., 2010.

Y. Koren. The emergence of reconfigurable manufacturing systems (RMSs). In *Reconfigurable Manufacturing Systems : From Design to Implementation.* Springer International Publishing, 2020.

Y. Koren and M. Shpitalni. Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems*, 29(4) :130–141, 2010.

Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. Reconfigurable manufacturing systems. *CIRP Annals*, 48(2) :527–540, 1999.

Y. Koren, W. Wang, and X. Gu. Value creation through design for scalability of reconfigurable manufacturing systems. *International Journal of Production Research*, 55(5) :1227–1242, 2016.

J. Koskinen, C. Raduly-Baka, M. Johnsson, and O. S. Nevalainen. Rolling horizon production scheduling of multi-model PCBs for several assembly lines. *International Journal of Production Research*, 58(4) :1052–1073, 2020.

P. Kostal and K. Velisek. *Flexible Manufacturing System.* Zenodo, 2011.

A. Kumar, L.N. Pattanaik, and R. Agrawal. Optimal sequence planning for multi-model reconfigurable assembly systems. *The International Journal of Advanced Manufacturing Technology*, 100(5-8) :1719–1730, 2018.

A.H. Land and A.G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3) :497, 1960.

R.G. Landers, B.-K. Min, and Y. Koren. Reconfigurable machine tools. *CIRP Annals*, 50(1) : 269–274, 2001.

S.D. Lapierre, L. Debargis, and F. Soumis. Balancing printed circuit board assembly line systems. *International Journal of Production Research*, 38(16) :3899–3911, 2000.

L. Liao. Construction and comparison of multi-model and mixed-model assembly lines balancing problems with bi-objective. *Journal of Industrial and Production Engineering*, 31(8) :483–490, 2014.

M. Liu, L. An, J. Zhang, F. Chu, and C. Chu. Energy-oriented bi-objective optimisation for a multi-module reconfigurable manufacturing system. *International Journal of Production Research*, 57(19) :5974–5995, 2018.

S.-T. Liu. A fuzzy DEA/AR approach to the selection of flexible manufacturing systems. *Computers & Industrial Engineering*, 54(1) :66–76, 2008.

F. Makssoud, O. Battaïa, and A. Dolgui. An exact optimization approach for a transfer line reconfiguration problem. *The International Journal of Advanced Manufacturing Technology*, 72(5-8) :717–727, 2014.

F. Makssoud, O. Battaïa, A. Dolgui, K. Mpofu, and O. Olabanji. Re-balancing problem for assembly lines : new mathematical model and exact solution method. *Assembly Automation*, 35(1) :16–21, 2015.

M. Maniraj, V. Pakkirisamy, and P. Parthiban. Optimisation of process plans in reconfigurable manufacturing systems using ant colony technique. *International Journal of Enterprise Network Management*, 6(2) :125–138, 2014.

M. Maniraj, V. Pakkirisamy, and R. Jeyapaul. An ant colony optimization–based approach for a single-product flow-line reconfigurable manufacturing systems. *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture*, 231(7) :1229–1236, 2015.

S.K. Moghaddam, M. Houshmand, and O.F. Valilai. Configuration design in scalable reconfigurable manufacturing systems (RMS) : a case of single-product flow line (SPFL). *International Journal of Production Research*, 56(11) :3932–3954, 2017.

S.K. Moghaddam, M. Houshmand, K. Saitou, and O.F. Valilai. Configuration design of scalable reconfigurable manufacturing systems for part family. *International Journal of Production Research*, 58(10) :2974–2996, 2020.

F. Musharavati and A.M.S. Hamouda. Simulated annealing with auxiliary knowledge for process planning optimization in reconfigurable manufacturing. *Robotics and Computer-Integrated Manufacturing*, 28(2) :113–131, 2012a.

F. Musharavati and A.M.S. Hamouda. Enhanced simulated-annealing-based algorithms and their applications to process planning in reconfigurable manufacturing systems. *Advances in Engineering Software*, 45(1) :80–90, 2012b.

A. Otto, C. Otto, and A. Scholl. Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing. *European Journal of Operational Research*, 228(1) :33–45, 2013.

Z.J. Pasek. Challenges in the design of reconfigurable machine tools. In *Reconfigurable Manufacturing Systems and Transformable Factories*, pages 141–154. Springer Berlin Heidelberg, 2006.

R. Pastor, C. Andrés, A. Duran, and M. Pérez. Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the task dispersion. *Journal of the Operational Research Society*, 53(12) :1317–1323, 2002.

L.N. Pattanaik, P.K. Jain, and N.K. Mehta. Cell formation in the presence of reconfigurable machines. *The International Journal of Advanced Manufacturing Technology*, 34(3-4) :335–345, 2006.

J.H. Patterson and J.J. Albracht. Assembly-line balancing : Zero-one programming with Fibonacci search. *Operations Research*, 23(1) :166–172, 1975.

J. Pereira. Modelling and solving a cost-oriented resource-constrained multi-model assembly line balancing problem. *International Journal of Production Research*, 56(11) :3994–4016, 2018.

A. Pirogov, A. Rossi, E. Gurevsky, and A. Dolgui. Search space reduction in MILP approaches for the robust balancing of transfer lines. In *Proceedings of the 17th International Workshop on Project Management and Scheduling (PMS 2020/2021)*, 2021.

S. Qu and Z. Jiang. A memetic algorithm approach for batch-model assembly line balancing problem of sub-block in shipbuilding. *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture*, 228(10) :1290–1304, 2014.

M. Rabbani, M. Samavati, M.S. Ziaee, and H. Rafiei. Reconfigurable dynamic cellular manufacturing system : A new bi-objective mathematical model. *RAIRO - Operations Research*, 48(1) :75–102, 2014.

P. Renna and M. Ambrico. Design and reconfiguration models for dynamic cellular manufacturing to handle market changes. *International Journal of Computer Integrated Manufacturing*, 28(2) :170–186, 2014.

M. Rheault, J.R. Drolet, and G. Abdulnour. Physically reconfigurable virtual cells : A dynamic model for a highly dynamic environment. *Computers & Industrial Engineering*, 29(1-4) : 221–225, 1995.

M.E. Salveson. The assembly line balancing problem. *Journal of Industrial Engineering*, 6(4) : 18–25, 1955.

E. Sancı and M. Azizoğlu. Rebalancing the assembly lines : exact solution approaches. *International Journal of Production Research*, 55(20) :5991–6010, 2017.

L.K. Saxena and P.K. Jain. A model and optimisation approach for reconfigurable manufacturing system configuration design. *International Journal of Production Research*, 50(12) :3359–3381, 2012.

A. Scholl. *Balancing and Sequencing of Assembly Lines*. Physica-Verlag HD, 1999.

A. Scholl and C. Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168(3) :666–693, 2006.

A. Scholl and R. Klein. SALOME : A bidirectional branch-and-bound procedure for assembly line balancing. *INFORMS Journal on Computing*, 9(4) :319–334, 1997.

A.I. Shabaka and H.A. ElMaraghy. A model for generating optimal process plans in RMS. *International Journal of Computer Integrated Manufacturing*, 21(2) :180–194, 2008.

J. Shang and T. Sueyoshi. A unified framework for the selection of a flexible manufacturing system. *European Journal of Operational Research*, 85(2) :297–315, 1995.

P. Sivasankaran and P. Shahabudeen. Literature review of assembly line balancing problems. *The International Journal of Advanced Manufacturing Technology*, 73(9-12) :1665–1694, 2014.

P. Spicer and H.J. Carlo. Integrating reconfiguration cost into the design of multi-period scalable reconfigurable manufacturing systems. *Journal of Manufacturing Science and Engineering*, 129(1) :202–210, 2007.

P. Spicer, D. Yip-Hoi, and Y. Koren. Scalable reconfigurable equipment design principles. *International Journal of Production Research*, 43(22) :4839–4852, 2005.

R. Tavakkoli-Moghaddam, M Ranjbar-Bourani, G.R. Amin, and A. Siadat. A cell formation problem considering machine utilization and alternative process routes by scatter search. *Journal of Intelligent Manufacturing*, 23(4) :1127–1139, 2010.

S. Topaloglu, L. Salum, and A.A. Supciller. Rule-based modeling and constraint programming based solution of the assembly line balancing problem. *Expert Systems with Applications*, 2012.

F.A. Touzout and L. Benyoucef. Multi-objective sustainable process plan generation in a reconfigurable manufacturing environment : exact and adapted evolutionary approaches. *International Journal of Production Research*, 57(8) :1–17, 2018.

F.A. Touzout and L. Benyoucef. Multi-objective multi-unit process plan generation in a reconfigurable manufacturing environment : a comparative study of three hybrid metaheuristics. *International Journal of Production Research*, 57(24) :1–16, 2019.

D. Tremblet, A.R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui. Balancing multi-model assembly lines in a reconfigurable environment. 2021. In *European Conference on Operational Research (EURO 2021)*.

D. Tremblet, A.R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui. Balancing multi-model assembly lines in a reconfigurable environment with unknown order of product arrival. *Journal of the Operational Research Society (soumis)*, xxxxb.

J.I. van Zante-de Fokkert and T.G. de Kok. The mixed and multi model line balancing problem : a comparison. *European Journal of Operational Research*, 100(3) :399–412, 1997.

W. Wang and Y. Koren. Scalability planning for reconfigurable manufacturing systems. *Journal of Manufacturing Systems*, 31(2) :83–91, 2012.

T.S. Wee and M.J. Magazine. Assembly line balancing as generalized bin packing. *Operations Research Letters*, 1(2) :56–58, 1982.

U. Wemmerlov and D.J. Johnson. Cellular manufacturing at 46 user plants : Implementation experiences and performance improvements. *International Journal of Production Research*, 35(1) :29–49, 1997.

L. Wolsey. *Integer Programming*. Wiley, 1998.

N. Xie, A. Li, and W. Xue. Cooperative optimization of reconfigurable machine tool configurations and production process plan. *Chinese Journal of Mechanical Engineering*, 25(5) : 982–989, 2012.

C. Yang, J. Gao, and L. Sun. A multi-objective genetic algorithm for mixed-model assembly line rebalancing. *Computers & Industrial Engineering*, 65(1) :109–116, 2013.

H. Ye and M. Liang. Simultaneous modular product scheduling and manufacturing cell reconfiguration using a genetic algorithm. *Journal of Manufacturing Science and Engineering*, 128 (4) :984–995, 2006.

A.R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui. Optimizing task reassignments in the design of reconfigurable manufacturing lines. 2020. In *21st Annual Conference of the French Society for Operations Research and Decision Aid (ROADEF 2020)*.

A.R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui. Minimizing task reassignments in the design of reconfigurable manufacturing lines with space restrictions. *IFAC-PapersOnLine*, 53(2) :10437–10442, 2021a.

A.R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui. Modular equipment optimization in the design of multi-product reconfigurable manufacturing systems. 2021b. In *17th International Workshop on Project Management and Scheduling (PMS 2020/2021)*.

A.R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui. A milp-based heuristic for minimizing the number of reassignments under balancing multi-model reconfigurable manufacturing lines. 2021c. In *22nd Annual Conference of the French Society for Operations Research and Decision Aid (ROADEF 2021)*.

A.R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui. Reconfigurable manufacturing systems from an optimisation perspective : a focused review of literature. *International Journal of Production Research*, 59(21) :6400–6418, 2021d.

A.R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui. Minimizing task reassignments under balancing multi-product reconfigurable manufacturing lines. *Computers & Industrial Engineering (soumis)*, xxxxa.

A.M.A. Youssef and H.A. ElMaraghy. Modelling and optimization of multiple-aspect RMS configurations. *International Journal of Production Research*, 44(22) :4929–4958, 2006.

A.M.A. Youssef and H.A. ElMaraghy. Optimal configuration selection for reconfigurable manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 19(2) :67–106, 2007.

A.M.A. Youssef and H.A. ElMaraghy. Performance analysis of manufacturing systems composed of modular machines using the universal generating function. *Journal of Manufacturing Systems*, 27(2) :55–69, 2008a.

A.M.A. Youssef and H.A. ElMaraghy. Availability consideration in the optimal selection of multiple-aspect RMS configurations. *International Journal of Production Research*, 46(21) : 5849–5882, 2008b.

H. Yu and W. Shi. A genetic algorithm for multi-model assembly line balancing problem. In *IEEE International Symposium on Assembly and Manufacturing (ISAM 2013)*, pages 369–371, 2013.

J.-M. Yu, H.-H. Doh, H.-W. Kim, J.-S. Kim, D.-H. Lee, and S.-H. Nam. Iterative algorithms for part grouping and loading in cellular reconfigurable manufacturing systems. *Journal of the Operational Research Society*, 63(12) :1635–1644, 2012.

J.-M. Yu, H.-H. Doh, J.-S. Kim, Y.-J. Kwon, D.-H. Lee, and S.-H. Nam. Input sequencing and scheduling for a reconfigurable manufacturing system with a limited number of fixtures. *The International Journal of Advanced Manufacturing Technology*, 67(1-4) :157–169, 2013.

F. Zhang, Y.F. Zhang, and A.Y.C. Nee. Using genetic algorithms in process planning for job shop machining. *IEEE Transactions on Evolutionary Computation*, 1(4) :278–289, 1997.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Outils d'aide à la décision pour la mise en place et la reconfiguration dynamique des systèmes de production reconfigurables

Mot clés : Production, reconfigurabilité, modularité, multi-produit, équilibrage de ligne, optimisation.

Résumé : Les travaux de recherche effectués dans cette thèse traitent de l'optimisation dans la conception de lignes de production reconfigurables multi-produits. Ce type de lignes est caractérisé par leur capacité à traiter différents produits, où chacun d'entre eux nécessite une configuration adéquate de la ligne. Le passage d'une configuration de produit vers une autre implique que la ligne doit être reconfigurée en réaffectant certains tâches/modules entre les stations/machines disponibles. Ainsi, étant donné un ensemble de produits, les deux problèmes d'optimisation suivants sont étudiés. Le premier consiste à concevoir une configuration de ligne admissible pour chaque produit de telle sorte que le nombre de réaffectations de tâches soit minimisé. Le deuxième problème, quant à lui, vise à identifier le nombre minimal de modules nécessaire pour fabriquer tous les produits. Pour ces deux problèmes, nous avons initialement proposé une formulation PLNE. Ensuite, nous avons développé une approche heuristique, nommée Halt-and-Fix, pour le premier problème, et une méthode de décomposition pour le second. Ces deux problèmes ouvrent la voie à des perspectives de recherche intéressantes.

Title: Decision support tools for the design and dynamic reconfiguration of reconfigurable manufacturing systems

Keywords: Manufacturing, reconfigurability, modularity, multi-product, line balancing, optimization.

Abstract: The research work conducted in this thesis deals with the optimization in the design of multi-product reconfigurable lines. Such lines are characterized by their ability to handle different products, where each of them requires an appropriate line configuration. Moving from one product configuration to another means that the line has to be reconfigured by reassigning some tasks/modules between the available stations/machines. Thus, given a set of products, the following two optimization problems are studied. The first one consists in designing an admissible line configuration for each product so that the number of task reassignments is minimized when switching between configurations. The second problem aims at identifying the minimum number of modules needed to manufacture all the products. For these two problems, we have proposed, at first, a MILP formulation. Subsequently, we have developed a heuristic approach, called Halt-and-Fix, for the first problem, and a decomposition method for the second. These two problems open the way to interesting research perspectives.