Combinatorial optimization for the configuration of workforce and equipment in reconfigurable assembly lines

Seyyed Ehsan Hashemi Petroodi

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE
MINES-TÉLÉCOM ATLANTIQUE BRETAGNE
PAYS-DE-LA-LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies
de l'Information et de la Communication
Spécialité : *Automatique, productique et robotique*

Par
Seyyed Ehsan HASHEMI PETROODI

Optimisation combinatoire pour l'affectation des opérateurs et la configuration des équipements dans les lignes d'assemblage reconfigurables

Combinatorial optimization for the configuration of workforce and equipment in reconfigurable assembly lines

Thèse présentée et soutenue à « IMT-Atlantique, Nantes, France », le « 09/12/2021 »
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes (LS2N)
Thèse N° : 2021IMTA0266

Rapporteurs avant soutenance :

Olga BATTAIA Professeur à KEDGE Business School, Bordeaux, France
Lyes BENYOUCEF Professeur à Aix-Marseille Université, Marseille, France

Composition du Jury :

Président :	Reza TAVAKKOLI-MOGHADDAM	Professeur à l'Université de Téhéran, Iran
Examinateurs :	Olga BATTAIA	Professeur à KEDGE Business School, Bordeaux, France
	Lyes BENYOUCEF	Professeur à Aix-Marseille Université, Marseille, France
	Alena OTTO	Professeur à l'Université de Passau, Allemagne
	Farouk YALAOUI	Professeur à l'Université de Technologie de Troyes, France
Dir. de thèse :	Alexandre DOLGUI	Professeur à IMT Atlantique, Nantes, France
Co-encadrants. de thèse :	Simon THEVENIN	Maitre assistant à IMT Atlantique, Nantes, France
	Sergey KOVALEY	Professeur associé à INSEEC U. Research Center, Lyon, France

Invité(s) :

Mikhail Y. KOVALYOV Professeur, Membre-correspondant de National Academy of Sciences of Belarus, Minsk, Bélarus
Ali SIADAT Professeur à École Nationale Supérieure d'Arts et Métiers, Metz, France

Acknowledgment

My deepest gratitude goes to my supervisor Prof. Alexandre Dolgui. Working with him was an honor. I have learned a lot from him both professionally and personally. His constant support and encouragement along with his willingness and trust in allowing me to seek my interests and have made my whole Ph.D. program an enjoyable experience. Many thanks also go to my co-supervisors Dr. Simon Thevenin and Dr. Sergey Kovalev for their scientific attitude, such as the effort to fundamentally comprehend a subject or a method, and relentless simplifications of a result in order to make a theoretical statement accessible to a greater audience, were constant sources of my surprise during our collaborations.

I have to also thank Prof. Mikhail Y. Kovalyov for the fruitful collaboration that we had during my PhD studies

I would like to express my sincere gratitude to the defense committee members Prof. Olga Battaïa, Prof. Alena Otto, Prof. Lyes Beyousef, Prof. Reza Tavakkoli-Moghaddam, and Prof. Farouk Yalaoui for their constructive criticism and feedback. Also, many thanks to Prof. Ali Siadat and Prof. Mikhail Y. Kovalyov for accepting to participate in my Ph.D. defense.

I would like to thank to all my colleagues at SLP and my friends for their support and the nice moments we had together.

Last but not the least, my family deserves endless gratitude: my parents and my brothers for their tremendous support and hope they had given to me. I love you all !

“What you seek is seeking you.”

Rumi

To Elaheh & Elisa.

RÉSUMÉ EN FRANÇAIS

Les entreprises industrielles sont confrontées à une incertitude croissante de la demande du marché, à des changements brusques des besoins des clients, à une grande variété de produits et à des cycles de vie courts des produits. Par conséquent, les entreprises manufacturières recherchent des systèmes de fabrication ajustables et adaptables afin de passer de la production de masse à la personnalisation de masse. Ces systèmes de fabrication permettent aux entreprises de répondre rapidement aux changements du marché et de la technologie, et de lancer fréquemment des produits nouveaux ou mis à jour. La réalisation de cet objectif dépend de la reconfigurabilité, de l'adaptabilité et de la flexibilité des systèmes de fabrication. Le concept de système de fabrication reconfigurable a été proposé par Koren et al. [1999]. Il s'agit d'un système de fabrication dans lequel les ressources peuvent être réorganisées et remplacées rapidement pour modifier la capacité de production. La littérature sur les systèmes de fabrication capables de produire divers produits peut être classée en publications sur les systèmes de fabrication cellulaires, flexibles et reconfigurables. Les lignes d'assemblage peuvent également être cellulaires, flexibles et reconfigurables. Cependant, il existe d'autres classifications des lignes d'assemblage. Les différents types de lignes d'assemblage et de systèmes de fabrication seront discutés en détail à travers l'état de l'art proposé dans cette thèse.

Un système de fabrication capable d'effectuer une grande variété de tâches nécessite différents types de ressources. Les systèmes entièrement automatisés utilisant des robots possèdent une vitesse, une précision, une infatigabilité et une force élevées, mais ils sont chers. L'automatisation des systèmes de fabrication est une tendance constante dans le secteur industriel. Dans le même temps, pour de nombreuses industries, la transition vers un système entièrement automatisé reste un défi insurmontable. L'industrie 4.0 favorise l'adoption de robots collaboratifs, appelés cobots, comme les robots d'assemblage à deux bras, par exemple. Ils conduisent à un système de fabrication, où opérateurs humains et robots travaillent côte à côte. En effet, les récentes avancées en matière d'intelligence artificielle et de dispositifs de détection ont donné naissance à ce nouveau type de robots capables de collaborer avec les humains et d'effectuer une grande variété de tâches [Olsen and Tomlin, 2020]. D'autre part, les opérateurs humains sont intelligents, créatifs, flexibles et capables de travailler avec différents outils dans différentes situations. Il existe un nombre croissant de publications sur la reconfiguration des machines et des équipements. En revanche, la reconfiguration des opérateurs n'est pas assez analysée. Pourtant, les humains sont flexibles par nature, et ils représentent une opportunité d'améliorer la flexibilité des systèmes de fabrication. Contrairement aux machines, qui ne peuvent pas effectuer une tâche au-delà de leur prédestination (du moins jusqu'à un certain point), les opérateurs humains sont créatifs et capables de fonctionner avec différents outils et équipements. En outre, un opérateur peut gérer une situation non standard, là où une ressource automatisée échouerait. Par conséquent, les opérateurs augmentent la flexibilité et l'adaptabilité des systèmes de fabrication.

Les lignes d'assemblage représentent une sous-classe de systèmes de fabrication, dont la spécificité consiste en une nature d'atelier à flux et une production répétitive. Une ligne d'assemblage est caractérisée par un flux de produits passant par des stations de travail ou des ressources dans le temps. La disposition physique des stations, des ressources, de la composition du traitement et des équipements de transport peut être donnée, ou leur sélection peut être l'une des décisions à prendre. Les séquences de produits entrants peuvent être finies ou infinies, répétitives ou non, et elles peuvent être entièrement ou partiellement spécifiées dans un sens déterministe, stochastique ou autre. Les produits peuvent être identiques ou différents. Chaque produit nécessite l'exécution d'un ensemble de tâches. Chaque tâche peut nécessiter certaines quantités de ressources et/ou d'équipements supplémentaires. Une station peut exécuter une seule ou plusieurs tâches, de manière séquentielle ou en parallèle. L'une des principales caractéristiques du système est le moment où les produits sont déplacés vers et depuis les stations. Si ces mouvements sont effectués avec le même pas de temps, appelé temps de cycle ou takt, pour toutes les stations, alors le système est appelé "paced", sinon, il est appelé "un-paced". Un ensemble d'événements qui se produisent entre deux mouvements séquentiels de produits dans une ligne cadencée est appelé cycle de production. De nombreux problèmes de recherche opérationnelle peuvent être définis pour une ligne d'assemblage. De nombreuses publications récentes traitent des nouveaux problèmes de différents types de lignes d'assemblage. Les différentes lignes d'assemblage qui se distinguent par leur capacité à gérer la variété des produits, par exemple les lignes dédiées, à modèles multiples/mixtes, seront précisées par la revue de littérature de la thèse, voir la figure 1 dans l'introduction.

Une des étapes importantes de la conception d'une ligne d'assemblage est l'équilibrage de la ligne, c'est-à-dire l'affectation des tâches aux postes de travail en optimisant un critère donné : minimisation du temps de cycle, du coût total, du nombre de stations, etc. Les décisions sur l'affectation des tâches et des effectifs doivent être prises simultanément, car tout changement dans l'affectation des tâches peut impliquer des changements dans l'affectation des effectifs [Cortez and Costa, 2015]. Plusieurs hypothèses restrictives sont couramment formulées dans la littérature sur les problèmes d'équilibrage des lignes d'assemblage [Baybars, 1986, Scholl and Becker, 2006, Boysen et al., 2008, Battaïa and Dolgui, 2013] : affectation d'un seul opérateur à chaque station, production d'un seul modèle dans la ligne, opérateurs et tâches sont fixes, etc. Cependant, pour les tâches lourdes effectuées sur des produits de grande taille, comme dans l'industrie automobile, il est plus réaliste d'affecter plus d'un opérateur à chaque station [Michels et al., 2019]. En outre, le déplacement des opérateurs entre les stations permet d'adapter les capacités des stations à la séquence de production [Sikora et al., 2017]. En outre, la personnalisation de masse et les changements de marché obligent les entreprises à produire plusieurs modèles de produits au lieu d'un seul. La figure 2 dans l'introduction compare les lignes d'assemblage à modèles mixtes simples, à modèles mixtes multi-manned et à modèles mixtes multi-manned reconfigurables. La flexibilité des lignes d'assemblage à modèles mixtes avec plusieurs opérateurs mobiles aux stations de travail peut être améliorée par une affectation dynamique des tâches aux stations (plutôt qu'une affectation fixe classique).Outre l'ajustement de la capacité de la ligne à la combinaison de production, l'affectation dynamique des tâches permet de gérer les relations de priorité conflictuelles pour différents modèles de produits dans ces types de lignes.

La séquence de production a un impact important sur la performance d'une ligne d'assemblage à modèle mixte car les items nécessitent des temps de traitement différents à chaque station. Dans un contexte de fabrication, la séquence des items entrant dans une ligne de modèle mixte n'est souvent pas contrôlable à cause d'une sortie incertaine de l'étape de production en amont et à cause de la variation de la demande. Par conséquent, nous cherchons à dimensionner la main d'œuvre de la ligne (nombre d'opérateurs) dans le pire takt time possible [Mosadegh et al., 2020], soumis à un temps de cycle donné. Si la ligne respecte le temps de cycle dans le pire cas, elle respecte le temps de cycle pour tous les cas.

Motivation. Cette thèse est motivée par plusieurs défis de recherche observés dans la littérature récente sur l'équilibrage et la conception des lignes d'assemblage.

Les situations de marché actuelles obligent les entreprises manufacturières à utiliser des systèmes de fabrication capables de produire différents types de produits. De plus, selon la littérature sur les lignes d'assemblage à modèle unique, mixte et multiples. Dans la pratique, les fabricants ont besoin d'outils pour décider de la ligne qui mérite d'être installée. Une entreprise de fabrication pourrait soit installer une ligne d'assemblage unique à modèles multiples/mixtes, soit plusieurs lignes d'assemblage dédiées pour produire une variété de produits.Par conséquent, nous étudions d'abord la rentabilité d'une ligne unique à modèles multiples/mixtes par rapport à plusieurs lignes d'assemblage dédiées.

En outre, les défis et les tendances observés dans des publications récentes nous incitent à nous concentrer sur les lignes d'assemblage à modèles multiples/mixtes. L'ordre des produits entrant dans une ligne à modèles mixtes n'est souvent pas contrôlable car il dépend des incertitudes d'une étape de production en amont et d'une demande variable. L'ordre arbitraire des modèles de produits dans une ligne à modèles mixtes demande plus de flexibilité. De plus, ces lignes peuvent bénéficier du concept de reconfigurabilité avec le déplacement des ressources d'assemblage et la réaffectation des tâches. Il est intéressant d'étudier une situation de production dans laquelle l'ordre des produits est inconnu, les travailleurs peuvent passer d'une tâche à l'autre et chaque modèle de produit a son propre ensemble de tâches et de relations de précédence. Ces conditions favorisent l'application d'une affectation dynamique des tâches qui réajuste les ressources à la fin de chaque cycle. Par conséquent, la prise en compte des hypothèses mentionnées dans ces lignes ouvre de nombreux problèmes complexes à étudier dans la conception et l'équilibrage des lignes d'assemblage à modèles mixtes. Par exemple, une ligne d'assemblage à modèles mixtes et multi-manned pourrait être étudiée avec des opérateurs mobiles lorsque chaque modèle de produits a ses relations spécifiques de priorité des tâches, et une sorte d'affectation dynamique des tâches pourrait être requise dans ce cas.

Nous évaluons différents types d'assignation des tâches dans une chaîne de montage à modèles mixtes et multi-manned avec des opérateurs mobiles : fixe, dépendant du modèle et dynamique. L'affectation fixe des tâches signifie que la position et l'ordre des tâches restent les mêmes pour toute séquence de modèles de produits. Dans la politique dépendante du modèle, l'affectation des tâches change d'un produit à l'autre, mais pour un modèle de produit donné, elle reste la même. Dans l'affectation dynamique des tâches, les ressources peuvent être remplacées à chaque cycle afin de s'adapter à toute séquence de produits possible.

Une séquence de produits peut être prédéfinie avant le début de la période de production ou inconnue et infinie. Les problèmes étudiés sont principalement des problèmes d'optimisation robuste visant à minimiser le coût total des opérateurs et des équipements pour le pire cycle du processus de production.

Afin de surmonter les fluctuations et les incertitudes, une ligne à modèles mixtes doit avoir un haut degré de reconfigurabilité [Koren et al., 1999], c'est-à-dire une capacité à adapter rapidement les ressources, opérateurs humains et machines, aux modèles de produits entrants. La reconfigurabilité est obtenue en déplaçant les travailleurs d'une station à une autre à la fin de chaque cycle, tandis que les équipements peuvent être dupliqués dans les stations si nécessaire. Les problèmes proposés sont adaptés à différentes conditions de production et caractéristiques de modèles de produits.

Plusieurs approches d'optimisation sont développées pour modéliser et résoudre les problèmes proposés. Bien que ces approches aient été rarement implémentées dans la littérature sur les lignes d'assemblage à modèles mixtes, elles s'adaptent bien aux problèmes étudiés dans cette thèse. Chaque chapitre fournit une justification détaillée du choix de la méthode de solution. Les résultats expérimentaux détaillés et les perspectives de gestion dans tous les chapitres apportent des conseils utiles aux entreprises de fabrication.

Contribution. Cette thèse est structurée et contribue à la littérature sur l'équilibrage et la conception de lignes d'assemblage mixtes. Cette thèse contribue à la littérature sur les problèmes d'équilibrage et de conception des lignes d'assemblage mixtes avec des opérateurs mobiles. La thèse évalue l'impact de l'affectation dynamique des tâches et des opérateurs, et différentes politiques d'affectation des tâches (fixe, dépendante du modèle, dynamique) sont étudiées. Les hypothèses des problèmes sont réalistes et pratiques. Pour chacun des problèmes d'optimisation considérés, la thèse fournit une contribution méthodologique pour résoudre des instances de taille réaliste.

Les principales contributions de chaque chapitre sont soulignées comme suit :

— **Chapitre 1** présente une revue de littérature approfondie sur les problèmes de planification des opérateurs et d'équilibrage des lignes d'assemblage. Nous avons défini différents types de systèmes de production et d'assemblage, le problème de planification de la main-d'œuvre, le problème d'équilibrage de la ligne d'assemblage, et les concepts pertinents utilisés dans la thèse comme la flexibilité, la reconfigurabilité et les stratégies de reconfiguration des opérateurs. À la fin du chapitre, nous discutons de l'état actuel des connaissances dans la littérature connexe et découvrons plusieurs possibilités de recherches futures. Certaines d'entre elles ont été couvertes par cette thèse.

— **Chapitre 2** étudie un problème de sélection de configuration entre une ligne unique modèles multiples et plusieurs lignes dédiées. L'objectif de l'utilisation de l'une des deux configurations est de maximiser le profit total, sous réserve de la demande du produit et des contraintes de temps de fabrication. Le problème de sélection a été réduit à deux problèmes d'optimisation, pour l'un desquels un algorithme en temps polynomial est développé, et la dureté NP est prouvée pour l'autre. Un algorithme de programmation dynamique, une heuristique grégaire constructive, une heuristique aléatoire et un algorithme de recherche locale avec ascension de pente la plus raide sont présentés pour le problème NP-hard. Des expériences informatiques avec l'heuristique, l'algorithme de recherche locale et l'approche de la solution au problème de programmation linéaire en nombres entiers correspondant à l'aide d'un solveur commercial sont décrites. Les résultats démontrent la qualité appropriée des solutions heuristiques et de recherche locale. La méthodologie et le logiciel proposés peuvent être utilisés pour évaluer différents scénarios de données d'entrée tout en prenant une décision de sélection entre les deux configurations de fabrication. La demande et les prix de vente des produits, les temps de préparation et de fabrication, la demande et les annulations de production sont les paramètres qui affectent la décision de sélection. Quelques directions de recherche futures ont été proposées.

— **Chapitre 3** étudie l'impact de l'affectation des tâches en fonction du modèle, de la reconfiguration des opérateurs et de la duplication des équipements dans des lignes d'assemblage à modèles mixtes. La ligne étudiée est cadencée, et elle peut traiter différents modèles de produits avec différents ensembles de tâches et de relations de précédence.L'affectation des tâches et des travailleurs aux stations peut changer à chaque cycle, et l'objectif est de concevoir une ligne capable de gérer un ensemble prédéfini de situations correspondant à différents flux de produits entrant dans la ligne. Nous fournissons une nouvelle formulation de programmation linéaire en nombres entiers mixtes (MILP) pour minimiser les coûts d'opérateurs et d'équipement dans les lignes d'assemblage à modèles mixtes avec une affectation des tâches dépendant du modèle. Nous proposons une reformulation efficace de la MILP en nous appuyant sur l'approche de dualisation couramment utilisée en optimisation robuste. De plus, nous utilisons une matheuristique constructive (CM) et une heuristique de type " fixer et optimiser " (FOH) pour traiter les instances à grande échelle. Des expériences de calcul approfondies réalisées avec des repères bien connus de la littérature montrent que les approches proposées sont performantes en termes de qualité de solution et de temps de calcul.En outre, les résultats révèlent que l'affectation des tâches en fonction du modèle réduit considérablement le coût des équipements et le nombre d'opérateurs par rapport aux lignes d'assemblage classiques à modèle mixte avec affectation fixe des tâches et opérateurs mobiles.

— **Chapitre 4** étudie l'impact de l'affectation dynamique des tâches, de la reconfiguration des opérateurs et de la duplication des équipements dans des lignes d'assemblage à modèles mixtes. La ligne étudiée et le problème sont les mêmes que dans le chapitre 3. Les affectations des tâches et des opérateurs aux stations peuvent changer à chaque cycle en fonction de l'ordre du modèle de produit pour la période en cours. Ce chapitre fournit une formulation de programmation linéaire mixte en nombres entiers (MILP) basée sur le scénario pour minimiser les coûts des opérateurs et des équipements dans le pire des cas avec une affectation dynamique des tâches. Comme la génération de toutes les ordonnances possibles et la résolution de la MILP proposée pour toutes les ordonnances prennent du temps, un générateur de séquence est développé pour créer un ensemble d'ordonnances possibles de produits. Un modèle de simulation est développé pour évaluer le niveau de robustesse de la solution fournie par la MILP. Une approche basée sur la simulation est proposée pour améliorer la solution afin d'obtenir la solution la plus robuste à l'aide du modèle de simulation. L'approche prend séparément les décisions de conception et les décisions opérationnelles, dans lesquelles le modèle MILP conçoit la ligne tandis que la simulation vérifie les décisions opérationnelles (c'est-à-dire l'affectation des travailleurs et des tâches) pour une conception donnée. De plus, un algorithme de recherche locale est développé pour trouver plus rapidement la solution la plus robuste. Plusieurs instances du chapitre 3 sont résolues. Les résultats montrent une meilleure rentabilité de l'affectation dynamique des tâches par rapport aux affectations fixes et dépendantes du modèle. Cependant, les résultats dépendent dans une certaine mesure des paramètres d'entrée.

— **Chapitre 5** étend le problème étudié dans le chapitre 4 au cas d'une séquence infinie et inconnue de produits. En plus de l'affectation dynamique des tâches à chaque takt, il n'y a aucune information sur le modèle de produit entrant dans la ligne, ce qui crée un environnement hautement dynamique et incertain. En raison de sa capacité à gérer les incertitudes, un processus de décision de Markov (MDP) est appliqué pour modéliser le système. Le problème est abordé en utilisant deux critères. Le premier reflète un cas stochastique dans lequel le coût total attendu des travailleurs et des équipements sur tous les états (cycles) possibles est minimisé. Le second minimise le coût total des travailleurs et des équipements pour le pire état (cycle). Deux modèles MILP correspondants sont développés pour résoudre deux problèmes stochastiques et robustes. L'approche est capable de résoudre des instances de taille moyenne. Les travaux futurs pourraient se concentrer sur l'amélioration du temps de calcul par l'application de méthodes approximatives.

Cette thèse fournit des résultats utiles tant d'un point de vue pratique que théorique. Elle étudie les avantages et les inconvénients de l'utilisation de plusieurs lignes dédiées ou d'une seule ligne multi-modèles. Elle étudie l'impact des affectations de tâches fixes, dépendantes du modèle et dynamiques dans différentes situations pratiques. Les approches de solution proposées intègrent les avantages de la qualité de solution des méthodes exactes et les temps de calcul rapides des heuristiques.

Plusieurs pistes de recherche futures découlent des problèmes étudiés. Nos résultats montrent que la stratégie d'affectation dynamique des tâches et le déplacement des opérateurs réduisent de manière significative le coût d'une ligne d'assemblage de modèle mixte. Une direction de recherche future consiste à concevoir des approches d'optimisation efficaces (par exemple, des heuristiques) pour traiter des instances de grande taille et réduire le temps de calcul. Outre le déplacement des opérateurs, d'autres stratégies de reconfiguration des opérateurs peuvent être utilisées, telles que l'utilisation de travailleurs utilitaires et temporaires. Un environnement collaboratif homme-robot est également d'un grand intérêt et présente un potentiel de recherche prometteur. L'aspect du bien-être et de la sécurité des opérateurs est de la plus haute importance. Par conséquent, les études futures sont invitées à prendre en considération la qualité ergonomique de la conception et de l'équilibrage de la ligne. Enfin, les recherches futures pourraient évaluer l'impact des nouvelles technologies, telles que les dispositifs intelligents, les caméras, les capteurs, la téléopération, l'échange de messages et la réalité augmentée, sur l'environnement de la ligne d'assemblage.

TABLE OF CONTENTS

RÉSUMÉ EN FRANÇAIS **9**

Introduction **29**

0.1 Context . . . 29
0.2 Motivation . . . 32
0.3 Contributions . . . 33

1 Literature review **35**

1.1 Introduction . . . 35
1.2 Production systems . . . 35
1.2.1 Dedicated Manufacturing System (DMS) . . . 35
1.2.2 Flexible Manufacturing System (FMS) . . . 36
1.2.3 Cellular Manufacturing System (CMS) . . . 36
1.2.4 Reconfigurable Manufacturing System (RMS) . . . 36
1.3 Assembly line characteristics . . . 37
1.3.1 Product variety in assembly lines . . . 38
1.3.2 Resource variety in assembly lines . . . 38
1.4 Assembly line balancing . . . 40
1.5 Workforce planning in production systems . . . 43
1.6 Reconfigurability and flexibility . . . 45
1.7 Workforce reconfiguration strategies in production systems . . . 47
1.7.1 Utility workers . . . 48
1.7.2 Temporary workers . . . 50
1.7.3 Walking workers . . . 51
1.7.4 Bucket brigades . . . 53
1.7.5 Cross-trained workers . . . 56
1.7.6 Analysis of workforce reconfiguration strategies . . . 58
1.8 Conclusion . . . 64

2 A single multi-model line versus multiple dedicated lines **67**
2.1 Introduction 67
2.2 Problem formulation 69
2.3 Solution of the problem OPT1 72
2.3.1 Algorithm S 73
2.4 Solution of the problem OPT2 75
2.4.1 Dynamic programming 76
2.4.2 Greedy and randomized greedy algorithms 78
2.4.3 Local search algorithm 78
2.5 Computational experiments 81
2.5.1 Instances generation 82
2.5.2 Computational results 83
2.5.3 Managerial insights 84
2.6 Conclusions 87

3 Model-dependent task assignment in mixed-model assembly lines with walking workers **91**
3.1 Introduction 91
3.2 Problem description and formulation 94
3.2.1 Description of $MALBP-W$ 94
3.2.2 Illustrative example 96
3.2.3 Mathematical model for fixed task assignment 98
3.2.4 Mathematical model for model-dependent task assignment 100
3.3 Optimization approaches 100
3.3.1 MILP reformulation 100
3.3.2 Constructive matheuristic (CM) 104
3.3.3 Fix-and-optimize heuristic (FOH) 105
3.3.4 Fix^h heuristic 106
3.4 Computational experiments and results 107
3.4.1 Instances generation 107
3.4.2 Analysis of the heuristics 109
3.4.3 Performance of the optimization approaches 110
3.4.4 Managerial insights 111
3.5 Conclusion 113

4 Dynamic task assignment in mixed-model assembly lines with walking workers 115
4.1 Introduction 115
4.2 Problem description 117
4.2.1 Illustrative example 119
4.2.2 Mathematical formulation of $MILP^{Dyn}$ 119
4.3 Solution approach 122
4.3.1 Product order generation 122
4.3.2 Simulation model 123
4.3.3 Simulation-based optimization (SO) 123
4.3.4 Local search algorithm (LS) 124
4.4 Computational experiments and results 127
4.4.1 Performance of the approaches 128
4.4.2 Managerial insights 130
4.5 Conclusion 132

5 Markov decision process for dynamic task assignment in mixed-model assembly lines under uncertainty 135
5.1 Introduction 135
5.2 Problem description 137
5.3 Markov Decision Process (MDP) application 140
5.3.1 Markov Decision Process (MDP) model 140
5.3.2 Stochastic model MDP^{Sto} 142
5.3.3 Robust model MDP^{Ro} 143
5.3.4 An example of an MDP graph. 144
5.4 Fixed task assignment policies and algorithmic improvements 145
5.4.1 Fixed task assignment policy 145
5.4.2 Reduction rules for states and actions 146
5.4.3 Decomposed transition 148
5.5 Computational experiments and results 150
5.5.1 Instances generation 150
5.5.2 Analysis of the MDP models 153
5.5.3 Managerial insights 155
5.6 Conclusion 158

Conclusion **161**

Publications **165**

Bibliography **167**

List of Figures

1 Single, multi, mixed-model assembly line. 30
2 Simple, multi-manned and reconfigurable multi-manned mixed-model assembly lines. 31

1.1 Main characteristics of the three manufacturing systems: manual, fully automated, and human-robot (hybrid). 39
1.2 Relationship between human/automated resources and the evolution of manufacturing system. 46
1.3 Various workfroce reconfiguration strategies in various production systems. 48
1.4 The number of papers based on both workforce reconfiguration strategy and manufacturing system's type. 59
1.5 Compliance of the workforce reconfiguration related studies to the importance of system's reconfiguration for different manufacturing systems. . . . 62

2.1 Solution of the example problem. $P_1^* = 164280$, $P_2^* = 162080$. 71

3.1 The precedence graphs and tasks processing times of the simple example. . 96
3.2 The optimal solution of $MALBP - W^{Fix}$ and $MALBP - W^{Md}$ in the simple example. 97
3.3 An example of changing pictures of the line for product order (B-A). . . . 98
3.4 A small example showing how reformulation of MILP works based on the MCFP network. 103

4.1 The optimal solution of $MALBP - W^{Dyn}$ in the simple example. 120
4.2 Simulation model and simulation-based optimization approach. 124
4.3 Generation of a new order in the neighborhood. 125
4.4 A schema of the proposed local search algorithm. 127

5.1 The optimal solution of $MALBP - W^{Dyn}$ in the simple example. 140
5.2 The MDP graph for the presented illustrative example and the optimal solution of MDP^{Sto} and MDP^{Ro}. 145

5.3 The framework of the proposed methodology. 147
5.4 Illustrative example of the decomposition process. 150

LIST OF TABLES

1.1 Articles related to utility workers. 49
1.2 Studies concentrating on temporary workers. 50
1.3 Studies considering walking workers. 52
1.4 Studies considering bucket brigades. 55
1.5 Studies considering cross-trained workers. 57
1.6 Current state of applying workforce reconfiguration strategies to different manufacturing systems. 62

2.1 Running time (seconds) and solution quality. 84
2.2 Dependencies of solution characteristics of the input data, $(T, F) = (52, 10)$. 86
2.3 Computational complexity and algorithms 88

3.1 Compatibility between tasks and equipment, and the cost of equipment at each station. 97
3.2 Task selection rules considered in constructive matheuristic (CM) 105
3.3 Station selection rules applied in FOH 106
3.4 Solution quality of the heuristics. 109
3.5 Solution quality of optimization approaches depending on the instances' size. (I, S, O) stands for the number of product models, stations, and tasks, respectively. 110
3.6 Solution quality of optimization approaches depending on the cost of workers. 111
3.7 Average computational time of optimization approaches. 111
3.8 Average computational time of optimization approaches depending on the cost of workers. 111
3.9 The impact of the worker cost on the number of workers, equipment cost and duplication, and cost saving via $MALBP - W^{Md}$. 112
3.10 The influence of different classes of instances on the equipment cost and the number of workers and equipment duplications. 112
3.11 Cost saving via $MALBP - W^{Md}$ as compared to $MALBP - W^{Fix}$. 113

4.1 Results of $MILP^{Dyn}$ and simulation with longer product orders. 129
4.2 The performance of the proposed simulation-based optimization (SO) approach. 129
4.3 The performance of the proposed Local Search (LS) algorithm. 130
4.4 The impact of worker cost on cost saving via $MALBP-W^{Dyn}$ as compared to $MALBP-W^{Md}$ and $MALBP-W^{Fix}$. 131
4.5 The impact of different classes of instances on cost saving via $MALBP-W^{Dyn}$ as compared to $MALBP-W^{Md}$ and $MALBP-W^{Fix}$. 131
4.6 The impact of the worker cost on the equipment cost, number of workers, and equipment duplication. 131
4.7 The impact of different classes of instances on the equipment cost, number of workers, and equipment duplication. 132

5.1 The number of actions and states, and the computational times based on the size of instances. 153
5.2 The number of actions and states, and the computational times based on process time variety of products. 154
5.3 Analogy of the solution quality of fixed, model dependent, and dynamic task assignment. 154
5.4 The impact of the cost of workers on the solution quality of fixed, model dependent, and dynamic task assignment. 155
5.5 The impact of classes of instances on the solution quality of fixed, model dependent, and dynamic task assignment. 156
5.6 The impact of the cost of workers on the cost of equipment and the number of workers in the robust model MDP^{Ro}. 157
5.7 The impact of the cost of workers on the cost of equipment and the number of workers in the stochastic model MDP^{Sto}. 157
5.8 The impact of the classes of instances on the cost of equipment and the number of workers in the robust model MDP^{Ro}. 158
5.9 The impact of the classes of instances on the cost of equipment and the number of workers in the stochastic model MDP^{Sto}. 158

LIST OF ALGORITHMS

1 Algorithm S 73
2 Dynamic programming (DP) 77
3 Greedy algorithm 78
4 Local search algorithm 79

5 Constructive matheuristic (CM) 106
6 Fix-and-optimize heuristic (FOH) 107

ABBREVIATIONS

CM : Constructive Matheuristic
CMS: Cellular Manufacturing System
DML : Dedicated Manufacturing Line
DMS: Dedicated Manufacturing System
FMS : Flexible Manufacturing System
FOH : Fixed-Optimize Heuristic
MALBP-W : Multi-Manned Mixed-Model Assembly Line Balancing Problem with Walking Workers
MCFP : Minimum Cost Flow Problem
MDP : Markov Decision Process
MILP : Mixed Integer Linear Programming
MMAL : Manual Mixed-model Assembly line
MML: Multi-Model Manufacturing Line
LP : Linear Programming
LS : Local Search
RMS: Reconfigurable Manufacturing System
SG : Sequence Generator
SO : Simulation-based Optimization

Introduction

0.1 Context

Industrial companies face an increasing uncertainty in the market demand, abrupt changes in customer needs, large product variety, and short product life cycles. Thereby, manufacturing companies search for adjustable and adaptable manufacturing systems in order to switch from mass production to mass customization. This manufacturing systems allow companies to quickly react to changes in market and technology, and launch new or updated products frequently. The achievement of this goal depends on reconfigurability, adaptability and flexibility of manufacturing systems. The concept of reconfigurable manufacturing system was proposed by Koren et al. [1999]. It is a manufacturing system in which resources can be rearranged, and replaced quickly to change the production capacity. The literature on manufacturing systems that can handle various products can be categorized into publications on cellular, flexible, and reconfigurable manufacturing systems. Assembly lines also can be cellular, flexible, and reconfigurable. However, other classifications of assembly lines exist. Different types of assembly lines and manufacturing systems will be further discussed in details through the literature reviewed in this thesis.

A manufacturing system able to perform a high variety of tasks requires different types of resources. Fully automated systems using robots possess high speed, accuracy, tirelessness, and force, but they are expensive. Automation of manufacturing systems is an ongoing trend in the industrial sector. At the same time, for many industries, the transition to a fully automated system remains an insurmountable challenge. Industry 4.0 fosters the adoption of collaborative robots, called cobots, such as dual-arm assembly robots, for example. They lead to a manufacturing system, where humans and robots work side by side. Indeed, the recent advances in artificial intelligence and sensor devices gave rise to this new type of robots able to collaborate with humans and perform a wide variety of tasks [Olsen and Tomlin, 2020]. On the other hand, human workers are intelligent, creative, flexible, and able to work with different tools in different situations. There is a growing amount of literature on the reconfiguration of machines and equipment. In contrast, workforce reconfiguration is not enough analyzed. Nevertheless, humans are flexible by nature, and they represent an opportunity to enhance the flexibility of manufacturing systems. Unlike machines, which cannot perform a task beyond the scope of their predestination (at least to a certain degree), human workers are creative and able to operate with different tools and equipment. Moreover, a worker can handle a non-standard situation, where an automated resource would fail. Thus, the workers increase the flexibility and adaptability of manufacturing systems.

Assembly lines represent a subclass of manufacturing systems, whose specificity consists in a flow shop nature and repetitive production. An assembly line is characterized by a flow of products through workstations or resources over time. Physical layout of stations, resources, processing composition, and transporting equipment can be given, or their selection can be one of the decisions to be made. The incoming product sequences can be finite or infinite, repetitive or not, and they can be fully or partially specified in the deterministic, stochastic or any other sense. The products can be the same or different. Each product requires a set of tasks to be performed. Each task can require certain quantities of additional resources and/or equipment. A station can execute a single task or multiple tasks, sequentially or in parallel. One of the main system's characteristics is the timing of product moves to and from stations. If these moves are made with the same time step, called cycle time or takt, for all stations, then the system is called paced, otherwise, it is called un-paced. A collection of events that happen between two sequential product moves in a paced line is called a production cycle. Different assembly lines distinguished by their ability to handle product variety, e.g. dedicated, multi- or mixed-model lines, will be further clarified through the literature review of the thesis, see Figure 1.

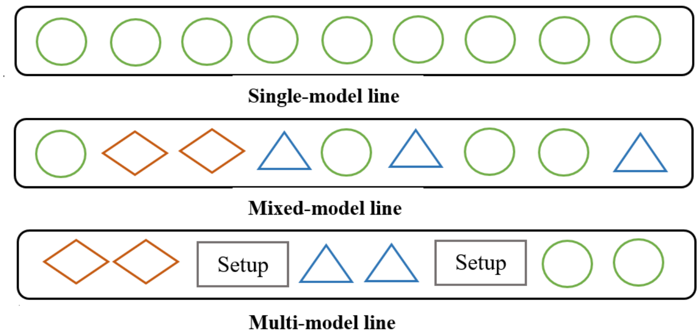

Figure 1 – Single, multi, mixed-model assembly line.

One of the important assembly line design steps is the line balancing, i.e., the assignment of tasks to workstations optimizing a given criterion: minimizing takt time, total cost, number of stations, etc. Decisions on task and workforce assignments have to be made simultaneously, since any change in task assignment may imply changes in workforce assignment [Cortez and Costa, 2015]. Several restrictive assumptions are commonly made in the literature on assembly line balancing problems [Baybars, 1986, Scholl and Becker, 2006, Boysen et al., 2008, Battaïa and Dolgui, 2013]: allocating only one worker to each station, producing only a single model in the line, workers and tasks are fixed, etc. However, for heavy tasks performed on large-size products, like in the automotive industry, assigning more than one worker to each station is more realistic [Michels et al., 2019]. Furthermore, moving workers between stations adapts the stations' capacities to the production sequence [Sikora et al., 2017]. In addition, mass customization and market changes force companies to produce multiple product models instead of a single one. Figure 2 compares simple, multi-manned, and reconfigurable multi-manned mixed-model assembly lines. The flexibility of mixed-model assembly lines with multiple moving workers at workstations can be enhanced by dynamic task assignment to stations (rather than a classical fixed one). Beside adjusting the capacity of the line to the production mix, dynamic task assignment allows handling conflicting precedence relations for different product models in such lines.

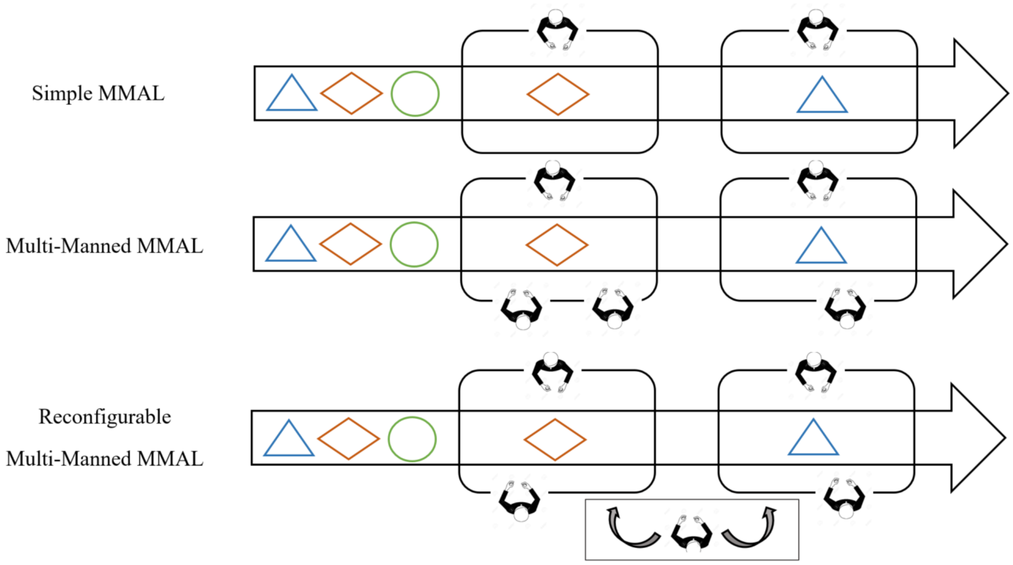

Figure 2 – Simple, multi-manned and reconfigurable multi-manned mixed-model assembly lines.

The production sequence has a large impact on the performance of a mixed-model assembly line because items require different processing times at each station. In a manufacturing context, the sequence of the items entering a mixed model line is often not controllable due to an uncertain output from the upstream production step and due to demand variation. Therefore, we aim to dimension the workforce of the line (number of workers) in the worst possible takt [Mosadegh et al., 2020], subject to a given takt time. If the line respects the takt time in the worst case, it respects the takt time for all cases.

0.2 Motivation

This thesis is motivated by several research challenges observed in the recent literature on assembly line balancing and design.

Today's market situations force manufacturing companies to employ manufacturing systems able to produce different product types. Moreover, there are different studies in the literature on single, mixed, and multi-model assembly lines. In practice, manufacturers need tools to decide which line is worth to be installed. A manufacturing company could either install a unique multi/mixed model assembly line or multiple dedicated assembly lines to produce a variety of products. Therefore, in the beginning, we study the profitability of a unique multi-model line versus multiple dedicated assembly lines.

In addition, challenges and trends observed in recent publications motivate us to focus on multi/mixed-model assembly lines. The order of products entering a mixed-model line is often not controllable as it depends on the uncertainties of an upstream production step and variable demand. The arbitrary order of product models in a mixed-model line requires more flexibility. Moreover, these lines can benefit from the concept of reconfigurability with moving assembly resources and re-assigning tasks. It is interesting to study a production situation in which the product order is unknown, workers can move from one task to another and each product model has its own set of tasks and precedence relationships. Such conditions favor the application of a dynamic task assignment that re-adjusts resources at the end of each takt. Therefore, considering the mentioned assumptions in such lines open many complex challenging problems to study in mixed-model assembly line design and balancing. For example, a multi-manned mixed-model assembly line could be studied with walking workers when each model of products has its specific task precedence relationships, and a kind of dynamic task assignment could be required in this case.

We evaluate different types of task assignments in a multi-manned mixed-model assembly line with walking workers: fixed, model-dependent, and dynamic. Fixed task assignment means that tasks' positions and order remains the same for any sequence of product models. In the model-dependent policy, the task assignment changes from one product to another, but for a given product model remains the same. In dynamic task assignment resources can be replaced every takt in order to adjust to any possible orders of products.

A product sequence can be pre-defined before the beginning of the production period or unknown and infinite. The studied problems are mainly robust optimization problems aiming to minimize the total cost of workers and equipment for the worst takt in the production process.

In order to overcome fluctuations and uncertainties, a mixed-model line should have a high degree of reconfigurability [Koren et al., 1999], i.e. an ability to quickly adapt the resources, human operators and machines, to incoming product models. Reconfigurability is achieved by moving workers from one station to another station at the end of each takt, while equipment can be duplicated in stations if needed. The proposed problems are adapted to different production conditions and product model features.

Several optimization approaches are developed to model and solve the proposed problems. While these approaches have been rarely implemented in the literature on mixed-model assembly lines, they fit well to the problems studied in this thesis. Each chapter provides a detailed justification of the solution method's choice. The extensive experimental results and managerial insights in all chapters bring useful hints for manufacturing companies.

0.3 Contributions

This section describes how the thesis is structured and contributes to the literature on mixed-model assembly line balancing and design. This thesis contributes to the literature on multi-manned mixed-model assembly line balancing and design problems with walking workers. The thesis evaluates the impact of dynamic task and worker assignment, and different task assignment policies (fixed, model-dependent, dynamic) are studied. Problems' assumptions are realistic and practical. For each of the considered optimization problem, the thesis provides methodological contribution to solve realistic size instances.

The main contributions of each chapter are highlighted as follows:

— **Chapter 1** presents a comprehensive literature review on workforce planning and manual line balancing problems, it introduces different production systems, and it provides the concepts of assembly lines, flexibility, and reconfigurability. Five workforce reconfiguration strategies are extracted from the literature: the use of utility, temporary, walking, cross-trained workers, and bucket brigades. This chapter analyzes these strategies in the context of manual assembly lines (single, mixed and multi-model) and manufacturing systems (dedicated, cellular, flexible, and reconfigurable).

- **Chapter 2** evaluates the profitability of multiple dedicated assembly lines versus a single multi-model assembly line. Several algorithms are developed to solve the defined problems. Regarding the NP-hardness of the studied problem, some heuristics are constructed to improve the solutions' quality. The computational results and managerial insights are demonstrated.
- **Chapter 3** studies the impact of model-dependent task assignment on the design of a multi-manned mixed-model assembly line with walking workers. A set of given product orders is considered. A robust mixed-model assembly line balancing problem is studied. It consists in the total production cost minimization for the worst case. A Mixed-Integer Linear Programming (MILP), the reformulated version of the proposed MILP, and some heuristics are developed. The results are compared to the ones for the fixed task assignment policy. Managerial insights are provided.
- **Chapter 4** studies the dynamic task assignment in the same line as in Chapter 3. The same robust optimization problem is also taken into account for a given set of product model sequences. An MILP model, a simulation-based optimization approach and a local search algorithm using the greedy and descent algorithms are developed to solve the problem. The results and managerial insights are provided.
- **Chapter 5** adds another layer of complexity by considering a dynamic task assignment in a line with an infinite unknown product sequence. Two stochastic and robust models are studied. Due to its ability to tackle uncertain dynamic problems, a Markov Decision Process (MDP) solution method is chosen. The MDP is enhanced by using some reduction rules and a decomposition algorithm. It is consequently solved by exact methods using two MILP models corresponding to the stochastic and robust problems. The computational results and managerial insights are reported.

Chapter 1

Literature review

1.1 Introduction

This chapter presents a literature review on the main topics addressed in the thesis. Different types of manufacturing systems are described. Their characteristics are shown in details. Proposed classifications are centered around flexibility, reconfigurability, and variety of product types. The chapter identifies optimization challenges and problems in the view of manual production systems' reconfigurability. Several workforce reconfiguration strategies in various production environments are identified. Future research directions are proposed. Some of them are initiated within this thesis.

1.2 Production systems

A manufacturing system can be characterized by the variability of manufactured products [Dolgui and Proth, 2010]. In general, a system can be either dedicated to a single product type, in which case it is called a dedicated manufacturing system, or it can be designed to produce multiple product types. Besides, manufacturing systems vary with regard to their layout and the level of the flexibility. The literature on manufacturing systems that can handle various products can be categorized into publications on cellular, flexible, and reconfigurable manufacturing systems. The mentioned manufacturing systems are detailed through the following subsections:

1.2.1 Dedicated Manufacturing System (DMS)

A DMS is a mass production system because it focuses on a high volume and low variety of products. Thereby, a DMS is characterized by relatively low costs and high throughput. Some examples are the transfer lines in automotive industry for machining cylinder blocks [Dolgui and Ihnatsenka, 2009, Dolgui et al., 2009]. The fixed structure of a DMS does not allow to increase the product variety or the throughput. The only way to enhance the flexibility of a DMS is to use several DMSs in parallel, where each DMS handles a specific product type (e.g., [Özcan et al., 2010]). A DMS can be reconfigured for new products, but it is costly and time consuming [Makssoud et al., 2014, 2020].

1.2.2 Flexible Manufacturing System (FMS)

An FMS can be efficient in situations where new products are introduced frequently, and companies are shifting from low-mix high-volume to low-volume high-mix production, thus, require more flexibility. A high level of flexibility such that the new product requirements are adapted easily and quickly lead to the high initial investment for an FMS. An FMS is equipped with computer numerical control (CNC) machines connected by an automatic material handling system, where the numerical control is easily changed to process different tasks [ElMaraghy, 2005].

1.2.3 Cellular Manufacturing System (CMS)

A CMS is an implementation of the Group Technology principles [Rajamani et al., 1990, Singh, 1993, Askin, 2013]. A CMS comprises multiple cells, where each cell consists of a set of machines. Each cell is dedicated to the production of a given part family, where each family contains some parts with similar manufacturing requirements. Usually, the machine layout of the same cell is U-shaped to facilitate movements of the worker assigned to stations of the opposite sides of the cell.

1.2.4 Reconfigurable Manufacturing System (RMS)

The concept of an RMS, introduced by Koren et al. [1999] is based on physical reconfiguration of equipment and other resources. Such system is able to adjust to different products of a product family. It is less costly than an FMS and offer a trade-off between the high throughput of a DMS and the universality of an FMS. An RMS is composed of the components such as workforce, machines, tools and material handling devices that can be easily added, removed or replaced. This system permits two levels of reconfiguration: (1) the system level, which changes connections between the components, (2) the components level, which changes the functionality of a component. Thanks to its ability to change the components, the RMS reduces setups and is able to change and adjust the production capacity and functionalities. Bortolini et al. [2018] give a comprehensive review on RMS research trends. They link reconfigurable manufacturing with Industry 4.0 technologies.

Note that FMS and RMS flexibilities are based on different concepts. An FMS is able to change its functionality without changing its physical configuration (except for tools), whereas an RMS is able to change its functionality by changing its physical configuration, modules and pieces of equipment. Both abilities are assumed to be cost effective.

While an FMS is highly flexible, but they have a limited capacity. A DMS is highly productive, but not flexible. Besides, a CMS can be considered as a compromise by using several dedicated cells instead of a sole DMS. However, their applicability is limited by the necessity of a rather predictable demand and a long life-cycle of the manufactured products [Benjaafar et al., 2002]. An FMS does not have such constraints, but it is also costly, less productive and more complex. An RMS is less costly than an FMS, and it provides a customized flexibility when compared to the general flexibility existing in an FMS [ElMaraghy, 2005]. In other words, an RMS creates the capacity and functionality that is needed, when it is needed. Thereby, in terms of capacity and functionality, an RMS may be placed between a DMS and an FMS [Mehrabi et al., 2000].

1.3 Assembly line characteristics

Assembly lines represent a subclass of manufacturing systems, whose specificity consists in their flow shop nature and repetitive production. An assembly line is characterized by a flow of products through workstations or resources over time. Physical layout of stations, resources, processing composition, and transporting equipment can be given, or their selection can be one of the decisions to be made. The incoming product sequences can be finite or infinite, repetitive or not, and they can be fully or partially specified in the deterministic, stochastic or any other sense. The products can be the same or different. Each product requires a set of tasks to be performed. Each task can require certain quantities of additional resources and/or equipment. A station can execute a single task or multiple tasks, sequentially or in parallel. One of the main system's characteristics is the timing of product moves to and from stations. If these moves are made with the same time step, called cycle time or takt, for all stations, then the system is called paced, otherwise, it is called un-paced. A collection of events that happens between two sequential product moves in a paced line is called a production cycle. Many operation research problems can be defined for an assembly line that this thesis converses about a new problem of an assembly line.

Historically, the first assembly lines were fully manual lines where only human workers could work in the line. Moreover, many of such traditional assembly lines were able to produce a single type of products through a mass production system. Lately, steam and electric engines conveyors where employed to move parts from one workstation to another. Today, by developing the technology, many machines and robots are used in assembly lines to perform different assembly tasks. Recent lines are mainly fully automated or semi-automated using both robots and humans, however manual lines still exist. In addition, because of increasing mass customization, many recent companies are transferring their lines from the single model line to the multiple model line. Different types of assembly lines are further clarified within following subsections.

1.3.1 Product variety in assembly lines

A lot of researches are dedicated to assembly line balancing and configuration problems, see the review papers of [Rekiek et al., 2002, Boysen et al., 2008, Battaïa and Dolgui, 2013]. As mentioned, assembly lines represent a subclass of manufacturing systems, whose specificity consists in their flow shop nature and repetitive production. Assembly lines can be DMS, FMS, CMS or RMS. However, in terms of variety of products assembled on the line, assembly lines are commonly classified as dedicated, multi- or mixed-model lines [Bellgran and Säfsten, 2009]. Often manual assembly lines are studied. On multi-model manual assembly lines products of the same type are manufactured in batches, allowing a high level of productivity to the expense of low reactivity in product type changes. On mixed-model assembly lines (MMAL), products of different types can be produced in an arbitrary order, which increases the level of flexibility compared to the multi-model assembly lines. Dedicated assembly lines have the same properties as a DMS. They are designed to assemble a single product type with high throughput. A schema of assembly line variety is demonstrated in Figure 1 in the Introduction.

1.3.2 Resource variety in assembly lines

Moreover, in terms of resources using in assembly lines, they are divided to three main types: manual, automated, and hybrid lines. A manufacturing system able to perform a high variety of tasks requires different types of resources. In this context, manufacturing and assembly systems can be classified into three main types: automated, manual, and hybrid lines. Fully automated systems using robots possess high speed, accuracy, tirelessness, and force, but they are expensive. On the other hand, human workers are intelligent, creative, flexible, and able to work with different tools in different situations. A combination of these resources forms a human-machine/robot (hybrid) system, where humans and robots perform a variety of tasks (manual, automated, and hybrid tasks) in a shared workspace. A Human-robot collaborative (HRC) system refers to a common work space, where robots and workforce collaborate to jointly process a product. Such robots, able to collaborate with workers, are called cobots. HRC systems offer an alternative to fully manual workstations, and it results in workstations gathering the strengths of both humans and robots. Typically, manufacturers introduce cobots in their production system to improve the level of safety, ergonomics, quality, flexibility, and reconfigurability [Krüger et al., 2009, 2011, Koppenborg et al., 2017, Elprama et al., 2017].

More generally, Figure 1.1 [Hashemi-Petroodi et al., 2020c] shows the strengths of humans and robots, and the collaboration between humans and robots, which creates workstations with a mixture of these advantages [Tsarouchi et al., 2016]. Humans have learning and cognitive skills to enhance their ability for performing various tasks. Intelligence and creativeness make them the most flexible resource in manufacturing systems. On the other hand, robots are able to perform a much higher volume of manufacturing tasks thanks to their force, tirelessness, speed, accuracy, and repeatability. These features are the main advantages of humans and robots, whose combination leads to a higher level of productivity, ergonomics, safety, flexibility, and reconfigurability.

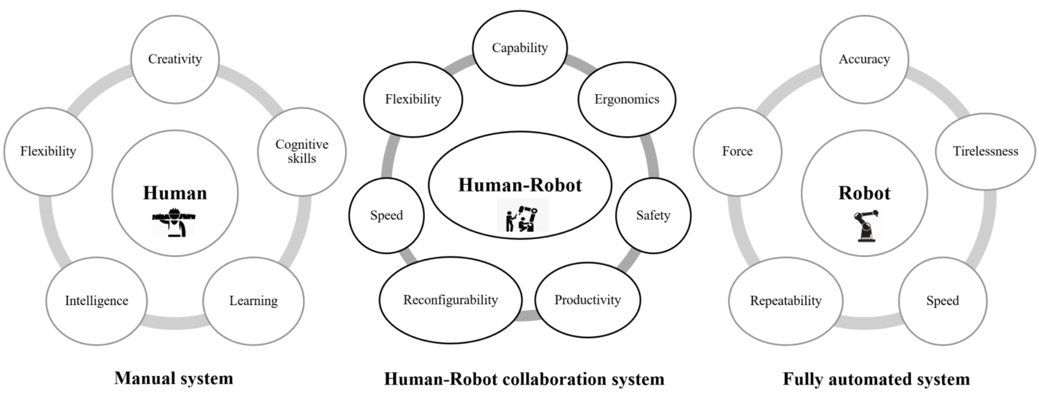

Figure 1.1 – Main characteristics of the three manufacturing systems: manual, fully automated, and human-robot (hybrid).

1.4 Assembly line balancing

Many operations management and optimization problems rise in context of optimizing the performance of assembly lines. Several optimization problems relating to the assembly line design are: process planning (discover the optimal production process, including the set of resource-task-station assignments to produce a single/multiple product, among all alternative generated processes), line configuration (straight, parallel, U-shape lines and etc.), line balancing (task assignment to stations/resources), workforce assignment (similar to line balancing for task assignment to workers), scheduling (task sequencing), equipment selection, product model sequencing (sequence of product models in multi/mixed-model lines). Obviously, it is not possible to consider all these problems in this thesis. Therefore, this thesis mostly focus on the mixed-model line balancing and workforce assignment, simultaneously. Decisions on tasks and workforce assignment have to be made simultaneously, since any change in task assignment may imply changes in workforce assignment [Cortez and Costa, 2015]. However, we study a configuration selection problem with the goal of maximizing the profit of a single multi-model line versus multiple single model line in Chapter 2).

One of the important steps of assembly line design is the line balancing: assignment of tasks to workstations optimizing a given criterion (minimizing takt time or number of workstations, etc.). As one of the initial studies on the assembly line balancing problem (ALBP), Baybars [1986] proposed a Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem (SALBP). Traditionally, two main types of SALBPs were introduced in the literature: SALBP-1, where the goal of SALBP is minimization of the number of workstations for a given cycle time, and SALBP-2, where the goal is minimization of cycle time for a given number of work stations. Later on, SALBPs have been extended in the literature by considering product varieties, different line configurations, and sequence of tasks and as well as product models.

Several restrictive assumptions are commonly made in the literature on assembly line balancing problems in the literature [Baybars, 1986, Scholl and Becker, 2006, Boysen et al., 2008, Battaïa and Dolgui, 2013]: allocating only one worker to each station, producing only a single model in the line, workers and tasks are fixed, etc. However, for heavy tasks of large-size products, like in the automotive industry, assigning more than one worker to each station is more realistic [Lopes et al., 2020]. Some researchers studied multi-manned line balancing problems under different assumptions [Dimitriadis, 2006, Michels et al., 2019, Becker and Scholl, 2009, Kellegöz, 2017]. Furthermore, moving workers between stations adapts the stations' capacities to the production sequence [Sikora et al., 2017]. The flexibility of assembly lines with multiple moving workers at workstations can be enhanced by a kind of dynamic task assignment to stations, which allows conflicting precedence relations to be handled. According to the product variety in assembly lines, ignoring the proposed restrictive assumptions and giving more flexibility look more interesting to be applied in mixed/multi-model assembly lines in compare to single model lines. This is since a multi-manned mixed-model assembly line (MMAL) with walking workers, where product models with different precedence graphs and task process times can enter the line in an arbitrary order, can benefit well from the concept of reconfigurability to well adjust and adapt the line's capacity to production requirements.

As mentioned in Boysen et al. [2009], assembly line balancing (task assignment) and design (resource assignment) are crucial steps for an MMAL. These problems have been studied by many researchers [Bukchin et al., 2002, Choi, 2009, Dolgui and Proth, 2010, Alghazi and Kurz, 2018, Samouei and Fattahi, 2018]. Workforce and task assignment are usually taken into account as decision variables to balance an MMAL [Choi, 2009, Gebennini et al., 2018, Moreira et al., 2015]. For example, Lee and Vairaktarakis [1997] developed some heuristics for workforce minimization in a workforce planning problem in an MMAL. Choi [2009] also proposed a goal programming approach to tackle a task allocation problem in an MMAL. The author tried to find the best trade-off between processing times of the assigned tasks and the workers' physical workload, but he assumed that workers were not allowed to move between stations.

Overall, the literature of MMAL design and balancing is centered around workforce and tasks assignments. Workers can either be fixed at each station [Biele and Mönch, 2018] or move along the line and be re-assigned to other stations/tasks based on their skill sets or product requirements [Naderi et al., 2019]. The flexibility of walking workers in an assembly line improves the reconfigurability of the line [Hashemi-Petroodi et al., 2020a]. In several studies on workforce assignment problems, workers are allowed to move. Battaïa et al. [2015] assumed that workers are allowed to move between stations after finishing a non-preemptive task, and the processing time of a task depends on the number of workers assigned to the corresponding station. The authors built a linear programming model and constructive heuristics to solve a workforce minimization problem in a paced assembly line, motivated by an automotive industry case. The goal was to find a workforce assignment, which minimizes the maximal number of workers used in all production cycles. Following this study, Dolgui et al. [2018] developed a mixed integer linear program and some constructive heuristics to solve the problem in the particular case where the precedence graph is disconnected. The problem was further generalized in Delorme et al. [2019] by including the search for an optimal product sequence. An integer linear programming model, an enumeration and a model dependent programming algorithms were proposed, and polynomial algorithms were presented for special cases with two product types and two workstations.

From another perspective, tasks assignment to stations can be either known [Battaïa et al., 2015], fixed [Özcan et al., 2010, Sikora et al., 2017], or dynamic [Kucukkoc and Zhang, 2014, Hashemi-Petroodi et al., 2020b]. A few publications exist on dynamic task assignment in MMAL balancing for different production environments with fixed workers, such as parallel two-sided MMAL [Kucukkoc and Zhang, 2014], or traditional one-sided MMAL [Choi, 2009]. For instance, in Kucukkoc and Zhang [2014], the assignment of tasks to stations depends on the production cycle, where a production cycle corresponds to a certain combination of product types present in the line. In the literature on MMAL balancing with moving workers, tasks assigned to stations are either fixed or given [Battaïa et al., 2015, Delorme et al., 2019, Dolgui et al., 2018, Hwang and Katayama, 2010]. Recently, Hashemi-Petroodi et al. [2020b] studied the impact of dynamic task assignment on MMAL balancing. However, the concept of dynamic task assignment in Hashemi-Petroodi et al. [2020b] is different from the task reassignment in this study. Hashemi-Petroodi et al. [2020b] considered that tasks can be reassigned at each takt depending on the product sequence and on the state of the line. In the present work, task reassignment is model-dependent, but it does not depend on the takt nor on the sequence of product models. In addition, Hashemi-Petroodi et al. [2020b] only present only a scenario-based model limited to small scale instances, whereas the present work provides a set of more advanced optimization methods to solve larger instances. To the best of our knowledge, there is no existing work on MMAL balancing and design with walking workers and task reassignment. The present work aims to fill this gap. Decisions on tasks and workforce assignment have to be made simultaneously, since any change in task assignment may imply changes in workforce assignment [Cortez and Costa, 2015].

In the literature on assembly lines, several objective functions have been optimized using different optimization methods. Different solution approaches, either exact or approximate methods are developed to solve line balancing problems. Several well-used exact methods include mathematical programming (i.e. integer linear programming, non-linear programming, etc.) models [Battaïa et al., 2015, Biele and Mönch, 2018, Dolgui et al., 2018, Naderi et al., 2019, Taube and Minner, 2018], other exact methods (i.e. branch and bound, constraint programming, dynamic programming, goal programming, benders' decomposition, etc.) [Pereira and Álvarez-Miranda, 2018, Bukchin and Rabinowitch, 2006, Li and Gao, 2014, Alghazi and Kurz, 2018, Delorme et al., 2019, Choi, 2009, Michels et al., 2019]. Exact methods are useful to better understand the problem and solve small size problems, optimally. However, such methods often require an excessive computer time to solve large scale problems. In contrast, approximate algorithms can provide results significantly faster than exact methods, but they do not guarantee to reveal the optimal solutions. Various (meta-)heuristics have been developed in the literature on assembly lines [Pereira and Álvarez-Miranda, 2018, Biele and Mönch, 2018, Dolgui et al., 2018, Li and Gao, 2014, Özcan et al., 2010, Samouei and Fattahi, 2018, AkpıNar et al., 2013, Kucukkoc and Zhang, 2014, Saif et al., 2019, Samouei and Fattahi, 2018].

1.5 Workforce planning in production systems

In the manufacturing context, workforce planning consists in determining the workforce capacity and assigning workers to the tasks. Workforce planning problems vary significantly depending on the nature of the items to produce, the type of the manufacturing system, the decisions to be made, and the optimization criteria. Studies on workforce planning in manufacturing systems were initiated by Akagi et al. [1983] and Shttjb [1984]. Most of the other studies were done in the context of production scheduling problems. Several articles provided the states of the art on the existing advances in workforce planning research [Ammar et al., 2013, De Bruecker et al., 2015].

Traditionally, a task is an indivisible amount of work to be performed on a product item. While mass production led to the design of production lines, which repetitively manufacture large series of the same item, mass customization drives towards multi-item manufacturing systems. When multiple items are produced, their sequences can be finite or infinite, repetitive or not, fully or partially specified.

In most studies, a single worker performs each task, and the task processing time is fixed. However, some studies consider that the processing time of a task depends on the quantity and characteristics of the assigned resources including workforce [Battaïa et al., 2015]. These resource-dependent processing times can be deterministic, stochastic or uncertain due to, for example, the resources unavailability or production failures.

Industrial components (e.g., workforce) possess different skills, and thus they can perform different tasks. This led to the classification of workforce assignment problems in two categories: homogeneous workers or heterogeneous workers. In a manufacturing system with homogeneous workers, all workers are the same with respect to their skills, physical abilities, labor costs, or any other characteristics. Consequently, each worker can perform all the tasks. On the contrary, in a system with a heterogeneous workforce, workers have different skills or skill levels, which creates resource-task assignment restrictions. Typically, in workforce planning problems with heterogeneous resources, each operator is associated with a skill set [Wittrock, 1992], and an assignment is feasible if the assigned task is covered by the operator's skill. These skill-sets may be identical, non-identical or in some cases overlapping from one resource to another. Homogeneity has the same definition and application for machines and robots [Kim and Lee, 1998, Jones et al., 2006]. However, the problems with heterogeneous resources become more complex when workstations include humans and machines/robots at the same time since the processing of a task requires to join the skills of robots and operators.

In practice, workforce skills change with time, because of learning and forgetting effects. In addition, companies can control the skill-sets of the workforce through cross-training strategies. Workforce homogeneity or heterogeneity hardly depends on the task proficiencies. Note that cross-training, learning and forgetting effects are defined for both heterogeneous and homogeneous workforce. Learning and forgetting effects are the same for all workers of a homogeneous workforce, while for a heterogeneous workforce they can be different from a worker to another. There are several studies concerning the impact of cross-training strategies on both heterogeneous and homogeneous workforce [Shafer et al., 2001, Süer and Tummaluri, 2008]. Heterogeneous workforce planning can be viewed as a selection of equipment alternatives for workstations, where different pieces of equipment result in different productivity and cost. In other words, the equipment selection problem is equivalent to a worker selection problem, where workers with different qualifications in terms of production speed or quality are available and are paid according to their qualifications [Akagi et al., 1983, Wilson, 1986]. Assembly line balancing models with equipment alternatives are addressed by [Bukchin and Tzur, 2000, Bukchin and Rubinovitz, 2003], see also the survey by [Battaïa and Dolgui, 2013].

The shop floor's structure has a critical impact on workforce planning. In the classical flow shop setting, all tasks have the same routing from the first workstation to the last. However, in more complicated manufacturing systems, called job shops, tasks have different processing routes through the workstations. Some systems are constrained to process a single task per station, whereas others can perform several tasks sequentially or in parallel. In addition, industrial resources can induce various constraints on the task's allocation, such as space and time constraints. Finally, the timespan between two consecutive items moved from one station to the next one is a crucial characteristic of a manufacturing system. In a paced system, these moves follow the same time step, called cycle or takt time, for all stations. An essential characteristic of a paced production system is that the stations have no buffer to stock the incoming or outgoing products. In un-paced systems, buffers with a limited capacity are set between the stations.

Workforce planning is often combined with design, planning, and scheduling of a whole manufacturing system. In most cases, the physical layout and the composition of the processing and transporting equipment are decided before the workforce planning. However, several works have considered the case where the equipment is selected along with the workforce planning.

An optimization criterion is chosen depending on the need of the decision maker. The typical criteria related to the workforce are minimization of the labor costs, the number of workers in each production cycle, the ergonomic risks, the maximum workload, the workers' traveling distance, and maximization of the work variability and smoothness of the workload. Workforce planning aims to optimize efficiency criteria in different manufacturing systems: minimization of the cycle time, a function of the product completion times (usually in the case of the non-repetitive production), the equipment costs, the cost of the additional resources, maximization of the number of completed products per time unit. Sometimes, these criteria are replaced with the constraints limiting their values.

Note that, contrarily to the service industries, the workers in a manufacturing system follow regular shifts, and the workforce planning decisions usually do not account for the same work constraints and regulations.

1.6 Reconfigurability and flexibility

Flexibility is defined as the capacity of a manufacturing system to change into a variety of states and functions in order to respond to changing requirements with a little penalty of time, cost, or performance [De Toni and Tonchia, 1998]. On the other hand, reconfigurability is the capability to quickly provide a customized flexibility via equipment modularity when needed to meet market requirements. This kind of customized flexibility, in comparison with general flexibility, is specifically addressed for the production of a part family [Koren et al., 1999].

Figure 1.2 [Hashemi-Petroodi et al., 2020c] shows the evolutions of the automation level in manufacturing systems and their impact on the characteristics of the system [ElMaraghy, 2005]. Manufacturing systems evolved from mass manufacturing systems to flexible manufacturing systems (FMSs), and more recently, to reconfigurable manufacturing systems (RMSs) [Koren et al., 1999]. Concurrently, technological progress modified manufacturing systems from fully manual toward almost fully and hard automated, then to flexible, and subsequently to hybrid automated systems. The implementation of fully automated systems converted manual systems into hard automated systems with automated transfer lines. Later, the need for flexibility led to flexible automated systems (e.g., CNC machines in a flexible manufacturing system (FMS)). Finally, the need for reconfigurability fosters the integration of humans and robots in a hybrid automated manufacturing system.

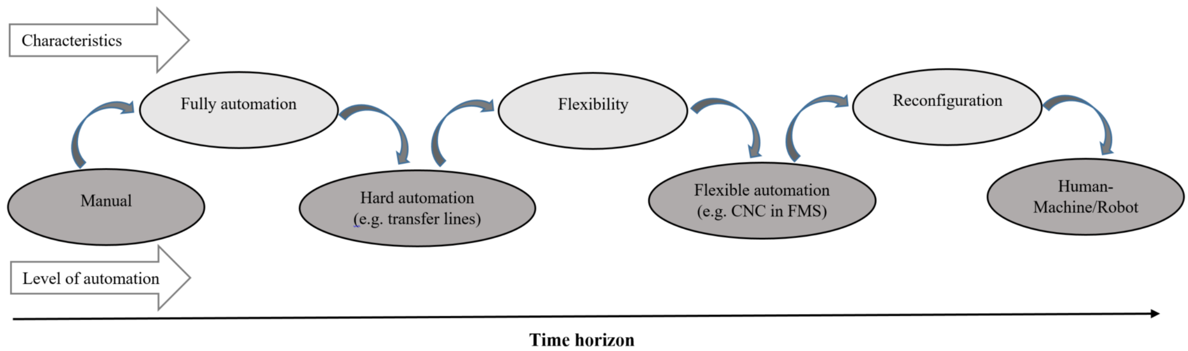

Figure 1.2 – Relationship between human/automated resources and the evolution of manufacturing system.

The design of flexible or reconfigurable automated systems is a hard task. The development of computer numerical control (CNC) machines helped to tackle this challenge with flexible automation into the manufacturing systems (e.g. FMSs). The recent advances in robotics led to the creation of machines able to process a large variety of tasks, and these new robots will further extend the flexibility of automated manufacturing systems. Nevertheless, human workers remain the most flexible component of a manufacturing system. In the mass customization area, manufacturing systems must rely on the human to attain the desired flexibility level [ElMaraghy, 2005].

The adoption of robots in manufacturing systems creates new operations management challenges. For instance, robots and workers have different capacities, and this affects the allocation of the resources in the system and the (re-)assignment of tasks to the workforce/robots. Besides, ergonomics and risk assessments are of critical importance in HRC manufacturing systems. Consequently, the model to design, plan, and schedule the tasks must account for ergonomics and safety constraints. Finally, hybrid human-robot systems can be very flexible and reconfigurable if designed appropriately. The next section presents how the existing literature accounts for these characteristics.

The flexibility of a manufacturing system depends on its resources. Humans and cobots have complementary skills. Humans represent the most flexible resource thanks to their learning, training, and cognitive skills. Cobots are adapted to repetitive work, they are more precise and can handle heavy parts without fatigue. Consequently, the human-robot interaction increases the flexibility of manufacturing systems. For instance, Rahman and Wang [2018] showed the improvement of productivity and quality in automative, aerospace, and electronic industries using flexible assembly lines with cobots.

There is a growing amount of literature on the reconfiguration of machines and equipment. In contrast, workforce reconfiguration is not enough analyzed. Nevertheless, humans are flexible by nature, and they represent an opportunity to enhance the flexibility of manufacturing systems. Unlike machines, which cannot perform a task beyond the scope of their predestination (at least to a certain degree), human workers are creative and able to operate with different tools and equipment. Moreover, a worker can handle a non-standard situation, where an automated resource would fail. Thus, the workers increase the flexibility and adaptability of manufacturing systems.

1.7 Workforce reconfiguration strategies in production systems

This section presents a classification of workforce reconfiguration strategies which is motivated by the fact that there is a growing amount of literature on the reconfiguration of machines and equipment, but not that much focus on workforce reconfiguration. The proposed classification of workforce reconfiguration strategies is based on the concept of reconfiguration of manufacturing systems [Koren et al., 1999, Mehrabi et al., 2000]. A manufacturing system is called reconfigurable if it can modify its specific process capabilities, and subsequently adjust the production capacity to quickly respond to changes in the market demand. In an RMS, it is easy to add, remove, or interchange the components. In other words, the reconfiguration creates the capacity and functionality, which is needed, when it is needed.

The following subsections are dedicated to five workforce reconfiguration strategies studied in the literature. For every research paper related to a workforce reconfiguration strategy, we mention the studied problem's criterion, the type of the manufacturing system and the solution approach. The aim of this section is to know for which manufacturing systems workforce reconfiguration strategies were already studied in literature and for which this is still an open issue. A frame of the classification is demonstrated in Figure 1.3 [Hashemi-Petroodi et al., 2020a].

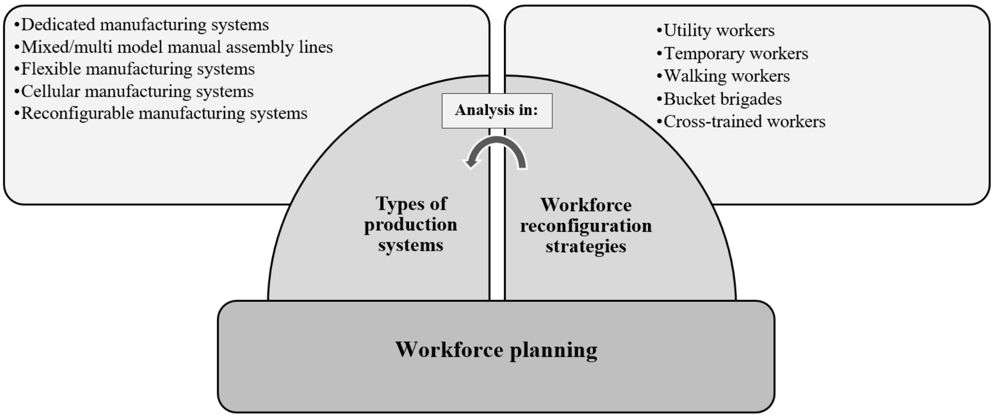

Figure 1.3 – Various workfroce reconfiguration strategies in various production systems.

1.7.1 Utility workers

A task which cannot be executed completely within the workstation's takt time is called a utility work. It may create problems such as line stoppages, increased stocks of unfinished goods between stations, insufficient productivity and, as a result, unsatisfied demand. Utility workers assist permanent workers to complete such tasks. The problems are in designing algorithms to assign utility workers to the tasks. Most of them are scheduling problems. An assignment of utility workers to utility work can be considered as a reconfiguration of a manufacturing system as the allocation of workforce resources may vary from one cycle to another or from one product sequence to another. Table 1.1 presents the classification of the major studies related to the concept of utility workers, in which MMAL stands for mixed-model manual assembly lines.

In these studies, a utility work mostly leads to line stoppages and increased workload.

Table 1.1 – Articles related to utility workers.

Paper	Minimization of	Type of the system	Solution approach
Hyun et al. (1998)	Utility work and setup cost	Straight MMAL	Genetic algorithm
Celano et al. (2004)	Total stoppage time	U-shape MMAL	Genetic algorithm
Yoo et al. (2005)	Weighted sum of line stops and idle time	Straight MMAL	Simulated annealing and Tabu search
Boysen et al. (2011)	Number of overload situations	Straight MMAL	Exact and heuristics
Cevikcan & Durmusoglu (2011)	Total utility work and utility worker transfers	Straight MMAL	Meta-heuristics and local search
Li & Gao (2014)	Total regular and overtime labor costs	Straight MMAL	Heuristic and branch-and-bound-and-remember algorithm
Cortez & Costa (2015)	Utility work needed	Straight MMAL	Mixed integer programming and heuristics
Faccio et al. (2016)	Number of workers and work overload	Straight MMAL	Hierarchical approach
Aroui et al. (2017)	Total work overload	Straight MMAL	Mixed integer linear programming, simulated annealing, genetic algorithm

The studied problems are related to product sequencing [Yoo et al., 2005, Boysen et al., 2011, Cevikcan and Durmusoglu, 2011, Cortez and Costa, 2015], line balancing [Li and Gao, 2014] and both sequencing and balancing [Faccio et al., 2016]. One can notice that all these studies consider an MMAL. This is expected since assembly lines rely mainly on manual operations. Besides, product differentiation is often done in the assembly step, and assembly lines must be reconfigurable. The solution methods, which are mostly composed of heuristics and meta-heuristics, reflect, on one hand, the complexity of the studied problems and, on the other hand, emphasize the importance of solution times.

These studies present various ideas of how utility workers may assist regular workers: sequentially, in parallel or replacing a regular worker completely. In Celano et al. [2004], if a task is not completed on time, a utility worker intervenes and assists the regular worker in completing the task. Boysen et al. [2011] study the case where utility workers do not help, but rather replace regular workers to finish the task. Regular workers, in turn, start processing the next part. Utility workers that operate in parallel or after regular workers in the same cycle are called “jolly workers” [Faccio et al., 2016]. A kind of utility workers is considered in Aroui et al. [2017], where some workers work besides regular workers to minimize the overloading. Line balancing with a demand changing from shift to shift, both in terms of volume and product mix, is considered in Li and Gao [2014]. Cortez and Costa [2015] study a case, where heterogeneous regular workers are assisted by utility workers able to perform any task.

1.7.2 Temporary workers

Temporary workers can be used to help permanent workers. As the temporary workers are in most cases, less skillful than regular workers, they usually perform only a specific subset of tasks. Temporary workers improve the adaptability and, therefore, responsiveness of a manufacturing system in case of a high seasonal or uncertain demand [De Bruecker et al., 2015, Corominas et al., 2008, Francas et al., 2011]. Table 1.2 summarizes the recent literature, which concentrates on the use of temporary workers and corresponding optimization problems.

Table 1.2 – Studies concentrating on temporary workers.

Paper	Criteria	Type of the system	Solution approach
Stratman et al. (2004)	Minimization of the total cost	Straight MMAL	Discrete event simulation
Techawiboonwong et al. (2006)	Minimization of workforce-related and inventory costs	Straight MMAL	Mixed integer programming
Corominas et al. (2008)	Minimization of the number of temporary workers	Straight single-model assembly line	Integer linear programming
Widyadana (2009)	Minimization of the number of temporary workers and the cycle time	U-shape single-model assembly line	Goal programming
Francas et al. (2011)	Maximization of the difference of expected second-stage profits and first-stage investment costs	Straight multi-model assembly line	Two-stage stochastic model
Manavizadeh et al. (2013)	Minimization of the total weighted idle time, workload imbalance, uneven distribution of idle time	U-shape MMAL	Simulated annealing
Buyukkaramikli et al. (2013)	Minimization of the flexible crew cost	Parallel single-model assembly line	Transient behaviour analysis of multi-server queues
Kim et al. (2018)	Minimization of the total operating and workers cost, the cycle time, and work overload	Straight MMAL	Integer and mixed integer linear programming and hybrid genetic algorithm

Several researchers proposed solutions to workforce assignment problems where temporary and permanent workers have different skill levels [Stratman et al., 2004, Techawiboonwong et al., 2006, Corominas et al., 2008, Manavizadeh et al., 2013, Kim et al., 2018]. For example, in [Stratman et al., 2004] it was showed that allocating skilled permanent workers upstream of the production process leads to a better cost efficiency. In Buyukkaramikli et al. [2013], the authors compared the hiring of temporary and permanent workers in a make-to-order production system. The cost incurred for a temporary crew is higher than the one for a permanent crew. However, it decreases as the length of the hiring period increases. The results showed that the highest cost reduction is achieved when the cost of a flexible crew equals the cost of a permanent crew.

In terms of the layout, Widyadana [2009] studied a MMAL balancing problem with permanent and temporary workers, and they show that a U-shape line provides better results compared to a straight line.

1.7.3 Walking workers

Walking workers are not fixed to a given workplace and may follow the processed product until its last task. Upon completion, they return upstream to start processing a new product unit [Al-Zuheri et al., 2014]. Several studies, Bischak [1996], Deepak et al. [2017] showed that moving workers, whose dynamic reassignment allows increasing the workforce resource where and when needed, improve the performance of production lines and provide larger throughput, larger resource utilization, and less work in process.

A walking worker can be skilled or unskilled, temporary or permanent. Chen et al. [2016] considered a so called “chasing-overtaking” production line, in which workers with high efficiency are allowed to overtake workers with low efficiency at workstations. The conducted simulation showed the superiority of the “chasing-overtaking” production line over traditional and bucket brigade (see Section 1.7.4) production lines in terms of production capacity and resource utilization. In Pröpster et al. [2015], workforce-related reconfigurability is expressed in two ways: drifting of workers within a station and so-called “jumpers”, i.e. workers able to intervene to any station if necessary. Table 1.3 presents only some resent papers related to walking workers, classified by content/criteria, production system's type and solution approach. However, there are many papers in the literature related to walking workers which are mentioned in Hashemi-Petroodi et al. [2020a].

Mahdavi et al. [2010], Soolaki [2012] studied workforce assignment problems in a dynamic CMS with reconfiguration, i.e., adding, removing and changing machines between cells. In these studies, workers can be removed from one cell and assigned to another cell in each time period. A similar production line configuration, in which workers can move from one station to another after completing a task in the MMAL, was considered in Battaïa et al. [2015], Dolgui et al. [2018], Delorme et al. [2019]. This movement changes the number of workers assigned to the tasks at stations, which, in turn, either increases or decreases corresponding task processing times. The objective was to find an optimal scheduling of worker moves among stations minimizing the number of workers while respecting the line takt time. Most studies confirm that skilled walking workers improve the manufacturing system's performance and responsiveness. The reconfigurabity increases as well, as they shift productive capacity from one workplace to another in order to adapt it to the current situation in a production system.

Table 1.3 – Studies considering walking workers.

Paper	Content or criteria	Type of the system	Solution approach
Cevikcan (2016)	Minimization of the number of workforce and maximization of the workload smoothness	Straight Multi-model assembly line	Heuristics
Kucukkoc & Zhang (2016)	Minimization of the weighted summation of line length and the number of workstations	Parallel two-sided MMAL	Agent-based ant colony
Al-Zuheri et al. (2016)	Minimization of the total cost	U-shape MMAL	Genetic algorithm
Vairaktarakis et al. (2016)	Levelling criteria: maximization of the workforce size and minimization of the maximum workforce fluctuation	Straight MMAL	Heuristics
Liu et al. (2016)	Minimization of backorder and holding costs	Dynamic Cellular manufacturing system	Meta-heuristics
Kellegöz (2017)	Minimization of the number of workers and stations opened in the line	Straight single-model assembly line	Simulated annealing
Deepak et al. (2017)	Maximization of the resource utilization and minimization of the work in process	Straight single-model assembly line	Simulation
Stadnicka et al. (2017)	Minimization of the walking path of the workers	Straight single-model assembly line	Simulation
Sikora et al. (2017)	Minimization of the cycle time	Straight MMAL	Mixed integer linear programming
Kuo & Liu (2017)	Minimization of the number of workers	Cellular manufacturing system	Mixed integer linear programming
Feng et al. (2017)	Minimization of the total cost	Cellular manufacturing system	Particle swarm optimization
Lian et al. (2018)	Minimization of the deviations from the average workload of cells and workers' number	Cellular manufacturing system	Non-dominated sorting genetic algorithm
Baykasoğlu et al. (2018)	Minimization of machine and worker duplication costs	Cellular manufacturing system	Integer and constraint programing
Biele & Mönch (2018)	Minimization of labor and inventory costs	Straight MMAL	Random-key genetic algorithm
Dolgui et al. (2018)	Minimization of the maximum number of workers	Straight MMAL	Mixed-integer linear programming and heuristics
Gebennini et al. (2018)	Minimization of the workers' walking cost and ergonomic risks of scheduled jobs	Straight single-model assembly line	Mixed integer linear programming
Naderi et al. (2019)	Minimization of the number of workers	Five-sided MMAL	Benders' decomposition
Delorme et al. (2019)	Minimization of the maximum number of workers	Straight MMAL	Integer linear programming and dynamic programming
Méndez-Vázquez & Nembhard (2019)	Estimation of system's productivity in four scenarios related to its configuration	Cellular manufacturing system	Simulation

In many researches, workers are assigned to stations based on their skill levels [Nakade and Ohno, 1999, Wang et al., 2007, Nakade and Nishiwaki, 2008, Al-Zuheri et al., 2013, Egilmez et al., 2014, Dalle Mura and Dini, 2016, Lian et al., 2018, Méndez-Vázquez and Nembhard, 2019]. Indeed, workers must be properly trained to perform multiple or complicated tasks efficiently. Learning by doing repetitive tasks usually reduces processing times, whereas long periods between two successive similar tasks lead to forgetting and increase processing times. The worker assignment taking into account learning and forgetting effects has drawn a certain attention from researchers [Anzanello and Fogliatto, 2007, Thongsanit et al., 2010, Wang et al., 2013, Liu et al., 2016].

In Battini et al. [2007], the authors studied a semi-automated line, where workers perform tasks on a multi-turn rotation table. Sikora et al. [2017] provided some real case studies with human workers and robots, assignment restrictions, zoning constraints, tasks executed by machines and common tasks requiring at least two workers.

Bock et al. [2006] used workers' movement in a real time control of an MMAL to deal with disruptions caused by a worker's absence, material bottleneck, or machine breakdown, among others. Al-Zuheri et al. [2016] studied the impact of distances between workstations, number of stations, layout design and a workload assigning method on ergonomic measures including energy expenditure and walking time to standing position working time ratio. In Yang et al. [2013] both tasks and workers are allowed to be reassigned to other stations when a change of the demand occurs. Battaïa et al. [2015] studied a workforce planning problem, in which workers are allowed to move between stations after finishing a task.

A combination of moving and temporary workers was considered by [Francas et al., 2011]. The authors proved that temporary workers always decrease the investment in regular workers. It was also shown that, in spite of a possible increase of investment on moving the regular workers due to a positive influence on labor utilization, moving workers enhance the efficiency of temporary workers. Thus, an industrial company may benefit from a right combination of temporary and moving workers.

1.7.4 Bucket brigades

Bartholdi III and Eisenstein [1996] introduced a self-balancing approach for flow shop manufacturing systems, called "bucket brigade" (BB). A bucket brigade is an organization of workforce movement, where the number of workers is lower than the number of stations and a worker follows the part from one station to the next until he/she meets his/her successor. Once a worker meets his/her successor, the successor takes over the work on the product, and the worker moves upstream to take over the part of his/her predecessor and so on. Bartholdi III and Eisenstein [1996] demonstrated that sequencing workers from the slowest to the fastest leads to a stable partition of work making the bucket brigade self-balancing.

In a survey paper, Bratcu and Dolgui [2005] pointed out the main advantage of bucket brigades, namely, their adaptability to changing operational conditions like task times, product mix, spatial configuration modifications, etc. Moreover, their relatively easy implementation reduces the design and control effort, making the corresponding reconfiguration strategy popular among practitioners.

Despite the deterministic nature of the basic bucket brigade model, it can have a chaotic behavior that negatively influences the performance of an assembly line [Bartholdi III et al., 2009]. Indeed, the hand-offs can be unpredictable when workers are interrupted at any time or any position of the line. In the initial model the return velocity was considered as infinite. Song et al. [2011] studied bucket brigades with limited return velocities and analyzed their impact on the line's stability and productivity. They demonstrated that bucket brigades with the same return velocity are self-balanced and that the line's productivity is directly proportional to the value of return velocity.

Lim [2011] introduced the concept of cellular bucket brigade (CBB), where the workers operate in aisles with production lines on both sides. A worker performs tasks at one side of the line moving in one direction, but when he/she reaches his/her successor, this worker executes tasks at the other side of the line, moving in the other direction. Thus, unproductive traveling times are reduced. Lim [2011] proposed simple rules for work sharing and a sufficient condition for self-balancing. Numerical experiments showed a 30% to 50% increase in throughput compared to the traditional bucket brigade model [Lim, 2012, 2017].

Sriram et al. [2014] considered a bucket brigade approach in a U-shape assembly line with buffers. They proposed a new control protocol for bucket brigades. By using a discrete events simulation and an optimization model, the authors determined optimal buffer locations and buffer control levels associated with each worker maximizing the line throughput. A buffer level is the amount of excess production capacity in a production line that is included to ensure that production goals are met in the event of downtime. Lim and Wu [2014] proposed some simple cellular bucket brigade rules to coordinate workers in a U-shape assembly line with stations in which at most one worker is allowed to operate at a station. The goal was to maximize the productivity of the line. The simulation results show that the number of stations has a critical impact on the performance of a cellular bucket brigade.

Table 1.4 contains recent papers on bucket brigades, where BB and CBB stand for bucket brigades and cellular bucket brigades, respectively. Other studies are mentioned in Hashemi-Petroodi et al. [2020a].

Table 1.4 – Studies considering bucket brigades.

Paper	**Content or criteria**	**Type of system**	**Solution approach**	**CBB/BB**
Lim & Yang (2009)	Maximization of the throughput	General manufacturing system	Heuristic (simulation)	BB
Quintana et al. (2009)	Maximization of the machine availability and utilization	General manufacturing system	Simulation	BB
Koo (2009)	Maximization of the workers' productivity	Order picking system	Simulation	BB
Wang et al. (2009)	Minimization of in-process waiting times	U-shape MMAL	Simulation and mathematical modelling	BB
Wang et al. (2010)	Minimization of in-process waiting and traveling times	Assembly line	Mathematical modelling	BB
Song et al. (2011)	Maximization of productivity and production stability	General manufacturing system	Heuristic	BB
Webster et al. (2012)	Maximization of the throughput	Order picking line	Discrete event simulation	BB
Lim et al. (2011)	Minimization of the unproductive travel. Simple rules leading to the line's self-balancing	Generalized assembly line	Numerical simulation	CBB
Lim (2012)	Minimization of the unproductive travel. Simple rules leading to the line's self-balancing	Order picking line	Numerical simulation	CBB
Sriram et al. (2014)	Maximization of the throughput	U-shape cellular manufacturing system	Discrete event simulation	CBB
Lim & Wu (2014)	Minimization of the unproductive travel. Simple rules leading to the line's self-balancing	Generalized assembly line	Numerical simulation	CBB
Lim (2017)	Minimization of the unproductive travel. Impact of hand-off times on the CBB performance	Generalized assembly line	Numerical simulation	CBB
Zhou et al. (2017)	Minimization of the unproductive travel	Generalized assembly line	Mathematical modelling and simulation	CBB

1.7.5 Cross-trained workers

A cross-trained worker is a worker able to perform multiple tasks in various locations of a manufacturing system when needed [Ebeling and Lee, 1994]. Compared to walking or bucket brigade workers, who are initially trained to perform multiple tasks and whose movement is planned, cross-trained workers are specialized on specific tasks but also trained to perform other tasks in case of an unplanned necessity. Such unplanned necessities include an ill operator, a change of product mix, or a change in the demand of specific products. Workers' cross-training improves their understanding of the whole production process and tends to increase the overall quality of the manufactured products.

A cross-trained worker's timely response to unplanned situations enhances the flexibility of a manufacturing system. On the other hand, cross-training is costly, and it can increase the production time. To mitigate these shortcomings, several strategies for efficient cross-training were introduced. In the chain cross-training strategy [Inman et al., 2004], workers are trained to execute a secondary task, and tasks are allocated to the workers in a chain. For example, worker A performs tasks 1 and 2, worker B executes tasks 2 and 3, and so on, where the latter task for each worker is the secondary task. Hopp et al. [2004] proposed two other strategies, namely, cherry-picking and skill chaining. In cherry-picking, cross-trained workers assist their colleagues in a bottleneck station to increase the system's throughput. Such strategy implies a higher investment in workers' cross-training. Skill chaining reduces cross-training costs since only workers from an adjacent station assist directly at the bottleneck station. Others assist indirectly by taking part of the work of the following or preceding station. More details on skill chaining with cross-trained workforce are presented in Tekin et al. [2002]. A summary of the recent studies on problems with cross-trained workers is given in Table 1.5. Other studies are mentioned in Hashemi-Petroodi et al. [2020a].

The positive impact of using cross-trained workers on the production system's performance was proved in numerous studies. For example, in Sayın and Karabatı [2007], the authors proposed a simulation model to analyze the impact of some parameters on the utility and skill improvement. These parameters include the number of workers, departments, demand for workers, learning speed, demand variation, etc. The authors suggest that cross-training and skill improvement lead to higher system's productivity. Davis et al. [2009] showed that an extensive cross-training improves the performance under high workload variation conditions. However, in the case of insufficient capacity of equipment in a job shop manufacturing system, additional training expenses are not justified by

Table 1.5 – Studies considering cross-trained workers.

Paper	Criteria	Type of system	Solution approach
Yue et al. (2008)	Maximization of the system's efficiency	Job shop	Simulation
Kaku et al. (2008)	Maximization of the productivity, minimization of the inventory and stock outs	U-shape MMAL and Cellular manufacturing system	Heuristic (human-factor-based training approach) and simulation
Davis et al. (2009)	Minimization of the workload imbalance	Job shop	Simulation
Aryanezhad et al. (2009)	Minimization of the total cost including production, hiring, firing, and training costs	Dynamic cellular manufacturing system	Linear programming
Bokhorst & Gaalman (2009)	Maximization of the productivity	Job shop	Simulation
Satoglu & Suresh (2009)	Minimization of cross-training, hiring, firing, and over-assignment of workers to more than one cell	Hybrid (adapted both to high/stable and low/sporadic demand) cellular manufacturing system	Goal programming
Campbell (2011)	Maximization of the workers' utility in the departments	General service system	Two-stage stochastic approach
Easton (2011, 2014)	Minimization of the labor cost, maximization of the service level	General service system	Two-stage stochastic approach
Kim & Nembhard (2013)	Minimization of the number of workers	Parallel MMAL	Data mining technique
Xu et al. (2015)	Minimization of the total workforce-related cost and maximization of the customer satisfaction	General service system	Binary programing and non-dominated sorting genetic algorithm
Yang & Gao (2016)	Minimization of the number of skill zones (stations)	Straight MMAL	Branch-and-bound
Wu et al. (2018)	Minimization of the training cost, maximization of the workload balance	Cellular manufacturing system	Particle swarm optimization and artificial bee colony
Chu et al. (2019)	Minimization of the costs related to the workers' training, assignment, and workload imbalance	Cellular manufacturing system	Adaptive memetic differential search algorithm

the marginal improvement related to cross-training. The impact of cross-trained workers' learning and forgetting effects on the performance of manufacturing systems were also investigated by [McCreery and Krajewski, 1999, Kim and Nembhard, 2013, Chu et al., 2019].

Several studies on using cross-trained workforce in dual resource constrained (DRC) production systems were conducted, for example in Yue et al. [2008], Bokhorst et al. [2004], Hottenstein and Bowman [1998], ElMaraghy et al. [2000], Davis et al. [2009], Satoglu and Suresh [2009], Xu et al. [2015]. DRC system is a manufacturing system, which is not only constrained by machine capacity, but also by workforce capacity. Cross-trained workforce is also largely used in CMS [Slomp et al., 2005, Kaku et al., 2008, Aryanezhad et al., 2009, Wu et al., 2018, Chu et al., 2019].

1.7.6 Analysis of workforce reconfiguration strategies

This subsection reviews the literature on different types of manufacturing systems and certifies the relative significance of workforce reconfiguration strategies for each of them. It is interesting to see how the five workforce reconfiguration strategies, have been studied across different types of manufacturing systems. Figure 1.4 [Hashemi-Petroodi et al., 2020a] gives the number of papers based on both the workforce reconfiguration strategies and manufacturing system's types. This figure indicates that most of the studies consider mixed-model manual assembly lines and emphasize the importance of utility, walking and cross-trained workers in the system's reconfigurability.

A large number of studies exist on workforce assignment in a DMS, see for example [Sungur and Yavuz, 2015, Lai et al., 2019]. Several researchers studied workforce assignment for a single-model assembly line [Nakade and Ohno, 1999, Miralles et al., 2008, Moreira and Costa, 2009, Chaves et al., 2007, Anzanello and Fogliatto, 2007, Thongsanit et al., 2010]. In contrast, due to the fact that only one product can be produced by a DMS, there are only few studies regarding workforce reconfigurability (e.g., [Corominas et al., 2008, Moreira and Costa, 2009, Gebennini et al., 2018]. On the other hand, a large body of literature is dedicated to workforce assignment problems related to mixed/multi-model assembly lines (e.g., [Battaïa et al., 2015, Delorme et al., 2019], since such lines are usually manual. In most cases, see Figure 2, the line's adaptability is achieved by walking workers, who, upon completion of a task, can be assigned to another task at another station. On the one hand, walking workers allow to have a necessary minimal amount of workers to accomplish the task and therefore keep the production going. On the other

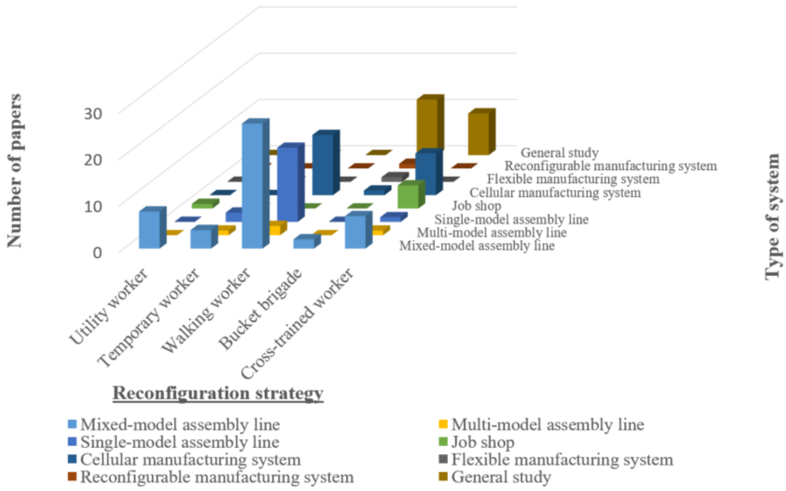

Figure 1.4 – The number of papers based on both workforce reconfiguration strategy and manufacturing system's type.

hand, this strategy can decrease the task's processing time and consequently increase the line's productivity. In many studies, the number of workers and workforce-related costs (e.g. cost of temporary workers' hiring, cross-training cost) are a part of the problem's criterion. In such case, the cycle time criterion is usually replaced by the corresponding constraint, limiting the value of a station time.

A CMS can be viewed as a collection of several assembly lines (cells), each of which is designed to process only a specific set of products. Thus, a CMS represents a mixture of flow and job shop systems. Compared to a static situation, where the demand volume and product mix are known, a multi-period problem with changing demand volume and product mix requires a CMS to be robust and adaptive. Historically, the re-assignment of machines between cells (adding, removing and swapping) was the first type of CMS's reconfiguration, see for example [Safaei et al., 2008, Papaioannou and Wilson, 2010]. In a CMS involving workers, cells quite often have a U-shaped layout, allowing workers assigned to a cell to move from one station (machine) to another in a short time [Schrader and Elshennawy, 2000]. In a quickly changing dynamic environment, the adaptability of a CMS can be increased by using the workforce reconfiguration strategies. While many studies on multi-period dynamic CMS with workforce considered the possibility of workers' firing [Satoglu and Suresh, 2009, Mahdavi et al., 2010], some of the workforce reconfiguration strategies can provide an alternative, in which the number of workers do not change. Thus, utility, moving or cross-trained workers can travel between cells, providing necessary skills and manpower when and where needed without demoralizing layoffs related to a sudden drop in demand, for example. On the other hand, training costs incurred by these strategies can be relatively high. An adequate trade-off between using these strategies and changing the number of workers should be made.

In general, the literature on workforce in FMS is poor, since a long time they were considered as fully automated systems, mainly composed of CNC machines and robots. It is extremely hard to find even the keyword "workforce" or "workers" in the FMS-related literature, which is itself quite scarce. Sometimes researchers describe another system, using the term FMS. For example, Cronin et al. [2019] call an assembly line an FMS. Bortolini et al. [2019] use the term FMS to denote a CMS. Lee et al. [2020] used the term FMS in its conventional meaning. The authors considered workers, who load parts of different type on a pallet, which is then released into the system, composed of numerical control machines and the central buffer. The workers also unload the pallets. The studied problem consists in minimizing the total tardiness, taking into account, among other constraints, workers' availability times.

Due to its complexity, an FMS requires the presence of a highly skilled personnel to control the production process [Mehrabi et al., 2002]. It comes at a cost and urges a company to reduce the number of such operators as much as possible, taking into account the high cost of an FMS itself. Therefore, it can be concluded that the scope of workforce reconfiguration strategies' application to an FMS was extremely small. Nevertheless, the new tendencies consist in adding workers into FMS to decrease the cost and increase the reliability, thus the workforce planning problems also concern FMS.

Workforce planning in RMS has been generally ignored by the researchers. Only a few papers shed light on this aspect. Askin and Huang [1997] developed two integer programming models to assign workers and determine their individual training programs. Peruzzini and Pellicciari [2017] claimed that in order to create an effective smart factory context (e.g., a FMS or a RMS), human performance should be taken into account and managed in the most efficient way. In the paper [Gyulai et al., 2017], the authors proposed a method to minimize the number of workers in a reconfigurable assembly system with constraint programming and genetic algorithms. Harari et al. [2018] took into account the human resource as a component of the design process of flexible and reconfigurable assembly systems. Andersen et al. [2018] demonstrated that convertibility, i.e. ability to change the functionality of a system to meet new production requirements, is easier to implement in a high-level manual production than in a less manual manufacturing system.

Noticeably, convertibility is one of the main characteristics of RMS [Koren et al., 1999]. A flexible workforce increases the convertibility of the manufacturing system.

The following differences between FMS and RMS lead to consider that workforce planning problems in RMS represent a promising research avenue. Firstly, RMS is less automated than FMS. Secondly, in contrast to the numerical control flexibility of FMS, the main principle of RMS is a physical reconfiguration of resources. Finally, RMSs are mixed systems with CNC machines, Reconfigurable Machines Tools (RMTs), traditional machines, collaborative robots (cobots) and reconfigurables workstations where workers play an important role. Workforce is one of the main resources in an RMS, and the principles of RMS foster its reconfiguration. Surprisingly, there are only few studies on workforce reconfiguration in RMS. In contrast to machines, human workers are naturally flexible and able to perform a task, which is not necessarily related to the scope of their predestination. A human worker can handle a non-standard situation, in which a machine would definitely fail, because, in case of such situation, it has no predetermined procedure to follow. Even though a recent progress in artificial intelligence may mitigate this flow, the aspect of cost of such smart and adaptive machines cannot be ignored. Usually the worker's training required to improve or acquire certain skills and, therefore, increase his or her flexibility, is cheaper than building a new functionality of a machine. Considering these factors, an application of workforce reconfiguration strategies in RMS represent an interesting research direction for future studies.

In contrast to the large number of studies on workforce planning in single/mixed/multi model assembly lines and CMS, the corresponding literature related to FMS and RMS is poor. Figure 1.5 [Hashemi-Petroodi et al., 2020a] positions different manufacturing systems according to two factors: the amount of literature associated with workforce reconfiguration and the importance of system's reconfigurability.

Figure 1.5 shows the mismatch between the high importance of reconfigurability for an RMS and the scarcity of corresponding studies related to workforce reconfiguration, thus emphasizing an interest in such research. In order to enhance the contrast and logical connection between different workforce reconfiguration strategies and the different types of manufacturing system, Table 1.6 presents them with regard to the existing literature ($\times$) and the open issues for future studies (?). The papers existing in the literature have been presented before, and several future research directions are proposed to be taken into account in future researches. In the current state, FMSs and RMSs use only equipment flexibility and reconfigurability. For FMSs, this can be explained by the fact that

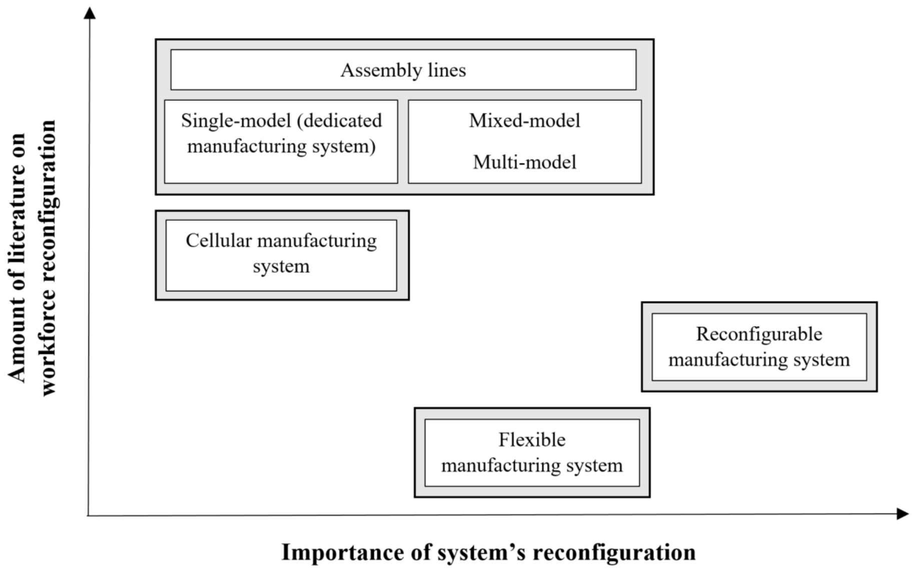

Figure 1.5 – Compliance of the workforce reconfiguration related studies to the importance of system's reconfiguration for different manufacturing systems.

they were considered in the past as fully automated systems. Nevertheless, the last tendencies in industry consist in adding human workers into the FMS. For RMS, this is even more important and crucial, because RMSs are mixed systems with CNC machines, RMT, traditional machines-tools, cobots and reconfigurable workstations, etc. The study of reconfigurability of RMS, based on both machine and workforce reconfigurations is a challenging research issue. The advantages of workforce flexibility and how workforce flexibility can improve the overall adaptability of production systems in the case of FMS and RMS are not studied in literature and can be new promising research directions.

Table 1.6 – Current state of applying workforce reconfiguration strategies to different manufacturing systems.

	Dedicated manufacturing system and mixed/multi-model assembly line	**Cellular manufacturing system**	**Flexible and reconfigurable manufacturing system**
Utility workers	×	?	?
Temporary workers	×	?	?
Walking workers	×	×	?
Bucket brigades	×	×	?
Cross-trained workers	×	×	?

Numerous studies show that workforce reconfiguration strategies have a positive impact on the manufacturing system's efficiency. For instance, the use of utility workers reduces production stoppages, and it decreases the stocks of unfinished goods. Temporary workers help coping with sudden demand increases. Walking workers allow to adjust capacity to different combinations of unfinished goods located in a manufacturing system at a certain moment of time. Bucket brigades provide an easy-to-implement worker assignment rule able to adapt to fluctuating operational conditions. Cross-trained workers apply their broad skills in order to react to unplanned situations.

However, the use of such strategies comes at a cost and may lead to certain side effects like the increased workers' stress and overload. Using temporary or cross-trained workers may bring several advantages for a company, such as an increased productivity and responsiveness [Stratman et al., 2004, Sayın and Karabatı, 2007]. However, the implementation of these strategies incurs the increased cost of hiring and training, which may not necessarily be reasonable. The use of walking workers increases the input of a manufacturing system, but an excessive overload of such workers may lead to a fatigue and stress, which, in turn, negatively affects the system's performance. In fact, manufacturers need to properly trade-off advantages against the disadvantages caused by these strategies. For example, Slomp et al. [2005] try to find the best possible trade-off between the operating costs of a manufacturing cell, related to the workload of the most charged worker, and the cross-training costs.

Most studies on manufacturing systems with workforce consider the criteria of efficiency, throughput and costs. Ergonomic side effects such as fatigue, injuries, absenteeism and stress, caused by overload, frequent task change, movement or inadequate work space organization, are not yet sufficiently studied. However, this issue becomes more and more relevant in the recent publications on workforce planning. These studies take into account workers' fatigue through repetitive movements [Asensio-Cuesta et al., 2012], metabolic energy expense [Al-Zuheri et al., 2016], risks and psychological costs of the heavy tasks [Gebennini et al., 2018]. Otto and Battaïa [2017] surveyed the literature on optimization methods for assembly line balancing and job rotation scheduling, which takes into account physical ergonomic risks. In those studies, ergonomic risks are either included in the objective function or represented as constraints. This survey might be useful for the future studies in this direction. Besides, future studies on ergonomic risks in workforce planning and assignment could benefit from consideration of the workers' cognitive load and its measuring methods.

Specific industrial situations favor a certain workforce reconfiguration strategy. For example, using temporary workers can be useful for a company that produces seasonal products [Corominas et al., 2008], while bucket brigades, thanks to their self-balancing nature and relatively easy implementation, are especially useful in case of short life cycle products manufacturin [Bartholdi III and Eisenstein, 1996]. At the same time, these strategies are closely connected to each other in practice. For instance, bucket brigades and cross-trained workers can be seen as a kind of walking workers. Workers' movement in bucket brigades follows the constant simple rules, while cross-trained workers move from one station to another in case of necessit [Ebeling and Lee, 1994]. In fact, a proper combination of strategies may provide better results than implementing only one. For example, Cevikcan and Durmusoglu [2011] and Francas et al. [2011] found the benefits of using moving workers in combination with temporary and utility workers.

Researchers studying operations management problems have paid a little attention to such hybrid systems. However, using collaborative robots, so-called cobots, helps manufacturing systems to improve their efficiency combining the advantages of workforce (e.g. flexibility, creativity, trainability, intelligence) and the advantages of robots, such as force, accuracy, tirelessness and speed [Hashemi-Petroodi et al., 2020c]. There are several ways of interaction between robots and humans in a hybrid system that affect the control, balancing and planning of a manufacturing system: independent, simultaneous, sequential, and supportive [El Zaatari et al., 2019]. A heavy task, which is dangerous for a worker, can be performed by a robot, while workers can perform certain delicate tasks requiring less force but more flexibility. In order to avoid the monotony of habitual operations, certain safe tasks can be from time to time performed or assisted by workers. In modern quickly changing market conditions, hybrid human-robot manufacturing systems must be adaptive, which requires a high degree of reconfigurability. Such a reconfiguration does not only concern the robots but also the workforce, and the use of utility, temporary, cross-trained, moving workers or bucket brigades would allow a timely and efficient adjustment of resources. The specificity of workforce reconfiguration strategies in such system consists in the consideration of inevitable human-robot interaction, and it opens some promising research directions.

1.8 Conclusion

The rise of mass customization and shortening product life cycles drive industrial companies to employ manufacturing systems with a high level of reconfigurability needed to adapt to quickly changing market conditions. The core interest of workforce planning lies in the workforce's ability to enhance the manufacturing system's reconfigurability. The current study provides a literature review on the research related to workforce reconfiguration. The studies are classified according to five workforce reconfiguration strategies: the use of utility, temporary, moving, cross-trained workers and bucket brigades. These strategies are presented in the context of different manufacturing system types: dedicated, flexible, cellular, reconfigurable manufacturing systems and assembly lines.

The review ascertains that most of the studies are dedicated to assembly lines with workers, since they are often used in practice. The number of papers with the keyword "assembly line" significantly exceeds the one with the keywords "flexible", "cellular" or, to the less extent, "reconfigurable" manufacturing systems. However, they are not mutually exclusive. For example, an assembly line that has a customized flexibility, changeable workstation structures, product variety and reconfigurable workforce, can be considered as a reconfigurable manufacturing system.

The literature analysis reveals a lack of study on workforce reconfiguration in reconfigurable manufacturing systems. In spite of a significant amount of literature on reconfigurable machines and tools, a combined approach integrating machines' and workers' reconfiguration has not been studied yet. Unlike a flexible manufacturing system, a reconfigurable manufacturing system is not fully automated. Therefore, a joint analysis of machine and workforce reconfigurations in a reconfigurable manufacturing system can enhance its adaptability and robustness. Moreover, the flexibility of such environments can benefit from dynamic concepts, such different dynamic task assignment policies which have been discussed in the chapter. Therefore, in this thesis in addition to the comparison between dedicated and multi/mixed model assembly lines, we focus on both design and operational (line balancing) decisions in multi-manned mixed-model assembly lines with walking workers and different task assignment policies.

Several major avenues for future research are identified. Some important ones are covered within the following chapters of this thesis, such as: studying configuration selection between multiple dedicated lines and a single multi-model line, design and balancing a multi-manned mixed-model assembly line with walking workers and different types of task assignment to improve the reconfigurability and flexibility of such environments. Other research directions are highlighted. The first consists in the consideration of ergonomic aspect. The second suggests applying a proper combination of several workforce strategies. The third calls to consider workforce strategies in an emerging human-robot collaborative environment. The fourth consists in studying the influence of the new technologies, such as smart devices, cameras, sensors, teleoperation, message exchange and augmented reality, on a manufacturing system employing both automated resources and human workers.

This chapter results a conference paper [Dolgui et al., 2019] and also two journal publications [Hashemi-Petroodi et al., 2020c,a].

Chapter 2

A single multi-model line versus multiple dedicated lines

2.1 Introduction

This chapter considers two types of manufacturing systems with respect to the ability of processing different products on the same line – Multi-model Manufacturing Line (MML) and Dedicated Manufacturing Lines (DML). DML appeared in the beginning of 20th century and they are still popular. The well-known representatives of MML are Flexible Manufacturing Systems (FMS) and Reconfigurable Manufacturing Systems (RMS). The requirement of flexibility for manufacturing systems appeared in 1950s and it was firstly addressed by FMS, which are able to produce a variety of products by means of Computer Numerical Control (CNC) machines with multiple tools. In such systems, it is possible to change tools and numerical control programs and thus, to produce a variety of products. Nevertheless, FMSs are complex and require costly equipment. RMS appeared few decades later. They are capable to quickly respond to market changes by selecting appropriate modules of the production equipment, adjusting other required resources and process new products in a cost-effective way. RMS capital costs are not as high as the costs of FMS but they are less flexible and dedicated to a smaller variety of products. Both RMS and FMS are designed to produce a set of different products at the same line.

RMS or FMS, and MML in general, may not always be the best choice for production environment from the economical point of view. When product types and their demands are sufficiently well specified in a long-term framework, DML, which are based on a fixed equipment, can be more profitable. Deciding which manufacturing configuration – DML or MML – is economically more preferable is the problem studied in this chapter.

A prerequisite of the applicability of our results is a sufficiently reliable forecast of the set of the product types to be manufactured and their manufacturing and economic characteristics. Demand forecasting techniques are described, for example, by Baecke et al.

[2017], Dombi et al. [2018] and Kück and Freitag [2021]. Chou et al. [2010] and Díaz et al. [2020] propose forecasting methods for the manufacturing cost and selling price, and Greig et al. [2018] and Wang et al. [2018] for the system performance and task times.

Depending on the planning horizon, fixed purchase and implementation costs of the manufacturing systems, their uncertain and time-dependent operating, depreciation and maintenance costs, product demands, selling prices and manufacturing and setup times, one of the two above mentioned manufacturing system options can be more profitable than the other. However, this relation can change for different input data. The decision discussed in this chapter specifies which of the two configurations is economically more preferable for the input data given by the decision makers.

The values of the input parameters can be nominal values, statistically average values, worst-case values, or values provided by the experts. Our results can be used to evaluate possible future scenarios if the manufacturing configuration selection problem is solved by the scenario-based approach, which is elaborated by stochastic programming, see Shapiro et al. [2014]. In the model, detailed configuration design is not considered, nor the impact of input data uncertainty on the best system configurations. Addressing this gap is an interesting topic for future research work.

A more detailed formulation of the studied decision problem is given in Section 2.2. The problem is reduced to two optimization problems Opt1 and Opt2, which are considered in Sections 2.3 and 2.4, respectively. We establish that, while Opt1 is polynomially solvable, Opt2 is NP-hard. Problems Opt1 and Opt2 can be interpreted as capacitated economic lot-sizing problems, knapsack type problems or single and parallel machine family scheduling problems. Optimization problems of these types are intensively studied in the literature, see, for example, Wagner and Whitin [1958], Rosling [1993] and Van Hoesel and Wagelmans [1996] for the lot-sizing problems, Kellerer et al. [2004] and Martello and Toth [1990] for knapsack type problems, Potts and Kovalyov [2000] and Allahverdi et al. [2008] for family scheduling problems. However, the specific problems Opt1 and Opt2 have never been studied before. An integer linear programming formulation, a dynamic programming algorithm exponential in the number of product types, a constructive greedy heuristic, a randomized heuristic and a local search algorithm with steepest ascent hill climbing are presented for the NP-hard problem. Computer experiments in Section 2.5 demonstrate sufficiently good quality of the heuristic and local search solutions. The chapter completes with a short summary of the results and suggestions for future research in in Section 2.6.

2.2 Problem formulation

We Consider a business project which includes manufacturing and selling products of F types over T time periods, which can be days, weeks, decades, months or quarters, and the length of one time period is one time unit. For the project implementation, there are two variants of the manufacturing configuration, denoted as M_1 and M_2. In M_1, there is a dedicated line for each product type f, $f = 1, \ldots, F$, and in M_2, there is a single line for all F product types. Manufacturing and selling are assumed to begin in period 1.

Each product type f is associated with a selling price p_{tf} per product unit and a *demand* of d_{tf} product units in time period t such that the number of manufactured product units in the first t time periods does not exceed the cumulative demand $D_{tf} = \sum_{\tau=1}^{t} d_{\tau f}$, $t = 1, \ldots, T$. If the demand values are known only for long time periods such as quarters or years, then the demand values for smaller time periods can be determined based on the demand function profile of the same or similar business. For example, if the year demand is A and the demand function is constant, then the month demand can be calculated as $\frac{A}{12}$.

Product units are assumed to be sold in the time period of their manufacturing. The time profile of the price values p_{tf} can be any. If late demand satisfaction does not affect the product price and there is an inflation, then $p_{\tau f} < p_{tf}$ can be satisfied for $\tau < t$. If late demand satisfaction implies lower price and there is no inflation, then the opposite inequality $p_{\tau f} > p_{tf}$ can be satisfied for $\tau < t$. The *Group Technology* approach is employed in each time period, according to which at most one non-empty *batch* of each product type f can be manufactured in each time period. Manufacturing any non-empty batch of product type f by configuration M_k, $k \in \{1, 2\}$, requires a *setup* of time s_{kf}, $0 \le s_{kf} < 1$, $f = 1, \ldots, F$, where number 1 represents the length of one time period. Manufacturing configuration M_k is associated with its purchase and implementation cost c_k, operating, depreciation and maintenance cost o_{tk} in time period t, and manufacturing time m_{kf}, $0 < m_{kf} \le 1 - s_{kf}$, of one unit of type f product, $f = 1, \ldots, F$, $k \in \{1, 2\}$. Parameters p_{tf}, d_{tf}, c_k and o_{tk} are non-negative integer numbers, and parameters m_{kf} and s_{kf} are rational numbers, $f = 1, \ldots, F$, $t = 1, \ldots, T$, $k = 1, 2$. Cumulative demands satisfy $0 \le D_{1f} \le \cdots \le D_{Tf}$, $f = 1, \ldots, F$.

Decision variables x_{tf} and y_{tf} are introduced which are the numbers of product units of type f to be manufactured in time period t by configurations M_1 and M_2, respectively, $f = 1, \ldots, F$, $t = 1, \ldots, T$. Denote by x and y matrices with entries x_{tf} and y_{tf} and by sgn(·) function taking values 0 and 1 for zero and positive argument, respectively. Denote by Z_0 the set of non-negative integer numbers. We consider the following two auxiliary optimization problems, denoted as OPT1 and OPT2, which are to maximize the total price of all products manufactured in T time periods, subject to the demand and manufacturing time constraints for configurations M_1 and M_2, respectively.

Problem OPT1:

$$\max_x \sum_{t=1}^{T} \sum_{f=1}^{F} p_{tf} x_{tf}, \text{ subject to}$$

$$\sum_{\tau=1}^{t} x_{\tau f} \leq D_{tf}, \ t = 1, \ldots, T, \ f = 1, \ldots, F,$$

$$s_{1f} \mathrm{sgn}(x_{tf}) + m_{1f} x_{tf} \leq 1, \ t = 1, \ldots, T, \ f = 1, \ldots, F, \tag{2.1}$$

$$x_{tf} \in Z_0, \ t = 1, \ldots, T, \ f = 1, \ldots, F.$$

Problem OPT2:

$$\max_y \sum_{t=1}^{T} \sum_{f=1}^{F} p_{tf} y_{tf}, \text{ subject to}$$

$$\sum_{\tau=1}^{t} y_{\tau f} \leq D_{tf}, \ t = 1, \ldots, T, \ f = 1, \ldots, F,$$

$$\sum_{f=1}^{F} (s_{2f} \mathrm{sgn}(y_{tf}) + m_{2f} y_{tf}) \leq 1, \ t = 1, \ldots, T, \tag{2.2}$$

$$y_{tf} \in Z_0, \ t = 1, \ldots, T, \ f = 1, \ldots, F.$$

The difference between the two problems is in the constraints (2.1) and (2.2). While the constraints (2.1) require that the total setup and manufacturing time fits each time period for each product type manufactured on a dedicated line, the constraints (2.2) do the same for the total setup and manufacturing time of all product types manufactured on a single multi-model line.

Remark 2. *Product units capacity of line f in the manufacturing configuration M_1 and any time period t is equal to $\left\lfloor \frac{1-s_{1f}}{m_{1f}} \right\rfloor$. Product units capacity of the manufacturing configuration M_2 and any time period t is a variable vector $(y_{t1}, \ldots, y_{tF})$ which satisfies (2.2). Decreasing setup and manufacturing times can increase these capacities, however, this should imply increasing the costs o_{tk} and c_k.*

Remark 2. *If product type f is not supposed to be produced in time period t, then* *we can set selling price $p_{tf} := 0$ before solving the problems* Opt1 *and* Opt2*, and re-set $x_{tf} := 0$ and $y_{tf} := 0$ in any feasible solution x or y of these problems. This change of x and y does not affect their feasibility and profitability. The proposed manipulations with the input and output data can be used to address the product portfolio changes over time.*

Denote by x^* and P_1^* optimal solution and its value for the problem Opt1, and denote by y^* and P_2^* optimal solution and its value for the problem Opt2. Observe that the value of $V_k^* := P_k^* - \sum_{t=1}^{T} o_{tk} - c_k$ is the total profit earned if configuration M_k is used during T time periods. The present problem is to determine which of the two relations $V_1^* < V_2^*$ or $V_1^* \geq V_2^*$ is satisfied. We call this problem Select. Obviously, this problem reduces to solving the problems Opt1 and Opt2. Solutions of these problems are described in Sections 2.3 and 2.4. An important note for the analysis of computational complexity of the studied problems is that the input size of any of the problems Select, Opt1 and Opt2 is $O(T \cdot F)$.

For an illustration, consider a small example of the problem Select with the following input data: $F = 3$, $T = 4$, selling prices are $p_{tf} = f(1 - 0.01t)10^3$, demand values are $d_{tf} = 12 - 2t$, $t = 1, \ldots, T$, $f = 1, \ldots, F$, setup times are $s_{1f} = \frac{2f}{10}$, $s_{2f} = \frac{f}{10}$, $f = 1, \ldots, F$, manufacturing times are $m_{1f} = \frac{1}{20}$ and $m_{2f} = \frac{1}{40}$. Optimal solutions of the problems Opt1 and Opt2 for these data are demonstrated in Figure 2.1 [Dolgui et al., 2021].

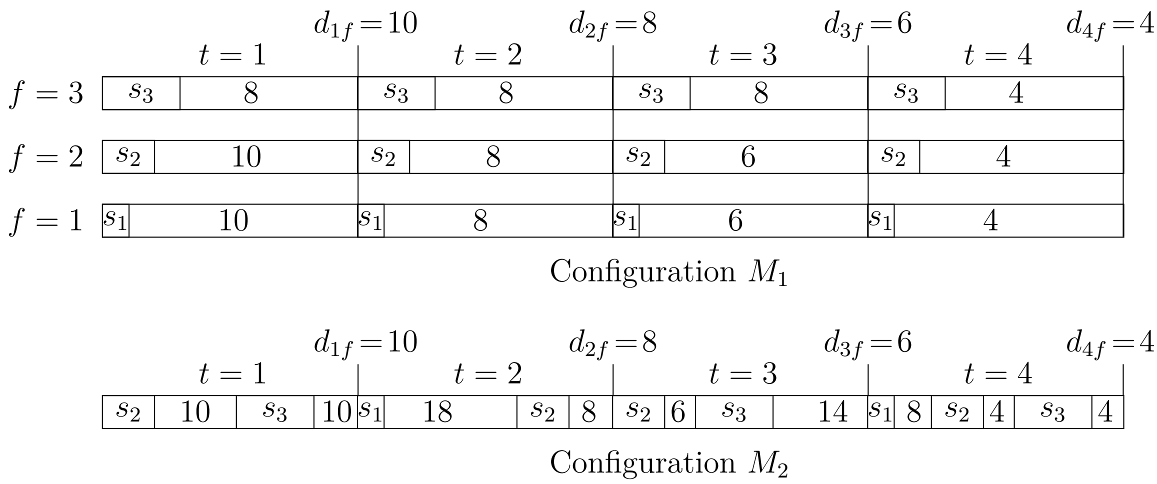

Figure 2.1 – Solution of the example problem. $P_1^* = 164280$, $P_2^* = 162080$.

There, the number of units of type f product immediately follows corresponding setup time s_f. For this example, the total prices $P_1^* = 164280$ and $P_2^* = 162080$ of the products to be manufactured are comparable for configurations M_1 and M_2. Therefore, the optimal selection decision mainly depends on the costs of the two configurations. Note that all the demands are satisfied in full in configuration M_1, while there is an unsatisfied demand of two units for the type 1 product in configuration M_2.

2.3 Solution of the problem Opt1

Due to the facts that 1) all the input parameters are positive numbers, 2) variables x_{tf} with different indices f are not involved in any same constraint and 3) they participate in the objective function to be maximized with positive coefficients, the problem Opt1 decomposes into F sub-problems differing from Opt1 in that the product type index f is fixed, $f = 1, \dots, F$. Therefore, it reduces to solving the following generic problem, denoted as Opt1(f). For given product type $f \in \{1, \dots, F\}$, denote $q_t = p_{tf}$, $a = s_{1f}$, $b = m_{1f}$, $G_t = D_{tf}$ and introduce variables $z_t = x_{tf}$, $t = 1, \dots, T$.

Problem Opt1(f):

$$\max_x \sum_{t=1}^{T} q_t z_t, \text{ subject to}$$

$$\sum_{\tau=1}^{t} z_\tau \le G_t, \ t = 1, \dots, T,$$

$$a \cdot \operatorname{sgn}(z_t) + b \cdot z_t \le 1, \ t = 1, \dots, T,$$

$$z_t \in Z_0, \ t = 1, \dots, T.$$

Denote $u = \lfloor \frac{1-a}{b} \rfloor$. Remark that $u \ge 0$. Furthermore, relations $a \cdot \operatorname{sgn}(z_t) + b \cdot z_t \le 1$, $t = 1, \dots, T$, are satisfied if and only if $z_t \le u$, $t = 1, \dots, T$. Therefore, the problem Opt1(f) can be re-formulated as follows.

Problem Opt1(f):

$$\max_x \sum_{t=1}^{T} q_t z_t, \text{ subject to}$$

$$\sum_{\tau=1}^{t} z_\tau \le G_t, \ t = 1, \dots, T,$$

$$z_t \in \{0, 1, \dots, u\}, \ t = 1, \dots, T.$$

Denote by $z^{(f)} = (z_1^{(f)}, z_2^{(f)}, \dots, z_n^{(f)})$ optimal solution of the problem Opt1(f). If $t \cdot u \le G_t$, $t = 1, \dots, T$, then $z^{(f)} = (u, \dots, u)$. Therefore, assume that $t \cdot u > G_t$ for at least one $t \in \{1, \dots, T\}$.

2.3.1 Algorithm S

Problem OPT1(f) is a maximization counterpart of a special case of the single machine scheduling problem, for which Janiak and Kovalyov [1996] developed an optimal $O(T \log T)$ time algorithm. This algorithm cannot be directly used to solve the problem OPT1(f). Let us show that OPT1(f) can be solved in $O(T \log T)$ time by the following algorithm **S** (see Algorithm 1), which is similar to the Modified Algorithm M in Janiak and Kovalyov [1996], but not the same.

Algorithm 1: Algorithm S

Step 1 Initiate time period counter as $t = 1$, total number of product units ($\sum z_\tau$) as $A = 0$, and perform Step 2.

Step 2 Set $z_t = \min\{u, G_t\}$ and $A := A + z_t$. If $A \leq G_t$, then go to Step 4, else perform Step 3.

Step 3 Select $h \in \{1, \ldots, t\}$ with $z_h > 0$ such that q_h is the smallest. If $z_h - (A - G_t) \geq 0$, then re-set $z_h := z_h - (A - G_t)$, $A := G_t$, and perform Step 4, else re-set $A := A - z_h$, $z_h := 0$ and repeat Step 3.

Step 4 If $t \leq T - 1$, then re-set $t := t + 1$ and perform Step 2, else output optimal solution z and stop.

Theorem 2.3.1. *Algorithm* **S** *is an optimal algorithm for the problem* OPT1*(f).*

Proof of Theorem 2.3.1. The proof proceeds in three stages.

Stage 1. Let r be an index such that $q_r = \max\{q_t \mid t = 1, \ldots, T\}$. We prove that there exists an optimal solution, in which $z_r^* = \min\{u, G_r\}$. Consider an optimal solution z^* and assume that $z_r^* < \min\{u, G_r\}$. If $z_t^* = 0$ for all $t \neq r$, then it is obvious that $z_r^* = \min\{u, G_r\}$. Therefore, it is sufficient to consider two sets $J_1 = \{t \mid 1 \leq t < r, z_t^* > 0\}$ and $J_2 = \{t \mid r < t \leq T, z_t^* > 0\}$ such that $J_1 \cup J_2 \neq \varnothing$, and investigate two cases: $J_2 \neq \varnothing$ and $J_2 = \varnothing$. In the case $J_2 \neq \varnothing$, modify z^* by re-setting $z_t^* := z_t^* - \delta_t$, $t \in J_2$, and $z_r^* := z_r^* + \sum_{t \in J_2} \delta_t$, where $\delta_t \geq 0$ are such that the new values of z_t^*, $t \in J_2$, are non-negative, the new value of z_r^* is maximized and it does not exceed $\min\{u, G_r\}$. Obviously, values δ_t exist, and the new solution z^* is optimal. If $z_r^* = \min\{u, G_r\}$ in the new solution, then we are done. If $z_r^* < \min\{u, G_r\}$, then $z_t^* = 0$, $t \in J_2$, in the new solution, and the case $J_2 \neq \varnothing$ reduces to the case $J_2 = \varnothing$.

In the case $J_1 \neq \varnothing$, $J_2 = \varnothing$, modify z^* by re-setting $z_t^* := z_t^* - \delta_t$, $t \in J_1$, and $z_r^* := z_r^* + \sum_{t \in J_1} \delta_t$, where $\delta_t \geq 0$ are such that the new values of z_t^*, $t \in J_1$, are non-negative, the new value of z_r^* is maximized and it does not exceed $\min\{u, G_r\}$. Obviously, values δ_t exist, and the new solution z^* is optimal. If $z_r^* = \min\{u, G_r\}$ in the new solution, then we are done. If $z_r^* < \min\{u, G_r\}$, then $z_t^* = 0$, $t \in J_1$, in the new solution, and, since $z_t^* = 0$ for $t \in J_2$, then $\sum_{t=1}^{T} z_t^* = z_r^*$. Therefore, $z_r^* = \min\{u, G_r\} \leq G_r$ must be satisfied.

Stage 2. Consider a *reduced problem* in which the part type r is removed, and cumulative demands G_t are re-set such that $G_t := G_t$ for $t = 1, \ldots, r-1$, and $G_t := G_t - \min\{u, G_r\}$ for $t = r+1, \ldots, T$. Obviously, any optimal solution of the reduced problem can be transformed into an optimal solution of the original problem $\text{OPT1}(f)$ by setting $z_r^* = \min\{u, G_r\}$.

Let us show that algorithm **S** outputs $z_r = \min\{u, G_r\}$. In iteration r of Step 2, $z_r = \min\{u, G_r\}$ is set. In Step 3, if value $z_h > 0$ is decreased, then it is selected such that q_h is the smallest. Therefore, in any iteration $t \geq r$ of Step 3, as soon as there is $z_\tau > 0$, $\tau \leq t$, $\tau \neq r$, the value $z_r = \min\{u, G_r\}$ does not change. If $z_\tau = 0$, $\tau = 1, \ldots, T$, $\tau \neq r$, in iteration T of Step 3, then $A = z_r = \min\{u, G_r\} \leq G_T$ and algorithm **S** outputs solution with $x_r = \min\{u, G_r\}$.

Note that algorithm **S** generates the same values $z_1, \ldots, z_{r-1}$ and A for both the original and the reduced problems up to the iteration r and its further calculations depend only on $z_1, \ldots, z_{r-1}$ and $G_t - A$, $t = r, r+1, \ldots, T$, which are the same in both cases for $t = r+1, \ldots, T$. Hence, algorithm **S** generates the same values $z_1, \ldots, z_{r-1}, z_{r+1}, \ldots, z_T$ for both the original and the reduced problems.

Stage 3. The theorem is proved by induction. Algorithm **S** is optimal for $T = 1$. Assume that it is optimal for $T - 1$ time periods. Therefore, it generates an optimal solution $(z_1, \ldots, z_{r-1}, z_{r+1}, \ldots, z_T)$ for the reduced problem. We proved in Stages 1 and 2 that $(z_1, \ldots, z_{r-1}, \min\{u, G_r\}, z_{r+1}, \ldots, z_T)$ is an optimal solution of the original problem. In Stage 2 we proved that algorithm **S** sets $z_r = \min\{u, G_r\}$ and generates the same values $z_1, \ldots, z_{r-1}, z_{r+1}, \ldots, z_T$ for both the original and the reduced problems, that is, it generates an optimal solution for the problem with T time periods, as it is required. □

Theorem 2.3.2. *Algorithm* **S** *can be implemented to run in $O(T \log T)$ time.*

Proof of Theorem 2.3.2. Steps 1, 2 and 4 obviously require $O(T)$ time. Regarding Step 3, values q_t such that $z_t > 0$ can be stored in a heap, see Cormen et al. [2001]. Inserting a new value q_t such that $z_t > 0$ to the heap takes $O(\log T)$ time. Finding the smallest element q_h in the heap takes $O(1)$ time. Calculation of the new value z_h in Step 3 takes $O(1)$ time. If the new value z_h is zero, then removal of q_h from the heap takes $O(\log T)$ time. The removed value is never inserted back. The number of iterations of Step 3 does not exceed $2T$ because in each iteration either t is increased by one or one value is removed from the heap. Therefore, Step 3 requires $O(T \log T)$ time.

An optimal solution of the problem Opt1 can be obtained by setting $x^*_{tf} = z^{(f)}_t$, $t = 1, \ldots, T$, $f = 1, \ldots, F$. Therefore, Opt1 can be solved in $O(F \cdot T \log T)$ time. □

2.4 Solution of the problem Opt2

We start this section with the NP-hardness proof.

Theorem 2.4.1. *Problem* Opt2 *is NP-hard even if* $T = 1$, $D_{tf} = 1$, *and* $s_{2f} = 0$, $t = 1, \ldots, T$, $f = 1, \ldots, F$.

Proof of Theorem 2.4.1. Special case of Opt2 in the formulation of this theorem can be represented as the following problem.

$$\max_{\alpha} \sum_{f=1}^{F} a_f \alpha_f, \text{ subject to}$$

$$\alpha_f \leq 1, \ f = 1, \ldots, F,$$

$$\sum_{f=1}^{F} e_f \alpha_f \leq 1,$$

$$\alpha_f \in Z_0, \ f = 1, \ldots, F,$$

where $a_f = p_{1f}$ and $e_f = m_{2f}$, $f = 1, \ldots, F$. Assume that $e_f = \frac{b_f}{E}$, where b_f, $f = 1, \ldots, F$, and E are positive integer numbers. Then the considered special case can be represented as the following problem.

$$\max_{\alpha} \sum_{f=1}^{F} a_f \alpha_f, \text{ subject to}$$

$$\sum_{f=1}^{F} b_f \alpha_f \leq E,$$

$$\alpha_f \in \{0, 1\}, \ f = 1, \ldots, F.$$

This problem is the well-known 0-1 knapsack problem, which is NP-hard [Karp, 1972]. □

The question whether the problem Opt2 is NP-hard in the strong sense remains open. Unlike Opt1, the problem Opt2 cannot be decomposed into F sub-problems due to the constraints (2.2): $\sum_{f=1}^{F}(s_{2f}\mathrm{sgn}(y_{tf})+m_{2f}y_{tf}) \le 1$, $t=1,\ldots,T$. Observe that these relations and the integrality of y_{tf} imply $y_{tf} \le u_f := \lfloor \frac{1-s_{2f}}{m_{2f}} \rfloor$, $f=1,\ldots,F$. Therefore, the problem Opt2 reduces to the following problem, for which we keep the same notation.

Problem Opt2:

$$\max_{y} \sum_{t=1}^{T}\sum_{f=1}^{F} p_{tf}y_{tf}, \text{ subject to}$$

$$\sum_{\tau=1}^{t} y_{\tau f} \le D_{tf},\ f=1,\ldots,F,\ t=1,\ldots,n,$$

$$\sum_{f=1}^{F}(s_{2f}\mathrm{sgn}(y_{tf})+m_{2f}y_{tf}) \le 1,\ t=1,\ldots,T,$$

$$y_{tf} \in \{0,1,\ldots,u_f\},\ f=1,\ldots,F,\ t=1,\ldots,T.$$

An Integer Linear Programming (ILP) formulation of the problem Opt2 can be given as follows. Introduce 0-1 variables γ_{tf} such that $\gamma_{tf}=\mathrm{sgn}(y_{tf})$ for all t and f.

ILP for Opt2:

$$\max_{y,\gamma} \sum_{t=1}^{T}\sum_{f=1}^{F} p_{tf}y_{tf}, \text{ subject to}$$

$$\sum_{\tau=1}^{t} y_{\tau f} \le D_{tf},\ f=1,\ldots,F,\ t=1,\ldots,n,$$

$$\sum_{f=1}^{F}(s_{2f}\gamma_{tf}+m_{2f}y_{tf}) \le 1,\ t=1,\ldots,T,$$

$$y_{tf} \le u_f\gamma_{tf},\ f=1,\ldots,F,\ t=1,\ldots,T,$$

$$\gamma_{tf} \le 1,\ f=1,\ldots,F,\ t=1,\ldots,T,$$

$$y_{tf} \in Z_0, \gamma_{tf} \in Z_0,\ f=1,\ldots,F,\ t=1,\ldots,T.$$

2.4.1 Dynamic programming

If $F=1$, then the problem Opt2 is the same as the problem Opt1(1), and therefore, it can be solved in $O(T\log T)$ time. We now show that the problem Opt2 is solvable in $O(\sum_{t=1}^{T}\prod_{f=1}^{F} D_{tf}\cdot u_f)$ time by a dynamic programming algorithm. Therefore, it is pseudo-polynomially solvable if the number product types F is a constant. The question whether it is polynomially solvable in this case is open.

In iteration t of our dynamic programming algorithm, denoted as **DP**, partial solutions are considered in which variables $y_{\tau f}$ are determined for $\tau = 1, \ldots, t$ and they are not determined for $\tau = t+1, \ldots, T$, $f = 1, \ldots, F$. Each such a partial solution in iteration t is associated with F-dimensional vector of *state variables (state)* $B = (B_1, \ldots, B_F)$, where $B_f = \sum_{\tau=1}^{t} y_{\tau f}$, $f = 1, \ldots, F$. Partial solution in the state B, for which the total price $\sum_{\tau=1}^{t} \sum_{f=1}^{F} p_{\tau f} y_{\tau f}$ is maximized, *dominates* all other partial solutions in the same state in the sense that if a partial solution in this state can be extended to an optimal solution of the problem OPT2, then the dominant solution can be extended in the same way to an optimal solution as well. Denote by $P_t(B)$ maximum total price of partial solutions in the state B of iteration t.

In addition to Algorithm **DP** (see Algorithm 2), a *greedy* algorithm, a *randomized* heuristic and a *local search* algorithm with *steepest ascent hill climbing* for the problem OPT2, which are denoted as algorithms **Greedy**, **Random** and **LocSearch**, respectively are developed. See, for example, Russell and Norvig [2002] and Edelkamp and Schrödl [2011] for the general descriptions of these methods.

Algorithm 2: Dynamic programming (DP)

Step 1 (Initialization) Set $P_0(0, \ldots, 0) = 0$, $P_0(B) = -\infty$ for $B \neq (0, \ldots, 0)$ and perform Step 2.

Step 2 (Recursion) For $t = 1, \ldots, T$ and $B_f \in 0, 1, \ldots, D_{tf}$, $f = 1, \ldots, F$, calculate

$$P_t(B_1, \ldots, B_F) = \max \Big\{ P_{t-1}(B_1 - y_{t1}, \ldots, B_F - y_{tF}) + \sum_{f=1}^{F} p_{tf} y_{tf} \ \mid \ y_{tf} \in \{0, 1, \ldots, u_f\},$$

$$f = 1, \ldots, F, \ \sum_{f=1}^{F} (s_{2f} \operatorname{sgn}(y_{tf}) + m_{2f} y_{tf}) \leq 1 \Big\}.$$

If $t = T$, then go to Step 3. Otherwise, re-set $t := t + 1$ and repeat Step 2.

Step 3 (Optimal solution) Calculate optimal solution value:

$$P^* = \max\{P_T(B_1, \ldots, B_F) \mid B_f = 0, 1, \ldots, D_{Tf}, f = 1, \ldots, F\},$$

and determine the corresponding optimal solution by backtracking.

The running time of the algorithm **DP** is determined by its Step 2. The number of states considered in iteration t of this step is $O(\prod_{f=1}^{F} D_{tf})$. For each state, the right-hand side of the recursion can be calculated in $O(\prod_{f=1}^{F} u_f)$ time. Therefore, the running time of the algorithm **DP** can be evaluated as $O(\sum_{t=1}^{T} \prod_{f=1}^{F} D_{tf} \cdot u_f)$, which is pseudo-polynomial if F is a constant.

2.4.2 Greedy and randomized greedy algorithms

We now present a *greedy* algorithm (Algorithm 3), a *randomized* heuristic as follows:

Algorithm 3: Greedy algorithm

Step 1 (Initialization) Calculate *priorities* of the product types as

$$r_f = \frac{\sum_{t=1}^{T} p_{tf} u_f}{s_{2f} + m_{2f} u_f}, \ f = 1, \dots, F.$$

Re-number product types such that $r_1 \geq \cdots \geq r_F$. Initiate time period counter $t = 0$. Denote the total number of product units of type f manufactured up to the current time period ($\sum_{\tau=1}^{t} y_{\tau f}$) as Q_f and initiate it as $Q_f = 0$, $f = 1, \dots, F$. Perform Step 2.

Step 2 (Construction of a "greedy" solution) If $t = T$, then output "greedy" solution y_{tf}, $f = 1, \dots, F$, $t = 1, \dots, T$, and stop. Otherwise, re-set $t := t + 1$. Denote total current setup and manufacturing time in period t as R and initiate it as $R = 0$. Initiate product type $f = 1$ and perform Step 3.

Step 3 (Determination of y_{tf}, $f = 1, \dots, F$) Determine maximum value $y_{tf} \in Z_0$ such that $Q_f + y_{tf} \leq D_{tf}$ and $R + s_{2f}\mathrm{sgn}(y_{tf}) + m_{2f} y_{tf} \leq 1$, that is, $y_{tf} = \max\left\{0, \min\left\{D_{tf} - Q_f, \left\lfloor \frac{1-(R+s_{2f})}{m_{2f}} \right\rfloor\right\}\right\}$. Re-set $Q_f := Q_f + y_{tf}$. If $f = F$, then perform Step 2, else re-set $R := R + s_{2f}\mathrm{sgn}(y_{tf}) + m_{2f} y_{tf}$, $f := f + 1$, and repeat Step 3.

The running time of the algorithm **Greedy** is $O(T \cdot F + F \log F)$. Algorithm **Random** differs from the algorithm **Greedy** only in that instead of the priority list such that $r_1 \geq \cdots \geq r_F$, a randomly generated list of product types is used. This algorithm is run as many times as the solution time limit permits, and the best solution is taken as the output.

2.4.3 Local search algorithm

The proposed LocSearch algorithm is explained in Algorithm 4.

Note that Steps 1-3 of the algorithm **LocSearch** can be repeated as many times as the objective function value is increased in Step 3. The algorithm can be stopped, and the best solution found can be put out if the solution time limit is exceeded. Furthermore,

Algorithm 4: Local search algorithm

Step 1 Initiate two pairs of indices (r^0, q^0) and (h^0, g^0) such that $(r^0, q^0) = (h^0, g^0) = (0, 0)$. Take a feasible solution y^0 of the problem OPT2 with value P^0 as an input. Solution delivered by algorithm **Greedy** or algorithm **Random** can be taken in the beginning. For every two ordered pairs of indices (r, q) and (h, g) such that $1 \le r \le h \le T$, $1 \le q \le F$, $1 \le g \le F$, $(r, q) \ne (h, g)$, $(r, q) \ne (r^0, q^0)$, $(h, g) \ne (h^0, g^0)$, perform Step 2.

Step 2 Solve problem OPT2, in which all but two variables are fixed such that $y_{tf} = y^0_{tf}$ for $(t, f) \notin \{(r, q), (h, g)\}$, $f = 1, \dots, F$, $t = 1, \dots, T$. Denote this problem, its optimal solution and its optimal objective function value as OPT2$((r, q), (h, g))$, $y^{(r,q),(h,g)}$ and $P^{(r,q),(h,g)}$, respectively.

Step 3 Determine two pairs (r^0, q^0) and (h^0, g^0) such that

$$P^{(r^0,q^0),(h^0,g^0)} = \max_{q=1,\dots,g,\ g=1,\dots,F,\ r,h=1,\dots,T,\ (r,q)\ne(h,g)} \{P^{(r,q),(h,g)}\}.$$

If $P^0 < P^{(r^0,q^0),(h^0,g^0)}$, then re-set $y^0 := y^{(r^0,q^0),(h^0,g^0)}$ and repeat Step 1, else output y^0 and stop.

observe that the problem OPT2$(t, (r, q), (h, g))$ in Step 2 of **LocSearch** can be formulated and solved as follows. We distinguish two cases: 1) $q \ne g$, $1 \le r \le h \le T$, and 2) $q = g$, $1 \le r < h \le T$.

Problem OPT2$((r, q), (h, g))$ for the case 1) $q \ne g$:

$$\max_{y_{rq}, y_{hg}} \{p_{rq} y_{rq} + p_{hg} y_{hg}\} + \sum_{(t,f)\notin\{(r,q),(h,g)\}} y_{tf} y^0_{tf}, \text{ subject to}$$

$$\sum_{\tau=1}^{r-1} y^0_{\tau q} + y_{rq} + \sum_{\tau=r+1}^{t} y^0_{\tau q} \le D_{tq},\ t = r, r+1, \dots, T,$$

$$\sum_{\tau=1}^{h-1} y^0_{\tau g} + y_{hg} + \sum_{\tau=h+1}^{t} y^0_{\tau g} \le D_{tg},\ t = h, h+1, \dots, T,$$

$$\sum_{f=1}^{q-1} \Big(s_{2f} \mathrm{sgn}(y^0_{rf}) + m_{2f} y^0_{rf}\Big) + \Big(s_{2q} \mathrm{sgn}(y_{rq}) + m_{2q} y_{rq}\Big) + \sum_{f=q+1}^{F} \Big(s_{2f} \mathrm{sgn}(y^0_{rf}) + m_{2f} y^0_{rf}\Big) \le 1,$$

$$\sum_{f=1}^{g-1} \Big(s_{2f} \mathrm{sgn}(y^0_{hf}) + m_{2f} y^0_{hf}\Big) + \Big(s_{2g} \mathrm{sgn}(y_{hg}) + m_{2g} y_{hg}\Big) + \sum_{f=g+1}^{F} \Big(s_{2f} \mathrm{sgn}(y^0_{hf}) + m_{2f} y^0_{hf}\Big) \le 1,$$

$$y_{rq}, y_{hg} \in Z_0.$$

Optimal solution of this problem is:

$$y_{rq}^{(r,q),(h,g)} = \max\Big\{0,$$

$$\min\Big\{\Big\lfloor \frac{1-\sum_{f=1}^{q-1}\Big(s_{2f}\mathrm{sgn}(y_{rf}^0)+m_{2f}y_{rf}^0\Big)-\sum_{f=q+1}^{F}\Big(s_{2f}\mathrm{sgn}(y_{rf}^0)+m_{2f}y_{rf}^0\Big)-s_{2q}}{m_{2q}}\Big\rfloor,$$

$$\min_{r\le t\le T}\Big\{D_{tq}-\sum_{\tau=1}^{r-1}y_{\tau q}^0-\sum_{\tau=r+1}^{t}y_{\tau q}^0\Big\}\Big\}\Big\},$$

$$y_{hg}^{(r,q),(h,g)} = \max\Big\{0,$$

$$\min\Big\{\Big\lfloor \frac{1-\sum_{f=1}^{g-1}\Big(s_{2f}\mathrm{sgn}(y_{hf}^0)+m_{2f}y_{hf}^0\Big)-\sum_{f=g+1}^{F}\Big(s_{2f}\mathrm{sgn}(y_{hf}^0)+m_{2f}y_{hf}^0\Big)-s_{2g}}{m_{2g}}\Big\rfloor,$$

$$\min_{h\le t\le T}\Big\{D_{tg}-\sum_{\tau=1}^{h-1}y_{\tau g}^0-\sum_{\tau=h+1}^{t}y_{\tau g}^0\Big\}\Big\}\Big\}.$$

Problem $\text{OPT2}((r,q),(h,g))$ **for the case 2)** $q=g$, $r<h$:

$$\max_{y_{rq},y_{hg}}\{p_{rq}y_{rq}+p_{hg}y_{hg}\}+\sum_{(t,f)\notin\{(r,q),(h,g)\}}p_{tf}y_{tf}^0,\ \text{subject to}$$

$$\sum_{\tau=1}^{r-1}y_{\tau q}^0+y_{rq}+\sum_{\tau=r+1}^{t}y_{\tau q}^0\le D_{tq},\ t=r,r+1,\dots,h-1,$$

$$\sum_{\tau=1}^{r-1}y_{\tau q}^0+y_{rq}+\sum_{\tau=r+1}^{h-1}y_{\tau q}^0+y_{hq}+\sum_{\tau=h+1}^{t}y_{\tau q}^0\le D_{tq},\ t=h,h+1,\dots,T,$$

$$\sum_{f=1}^{q-1}\Big(s_{2f}\mathrm{sgn}(y_{rf}^0)+m_{2f}y_{rf}^0\Big)+\Big(s_{2q}\mathrm{sgn}(y_{rq})+m_{2q}y_{rq}\Big)+\sum_{f=q+1}^{F}\Big(s_{2f}\mathrm{sgn}(y_{rf}^0)+m_{2f}y_{rf}^0\Big)\le 1,$$

$$\sum_{f=1}^{q-1}\Big(s_{2f}\mathrm{sgn}(y_{hf}^0)+m_{2f}y_{hf}^0\Big)+\Big(s_{2q}\mathrm{sgn}(y_{hq})+m_{2q}y_{hq}\Big)+\sum_{f=q+1}^{F}\Big(s_{2f}\mathrm{sgn}(y_{hf}^0)+m_{2f}y_{hf}^0\Big)\le 1,$$

$$y_{rq},y_{hq}\in Z_0.$$

Optimal solution of this problem can be determined as follows. If $p_{rq}\ge p_{hq}$ (recall that $q=g$), then, since the function to be maximized is linear, $y_{hq}^{(r,q),(h,q)}=0$ and the optimal value of the other variable is:

$$y_{rq}^{(r,q),(h,q)} = \max\Big\{0,$$

$$\min\Big\{\Big\lfloor \frac{1 - \sum_{f=1}^{q-1}\Big(s_{2f}\mathrm{sgn}(y_{rf}^0) + m_{2f}y_{rf}^0\Big) - \sum_{f=q+1}^{F}\Big(s_{2f}\mathrm{sgn}(y_{rf}^0) + m_{2f}y_{rf}^0\Big) - s_{2q}}{m_{2q}}\Big\rfloor,$$

$$\min_{r\le t\le h-1}\Big\{D_{tq} - \sum_{\tau=1}^{r-1} y_{\tau q}^0 - \sum_{\tau=r+1}^{t} y_{\tau q}^0\Big\}\Big\}\Big\}.$$

Else, if $p_{rq} < p_{hq}$, then, by the same reason, $y_{rq}^{(r,q),(h,q)} = 0$ and the optimal value of the other variable is:

$$y_{hq}^{(r,q),(h,q)} = \max\Big\{0,$$

$$\min\Big\{\Big\lfloor \frac{1 - \sum_{f=1}^{q-1}\Big(s_{2f}\mathrm{sgn}(y_{hf}^0) + m_{2f}y_{hf}^0\Big) - \sum_{f=q+1}^{F}\Big(s_{2f}\mathrm{sgn}(y_{hf}^0) + m_{2f}y_{hf}^0\Big) - s_{2q}}{m_{2q}}\Big\rfloor,$$

$$\min_{h\le t\le T}\Big\{D_{tq} - \sum_{\tau=1}^{r-1} y_{\tau q}^0 - \sum_{\tau=r+1}^{h-1} y_{\tau q}^0 - \sum_{\tau=h+1}^{t} y_{\tau q}^0\Big\}\Big\}\Big\}.$$

In either case 1) or case 2), the optimal solution value is:

$$P^{(r,q),(h,g)} = \sum_{(t,f)\notin\{(r,q),(h,g)\}} p_{tf}y_{tf}^0 + p_{rq}y_{rq}^{(r,q),(h,g)} + p_{hg}y_{hg}^{(r,q),(h,g)}.$$

By the above formulas, the problem $\textsc{Opt2}((r,q),(h,g))$ can be solved in $O(T+F)$ time.

We suggest to employ a *multi-start* version of the algorithm **LocSearch** such that it runs several times starting with different original solutions y^0. These solutions can be a solution delivered by the algorithm **Greedy** and solutions generated by the algorithm **Random**.

2.5 Computational experiments

This section explains how we generate a set of instances for the proposed problem SELECT. It also evaluates the performance of proposed optimization approaches, and provides managerial insights regarding the impact of effective parameters involving in the problem.

2.5.1 Instances generation

Computational experiments over randomly generated instances of the problem Select were performed. In the experiments, 27 series of instances are constructed. Each series is specified by the pair (T, F). We consider $T \in \{12, 52, 365\}$, where 12, 52 and 365 are the numbers of months, weeks and days in a year, and $F \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. For each series, 20 instances of the problem Select are built. For each instance of the same series, the following input data are generated.

— Purchase and implementation cost c_1 of manufacturing configuration M_1 (DML) is randomly selected as an integer number from the interval $[10^5, 10^6]$, and the same cost c_2 of M_2 (MML) is randomly selected as an integer number from the interval $[\lfloor 1.1c_1 \rfloor, \lfloor 1.2c_1 \rfloor]$. The cost c_2 of MML is on average 15% higher than the cost c_1 of DML.

— Operating, depreciation and maintenance cost o_{t1} of M_1 in time period t is randomly selected as an integer number from the interval $[\lfloor \frac{10^4}{T} \rfloor, \lfloor \frac{2 \cdot 10^4}{T} \rfloor]$, and the same cost o_{t2} of M_2 is randomly selected as an integer number from the interval $[\lfloor 1.1o_{t1} \rfloor, \lfloor 1.2o_{t1} \rfloor]$, $t = 1, \ldots, T$. These costs are higher for larger time units (week, month), and the cost o_{t2} is on average 15% higher than o_{t1}.

— Selling price $p_{tf} = \lceil v_f k_t \rceil$, where product type dependent parameter v_f is randomly selected as an integer number from the interval $[10^3, 10^4]$, and time period dependent parameter k_t is calculated as $k_t = (1.001)^t$ to account for the inflation, $f = 1, \ldots, F$, $t = 1, \ldots, T$.

— Demand d_{tf} is randomly selected as an integer number from the interval $[10, 100]$, $f = 1, \ldots, F$, $t = 1, \ldots, T$. Calculate the average demand for the product of any type in any time period: $A = \frac{\sum_{f=1}^{F} \sum_{t=1}^{T} d_{tf}}{TF}$.

— Setup time s_{1f} is randomly selected as a rational number from the interval $[\frac{1}{5T}, \frac{1}{T}]$, and setup time s_{2f} is randomly selected as a rational number from the interval $[1.1s_{1f}, 1.3s_{1f}]$, $f = 1, \ldots, F$. For both configurations, the setup time is longer for larger time units (week, month), and the setup time s_{2f} is on average 20% longer than s_{1f}, $f = 1, \ldots, F$.

— Manufacturing time m_{1f} is randomly generated as a rational number from the interval $[\frac{1-s_{1f}}{0.7A}, \frac{1-s_{1f}}{1.3A}]$, and manufacturing time m_{2f} is randomly generated as a rational number from the interval $[\frac{1-\sum_{f=1}^{F} s_{2f}}{0.7AF}, \frac{1-\sum_{f=1}^{F} s_{2f}}{1.3AF}]$, $f = 1, \ldots, F$. By this generation, the average capacity of any line in configuration M_1 is approximately equal to the average demand A, the average total capacity of this configuration is approximately equal to AF, and the average capacity of configuration M_2 is approximately equal to the same value AF.

2.5.2 Computational results

The problem Opt2 was solved by docplex Python API software combined with IBM ILOG CPLEX Optimization Studio and algorithms **Greedy**, **Random** and **LocSearch**, which were programmed in Python. The running time of CPLEX was limited by 10 minutes for each instance. Not every instance was solved by CPLEX to optimality. The running time of each of the algorithms **Random** and **LocSearch** was limited by the running time of CPLEX for the same instance. The experiments were run on a PC with Intel Pentium Core i7 Duo 1.9 - 2.11 GHz processor and 32 GB of RAM under MS Windows 10 Pro (64 bit). Table 2.1 contains the following information for each series (T, F).

— Average T^{ave}_{CPLEX} and worst (maximum) T^{max}_{CPLEX} running times of $CPLEX$ in seconds for Opt2 over 20 instances. The running time of **Greedy** was always less than 1 second.

— Average Δ^{ave}_{A} and worst (maximum) Δ^{max}_{A} relative deviation of the objective function value delivered by each algorithm $A \in \{Greedy, Random, LocSearch\}$ from the value of Opt2, delivered by CPLEX, over 20 instances, in percent. Algorithm **LocSearch** was run 30 times for each instance, starting with a solution delivered by the algorithm **Greedy** and 29 best solutions generated by algorithm **Random**.

For some instances our algorithms found better solutions than CPLEX in 10 minutes. Due to the definition of the multi-start algorithm **LocSearch**, this algorithm produces the best results among all the proposed algorithms.

The experimental results demonstrate that the average and worst solution qualities of the best heuristic solution with respect to the CPLEX solution do not exceed 2.24% and 8.93%, respectively. The quality of solutions of **Random** and **LocSearch** differ by at most 0.02%. The average and worst solution quality of **Greedy** can be as bad as 8.17% and 20.56%, which is still acceptable for practical purposes taking into account the input data uncertainty. All the proposed algorithms can be useful when many input data scenarios are required to be verified in order to make a choice between configurations M_1 (DML) and M_2 (MML). In this case, numerous applications of a MILP solver can take too much time.

Table 2.1 – Running time (seconds) and solution quality.

(T, F)	(12,2)	(12,3)	(12,4)	(12,5)	(12,6)	(12,7)	(12,8)	(12,9)	(12,10)
T^{ave}_{CPLEX}	1	1	9	8	2	2	2	1	1
T^{max}_{CPLEX}	2	2	101	118	8	4	4	4	2
Δ^{ave}_{Greedy}	1.94	1.82	2.28	2.28	1.95	2.12	1.35	1.86	1.65
Δ^{ave}_{Random}	1.91	1.75	2.25	2.14	1.69	2.05	1.22	1.58	1.36
$\Delta^{ave}_{LocSearch}$	1.89	1.74	2.24	2.13	1.68	2.04	1.21	1.57	1.34
Δ^{max}_{Greedy}	5.36	9.66	7.46	8.88	6.65	4.79	3.44	3.66	5.11
Δ^{max}_{Random}	5.36	8.93	7.46	6.99	6.65	4.75	3.43	3.29	4.13
$\Delta^{max}_{LocSearch}$	5.36	8.93	7.46	6.98	6.65	4.75	3.42	3.29	4.13
(T, F)	**(52,2)**	**(52,3)**	**(52,4)**	**(52,5)**	**(52,6)**	**(52,7)**	**(52,8)**	**(52,9)**	**(52,10)**
T^{ave}_{CPLEX}	155	572	587	600	600	600	600	600	600
T^{max}_{CPLEX}	600	600	600	600	600	600	600	600	600
Δ^{ave}_{Greedy}	0.77	1.03	1.01	1.40	1.27	1.52	1.63	1.64	1.96
Δ^{ave}_{Random}	0.77	1.01	1.00	1.38	1.25	1.48	1.53	1.58	1.88
$\Delta^{ave}_{LocSearch}$	0.76	1.00	1.00	1.38	1.25	1.47	1.53	1.57	1.88
T^{ave}_{CPLEX}	1.78	2.38	1.82	3.54	2.23	3.18	5.65	2.90	5.02
Δ^{max}_{Random}	1.78	2.38	1.82	3.54	2.22	2.88	5.10	2.79	4.41
$\Delta^{max}_{LocSearch}$	1.78	2.38	1.82	3.54	2.22	2.88	5.10	2.79	4.41
(T, F)	**(365,2)**	**(365,3)**	**(365,4)**	**(365,5)**	**(365,6)**	**(365,7)**	**(365,8)**	**(365,9)**	**(365,10)**
Δ^{max}_{Greedy}	514	600	600	600	600	600	600	600	600
T^{max}_{CPLEX}	600	600	600	600	600	600	600	600	600
Δ^{ave}_{Greedy}	1.56	1.64	1.34	1.06	0.75	1.09	2.51	3.49	8.17
Δ^{ave}_{Random}	1.56	1.62	1.33	1.05	0.73	1.07	0.77	0.48	0.89
$\Delta^{ave}_{LocSearch}$	1.56	1.62	1.33	1.05	0.73	1.07	0.77	0.48	0.89
Δ^{max}_{Greedy}	2.76	3.11	3.11	3.97	1.44	2.47	18.43	18.27	20.56
Δ^{max}_{Random}	2.76	3.11	3.11	3.97	1.43	2.39	1.58	2.93	2.66
$\Delta^{max}_{LocSearch}$	2.76	3.10	3.11	3.97	1.43	2.39	1.58	2.93	2.66

2.5.3 Managerial insights

We also conducted experiments to see a dependence of the optimal selection decision characteristics on the demand, setup time and selling price values, and demand and production cancellations. For the series $(T, F) = (52, 10)$, one instance denoted as I_{orig} (original instance) was randomly generated as described above but with close values V_1^* and V_2^* in order to find situations in which small variations of the input data cause the change of the optimal manufacturing configuration.

In the description below, $z_i \in \{-10, -5, 0, 5, 10\}$ are the deviation values in percent, $i = 1, 2$, and input data notations with upper index 0 apply to the original instance I_{orig}. Twenty new instances were randomly generated for each of the following seven instance families:

1) family $A^{z_1,z_2\%}_{setup}$ of instances with setup times $s_{kf} \in [\frac{100+z_1}{100} s^0_{kf}, \frac{100+z_2}{100} s^0_{kf}]$ for $k = 1, 2$ and all f,

2) family $A^{z_1,z_2\%}_{setup,1,\varnothing}$ of instances with setup times $s_{1f} \in [\frac{100+z_1}{100} s^0_{1f}, \frac{100+z_2}{100} s^0_{1f}]$ for all f (setups s_{2f} stay unchanged),

3) family $A^{z_1,z_2\%}_{setup,\varnothing,2}$ of instances with setup times $s_{2f} \in [\frac{100+z_1}{100} s^0_{2f}, \frac{100+z_2}{100} s^0_{2f}]$ for all f (setups s_{1f} stay unchanged),

4) family $A_{dem}^{z_1,z_2\%}$ of instances with demands $d_{ft} \in [\frac{100+z_1}{100}d_{ft}^0, \frac{100+z_2}{100}d_{ft}^0]$ for all f and t,

5) family $A_{price}^{z_1,z_2\%}$ of instances with selling prices $p_{ft} \in [\frac{100+z_1}{100}p_{ft}^0, \frac{100+z_2}{100}p_{ft}^0]$ for all f and t,

6) family $A_{dem.can}^{z}$ of instances, differing from I_{orig} only in that the demand values are re-set $d_{ft} := 0$ with probability z, $z \in \{0.05, 0.1, 0.2\}$, for all f and t (demand cancellation), and

7) family $A_{prod.can}^{z}$ of instances, differing from I_{orig} only in that product type f is not produced in time period t with probability z, $z \in \{0.05, 0.1, 0.2\}$, for all f and t (production cancellation). If product type f is not produced in time period t, then the selling price is re-set $p_{ft} := 0$ according to Remark 2.

We do not consider changes of the manufacturing times m_{kf} because, for the commonly used CNC machines, they are determined with a sufficient precision. Table 2.2 contains the following information for the original instance I_{orig}, $(z_1, z_2) \in \{(-10, -5), (-5, 0), (-5, 5), (0, 5), (5, 10)\}$, $z \in \{0.05, 0.1, 0.2\}$, and each of the instance families.

— Number $\#_{V_1^*<V_2^*}$ of instances in which configuration M_1 (DML) is less profitable than configuration M_2 (MML). For the row I_{orig}, $\#_{V_1^*<V_2^*} \in \{0, 1\}$.

— Number $\#_{V_1^*\geq V_2^*}$ of instances in which configuration M_1 is as good as configuration M_2. We have $\#_{V_1^*<V_2^*}+\#_{V_1^*\geq V_2^*} = 20$ for all rows but the row I_{orig}. For the row I_{orig}, $\#_{V_1^*\geq V_2^*} \in \{0, 1\}$ and $\#_{V_1^*<V_2^*}+\#_{V_1^*\geq V_2^*} = 1$.

— Minimal (over all instances) total product units capacity Cap_1 of configuration M_1, $Cap_1 = T\sum_{f=1}^{F}\left\lfloor\frac{1-s_{1f}}{m_{1f}}\right\rfloor$, and minimal (over all instances) y^*-*capacity* Cap_2^* of configuration M_2, defined via the optimal solution y^*, $Cap_2^* = \sum_{t=1}^{T}\sum_{f=1}^{F} y_{tf}^*$.

— Maximal and minimal (over all instances) values $V_1^* - V_2^*$.

— Maximal (over all instances) total demand dissatisfaction D_k^{dis} for configuration M_k, $k = 1, 2$. For a given instance, we define the total demand dissatisfaction as $D_1^{dis} = \sum_{f=1}^{F}\left(D_{Tf} - \sum_{t=1}^{T} x_{tf}^*\right)$ and $D_2^{dis} = \sum_{f=1}^{F}\left(D_{Tf} - \sum_{t=1}^{T} y_{tf}^*\right)$.

The considered minimal and maximal values of the above parameters can be used to predict possible changes of the selection decision depending of the input data variations.

The following managerial insights can be drawn from the data in Table **??**. For all the instance families, the y^*-capacity of M_2 (MML) is higher than the minimal total product units capacity of M_1 (DML), both configurations do not satisfy the total demand and configuration MML satisfies more demand than DML. For the original instance I_{orig}, configuration DML is better than MML. This relation does not change if the demand strictly decreases (for all instances of the families $A_{dem}^{-10,-5\%}$ and $A_{dem}^{-5,0\%}$), but it inverts for

Table 2.2 – Dependencies of solution characteristics of the input data, $(T, F) = (52, 10)$.

Input data	$\#_{V_1^* < V_2^*}$ **(M2 is better)**	$\#_{V_1^* \geq V_2^*}$ **(M1 is better)**	**Minimal** Cap_1	**Minimal** Cap_2	**Maximal** $V_1^* - V_2^*$	**Minimal** $V_1^* - V_2^*$	**Maximal** D_1^{dis}	**Maximal** D_2^{dis}
I_{orig}	0	1	24908	26716	229	229	4371	1228
$A_{dem}^{-10,-5\%}$	0	20	24908	24707	123105	116946	2675	884
$A_{dem}^{-5,0\%}$	0	20	24908	26095	39708	33999	3849	1092
$A_{dem}^{-5,5\%}$	5	15	24908	26722	4367	-5878	4485	1263
$A_{dem}^{0,5\%}$	20	0	24908	26857	-8490	-13708	4603	1335
$A_{dem}^{5,10\%}$	20	0	24908	28014	-85189	-89662	5955	1765
$A_{setup,1,2}^{-10,-5\%}$	20	0	24960	26731	-1476	-3887	4319	1216
$A_{setup,1,2}^{-5,0\%}$	18	2	24908	26716	728	-1978	4371	1231
$A_{setup,1,2}^{-5,5\%}$	6	14	24856	26704	2399	-2484	4423	1243
$A_{setup,1,2}^{0,5\%}$	3	17	24856	26689	2857	-592	4423	1258
$A_{setup,1,2}^{5,10\%}$	2	18	24752	26583	2592	-569	4475	1364
$A_{setup,1,\varnothing}^{-10,-5\%}$	0	20	24960	26719	4759	2623	4319	1228
$A_{setup,1,\varnothing}^{-5,0\%}$	0	20	24908	26719	2623	229	4371	1223
$A_{setup,1,\varnothing}^{-5,5\%}$	1	19	24856	26719	2623	-2065	4423	1228
$A_{setup,1,\varnothing}^{0,5\%}$	11	9	24856	26719	229	-2065	4423	1228
$A_{setup,1,\varnothing}^{5,10\%}$	20	0	24752	26719	-4003	-6062	4475	1228
$A_{setup,\varnothing,2}^{-10,-5\%}$	20	0	24908	26731	-4344	-6281	4371	1216
$A_{setup,\varnothing,2}^{-5,0\%}$	20	0	24908	26716	-873	-1978	4371	1231
$A_{setup,\varnothing,2}^{-5,5\%}$	8	12	24908	26704	1683	-1128	4371	1243
$A_{setup,\varnothing,2}^{0,5\%}$	0	20	24908	26689	3018	1702	4371	1258
$A_{setup,\varnothing,2}^{5,10\%}$	0	20	24908	26583	6946	5722	4371	1364
$A_{price}^{-10,-5\%}$	0	20	24908	26690	76327	61722	4371	1270
$A_{price}^{-5,0\%}$	9	11	24908	26718	11795	-5426	4371	1264
$A_{price}^{-5,5\%}$	13	7	24908	26677	19662	-17088	4371	1257
$A_{price}^{0,5\%}$	17	3	24908	26690	779	-12293	4371	1257
$A_{price}^{5,10\%}$	20	0	24908	26624	-58445	-76376	4371	1323
$A_{dem.can}^{0.05}$	0	20	24908	25567	78298	34833	3718	1085
$A_{dem.can}^{0.1}$	0	20	24908	24257	147091	103702	3149	967
$A_{dem.can}^{0.2}$	0	20	24908	22153	264953	200548	2216	658
$A_{prod.can}^{0.05}$	20	0	24908	26725	-24737	-56977	5617	1463
$A_{prod.can}^{0.1}$	20	0	24908	26586	-58291	-102397	5721	1602
$A_{prod.can}^{0.2}$	20	0	24908	26310	-95962	-168872	8457	1878

some instances if the demand can increase (family $A_{dem}^{-5,5\%}$) and it inverts for all instances of the families $A_{dem}^{0,5\%}$ and $A_{dem}^{5,10\%}$, in which the demand strictly increases. An explanation can be that the demand increase provides more opportunities for loading MML than for loading DML, and the new loading opportunities make MML better than DML.

Configuration DML remains better than MML if the setup times of DML strictly decrease (for all instances of the families $A_{setup,1,\varnothing}^{-10,-5\%}$ and $A_{setup,1,\varnothing}^{-5,0\%}$) or the setup times of MML strictly increase (for all instances of the families $A_{setup,\varnothing,2}^{0,5\%}$ and $A_{setup,\varnothing,2}^{5,10\%}$). Configuration MML becomes better than DML if all setup times of both configurations or the setup times of MML decrease substantially (families $A_{setup,1,2}^{-10,-5\%}$ and $A_{setup,\varnothing,2}^{-10,-5\%}$) or the setup times of DML increase substantially (family $A_{setup,1,\varnothing}^{5,10\%}$). For instances of the other families with setup time changes, either DML or MML can be optimal. These relations can again be explained by the changes of the loading opportunities for DML and MML, which are affected by the setup times.

With regard to the price changes, DML remains better than MML for all instances of the family $A_{price}^{-10,-5\%}$, in which prices reduce substantially and the relation changes for all instances of the family $A_{price}^{5,10\%}$, in which prices rise substantially. For instances of the other families with price changes, either DML or MML can be optimal. This observation can be explained by that MML is able to accommodate more units of expensive product types than DML.

Finally, if some demands are cancelled (families $A_{dem.can}^{0.05}$, $A_{dem.can}^{0.1}$ and $A_{dem.can}^{0.2}$), then DML remains optimal, and if some production is cancelled (families $A_{prod.can}^{0.05}$, $A_{prod.can}^{0.1}$ and $A_{prod.can}^{0.2}$), then MML becomes optimal. Reaction on the demand cancellation correlates with that on the demand values changes. Reaction on the production cancellation can be seen as counter-intuitive because the production cancellation is similar to the price reduction, in which case DML can remain optimal. However, production cancellation incurs zero prices for *some* product types and time periods, while price reduction applies uniformly to all product types and time periods. Perhaps, the expensive product types that MML is able to accommodate more than DML are affected insignificantly by the production cancellation.

2.6 Conclusions

Problem SELECT is introduced, which is to decide whether dedicated lines (DML) or a single multi-model line (MML) is economically more preferable for manufacturing products of several types in a given time period. The goal of employing any of the two configurations is to maximize the total profit, subject to the product demand and manufacturing time constraints. This problem is reduced to two optimization problems OPT1 and OPT2. A polynomial time algorithm is developed for OPT1, and NP-hardness is demonstrated for OPT2. The obtained computational complexity and algorithmic results are summarized in Table 2.3.

Table 2.3 – Computational complexity and algorithms

Problem	**Complexity and algorithms**	**Reference**
OPT 1	$O(F \cdot T \log T)$	Section 2.3
OPT1(f)	$O(T \log T)$	Algorithm **S**
OPT2	NP-hard, open for strong NP-hardness,	Theorem 2.4.1
	algorithms Greedy, Random, and LocSearch	Section 2.4
OPT2, $F = 1$	$O(T \log T)$	Section 2.4
OPT2, fixed $F \geq 2$	Open for ordinary NP-hardness,	Section 2.4
	pseudo-polynomially solvable	Algorithm **DP**

Computational experiment demonstrated that the quality of solutions delivered by the algorithms **Greedy**, **Random** and **LocSearch** is sufficiently good and that any of the two configurations (DML and MML) can be the optimal solution of the problem SELECT if they have similar product units capacities. Practically relevant modifications of the problem SELECT and techniques to obtain appropriate input data are of interest for the future research. The presented models, algorithms and software constitute a tool that can be used to evaluate different input data scenarios while making a selection decision between the production environments of DML and MML. The main application issue of this tool is the lack of the precise input data such as demands, prices, costs, setup times and manufacturing times for the future manufacturing and selling processes. Different approaches can be employed to obtain these data, but none of them is able to reliably predict the future. Therefore, we propose that the experts are used to obtain the input data and to evaluate the selection decisions for various input data scenarios.

A very interesting insight from this study corresponds to the impact of the setup time on the results, where the configuration MML becomes better than DML if all setup times of both configurations or the setup times of MML decrease substantially. If we ignore the setup time in MML, and consider an arbitrary order of entering products, MML can be treated as a mixed-model assembly line (MMAL). The current trends of increased mass customization, changing technologies and short product life cycles push manufacturing companies to produce multiple product models rather than a single one. An arbitrary order of products in MMAL rises an important challenge of designing and balancing such environment. In order to treat the incoming products and satisfy the targeted productivity, a MMAL should have an efficient dynamic resource/task re-assignment. The next chapters of the thesis will further study multi-manned mixed-model assembly lines, taking into account these aspects of reconfigurability.

The results of this chapter have been published in Dolgui et al. [2021].

Chapter 3

Model-dependent task assignment in mixed-model assembly lines with walking workers

3.1 Introduction

Mass customization and extensive changes in the market push manufacturing companies to employ mixed-model assembly lines [Manzini et al., 2018]. Short product life cycles, development of new technologies, frequent introduction of new products, and market fluctuations urge manufacturers to increase their adaptability and responsiveness. In this context, companies need to turn towards robust and efficient concepts of production system organization [Battaïa et al., 2018], like reconfigurability [Koren et al., 1999]. With a reconfigurable line, manufacturers can easily add, remove, or move manufacturing resources like machines, mobile robots, equipment, and workers. A manual mixed-model assembly line (MMAL) with walking workers can benefit from the concept of reconfigurability to adjust and adapt the line's capacity to production requirements.

As mentioned in Boysen et al. [2009], assembly line balancing (task assignment) and design (resource assignment) are crucial steps for an MMAL. One of the important steps of assembly line reconfiguration is line balancing: assignment of tasks to workstations under a given criterion (minimizing takt time, number of workstations, total cost, etc.). Several restrictive assumptions are commonly made in the literature on assembly line balancing problems [Baybars, 1986, Scholl and Becker, 2006, Boysen et al., 2008, Battaïa and Dolgui, 2013]: allocating only one worker to each station, producing only a single model in the line, workers and tasks are fixed, etc. However, for heavy tasks of large-size products, like in the automotive industry, assigning more than one worker to each station is more realistic [Lopes et al., 2020]. Multi-manned line balancing problems are often formulated using different restrictive assumptions [Dimitriadis, 2006, Michels et al., 2019, Becker and

Scholl, 2009, Kellegöz, 2017]. The movement of workers between stations adjusts stations' capacities to the production sequence [Sikora et al., 2017]. The flexibility of assembly lines with multiple moving workers at workstations can be enhanced by a model-dependent task assignment to stations. Note that model-dependent task assignment is required when each model of products has its specific task precedence relationships which may be conflicting.

This chapter deals with a multi-manned manual mixed-model assembly line balancing problem with walking workers (MALBP-W). The considered problem integrates the line design problem, consisting of equipment assignment to stations, and the task assignment problem, where tasks and workers are assigned to stations for a set of given product orders. Compared to the majority of studies in the literature, we consider that the production order entering the line is unknown, and we aim to design a line that can self-adjust to the products entering the line. More precisely, we provide a methodology to assign the equipment to the station and to select the number of workers so that the line meets the takt time for a prespecified set of pictures. As products enter the line, they consecutively occupy workstations creating different "pictures" of the line. By "picture" we mean the sequence of pairs station-product model that changes (product items shift towards the last station) every takt. As opposed to the concept of the line's picture, a product order can be defined as a sequence of product models entering the line whose number is not limited by the number of stations. For example, a product order can be $(B-A-A-C-A-C-C)$, where product B enters the line first, then two products A follow, and so on. Suppose that the line consists of four stations. One of the possible pictures of the line for such product order is: (station 1 - product C, station 2 - product A, station 3 - product A, station 4 - product B). In the following takt, the picture of the line is (station 1 - product A, station 2 - product C, station 3 - product A, station 4 - product A). The only product B present in the product order has left the line.

The order of products entering a mixed-model line is often not controllable as it depends on the uncertainties of an upstream production step and variable demand. We aim to provide methods to design a line that can adapt to the picture of the line. In each takt, the worker can be reassigned to a different station, and the assignment of task change depending on the item present in the station. The objective is to assign the equipment to the station and to select the number of workers in order to guarantee the line can meet the takt time for a prespecified set of pictures of the line. Therefore, the line is optimized for the worst picture of the line, since having enough resources for such picture guarantees a stable production flow delays. However, optimizing for the worst case may lead to over-conservative decisions. To control the conservatism, we assume that the user can specify the set of pictures of the line may encounter by providing the maximum number of units of each product model present in a picture. To define these restrictions, the user may rely on the demand mix, historical data, or expert's opinion.

The major contributions of this chapter are fourfold. First, we formulate the problem as a robust scenario-based mixed-integer linear programming model (MILP), where the cost related to workers and equipment is minimized for the worst picture. The line's reconfigurability is achieved by moving workers between stations and equipment duplication at stations. To the best of our knowledge, this study is the first to consider MALBP-W, and we provide a scenario-based MILP for the case with either model-dependent or fixed task assignment. Second, we show that the linear relaxation of the sub-problem that finds the number of workers required for the worst picture of the line yields an integer solution. As a consequence, we reformulate the MILP model using the dualization method commonly used in robust optimization. While the scenario-based MILP is unpractical because the number of scenarios/pictures is exponential in the number of stations, the reformulated MILP (RMILP) can solve practical size instances. Third, to solve large-size instances, we propose a constructive matheuristic (CM) and a fix-and-optimize heuristic (FOH). These approaches were rarely used in the literature on assembly line balancing. We may cite the work of Sun and Wang [2019], who applied a matheuristic approach for simple assembly line balancing problem. Lin and Ying [2016] proposed matheuristics approaches to solve a flowshop scheduling problem. Dang et al. [2021] developed a matheuristic for a parallel machine scheduling problem. The fix-and-optimize heuristic approach has not yet been applied to a MMAL balancing problem, although it was frequently used to solve lot sizing problems [Chen, 2015, Lang and Shen, 2011, Sahling et al., 2009, Helber and Sahling, 2010]. Finally, the performance of algorithms is evaluated in terms of solution quality and computational time through extensive computational experiments. In particular, we evaluate the impact of model-dependent task assignment to stations and compare it to the fixed task assignment. Our results suggest using the model-dependent task assignment because it results in a lower cost compared to a fixed task assignment. The difference of cost between these two assignments increases when the workforce cost and the problem size (the number of stations and types of products) increase.

The chapter is organized as follows: Section 3.2 formally defines the $MALBP-W$, and it provides a simple illustrative example as well as the mathematical model. Section 3.3 describes the proposed optimization approaches, including a transformed version of the mathematical model based on minimum cost flow problem, and two heuristics. Section 3.4 presents the computational results, discussion and managerial insights. The chapter ends with the conclusion and future research directions in Section 3.5.

3.2 Problem description and formulation

This section describes the multi-manned manual mixed-model assembly line balancing problem with walking workers ($MALBP{-}W$). In addition, it gives an illustrative example and the scenario-based mathematical formulation for the cases with fixed ($MALBP - W^{Fix}$) and model-dependent task assignment ($MALBP - W^{Md}$).

3.2.1 Description of $MALBP - W$

The problem consists of designing a mixed-model manual (manned) assembly line. The line contains a set $\mathcal{S} = \{1 \ldots S\}$ of sequentially located stations. The line assembles a set $\mathcal{I} = \{1 \ldots I\}$ of product models, which flow in any order through the line. The line is paced, and the items move from one station to the next at a regular time interval C, called takt time. At each takt, there is only one item at each station. Items of different types enter the line one by one, and all of them pass through all the stations. We denote the set of all tasks as $\mathcal{O}$. $\mathcal{O}$ refers to the unified set of tasks which some/all of them are common to different product models (possibly with different processing times). Each product model i requires a subset $\mathcal{O}_i$ of the unified set of all tasks $\mathcal{O}$. Note that if a task is model specific, it appears only in one of the $\mathcal{O}_i$. There is a set G_i of precedence relations (o, o') for each model i, where task o must be performed before task o'. The processing time p^l_{io} of task o performed on a model i depends on the number of workers l assigned to a station [Battaïa et al., 2015]. The proposed model is able to tackle any type of relation between process times and the number of workers (e.g. linear, non-linear relations). There is a limit l_{max} on the number of workers assigned to the same station. In addition, each task requires a certain equipment that has to be installed at a station. The set of equipment is denoted $\mathcal{E}$, and the requirements are represented with the parameter r_{oe}, whose value is equal to 1 if task o requires equipment e, and 0 otherwise. Each equipment has a certain ability to perform a set of tasks. If several tasks assigned to stations require the same type of equipment, it can be duplicated at stations [see Askin and Zhou, 1997, Tiacci and Mimmi, 2018, for example]. The objective is to minimize the sum of equipment and workforce costs. Each equipment e has a cost c_{se} at each station s. The equipment cost can be station dependent when installing some equipment pieces at some stations is more difficult and costly. However, our model can also handle the special case where equipment cost is not station dependent. As all workers are assumed to be identical and able to perform any task, the cost of workers α is the same for all workers.

The line is reconfigurable in the sense that workers can move from one station to another at any takt time, thus adapting the production capacity to the current load in each station. The workers' walking times compared to task processing times are negligible. This assumption is valid in industrial cases where walking times are sufficiently small compared to task processing times and cycle time, see for example Battaïa et al. [2015]. In addition, the considered product models in the studied mixed-model line are taken into account from a part family of products as a valid assumption in the concept of reconfigurable manufacturing systems.

In this work, we consider two variants of the problem $MALBP-W$. In $MALBP-W^{Fix}$, the task assignment to stations remains fixed for all product models. $MALBP-W^{Md}$ is similar to $MALBP-W^{Fix}$, but the task assignment in a station depends on the item in the station. In other words, in $MALBP-W^{Fix}$, the task assignment may change from one product model to another, but the task assignment is the same for several units of a given model.

The objective is to design a line that meets the takt time for the worst possible picture of the line. To define the set of pictures the line must cope with, we assume that the user can set the maximum number of units of each product model present on the line in any takt. This limit can be set based on a known ratio of demands for different products as in [Dolgui et al., 2018, Delorme et al., 2019], expert knowledge, or past data. For example, if at most one unit of product models A, B, and C can be present at 3 stations, only one station may be occupied by model A, one station by model B, and one station by model C in each takt. However, the proposed optimization approach can also handle the non-restricted case, where the maximum number of units of each product model present on the line is infinite.

Unlike other similar works [Taube and Minner, 2018, Cortez and Costa, 2015], this study does not consider the sequencing problem while assuming an arbitrary order of products in the mixed-model line. This assumption is valid in many environments containing the mixed-model line [Becker and Scholl, 2006, Bukchin et al., 2002]. Some existing works [Battaïa et al., 2015, Kucukkoc and Zhang, 2014, Delorme et al., 2019] provide methodologies to design and balance the line for a given set of production orders. As we consider fixed and model-dependent task assignments, given a set of possible orders of products, the number of workers is computed based on the pictures contained in the sequence. To give the user direct control over the line capacity, we let him/her restrict directly the possible line pictures.

Note that in the considered $MALBP-W$, product models can have different sets of tasks with different processing times and precedence graphs. The proposed solution approaches can also handle special cases, where product models require the same sets of tasks with the same precedence graph like in [Battaïa et al., 2015].

3.2.2 Illustrative example

This section illustrates the $MALBP-W$ on a simple example with two sequential stations. The example illustrates the impact of model-dependent task assignment on the number of workers, equipment assigned to the stations, and total cost of the workforce and equipment, while workers can move between stations at the end of each takt.

A picture of the line is denoted as $(1-i,...S-i')$, $i,i' \in \mathcal{I}$, and it determines the sequence of pairs station-product model in a certain takt. Since there are only two stations and two product models, the only possible pictures of the line (Pic.) are $(1-A, 2-B)$, $(1-B, 2-B)$, $(1-A, 2-A)$, and $(1-A, 2-B)$. Figure 3.1 shows precedence graphs and processing times for a common set of tasks $\{1, 2, .., 5\}$ for the two products. In Figure 3.1, processing time values correspond to the task durations when they are performed by a single worker (p^1_{io}). At most three workers can work at the same station simultaneously ($l_{max} = 3$). In this small example, we assume that the values of p^l_{io} are calculated by dividing p^1_{io} by the number of workers l assigned to each task o of each product model i, but our model can handle any processing time computations function. Processing time calculations are marked in blue color in Figure 3.2.

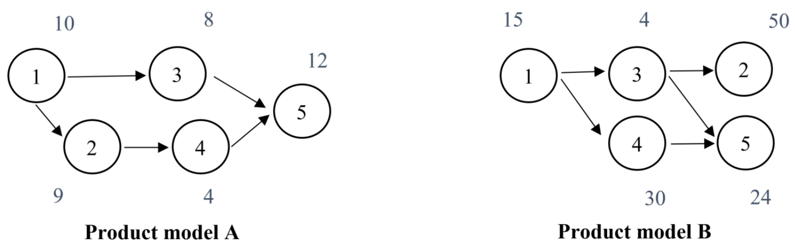

Figure 3.1 – The precedence graphs and tasks processing times of the simple example.

Table 3.1 shows the compatibility between equipment and tasks, and the cost of using the equipment at stations. Each equipment is able to perform a certain set of tasks. Note that the cost of equipment increases as the ability to perform a higher number of tasks increases. The cost of a worker is $\alpha = 500$, and the takt time is $C = 25$.

Table 3.1 – Compatibility between tasks and equipment, and the cost of equipment at each station.

	Task 1	Task 2	Task 3	Task 4	Task 5	Station 1	Station 2
Equipment 1	✓				✓	132	122
Equipment 2	✓		✓	✓	✓	172	148
Equipment 3	✓	✓	✓	✓	✓	224	200

Figure 3.2 presents the tasks, equipment and workers assigned to the stations for each picture of the line, the total processing time of each station marked in blue color, as well as the number of workers, equipment, and the total cost for the worst takt of both problems $MALBP-W^{Fix}$ and $MALBP-W^{Md}$. The optimal solution of $MALBP-W^{Fix}$ requires 6 workers and results in a total cost of 3372. The optimal solution to $MALBP-W^{Md}$ requires only 5 workers and gives a total cost 2872.

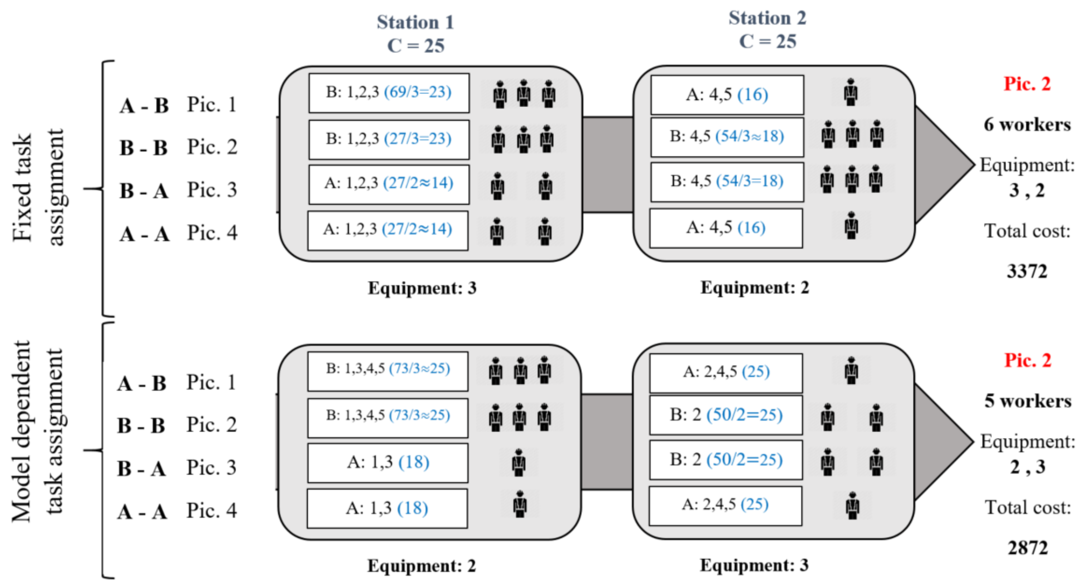

Figure 3.2 – The optimal solution of $MALBP-W^{Fix}$ and $MALBP-W^{Md}$ in the simple example.

Figure 3.3 shows the use of the solution for a given product order. Figure 3.3 clarify how pictures of the line change in every takt, depending on the product order. Here, only one order of products (B-A) is considered, where product B enter the line first, and then product A follows. Only three takts ($\mathcal{T} = 3$) and therefore only three pictures are shown for the considered order: $(1 - B, 2 - \varnothing)$, $(1 - A, 2 - B)$, $(1 - \varnothing, 2 - A)$.

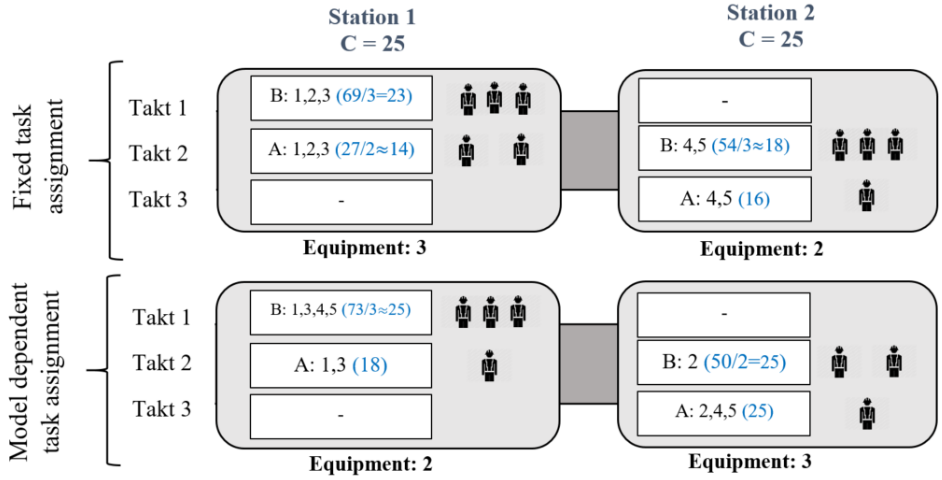

Figure 3.3 – An example of changing pictures of the line for product order (B-A).

3.2.3 Mathematical model for fixed task assignment

This section provides the mathematical formulation $MILP^{Fix}$ of the $MALBP - W$ with fixed workers. We denote K as the set of all possible pictures. Note that from a given picture of the line k, we can determine the station s_i^k where product model i is processed, as well as the model i_s^k processed at station s. Decision variables are as follows: Y is the number of workers to hire, W_{se} equal to 1 if equipment e is chosen for station s, and 0 otherwise, B_{sl}^k is equal to 1 if there are l workers in station s for picture k, and 0 otherwise; B_{oisl}^k is equal to 1 if there are l workers performing task o on model i at station s for picture k, and 0 otherwise; X_{soi} is equal to 1 if task o performed on model i is done at station s, and 0 otherwise.

The mathematical formulation of $MILP^{Fix}$ (3.1)-(3.12) is as follows.

$$\min \quad \alpha\, Y + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} c_{se} W_{se} \tag{3.1}$$

s.t.

$$\sum_{s\in\mathcal{S}} \sum_{l=1}^{l_{max}} l\ B_{sl}^{k} \ \le\ Y \qquad k \in K \tag{3.2}$$

$$\sum_{l=1}^{l_{max}} B_{sl}^{k} \ =\ 1 \qquad k \in K,\ s \in \mathcal{S} \tag{3.3}$$

$$\sum_{s\in\mathcal{S}} X_{soi} \ =\ 1 \qquad i \in \mathcal{I},\ o \in \mathcal{O}_i \tag{3.4}$$

$$X_{soi} = X_{soi'} \qquad i, i' \in \mathcal{I},\ o \in \mathcal{O}_i \cap \mathcal{O}_{i'},\ s \in \mathcal{S} \tag{3.5}$$

$$B_{oil}^{k} \ \le\ X_{soi} \qquad 1 \le l \le l_{max},\ k \in K,\ i \in \mathcal{I},\ o \in \mathcal{O}_i,\ s = s_i^k \tag{3.6}$$

$$B_{oisl}^{k} \ \le\ B_{sl}^{k} \qquad 1 \le l \le l_{max},\ k \in K,\ i \in \mathcal{I},\ o \in \mathcal{O}_i,\ s = s_i^k \tag{3.7}$$

$$B_{oisl}^{k} \ \ge\ B_{sl}^{k} + X_{soi} - 1 \qquad 1 \le l \le l_{max},\ k \in K,\ i \in \mathcal{I},\ o \in \mathcal{O}_i,\ s = s_i^k \tag{3.8}$$

$$\sum_{o\in\mathcal{O}_i} \sum_{l=1}^{l_{max}} p_{io}^{l}\ B_{oisl}^{k} \ \le\ C \qquad k \in K,\ s \in \mathcal{S},\ i = i_s^k \tag{3.9}$$

$$X_{soi} \ \le\ \sum_{e\in\mathcal{E}} r_{oe} W_{se} \qquad i \in \mathcal{I},\ o \in \mathcal{O}_i,\ s \in \mathcal{S} \tag{3.10}$$

$$\sum_{s\in S} s\ X_{soi} \ \le\ \sum_{s'\in S} s'\ X_{s'o'i} \qquad i \in \mathcal{I},\ (o, o') \in G_i \tag{3.11}$$

$$X_{soi},\ B_{sl}^{k},\ W_{se} \in \{0, 1\},\ Y \ge 0,\ B_{oisl}^{k} \le 1 \tag{3.12}$$

The objective function (3.1) is to minimize the costs associated with the workers and equipment, where α represents the labor cost (salary plus other charges) of a worker. Constraints (3.2) compute the total number of workers. Constraints (3.3) state that a single number of workers must be chosen for each station in each picture of the line. Constraints (3.4) ensure that each operation is assigned to a single station in each picture. Equations (3.5) force the tasks to remain fixed at stations for all the product models $i \in I$. Equations (3.6), (3.7), and (3.8) compute the value of B_{oisl}^{k} based on the values of B_{sl}^{k} and X_{soi}. Equations (3.9), (3.10), and (3.11) define the classical takt time, equipment, and precedence constraints, respectively.

3.2.4 Mathematical model for model-dependent task assignment

The mathematical formulation of $MILP^{Md}$ is similar to $MILP^{Fix}$, but without constraints (3.5), because the assignment of tasks to stations can dynamically change from one product model to another. Therefore, $MILP^{Md}$ corresponds to (3.1)- (3.4), (3.6) - (3.12).

3.3 Optimization approaches

As the number $|K|$ of pictures is naturally large, solving $MILP^{Fix}$ and $MILP^{Md}$ is time consuming. This section provides an efficient reformulation of the MILP, the constructive matheuristic (CM), and the fix-and-optimize heuristic (FOH). At the end of this section, we explain how $MILP^{Fix}$ can serve as a heuristic for $MILP^{Md}$.

3.3.1 MILP reformulation

$MALBP-W$ can be decomposed in two sub-problems. The first sub-problem assigns the tasks to a station, and it computes the minimum number of workers (Y_{is}) required to perform the tasks of product model i at station s within the takt time. The second sub-problem computes the number $f(Y_{11}, \ldots, Y_{IS})$ of workers in the worst picture. In this context, we show that the linear relaxation of the sub-problem is integer. Consequently, we can use the dualization method commonly used in robust optimization. As dualization transforms the maximization sub-problem into a minimization, it can be inserted into in the main problem. The mathematical formulation of the first sub-problem for $MALBP-W^{Fix}$ is given below.

$$\min \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} c_{se}W_{se} + f(Y_{11},\ldots,Y_{IS}) \tag{3.13}$$

s.t.

$$\begin{aligned}
Y_{is} &\geq \sum_{l=1}^{l_{max}} l\ B_{sl} && i\in\mathcal{I},\ s\in\mathcal{S} && (3.14)\\
\sum_{l=1}^{l_{max}} B_{sl} &= 1 && i\in\mathcal{I},\ s\in\mathcal{S} && (3.15)\\
\sum_{s\in\mathcal{S}} X_{soi} &= 1 && i\in\mathcal{I},\ o\in\mathcal{O}_i && (3.16)\\
X_{soi} &= X_{soi'} && i,i'\in\mathcal{I},\ o\in\mathcal{O}_i\cap\mathcal{O}_{i'},\ s\in\mathcal{S} && (3.17)\\
B_{oisl} &\leq X_{soi} && 1\leq l\leq l_{max},\ i\in\mathcal{I},\ o\in\mathcal{O}_i,\ s\in\mathcal{S} && (3.18)\\
B_{oisl} &\leq B_{sl} && 1\leq l\leq l_{max},\ i\in\mathcal{I},\ o\in\mathcal{O}_i,\ s\in\mathcal{S} && (3.19)\\
B_{oisl} &\geq B_{sl}+X_{soi}-1 && 1\leq l\leq l_{max},\ i\in\mathcal{I},\ o\in\mathcal{O}_i,\ s\in\mathcal{S} && (3.20)\\
\sum_{o\in\mathcal{O}_i}\sum_{l=1}^{l_{max}} p_{io}^{l}\ B_{oisl} &\leq C && i\in\mathcal{I},\ s\in\mathcal{S} && (3.21)\\
&(4.9)-(4.10)
\end{aligned}$$

For this sub-problem of $MALBP-W^{Md}$, constraints (3.17) must be removed from the above mathematical formulation.

Compared to the mathematical formulation (3.1)-(3.12), the above MILP ignores the production orders, $|K|$ pictures of the line, and these elements are considered in the sub-problem.

In this paragraph we explain the second sub-problem for workforce assignment to the stations. Let u_i be the maximum number of units for each product model i in a picture of the line. A new binary variable F_{is} is introduced and it is equal to 1 if model i is assigned to station s in the worst picture, and 0 otherwise. This assignment requires Y_{is} workers to process model i in station s. The function $f(Y_{11},\ldots,Y_{IS})$ corresponds to model (3.22) - (3.24). The objective (3.22) is to maximize the workforce cost to find the worst picture resulted from different product orders. Constraints (3.23) state that at most u_i model i are simultaneously present on the line. Constraints (3.24) state that each station is occupied by a single product model.

$$f(Y_{11}, \ldots, Y_{IS}) \quad = \max \quad \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} F_{is} \; \alpha Y_{is} \tag{3.22}$$

s.t.

$$\sum_{s \in \mathcal{S}} F_{is} \quad \leq \quad u_i \qquad\qquad i \in \mathcal{I} \tag{3.23}$$

$$\sum_{i \in \mathcal{I}} F_{is} \quad = \quad 1 \qquad\qquad s \in \mathcal{S} \tag{3.24}$$

Theorem 3.3.1. *The linear relaxation of sub-problem* (3.22) - (3.24) *yields a solution with integer values for variables F_{is}.*

Proof of Theorem 3.3.1. The proof shows that sub-problem (3.22) - (3.24) is equivalent to the minimum cost flow problem (MCFP) [Chassein and Kinscherff, 2019]. Given an oriented graph $G = (V, U)$ (where V and U are the nodes and edges, respectively), the MCFP decides the flow $f(n, m)$ on each edge $(n, m) \in U$ between two nodes $n, m \in V$. In each node, the input flow is equal to the output flow. The flow must respect the given capacity $c(n, m)$ of each edge. Each unit of the flow on edge (n, m) costs $a(n, m)$, and the objective is to minimize the total costs of the network.

Sub-problem (3.22) - (3.24) corresponds to the following MCFP: the set of nodes V contains a source node ss, a finish node ff, a node for each model $i \in I$, and a node for each station $s \in S$. The worst picture of the line (with the maximum workforce cost) occurs when all stations are occupied. There is an edge between source node ss and each model node ($i \in I$), and its capacity is the maximum number u_i of model i on the line. There is an edge between each model node i and each station node s that models the assignment of model i to station s. The capacity of this edge is equal to 1 since at each moment of time only one product model can be present at each station. Finally, there is an edge from each station node s to the final node ff, and its capacity is 1 since each station has at most one product model. The cost of all edges is set to 0, except for the cost of edge (i, s) which is set to the cost αY_{is} of the minimum number of workers required at station s to assemble model i within the takt time. $\square$

Figure 3.4 presents a simple example showing the process of obtaining a solution to the MILP problem based on the MCFP network. In this example, three product models $\{A, B, C\}$ enter a line containing three stations $\{St1, St2, St3\}$. The number of workers required to perform tasks on a single item of each product model at each station is written on the edges between product model and station nodes. The worst picture of the line is $(1-B, 2-B, 3-A)$, for which the total number of workers required is 11.

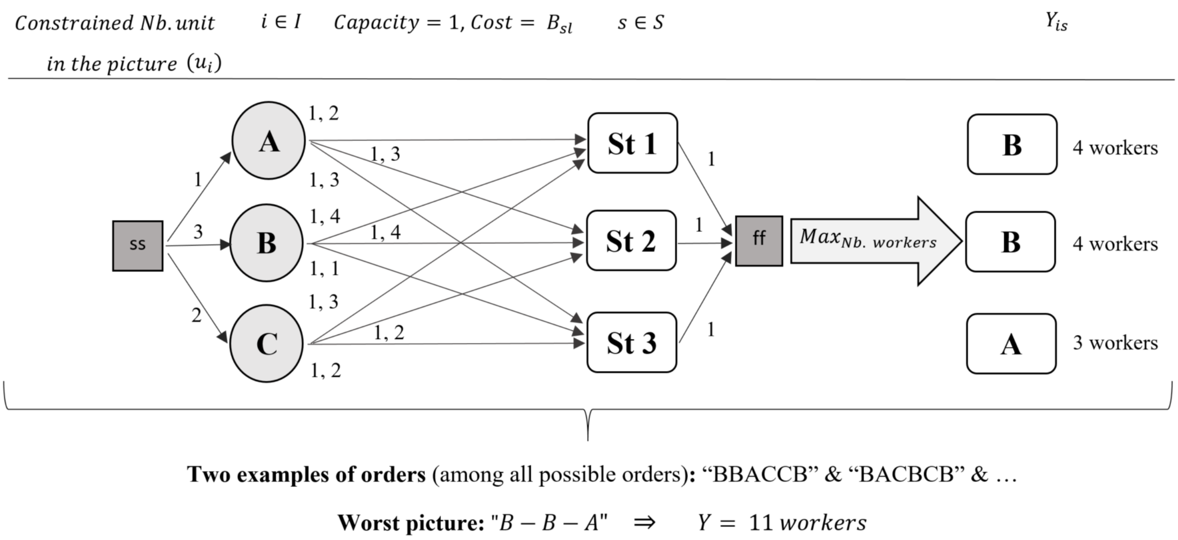

Figure 3.4 – A small example showing how reformulation of MILP works based on the MCFP network.

Dual programming for linearization: Equation (3.22) is a quadratic function, since both the product model assignment and the number of workers are multiplied as binary and integer decision variables, respectively. As the integrality constraint can be relaxed in (3.1)-(3.12), we can build a dual model of the sub-problem. The standard form, using auxiliary variable a_i (for each constraint i) is given as follows:

$$f(Y_{11}, \ldots, Y_{IS}) = \max \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} F_{is}\ \alpha Y_{is} \tag{3.25}$$

s.t.

$$\sum_{s \in \mathcal{S}} F_{is} + a_i = u_i \qquad i \in \mathcal{I} \tag{3.26}$$

$$\sum_{i \in \mathcal{I}} F_{is} = 1 \qquad s \in \mathcal{S} \tag{3.27}$$

The dual programming model is:

$$f(Y_{11},\dots,Y_{IS}) \quad = \min \quad \sum_{i\in\mathcal{I}} u_i \; M_i + \sum_{s\in\mathcal{S}} \; N_s \tag{3.28}$$

s.t.

$$M_i + N_s \quad \geq \quad \alpha Y_{is} \qquad i \in \mathcal{I} \quad s \in \mathcal{S} \tag{3.29}$$

$$M_i \quad \geq \quad 0 \qquad i \in \mathcal{I} \tag{3.30}$$

Here, M_i and N_s are the dual variables of (3.26)-(3.27), respectively.

Inserting the dual programming in the main problem (3.13) - (3.17) and (3.10) - (3.11) yields the reformulation of both $MILP^{Fix}$ and $MILP^{Md}$, and they are respectively denoted $RMILP^{Fix}$ (Equations 3.31) and $RMILP^{Md}$ (Equations 3.32).

$$\min \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} c_{se} W_{se} + \sum_{i\in\mathcal{I}} u_i \; M_i + \sum_{s\in\mathcal{S}} \; N_s \tag{3.31}$$

s.t.

$$(3.14) - (3.17) \& (3.10) - (3.11) \& (3.29) - (3.30)$$

$$\min \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} c_{se} W_{se} + \sum_{i\in\mathcal{I}} u_i \; M_i + \sum_{s\in\mathcal{S}} \; N_s \tag{3.32}$$

s.t.

$$(3.14) - (3.21) \& (3.10) - (3.11) \& (3.29) - (3.30)$$

3.3.2 Constructive matheuristic (CM)

The preliminary experiments showed that task assignments are computationally difficult to perform, while $RMILP^{Fix}$ and $RMILP^{Md}$ can be solved in a few seconds if variables X_{soi} are fixed. Therefore, our solution method first focuses on the assignment of tasks to stations (variable X_{soi}). Then, $RMILP^{Fix}$ and $RMILP^{Md}$ are solved assigning workers and equipment with fixed values of X_{soi}. This heuristic is called a constructive matheuristic (CM), since it involves both a heuristic algorithm and the reformulated mathematical model.

To calculate X_{soi} for each product model i, the heuristic starts with the assignment of the task without predecessors to the first station. If there is more than one task without predecessors, the following task is assigned according to one of the eight task selection rules described in Table 3.2. CM assigns tasks to the first station until the sum of the processing times reaches the takt time. However, CM considers the processing time of tasks performed by only one worker. The new takt time C_{new} is defined as follows: $\max(C, \frac{\sum_{o \in \mathcal{O}_i} p^1_{io}}{|S|}), i \in I$. In $MALBP-W$, multiple workers can work at a station with respect to the takt time. However, the CM stops assigning when the sum of processing times of tasks performed by only one worker reaches the takt time. To obtain a feasible task assignment by CM when the summation of task processing times for a product model is higher than the production time (summation of the takt time for all stations), the takt time is calculated by the division of total processing time by the number of stations. The heuristic continues to assign tasks in a similar manner until the last station. Values of X_{soi} are then given to $RMILPs$ to tackle the whole problem $MALBP-W$.

Table 3.2 – Task selection rules considered in constructive matheuristic (CM)

	Rules description
Rule 1	Largest processing time (LPT).
Rule 2	Smallest processing time (SPT).
Rule 3	Largest number of successors (LNS).
Rule 4	Smallest number of successors (SNS).
Rule 5	Largest processing time & Largest number of successors (LPTLNS).
Rule 6	Smallest processing time & Largest number of successors (SPTLNS).
Rule 7	Largest processing time & Smallest number of successors (LPTSNS).
Rule 8	Smallest processing time & Smallest number of successors (SPTSNS).

Algorithm 5 shows the basic steps of CM. A feasible solution is obtained in less than one second.

3.3.3 Fix-and-optimize heuristic (FOH)

The fix-and-optimize approach starts with the initial solution obtained by the CM, and it seeks to improve the task assignment. The initial solution is the best one among all solutions obtained using the rules from Table 3.2. In each iteration, FOH solves RMILP with most variables (X_{soi}) fixed to their values in a current solution, and another part of variables (X_{soi}) selected as a binary decision variable for re-optimization. To select this part of (X_{soi}), the tasks assigned to 2 or 3 adjacent stations become decision variables. These stations are selected randomly. Adjacent stations are selected in order to respect task precedence relations. Three rules are defined to select the stations randomly, see Table

Algorithm 5: Constructive matheuristic (CM)

Required: Precedence graph for each model i (G_i), task processing times for each model i performed by a single worker (p^1_{io}), and takt time (C).
Step 1: Compute the new takt time value (C_{new}).
Step 2: Set ($s := 1$) first. Start assigning tasks to stations for each product model (X_{soi}).
Step 3: For each model $i \in \mathcal{I}$, assign the task with no predecessor. If there is more than one task without predecessors, assign the task according to a selection rule proposed in Table 3.2.
Step 4: If C_{new} is exceeded, stop and pass to the next station.
Step 5: Repeat steps 3 and 4 with the same selection rule until all tasks are assigned to stations for all product models.
Step 6: Input the obtained values for X_{soi} to $RMILP^{Fix}$ or $RMILP^{Md}$ and solve. Obtain the greedy solution "*sol*".

3.3. The algorithm is stopped and the best current solution is saved if one of the following conditions is met: computational time has reached 1 hour; no improvement is observed after 20 iterations (in case of selecting 2 stations) or 10 iterations (in case of selecting 3 stations). These numbers are chosen based on several pre-computational experiments.

Table 3.3 – Station selection rules applied in FOH

	Rules description
Rule 1	Randomly select 2 adjacent stations ($2S$).
Rule 2	Randomly select 3 adjacent stations ($3S$).
Rule 3	Start with $2S$, and after 20 iteration without any improvement continue with $3S$ (2S3S).

Algorithm 6 provides the main steps of the FOH.

3.3.4 Fix^h heuristic

Note that a solution to $RMILP^{Fix}$ is a feasible but not optimal solution to $RMILP^{Md}$, because $MALBP-W^{Fix}$ corresponds to $MALBP-W^{Md}$ with the additional constraints (3.17). Fix^h uses $RMILP^{Fix}$ as a heuristic to solve $MALBP-W^{Md}$.

Algorithm 6: Fix-and-optimize heuristic (FOH)

Required: Parameter X_{soi} & solution sol (from Algorithm 5).

Step 1: Set the limit for number of iterations without any improvement, with the iteration counter "$Count$". The limit "$COUNT$" is set to 20 iterations (in case of selecting 2 stations) and to 10 iterations (in case of selecting 3 stations). Start the first iteration ($Count := 0$).

Step 2: Select adjacent stations for optimization with a selection rule proposed in Table 3.3, as a set (S') of stations.

Step 3: Consider all tasks assigned to the selected stations $s \in S'$ for each model $i \in I$, as binary decision variables X_{soi} for optimization. Keep the remaining values of X_{soi} known.

Step 4: Solve $RMILP^{Fix}$or $RMILP^{Md}$, and get a new solution sol^{new}

Step 5: If sol^{new} is better than solution sol, then set $sol = sol^{new}$ and re-start the iterations from the first one ($Count := 0$). Do the steps 2, 3 and 4 with the same selection rule. Otherwise go to the next iteration ($Count+ = 1$) and do the steps 2, 3 and 4 with the same selection rule.

Step 6: The algorithm stops either when it reaches "$COUNT$" iterations without improvement ($Count = COUNT$), or when 1 hour of computational time has passed.

3.4 Computational experiments and results

This section provides an adaptable data generation approach for $MALBP-W$ based on benchmark data generators from the literature. It evaluates the rules used in heuristics, analyzes the performance of each optimization approach, and provides managerial insights regarding the benefits of using the model-dependent task assignment. The problems are solved using IBM ILOG CPLEX Optimization Studio V12.10. The experiments were run on an Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz 2.11 GHz processor with 32 GB of RAM in MS Windows 10 Pro (64 bit) operational system. The computational time limit is set to 4 hours for each instance. The time limit for FOH is set to 1 hour.

3.4.1 Instances generation

To perform computational experiments, we extend the data generator, proposed by Otto et al. [2013] to the specificity of the problem in hand. Each of our instances merges I consecutive instances of Otto et al. [2013]. For example, our first generated instance contains the data of I first instances of Otto et al. [2013] and has I product models with different processing times and precedence graphs. The second instance contains the data of $\{2 \dots I+1\}$ first instances, and so on. There are two groups of instances, with 20 and 50 tasks, respectively. Note that the product models may have different processing times and precedence relationships between tasks. Herein, the processing time of tasks depends the number of workers, linearly, where $p_{io}^{l} = \dfrac{p_{io}^{1}}{|l|}$, $1 \leq l \leq l_{max}$, $i \in \mathcal{I}$, and $o \in \mathcal{O}_i$.

In order to provide an extensive analysis, eight classes of instances are considered:

1. Different sets of tasks with different precedence graphs and a restricted number of products' units (Fixed/Model-dependent).
2. Different sets of tasks with different precedence graphs and a non-restricted number of products' units (Fixed/Model-dependent).
3. Different sets of tasks with the same single precedence graph and a restricted number of products' units (Fixed/Model-dependent)
4. Different sets of tasks with the same single precedence graph and a non-restricted number of products' units (Fixed/Model-dependent)
5. The same set of tasks with different precedence graphs and a restricted number of products' units (Fixed/Model-dependent)
6. The same set of tasks with different precedence graphs and a non-restricted number of products' units (Fixed/Model-dependent)
7. The same set of tasks with the same single precedence graph and a restricted number of products' units (Fixed/Model-dependent)
8. The same set of tasks with the same single precedence graph and a non-restricted number of products' units (Fixed/Model-dependent)

Equipment costs at each station are generated randomly with a uniform distribution in the range $[100, 300]$. Three different values for workers' salary are considered: less, within, and more than the range for equipment costs ($\alpha = \{50, 200, 500\}$). Different number of stations and product models ($S, I = \{3, 5, 10\}$) are defined. The instances' sizes are determined by the 3-tuple (I, S, O), where I, S, and O represent the number of product models, stations, and tasks, respectively.

To generate the compatibility matrix, R_{oe} is set to 1 with probability $\frac{\bar{c}_e}{\bar{c}}$ (and 0 otherwise), where $\bar{c}_e$ is the average cost of equipment e (over all stations), and $\bar{c}$ is the average equipment cost (over all equipment and stations).

The takt time in the instances of Otto et al. [2013] is set to 1000. In $MALBP-W$, several workers may perform tasks in a station. It is reasonable to consider a reduced takt time since the processing time decreases with a higher number of workers. Here, the takt time can take the following values $\{1000, 500, 250\}$, and we only report results for "proper" instances, i.e. feasible and with more than one worker per station.

Single/different precedence graphs for product models: Otto et al. [2013] considered different precedence graphs for different product models. We only use their precedence graph for the first product model in the cases with the same single precedence graph (see classes 3,4,7,8).

Single/different set of tasks for product models: having the same set of tasks for all product models as Otto et al. [2013] raises no issues. In order to have different sets of tasks for product models, we randomly eliminate tasks in the range $[8, 12]$ for 20 task instances and in the range $[20, 30]$ for 50 task instances.

Restricted/non-restricted number of units of product models: In the restricted case, we consider a single unit restriction of all products ($u_i = 1$, for all $i \in I$). Thus, K includes all possible pictures of the line with a single unit of each product model ($|K| = I!/(I - S)!$). For the non-restricted case, $u_i = S$ for all $i \in I$ ($|K| = I^S$).

For each size and each class, 10 instances from all instances of Otto et al. [2013] are randomly selected. It leads to a total number of instances equal to 2640.

3.4.2 Analysis of the heuristics

Table 3.4 shows the average solution quality for problems $MALBP - W^{Fix}$ and $MALBP - W^{Md}$. It provides the average gap between either heuristics CM or FOH, and the best solution provided by $RMILPs$ for each selection rule. The best solution provided by $RMILPs$ refers to either the optimal solution of $RMILPs$ for optimally solved instances or to the best upper bound found by $RMILPs$ for non-solved instances within the considered time limit. The best rule used in CM is to assign the task with Smallest Processing Time and Smallest Number of Successor (SPTSNS) (22.5% for fixed and 18.9% for model-dependent task assignment). However, the difference among the rules used in CM is not significant. The best rule used in FOH is to start by selecting 2 adjacent stations and continue by selecting 3 stations as soon as there is no improvement of the solution (2S3S), see the bold values in Table 3.4 (-3.9% for fixed and -7.5% for model-dependent task assignment).

Table 3.4 – Solution quality of the heuristics.

MALBP-W	CM								FOH		
	LPT	SPT	LNS	SNS	LPTLNS	SPTLNS	LPTSNS	SPTSNS	2S	3S	2S3S
Fixed	24.0	23.2	23.5	24.0	23.8	23.6	23.3	**22.5**	-0.1	-3.6	**-3.9**
Model-dependent	20.0	**18.9**	19.4	19.8	19.8	19.5	19.2	**18.9**	-2.4	-7.3	**-7.5**

3.4.3 Performance of the optimization approaches

This subsection evaluates the performance of optimization approaches in terms of solution quality and computational time. Table 3.5 shows the number of instances solved by $RMILPs$, the average optimality/integrality gap provided by CPLEX, the average gap between the solutions obtained by heuristics CM, and FOH and the best solution found by $RMILPs$. Values Fix^h show the average relative gap between the model-dependent case solution value and the fixed one. It is computed using the formula (3.33). A positive value of Fix^h means that a better cost was obtained in the model-dependent case. For (5,5,50) and (10,10,50)-size instances fixed case solutions provided better values than model-dependent case solutions. In fact, $RMILP^{Md}$ reached the time limit, while $RMILP^{Fix}$ provided better solutions. The negative gap for (5,5,50)-size instances solved by FOH and (10,10,50)-size instances solved by CM and FOH means that RMILP reached the time limit, while heuristics provided better solutions. Both $RMILPs$ solve small instances with 3 stations and 3 product models to optimality. For larger instances with the number of tasks equal to 50 and the number of product models and stations equal to 10, CPLEX starts running out of memory.

$$Gap = 100 \; \frac{Cost(MALBP - W^{Fix}) - Cost(MALBP - W^{Md})}{Cost(MALBP - W^{Fix})} \tag{3.33}$$

Table 3.5 – Solution quality of optimization approaches depending on the instances' size. (I, S, O) stands for the number of product models, stations, and tasks, respectively.

Size	$MALBP-W^{Fix}$				$MALBP-W^{Md}$				
(I,S,O)	RMILP		CM	FOH	RMILP		CM	FOH	Fix^h
	N° solved	Opt. gap			N° solved	Opt. gap			
(3,3,20)	360/360	0.00	32.4	0.0	360/360	0.00	35.7	0.0	2.3
(3,3,50)	240/240	0.00	24.5	0.0	240/240	0.00	25.3	0.0	0.6
(5,5,20)	240/240	0.00	32.0	1.1	233/240	0.01	31.5	2.6	4.5
(5,5,50)	226/240	0.06	24.8	0.9	97/240	6.74	18.5	-3.1	-0.7
(10,10,50)	0/240	49.20	-19.3	-21.8	0/240	61.04	-31.0	-41.4	-36.9

Table 3.6 represents the same information as Table 3.5 but for different values of α, i.e. workers' salary. CM works better when α increases, while it is the other way round for FOH. The gap between model-dependent and fixed cases increases with growing α.

Table 3.7 shows the average computational times of MILPs, RMILPs, CM, FOH, Fix^h for different instances' sizes. Applying RMILP significantly improves computational times compared to MILP. CM provides a feasible and close to optimal solution in less than 1 second, whereas FOH provides a closer to optimal solution within few minutes.

Table 3.6 – Solution quality of optimization approaches depending on the cost of workers.

α	$MALBP-W^{Fix}$				$MALBP-W^{Md}$				
	RMILP		**CM**	**FOH**	**RMILP**		**CM**	**FOH**	**Fix^h**
	N° solved	**Opt. gap**			**N° solved**	**Opt. gap**			
50	360/440	1.10	41.3	-3.7	322/440	2.07	36.0	-8.1	-6.8
200	355/440	1.13	12.0	-4.5	300/440	2.22	0.9	-7.3	-4.7
500	351/440	1.11	10.3	-5.3	307/440	2.14	9.0	-7.7	-3.3

Table 3.7 – Average computational time of optimization approaches.

Size	$MALBP-W^{Fix}$				$MALBP-W^{Md}$			
(I,S,O)	**MILP**	**RMILP**	**CM**	**FOH**	**MILP**	**RMILP**	**CM**	**FOH**
(3,3,20)	31.4	0.6	<1	8.0	89.3	1.1	<1	22.2
(3,3,50)	701.1	8.4	<1	65.8	2202.3	12.8	<1	82.0
(5,5,20)	-	65.1	<1	39.4	-	1431.4	<1	96.9
(5,5,50)	-	2363.6	<1	281.7	-	9743.4	<1	515.5
(10,10,50)	-	14400.0	<1	338.2	-	14400.0	<1	1163.7

$MILP^{Fix}/RMILP^{Fix}$ are solved significantly faster than $MILP^{Md}$ $/RMILP^{Md}$ because the fixed case requires many fewer task assignment decisions.

Table 3.8 compares computational times for different worker costs (salaries). The computational time is sensitive to the cost of workers. The problem is hard to solve when α is in the range of equipment cost because it is hard to balance the workers and the equipment costs.

Table 3.8 – Average computational time of optimization approaches depending on the cost of workers.

α	**MALBP-W^{Fix}**			**MALBP-W^{Md}**		
	RMILP	**CM**	**FOH**	**RMILP**	**CM**	**FOH**
50	2877.3	<1	119.5	4131.7	<1	267.8
200	3147.8	<1	147.9	5007.8	<1	388.3
500	3126.9	<1	128.6	4820.9	<1	365.9

3.4.4 Managerial insights

The values reported in this section are either optimal solution values obtained by $RMILP^{Fix}$ and $RMILP^{Md}$ when instances are optimally solved or approximate solution values returned by FOH^{Fix} and FOH^{Md} when the instances are not optimally solved.

Table 3.9 shows that increasing the cost of workers increases the cost of equipment and the number of duplications, while the number of required workers decreases. Another observation is that increasing workers' cost makes the model-dependent task assignment more profitable than the fixed one (from 2% to 2.9%, then to 5%).

Table 3.10 shows the influence of different classes of instances on the equipment cost,

Table 3.9 – The impact of the worker cost on the number of workers, equipment cost and duplication, and cost saving via $MALBP-W^{Md}$.

α	MALBP-W^{Fix}			MALBP-W^{Md}			Fixed/Model-dependent
	Eq. Cost	N° Worker	N° Dup.	Eq. Cost	N° Worker	N° Dup.	Gap (%)
50	331.8	15.8	0.4	324.0	15.4	0.4	2.0
200	595.5	13.7	0.9	646.4	13.0	1.0	2.9
500	673.4	13.4	1.0	702.7	12.8	1.2	5.0

the number of workers and equipment duplications. It can be concluded that manufacturing companies may expect a less cost of workers but higher cost of equipment with both model-dependent and fixed cases, when they: 1) consider a restricted number of product model units in the line 2) produce products with different sets of tasks 3) produce products with the same precedence graph between the tasks. However, there is not a significant influence of producing products with the same or different precedence graph on the total number of workers and cost of equipment. On the other hand, companies can hire a less number of workers using a model-dependent task assignment. However, it might be counterbalanced by an increased equipment cost.

Table 3.10 – The influence of different classes of instances on the equipment cost and the number of workers and equipment duplications.

MALBP-W	MALBP-W^{Fix}			MALBP-W^{Md}		
	Eq. Cost	N° Worker	N° Dup.	Eq. Cost	N° Worker	N° Dup.
Restricted	550.3	12.7	0.8	569.0	12.2	0.9
Non-restricted	516.8	15.9	0.7	546.4	15.4	0.8
Same set of tasks	513.5	14.8	0.7	551.1	14.4	0.8
Different sets of tasks	553.7	13.9	0.9	564.3	13.1	0.9
Same precedence graph	537.2	14.2	0.7	558.3	13.7	0.8
Different precedence graphs	530.0	14.4	0.8	557.1	13.8	0.9

Table 3.11 shows cost saving advantages of $MALBP-W^{Md}$ over $MALBP-W^{Fix}$ for different classes of instances and instance sizes. Model-dependent task assignment performs better than the fixed case, especially when: 1) the number of product models increases (2.2% to 4.4% for 20 tasks, and 0.6% to 3.7% and 4.7% for 50 tasks); 2) the user considers restrictions on the number of product model units in the line (4% rather than 2.2%); 3) products have different sets of tasks (3.5% rather than 2.7%), and 4) products have different precedence graphs (3.7% rather than 2.6%).

Table 3.11 – Cost saving via $MALBP - W^{Md}$ as compared to $MALBP - W^{Fix}$.

MALBP-W	Size (I,S,O)					Average
	(3,3,20)	(3,3,50)	(5,5,20)	(5,5,50)	(10,10,50)	
Restricted	2.9	1.0	6.4	4.7	5.3	4.0
Non-restricted	1.8	0.3	2.4	2.7	4.0	2.2
Same set of tasks	3.8	0.6	4.3	2.1	2.7	2.7
Different sets of tasks	0.3	0.7	4.6	5.3	6.7	3.5
Same precedence graph	1.6	0.2	3.2	3.7	4.3	2.6
Different precedence graphs	3.1	1.1	5.6	3.7	5.0	3.7
Average	2.2	0.6	4.4	3.7	4.7	

3.5 Conclusion

This chapter focuses on a multi-manned manual mixed-model assembly line balancing problem with walking workers (MALBP-W). A new scenario based mixed-integer linear programming (MILP) model is built with the criterion of minimizing the total cost of workers and equipment. The proposed MILP is robust. It minimizes the total cost for the worst product model sequence. The equipment pieces can be duplicated depending on tasks assigned to stations. Two policies for task assignment are considered: fixed and model-dependent. In model-dependent policy, task assignments are model-dependent, which can change depending on the product model entering the line. Various states of the line and product models are taken into account, such as having restricted/non-restricted numbers of items of the same product model, considering same/different set(s) of tasks required for the products, and considering the same/different precedence graph(s) among tasks. The generic problem concerns a manned mixed-model assembly line, where products require different sets of tasks with different processing times, and different precedence relationships among tasks. The main challenge is to evaluate and compare the impact of fixed and model-dependent task assignments on solutions. A reformulated version of the proposed MILP is developed. The reformulation is based on the reduction to the minimum cost flow problem (MCFP). A dual programming approach is used to linearize the reformulated MILP. The reformulation significantly improves the computational time. Two heuristics, a constructive matheuristic (CM) and a fix-optimize heuristic (FOH) show their high efficiency both in terms of solution quality and computational time. Since a fixed policy is a special case of the model-dependent policy, it may serve as a heuristic for model-dependent task assignment. A set of instances is generated based on the benchmark data for the simple assembly line balancing problem in the literature. The computational results, including solution quality and computational time comparisons are shown and analyzed. Several managerial insights are highlighted. In general, model-dependent task assignment proves to be less costly than fixed task assignment.

In the next chapter, the current chapter's problem is addressed using a dynamic task assignment implying the change in tasks' allocation at the end of each takt. The results are compared to the ones obtained by applying model-dependent and fixed task assignments. Dynamic task assignment depends on the order of entering product models given from the set of possible product orders. Due to the difficulty of generating all possible product orders, the next chapter provides a sequence generator to create only a set of sequences. It also develops several optimization approaches to solve the proposed problem.

The results of this chapter have been submitted as in Hashemi-Petroodi et al. [2021b].

Chapter 4

Dynamic task assignment in mixed-model assembly lines with walking workers

4.1 Introduction

This chapter evaluates the impact of dynamic task assignment compared to the proposed model-dependent and fixed task assignments depicted in Chapter 3. In the dynamic task assignment strategy, tasks are assigned to stations at the beginning of each shift depending on the product order. Similarly to Chapter 3, we assume that the product model order is decided by the flow coming out of the upstream shopfloor, and it cannot be modified. However, the line can adapt to this flow by changing the task and workforce assignments. We recall that model-dependent task assignment strategy corresponds to the case where each product model has its own task assignment which remains the same regardless of the line's picture and already accomplished tasks. In the fixed task assignment strategy, the task assignment is the same for all product models.

The production sequence has a significant impact on the performance of mixed-model lines. Unlike the previous studies in which the product sequence is a decision variable [Mosadegh et al., 2020, Bautista and Cano, 2011] or given [Zhang et al., 2020], searching for the robust design of a mixed-model line has been rarely studied. There are many situations in which the production sequence in the assembly line is hardly controllable. For instance, a company may face a highly variable demand in an make-to-order environment [Bukchin et al., 2002]. Some works consider a set of possible product orders generated from the demand ratio and historical data [Battaïa et al., 2015]. Battaïa et al. [2015] design a cyclic sequence of products entering the line taking into account the product models' ratios in the total annual production. Battaïa et al. [2015] assumed that workers are allowed to move between stations after finishing a non-preemptive task, and that a task processing time depends on the number of workers assigned to the corresponding station. In contrast to our study, the assignment of tasks to stations and task sequence at each station are known and cannot be changed.

On the other hand, except for a few studies [Choi, 2009, Kucukkoc and Zhang, 2014], the majority of works on MMAL balancing with walking workers assume that task assignment is given [Battaïa et al., 2015, Delorme et al., 2019, Dolgui et al., 2018, Hwang and Katayama, 2010]. Regarding the task assignment, the existing studies mentioning dynamic task assignment imply a lower degree of reconfigurability compared to the current research. Thus, Choi [2009] considers a model-dependent task assignment in a simple MMAL with fixed workers. Kucukkoc and Zhang [2014] consider a dynamic change of task assignment at each production cycle, where a production cycle corresponds to a certain combination of models at stations. We have the same assumption, but unlike Kucukkoc and Zhang [2014] this work considers a wider range of product orders and also workforce assignment in the robust optimization of a multi-manned mixed-model line balancing problem with walking workers.

We aim to design a reconfigurable line which is able to adapt to any change in the market demand affecting the product order. The reconfigurability of such line is achieved by a re-arrangement of resources (walking workers and equipment duplications) and the corresponding task reassignment. This chapter considers the same problem as in Chapter 3, i.e. the multi-manned mixed-model assembly line balancing problem with walking workers $MALBP-W$. We consider the same instances as in Chapter 3. Unlike the study in Chapter 3, here, we assess the impact of dynamic task assignment for a set of product orders given at the beginning of a time period (e.g., a day, a week, a month). The studied $MALBP-W$ with such dynamic task assignment is called $MALBP-W^{Dyn}$. We propose a scenario-based Mixed-Integer Linear Programming (MILP) model to solve the problem for a set of generated product orders. The proposed MILP model is robust. It minimizes the costs of workers and equipment in the worst case/takt. To cope with a large number of possible product orders, we propose an approach that iteratively adds resource-consuming product orders to the line design found by solving an MILP problem. A local search algorithm is developed to find these orders for large-scale instances. Using a simulation process, we evaluate the line design's robustness. A larger number of product orders is generated. Feasibility and objective values are estimated using the number of workers and equipment assignment taken from the MILP problem's solution. The results show that the proposed optimization approach performs well in terms of computational time and solution quality. In addition, we provide some managerial insights that suggest using the dynamic task assignment strategy in order to significantly reduce the costs related to workers and equipment compared to the model-dependent and fixed task assignment strategies.

The chapter is organized as follows. Section 4.2 formally describes the considered problem, provides a simple example of a dynamic task assignment and also the scenario-based MILP model. Section 4.3 depicts the proposed optimization approaches, including the simulation model, simulation-based optimization, and the local search algorithm. Section 4.4 presents the computational results, and managerial insights. Finally, Section 4.5 summarizes the study and provides some future research directions.

4.2 Problem description

We consider another variant of $MALBP-W$, denoted as $MALBP-W^{Dyn}$, in which tasks can be re-assigned to stations at the beginning of each period when the product order unfolds.

The problem focuses on designing a mixed-model manual (multi-manned) assembly line. The line contains a set $\mathcal{S} = \{1 \dots S\}$ of sequential stations, and it assembles a set $\mathcal{I} = \{1 \dots I\}$ of product models. Items flow through the line in any possible sequence within a given set of finite sequences. We consider a paced line, where the items flow from one station to the next at a regular time interval C, called takt time. At each takt, there is only one item at each station. Product models enter the line one by one, and all items pass through all stations. We denote by $\mathcal{O}$ the set of all tasks. The product model i requires a subset $\mathcal{O}_i$ of tasks, and $\mathcal{O}_i$ may be equal to $\mathcal{O}$. There is a set G_i of precedence relations for each model i. G_i contains pairs (o, o') that represent precedence constraints, where task o must be performed before task o'. Eventually, the sets G_i may be identical for all models i. Each task o of product model i requires a processing time p_{io}^l which depends on the number of workers l assigned to the station [Battaïa et al., 2015]. In a multi-manned line, multiple workers can work at the same station simultaneously. Workers can move from one station to another at the end of each takt. It allows adapting the production capacity at stations to the given product order. The workers' walking times compared to task processing times are negligible. Note that the maximum number of workers operating at each station cannot exceed l_{max}. In addition, the processing of a task requires the installation of a piece of equipment at the station. The set of equipment is denoted $\mathcal{E}$, and the requirements are represented with the parameter r_{oe}, whose value is equal to 1 if task o requires equipment e, and 0 otherwise. To enhance task re-affectations, equipment pieces can be duplicated at stations. The objective is to minimize the sum of equipment and workforce costs. Each equipment e has a cost c_{se} depending on the station s where it is located. Since all workers are assumed to be identical and able to perform any task, the cost of workers α is the same for all workers.

At the design stage, the $MALBP-W^{Dyn}$ consists in finding the total number Y of workers to hire, and the position of the fixed equipment represented by the variable W_{se} equal to 1 if equipment e is chosen for station s, and 0 otherwise. At the operational stage, the product model order for the next period (e.g., a day, a week, or a month) is given, and $MALBP-W^{Dyn}$ assigns the tasks and workers to stations in each takt.

The line must be designed with respect to the takt time applied to all possible sequences and product models. We assume that a set of product orders is given at the beginning of a time horizon (e.g. a day, a week, or a month). It may contain successive items with high processing times at all stations. Therefore, in order to guarantee the production continuity, the required number of workers must be sufficiently high. To reduce the line's cost, the decision-maker may impose constraints on the incoming product orders. In practice, the decision-maker determines such constraints regarding the historical data. He/she may select may select only those product orders for which a scheduling tool always finds a feasible sequence. In this chapter, we consider two types of such constraints. However, the proposed solution approach is able to incorporate any possible constraint on the product order. The first type of constraints prevents the presence of more than u_i items of the product model i in a picture of the line. This type has been already addressed in Chapter 3. For example, to preserve the diversity of out-coming final products, the user may decide to have at most two items of models A, B, and C at 3 stations. The second type of constraints prevents having more than u'_i consecutive items of a product model i in the product order (as opposed to the line's picture limited by the number of stations). Such restrictions can be caused by producing a relatively rare customized/luxury product model together with more popular mass-production models. Due to the static nature of the studied problems $MALBP-W^{Md}$ and $MALBP-W^{Fix}$ in Chapter 3, these problems are not sensitive to such type of restrictive constraints (u'_i) as we have considered the first type (u_i).

Knowing the product order, the goal is to design a line for which the total cost of workers and equipment is minimized in the worst takt. Quite often, the order of product items entering the line is not controllable due to the uncertainty of output from the upstream production or delivery stage, and the demand fluctuation. In order to respect the desired productivity, avoid delays and disruptions, the number of workers must be sufficiently large to handle the worst possible takt.

In addition to $MALBP - W^{Dyn}$, results of $MALBP - W^{Fix}$ and $MALBP - W^{Md}$ solved by the proposed solution approach are also presented. It allows drawing better managerial insights. In $MALBP - W^{Fix}$, the task assignment to stations remains fixed for all models in any order, whereas task assignment can be different for each product model in $MALBP - W^{Md}$.

4.2.1 Illustrative example

To further clarify the problem, we provide an illustrative example. This example is similar to that of Chapter 3, in which Figure 3.1 and Table 3.1 provide the precedence graphs, processing times, the compatibility between equipment and tasks, and the cost of using the equipment at each station. The line has two stations, and it produces two product models A and B. The restrictions applied to the line's picture u_i and product order u'_i are the same: $u_A = u'_A = 1$ and $u_B = u'_B = 2$. With an operational horizon of 4 takts, these restrictions imply four product orders ($|\Omega| = 4$) each of which is composed of four products ($|\omega| = 4$): $\{A, B, B, A\}$, $\{B, A, B, B\}$, $\{A, B, A, B\}$, and $\{B, B, A, B\}$. For example, for the product order $\{B, B, A, B\}$, two items of product B enter the line first, then a product A and a product B follow. We assume that the cycle time is $C = 25$, and the cost of a worker is $\alpha = 500$. Figure 4.1 shows the optimal line design of $MALBP - W^{Dyn}$ (equipment, number of workers), along with the operational decisions (task and worker assignment) for the order $\{B, B, A, B\}$. This solution requires 4 workers and a total equipment and workforce cost of 2372. Note that dynamic task assignment strategy reduces the number of workers and the total cost compared to the optimal solutions of $MALBP - W^{Md}$ (5 workers and the total cost of 2872) and $MALBP - W^{Fix}$ (6 workers and the total cost of 3372) shown in Figure 3.2, Section 3.2.2, Chapter 3.

4.2.2 Mathematical formulation of $MILP^{Dyn}$

This section provides a mathematical formulation of $MILP^{Dyn}$ that relies on the set Ω of all possible product orders that respect the restrictive constraints. The length of each order ω is denoted by $|\omega|$. We denote by ω_j the model of the j^{th} item in the order ω

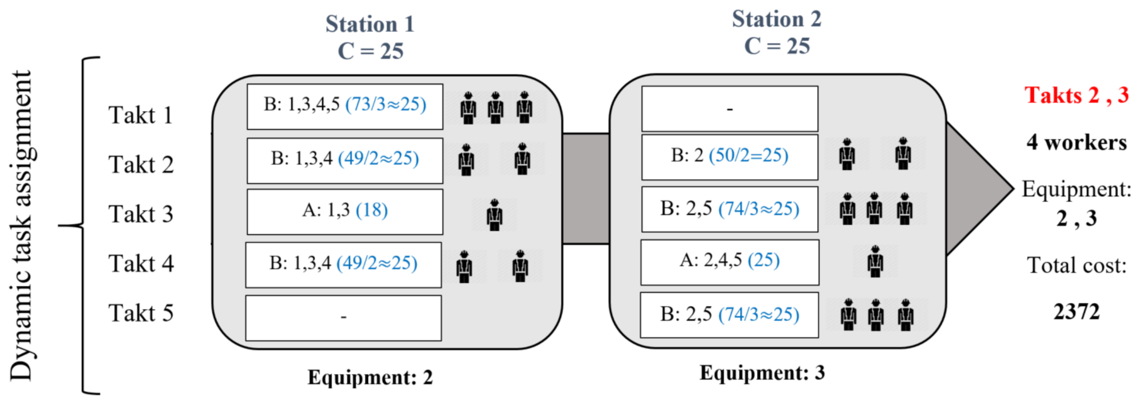

Figure 4.1 – The optimal solution of $MALBP - W^{Dyn}$ in the simple example.

$(\omega_j \in \mathcal{I})$. We can identify the station $s^{\omega}_{jt} = t - j + 1$ at which the j^{th} item (of a product model i in set I) is processed in each takt t, as well as the position $j^{t\omega}_{s} = t + s - 1$ in the order ω of the item processed at station s in takt t. The number of takts in the orders is denoted as $\mathcal{T}$.

The operational decision variables for assignment of workers and tasks are as follows:

— $B^{\omega t}_{sl}$ equals to 1 if there are l workers at station s in period t for order ω (with $l \in \mathcal{L} = \{1 \dots l_{max}\}$), and 0 otherwise.
— $B^{\omega t}_{ojsl}$ equals 1 if l workers perform task o of the j^{th} item at station s in period t for order ω, and 0 otherwise.
— X^{ω}_{soj} equals 1 if task o of item j is performed at station s for order ω, and 0 otherwise.

The scenario-based formulation of $MALBP - W^{Dyn}$ is given by (4.1)-(4.11).

$$\min \quad \alpha\, Y + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} c_{se} W_{se} \tag{4.1}$$

s.t.

$$\sum_{s\in\mathcal{S}} \sum_{l=1}^{l_{max}} l\; B^{\omega t}_{sl} \;\leq\; Y \qquad \omega\in\Omega,\ t\in\mathcal{T} \tag{4.2}$$

$$\sum_{l=1}^{l_{max}} B^{\omega t}_{sl} \;=\; 1 \qquad \omega\in\Omega,\ s\in\mathcal{S},\ t\in\mathcal{T} \tag{4.3}$$

$$\sum_{s\in\mathcal{S}} X^{\omega}_{soj} \;=\; 1 \qquad \omega\in\Omega,\ j\in\{1,\ldots,|\omega|\},\ i=\omega_j,\ o\in\mathcal{O}_i \tag{4.4}$$

$$B^{\omega t}_{ojsl} \;\leq\; X^{\omega}_{soj} \qquad l\in\mathcal{L},\ \omega\in\Omega,\ j\in\{1,\ldots,|\omega|\},\ t\in\mathcal{T},\ s=s^{\omega}_{jt},\ i=\omega_j,\ o\in\mathcal{O}_i \tag{4.5}$$

$$B^{\omega t}_{ojsl} \;\leq\; B^{\omega t}_{sl} \qquad l\in\mathcal{L},\ \omega\in\Omega,\ j\in\{1,\ldots,|\omega|\},\ t\in\mathcal{T},\ s=s^{\omega}_{jt},\ i=\omega_j,\ o\in\mathcal{O}_i \tag{4.6}$$

$$B^{\omega t}_{ojsl} \;\geq\; B^{\omega t}_{sl} + X^{\omega}_{soj} - 1 \qquad l\in\mathcal{L},\ \omega\in\Omega,\ j\in\{1,\ldots,|\omega|\},\ t\in\mathcal{T},\ s=s^{\omega}_{jt},\ i=\omega_j,\ o\in\mathcal{O}_i \tag{4.7}$$

$$\sum_{o\in O_i} \sum_{l=1}^{l_{max}} p^{l}_{io}\; B^{\omega t}_{ojsl} \;\leq\; C \qquad \omega\in\Omega,\ t\in\mathcal{T},\ s\in\mathcal{S},\ j=j^{t\omega}_{s},\ i=\omega_j \tag{4.8}$$

$$X^{\omega}_{soj} \;\leq\; \sum_{e\in\mathcal{E}} r_{oe} W_{se} \qquad \omega\in\Omega,\ s\in\mathcal{S},\ j\in\{1,\ldots,|\omega|\},\ i=\omega_j,\ o\in\mathcal{O}_i \tag{4.9}$$

$$\sum_{s\in S} s\; X^{\omega}_{soj} \;\leq\; \sum_{s'\in S} s'\; X^{\omega}_{s'o'j} \qquad \omega\in\Omega,\ j\in\{1,\ldots,|\omega|\},\ i=\omega_j,\ (o,o')\in G_i \tag{4.10}$$

$$X^{\omega}_{soj},\ B^{\omega t}_{sl},\ W_{se}\in\{0,1\},\ Y\geq 0,\ B^{\omega t}_{oisl}\leq 1 \tag{4.11}$$

The objective function (4.1) is to minimize the costs associated with workers and equipment, where α represents the labor cost (salary plus other charges) of a worker. Constraints (4.2) calculate the total number of workers needed in the line. Constraints (4.3) verify that a single number of workers is assigned at each station in each takt. Constraints (4.4) state that each task must be affected to a single station for each order. The constraints (4.5), (4.6), and (4.7) compute the value of $B^{\omega t}_{ojsl}$ based on the values of $B^{\omega t}_{sl}$ and X^{ω}_{soj}. Finally, inequations (4.8), (4.9) (4.10) represent the classical takt time, equipment, and precedence constraints, respectively.

4.3 Solution approach

The mathematical formulation of $MILP^{Dyn}$ cannot scale to large instances, since the number of possible orders grows exponentially with the number of takts in a period. This section presents a Sequence Generator (SG) to sample a set of orders respecting the defined restrictions u_i and u'_i. To study the robustness of the resulting model, this section introduces a simulation model. This simulation considers a large number of product orders that differ from the set of orders used in scenario-based MILP. As in the adversarial approach in robust optimization, this method iteratively builds a set of product orders. To speed up the solution approach, we propose a heuristic that consists of a greedy algorithm and a local search algorithm. The heuristic aims to find the worst order to be added to the set of considered orders.

4.3.1 Product order generation

This section describes a sampling tool to generate a set Ω of product orders, called Sequence Generator (SG), designed to create $|\Omega|$ orders with length $|\omega|$. The product orders are randomly generated, but they respect the user-defined restrictions on the number of units of each product model i in the picture of the line (u_i), and on the number of successive units of the same product model i (u'_i) in the product order. For each order ω, SG selects the item one by one, from the first to the last. At each iteration j, SG selects a model randomly (with a uniform distribution) among the models that can be placed in position j without violating the restrictions in the partial order $\omega_1, \ldots, \omega_j$.

For instance, we explain how the last product order of the proposed example in Section 3.2.2 ($\omega = \{B, B, A, B\}$) is created. We assume $u_A = u'_A = 1$ and $u_B = u'_B = 2$. Herein, both product models have the same probability 0.5 to randomly get the first position (e.g. product B is selected $\omega_1 = B$). For the second position when B is in the first position, both product models are allowed to get the position, randomly (e.g. product B is selected $\omega_2 = B$). Since we reach two consecutive items of product B and $u_B = u'_B = 2$, only product A can get into the third position (product A is selected $\omega_3 = A$). Similarly, since $\omega_3 = A$ and $u_A = u'_A = 1$, product model B surely occupies the fourth position (product B is selected $\omega_4 = B$). The final order is given as $\omega = \{B, B, A, B\}$.

4.3.2 Simulation model

The solution of $MILP^{Dyn}$ with just a sample of product orders may not be robust to the entire set of product orders since the sample may not contain the worst order. The solution of $MILP^{Dyn}$ provides a line design that includes the number Y^* of workers and the equipment assignment W^*_{se}. To evaluate the robustness of such lines, we simulate the operational stage, i.e. workers' and task assignments, for a large number of orders. For each order, we re-solve $MILP^{Dyn}$, but we assume that the maximum number of workers Y and the equipment assignment W_{se} are given. They are not decision variables anymore. We denote the resulting formulation as $RMILP^{Dyn}$. This model only seeks to find workers' and task assignments for a given product order. Note that we also consider the case where orders used for simulation are longer than the order used in $MILP^{Dyn}$. These orders are generated with SG. We solve $RMILP^{Dyn}$ for a reasonable number N of iterations. Among these N iterations, Nr_{yes} iterations are feasible with respect to the line design, while some of them can be infeasible because there is no reconfiguration (workers' and task assignments) respecting the takt time for these orders. The robustness *Rob* of a solution *sol* means the percentage of feasible iterations. It is computed as follows: $Rob = \dfrac{Nr_{yes}}{N} * 100$.

4.3.3 Simulation-based optimization (SO)

This section presents a simulation-based optimization (SO) approach that relies on the simulation to iteratively build the set of product orders considered in the optimization model. This approach initializes the set Ω of orders using SG. SO solves $MILP^{Dyn}$ to get a feasible line design, with a number of workers Y^* and an equipment assignment expressed by the values of variables W^*_{se}. SO simulates this line design for N iterations considering different orders. It re-solves $MILP^{Dyn}$ for the first infeasible order found in an iteration, and re-starts the iterations from the first iteration ($Iter = 0$). The approach stops either at a given time limit $Time$ or after passing the N^{th} iteration. Figure 4.2 provides a schema of the proposed simulation model, and as well as the simulation-based optimization approach. Note that for the simulation model, we only consider the first N iterations which is counted by $Iter'$ ($Iter'_{max} = N$).

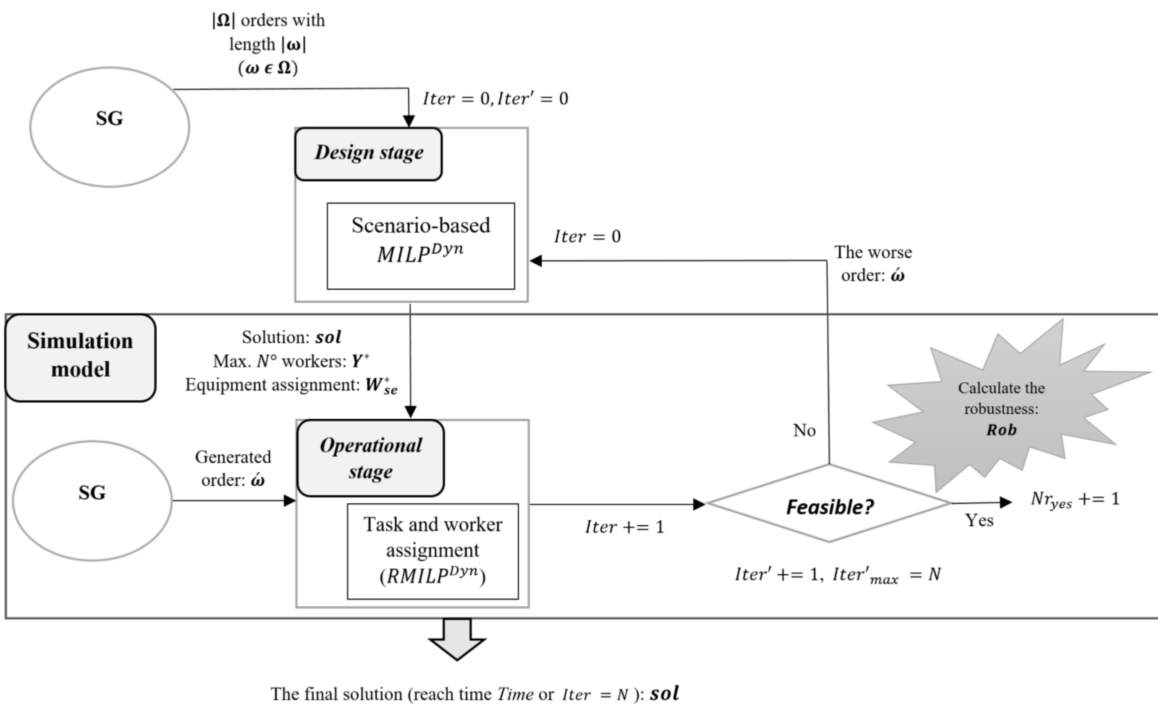

Figure 4.2 – Simulation model and simulation-based optimization approach.

4.3.4 Local search algorithm (LS)

This section presents a greedy algorithm and a local search algorithm to accelerate the convergence of the SO approach. The greedy algorithm generates a product order to initialize the set Ω' including an initial products order, and the local search replaces the random sequence generation in SO to aggressively search for the worst order.

Greedy algorithm prioritizes product models with large total processing times TP_i: $TP_i = \sum_{o \in \mathcal{O}_i} p^1_{io}$, for all $i \in \mathcal{I}$. Product model(s) with longest processing time(s) (e.g., a bottleneck/luxury product model) require(s) more workers in the line, and the worst order is likely to contain a large number of successive units of product model with the highest processing time. The greedy algorithm starts from an empty order, and adds product models one by one at the end of the current order. At each step, the greedy algorithm inserts the item with largest TP_i at the end of the partial order, subject to the given restrictions on successive items of different product models. Finally, it provides the order ω'' with $|\omega''|$ items. We assume a set Ω' of the order ω'', where $\Omega' = \{\omega''\}$.

The following example, illustrated in Figure 4.3, shows how the greedy algorithm creates the order $\omega'' = \{B, B, A, B\}$ containing $|\omega''| = 4$ items. Given the processing times of both product models A and B, it calculates $TP_A = \sum_{o \in \mathcal{O}_A} p^1_{Ao} = 43$ and $TP_B = \sum_{o \in \mathcal{O}_B} p^1_{Bo} = 123$. The algorithm prioritizes product B over product A since $TP_B > TP_A$. Then, respecting restrictions ($u_A = u'_A = 1$ and $u_B = u'_B = 2$), product model B with higher processing time gets the first position, again product B gets the second position, product A is located in the third position, and product B gets the last position. Note that, for instance after two items B, the product model B cannot get the third position, since $u_B = u'_B = 2$.

The descent algorithm is a local search method which iteratively improves an initial solution by moving from a solution to a neighbor solution. The neighborhood of a solution h contains solutions with a close structure from h, and it is often built by performing a move (i.e., a slight modification) on h. To find a product order with large cost, a move $m(j,i)$ changes the product model ω_j in position j to a model $i \neq \omega_j$. The neighborhood $N(\omega)$ of order ω contains all solutions obtained by applying $m(j,i)$ on ω for all $j \in \{1 \dots |\omega|\}$ and $i \in \mathcal{I}$, where $i \neq \omega_j$. If the resulting solution violates the restrictions on successive items of the same product model, it is removed from the neighborhood. In this work, the initial solution is the last product order inserted in Ω'. The first/initial order is generated by the greedy algorithm in the first iteration of the adverserial framework. For the following orders, it corresponds to the output of the last run of the proposed descent algorithm described below.

For instance, Figure 4.3 gives an example with a neighbor of the greedy order $\omega'' = \{B, B, A, B\}$. In the neighbor $\hat{\omega}$, the first item (product B) is replaced by product A since it respects the restrictions $u_A = u'_A = 1$ and $u_B = u'_B = 2$. However, the order $\bar{\omega}$ is not included in the neighborhood since it does not respect the restrictions.

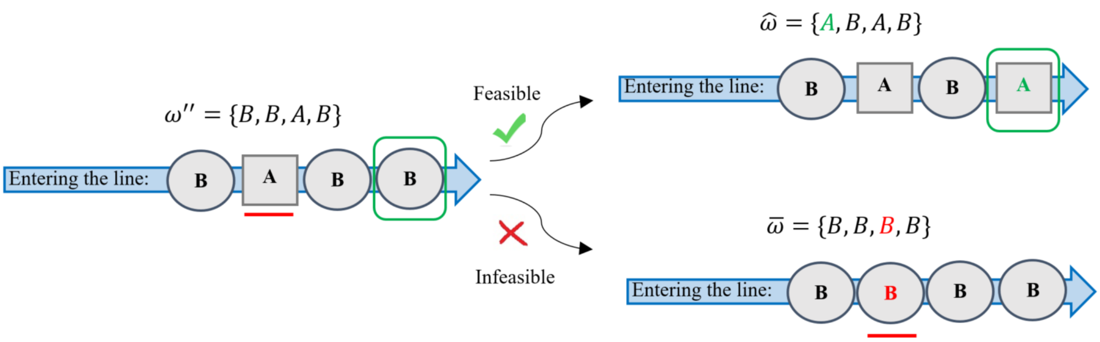

Figure 4.3 – Generation of a new order in the neighborhood.

The proposed local search (LS) algorithm first solves $MILP^{Dyn}$ considering the generated order ω'' obtained by the greedy algorithm to get the initial feasible line design with an equipment assignment corresponding to the values of variables W^*_{se}. The value of Y^* obtained from the solution to $MILP^{Dyn}$ is needed for a later stage. Then, LS creates the neighborhood of the order ω''. To evaluate the number of workers Y required for each neighbor order, descent re-solves $MILP^{Dyn}$ for each neighbor order $\hat{\omega}$, but we assume that the equipment assignment W_{se} is given (as W^*_{se}). It is not a decision variable anymore. We denote this reformulation as $RMILP^{Dyn}_{new}$. This model optimizes the number of required workers and finds the workers' and tasks' assignments for each neighbor order $\hat{\omega}$. The descent algorithm moves to the best neighbor at each iteration, and inserts the number of required workers Y^* obtained from the solution to $RMILP^{Dyn}_{new}$ to the set Y'. However, as soon as the algorithm finds an order $\hat{\omega}$ that provides a higher number of workers than the one initially obtained from $MILP^{Dyn}$ ($Max(Y') > Y^*$), it re-solves $MILP^{Dyn}$, and re-designs the line (finds the equipment assignment) considering an updated set Ω' with the order $\hat{\omega}$ added to the set Ω', i.e. $\Omega' = \Omega' \cup \{\hat{\omega}\}$. It updates the solution sol'. Then, the algorithm looks for the neighbor orders of the newly created order $\hat{\omega}$, and continues the process. The algorithm stops when it either reaches the time limit $Time$ or finishes creating the neighborhood, i.e. modifies all items in the order. Figure 4.4 shows a schema of the proposed LS.

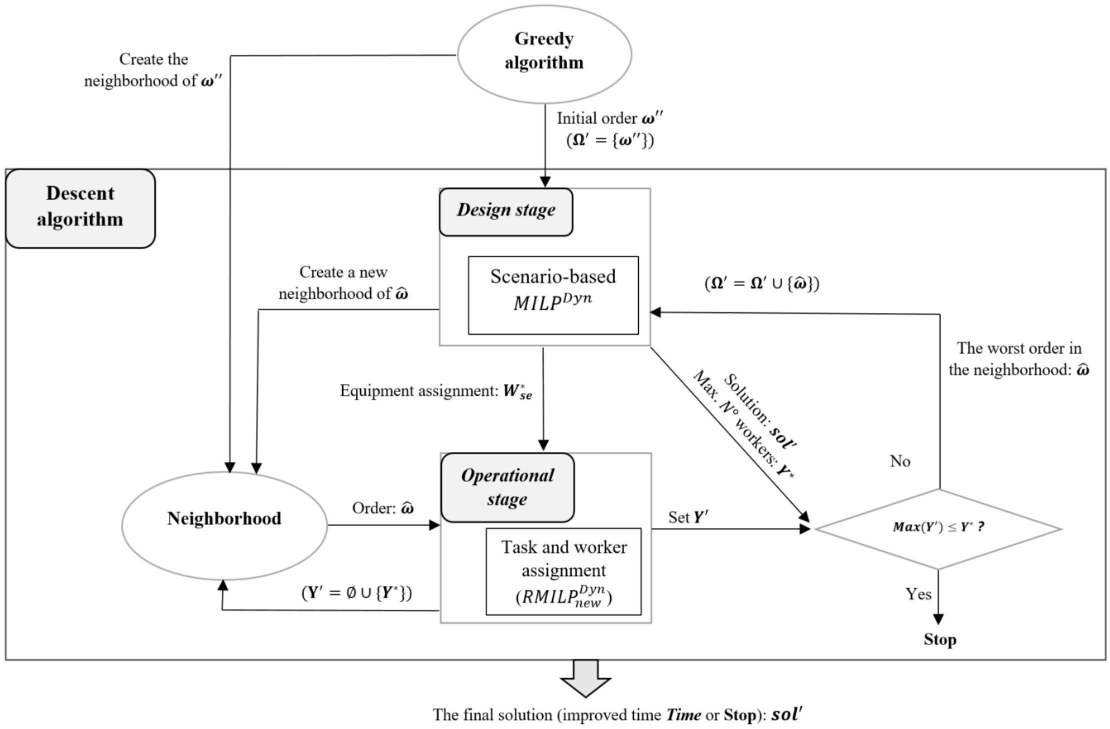

Figure 4.4 – A schema of the proposed local search algorithm.

4.4 Computational experiments and results

This section uses the same data generation approach as the one proposed in Chapter 3 based on benchmark data generators from the literature, except herein, we consider a new restrictive assumption on the successive units of the same product model in the product orders (u'_i).It evaluates the performance of each optimization approach in terms of robustness, computational time, and provides managerial insights on the use of the dynamic task assignment strategy for a given set of orders. The problems are solved using IBM ILOG CPLEX Optimization Studio V12.10. The experiments are run on an Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz 2.11 GHz processor with 32 GB of RAM in MS Windows 10 Pro (64 bit) operational system. The time limit of the Cplex is set to 4 hours, large enough to solve all instances optimally. The time limit $Time$ for the proposed approaches, simulation model, SO, and LS, is set to 1 hour.

To perform computational experiments, as mentioned, we use a subset of the instances generated in Chapter 3, except with slightly changes of restrictive assumptions. We consider two sizes $((I, S, O) = \{(3, 3, 20), (3, 3, 50)\})$, where the 3-tuple (I, S, O) gives the number I, S, and T of product models, stations, and tasks, respectively. For these sizes, we include the instances with two costs of the workers ($\alpha = \{50, 500\}$), both single/different precedence graph(s), and single/different set(s) of tasks for product models. In addition, to study the impact of the restrictive assumptions, we consider two different classes of instances as explained below.

Restricted/non-restricted number of units of product models: In the restricted case, we consider a single unit restriction of all product models ($u_i = 1$, for all $i \in I$), and the order cannot have more than 1 successive units of the same product model i ($u'_i = 1$, for all $i \in I$). Precisely, after a single item of each product model, another product model must enter the line. For the non-restricted case, $u_i = S$ (S is the number of stations) for all $i \in I$, and we assume the order also cannot have more than S successive units of the same product model i ($u'_i = S$, for all $i \in I$).

To solve $MILP^{Dyn}$ and also in SO, SG generates $|\Omega| = 5$ orders with two different lengths $|\omega| = 10$ and $|\omega| = 15$ items. Simulation model runs for $N = 100$ iterations. At each iteration, a new order with length $|\omega'| = 15$ and $|\omega'| = 20$ of items is taken into account for the instances with $|\omega| = 10$ and $|\omega| = 15$ of items, respectively. In addition, through the greedy algorithm and also at each iteration of the proposed descent algorithm, the orders with $|\omega''| = |\hat{\omega}| = 10$ items are generated.

For each size and each class, 10 instances from all instances of Otto et al. [2013] are randomly selected (the same instances in Chapter 3). It leads to a total number of instances around 320.

4.4.1 Performance of the approaches

This section evaluates the performance of the proposed approaches in terms of computational time. It also compares the obtained minimized costs to the ones found by solving $MALBP - W^{Md}$. Positive values in the columns "Dyn/Md Gap" show the percentage of cost saving obtained by using dynamic task assignment instead of model-dependent one. It is calculated by Equation (4.12). In addition, this subsection analysis the robustness of the methods. The column Rob provides the ratio of sequence in the simulation for which a valid task assignment exists with the line design provided by the considered method.

$$Gap = 100\, \frac{Cost(MALBP - W^{Md}) - Cost(MALBP - W^{Dyn})}{Cost(MALBP - W^{Md})} \tag{4.12}$$

Table 4.1 shows that increasing the length of generated orders increases the computational time. However, longer orders lead to more reliable solution since gap between the total cost of dynamic and model-dependent task assignments decreases (from 6.4% to 5.9%, or from 16.6% to 8.5%). As the total cost of $MALBP - W^{Md}$ is larger than the total cost of $MALBP - W^{Dyn}$, a decreasing gap indicates that the obtained optimal solutions (for the considered orders) are closer to the worst-case order and therefore they are more robust. Table 4.1 also shows the results of the simulation applied to the solution of $MILP^{Dyn}$. We observe that increasing the length of the generated orders increases the robustness *Rob* (percentage of feasible iterations) of the solution (from 98.9% to 99.2%, or from 96.2% to 97.6%).

Table 4.1 – Results of $MILP^{Dyn}$ and simulation with longer product orders.

Size	$\|\omega\|$	$\|\omega'\|$	$MILP^{Dyn}$		**Simulation**	
(I,S,O)			**CPU time (s)**	**Dyn/Md Gap (%)**	**CPU time (s)**	**Rob (%)**
(3,3,20)	10	15	372.0	6.4	58.4	98.9
	15	20	763.1	5.9	71.1	99.2
(3,3,50)	10	15	397.1	16.6	148.0	96.2
	15	20	1271.5	8.5	204.2	97.6

Table 4.2 shows how the proposed simulation-based optimization (SO) approach improves the solution found by solving $MILP^{Dyn}$. The re-solution of $MILP^{Dyn}$ using the infeasible order found by solving $RMILP^{Dyn}$ improves the robustness as the percentages of cost saving, obtained by using dynamic task assignment instead of model-dependent, shrinks. For example, for $(3,3,20)$)-size instances this percentage decreases from 6.4% to 4.1%. At the same time, this improvement using SO requires a larger computational time. For the same example of instances it passes from 372.0 to 568.2 seconds.

Table 4.2 – The performance of the proposed simulation-based optimization (SO) approach.

Size	$\|\omega\|$	$\|\omega'\|$	**SO**	
(I,S,O)			**CPU time (s)**	**Dyn/Md Gap (%)**
(3,3,20)	10	15	568.2	4.1
	15	20	1044.9	3.9
(3,3,50)	10	15	998.1	6.6
	15	20	1845.5	5

Table 4.3 shows that the proposed LS algorithm provides more robust solutions in shorter computation time. The percentages of cost saving, obtained by using dynamic task assignment instead of model-dependent, decreases, for example, from 3.9% (or 4.1%) to 3.8% for $(3,3,20)$)-size instances. At the same time computational times reduce from 1044.9 (or 568.2) to 27.8 seconds for this example of instances. The solutions' robustness

(percentage of feasible iterations) has also improved (99.3%) compared to the robustness of the solutions obtained from $MILP^{Dyn}$ in Table 4.1 (98.9% or 99.2%).

Table 4.3 – The performance of the proposed Local Search (LS) algorithm.

Size	**Heuristic**		**Simulation**	
(I,S,O)	**CPU time (s)**	**Dyn/Md Gap (%)**	**CPU time (s)**	**Rob′ (%)**
(3,3,20)	27.8	3.8	54.7	99.3
(3,3,50)	116.3	4.1	181.4	99.8

The goal is to find a solution able to guarantee optimality for any possible product order. Our results show that the proposed LS algorithm is an effective heuristic capable of solving the problem at hand in a reasonable time. At the same time, the computational results prove that dynamic task assignment policy provides better results compared to model-dependent task assignment.

4.4.2 Managerial insights

This section evaluates the impact of the costs of workers (α) and of the classes of instances on the efficiency of the proposed dynamic task assignment policy compared to the model-dependent and fixed ones proposed in Chapter 3. We evaluate the impact of these instance features in terms of number of workers, the cost of equipment, and equipment duplication in the resulting solutions.

All values reported in this section correspond to the optimal solution values of $MALBP-W^{Md}$ and $MALBP-W^{Fix}$ in Chapter 3, and to the most robust (worst) solutions obtained from the proposed approaches in this chapter (most of these solutions are from the LS algorithm). Note that, the positive values in the columns "Dyn/Fix Gap" and "Md/Fix Gap" are similar to the values in column "Dyn/Md Gap", but they give the percentage of cost saving obtained by using dynamic policy compared to fixed one (calculated by Equation (4.12) with $Cost(MALBP-W^{Fix})$ instead of $Cost(MALBP-W^{Md})$) and the cost saving obtained by using model-dependent policy compared to fixed one (calculated by Equation (3.33) in Chapter 3), respectively.

Tables 4.4 and 4.5 show the superiority of dynamic task assignment compared to the model-dependent and fixed ones, and also the superiority of the model-dependent policy compared to the fixed one, in terms of cost saving. Tables 4.4 shows increasing workers' cost makes the dynamic task assignment more profitable than the model-dependent (from 1.3% to 6.6%) and the fixed (from 1.9% to 9.3%) ones.

Table 4.4 – The impact of worker cost on cost saving via $MALBP-W^{Dyn}$ as compared to $MALBP-W^{Md}$ and $MALBP-W^{Fix}$.

α	**Dyn/Md Gap (%)**	**Dyn/Fix Gap (%)**	**Md/Fix Gap (%)**
50	1.3	1.9	0.5
500	6.6	9.3	2.3

Table 4.5 compares cost saving of $MALBP-W^{Dyn}$ over $MALBP-W^{Md}$, $MALBP-W^{Dyn}$ over $MALBP-W^{Fix}$, and $MALBP-W^{Md}$ over $MALBP-W^{Fix}$ for different classes of instances. For instance, the dynamic task assignment performs better than the model-dependent policy, especially when: 1) the user does not consider restrictions on the number of product model units in the picture of the line, however he/she restricts the order of products to not have more than S successive units of the same product model (4.5% rather than 3.4%); 2) products have different sets of tasks (4.9% rather than 3%), and 3) products have the same precedence graphs (4.6% rather than 3.3%).

Table 4.5 – The impact of different classes of instances on cost saving via $MALBP-W^{Dyn}$ as compared to $MALBP-W^{Md}$ and $MALBP-W^{Fix}$.

MALBP-W	**Dyn/Md Gap (%)**	**Dyn/Fix Gap (%)**	**Md/Fix Gap (%)**
Restricted	3.4	5.7	1.9
Non-restricted	4.5	5.4	1
Same set of tasks	3	5.5	2.3
Different sets of tasks	4.9	5.6	0.6
Same precedence graph	4.6	5.9	0.8
Different precedence graphs	3.3	5.2	2

Table 4.6 shows that increasing the cost of workers increases the cost of equipment and the number of duplications, while the number of required workers decreases. In addition, from both Tables 4.6 and 4.7, we can observe that the number of workers in the worst case of dynamic task assignment is less than the one in model-dependent case.

Table 4.6 – The impact of the worker cost on the equipment cost, number of workers, and equipment duplication.

α	$MALBP-W^{Dyn}$			$MALBP-W^{Md}$		
	Eq. Cost	**Nr. Worker**	**Nr. Dup.**	**Eq. Cost**	**Nr. Worker**	**Nr. Dup.**
50	218.6	8.9	0.1	216.9	9.3	0.1
500	441.2	7.1	0.2	462.5	7.9	0.2

Table 4.7 shows the impact of different classes of instances on the equipment cost, the number of workers and equipment duplications. We can conclude that the assembly line requires less workers but higher cost of equipment with both dynamic and model dependent policies, when they: 1) the number of units of the same product model units in

the line is restricted 2) products have different tasks. However, there is not a significant influence of producing products with the same or different precedence graph on the total number of workers and cost of equipment. On the other hand, companies can hire a less workers with the proposed dynamic task assignment policy.

Table 4.7 – The impact of different classes of instances on the equipment cost, number of workers, and equipment duplication.

MALBP-W	$MALBP-W^{Dyn}$			$MALBP-W^{Md}$		
	Eq. Cost	**Nr. Worker**	**Nr. Dup.**	**Eq. Cost**	**Nr. Worker**	**Nr. Dup.**
Restricted	329.8	7.3	0.2	330.3	7.6	0.2
Non-restricted	330.0	8.7	0.2	349.1	9.6	0.1
Same set of tasks	343.0	9.8	0.3	347.8	10.6	0.2
Different sets of tasks	316.8	6.2	0.1	331.6	6.6	0.1
Same precedence graph	330.3	8.0	0.2	348.9	8.7	0.1
different precedence graphs	329.5	8.0	0.2	330.6	8.6	0.2

4.5 Conclusion

This Chapter focuses on the same problem as in Chapter 3, i.e., the multi-manned manual mixed-model assembly line balancing problem with walking workers ($MALBP-W$). Herein, we study a dynamic task assignment, where tasks can be re-assigned to stations at each takt depending on the entering product model order. The problem is called $MALBP-W^{Dyn}$. Products can enter the line in an arbitrary order. A new scenario-based mixed-integer linear programming (MILP) model is developed with the criterion of minimizing the total cost of workers and equipment. The proposed MILP minimizes the total cost for the worst order of entering product models. Since a huge number of orders of products can be considered, we rely on sampling. A Sequence Generator (SG) is developed to create a set of product orders, and we use a simulation model to evaluate the solutions' robustness level. We try to improve the solutions' robustness by re-solving the problem every time a newly generated product order causes infeasibility. We also propose a local search algorithm designed to find the most robust solution by running through neighboring product orders.

The proposed model can handle various situations that cannot be considered in the classical MMAL balancing model, such as product models with different sets of tasks, with different processing times, and with different precedence relationships among tasks. We evaluate the solution quality and the computation time of the approaches with the same instances as in Chapter 3. The results show that dynamic task assignments provide a lower cost than model-dependent and fixed task assignments.

Following the results of this chapter, we aim to evaluate the same dynamic policy for task assignment in the same environment, but in the case where the order of products is not given. It is a practically useful assumption which reflects the uncertainties related to the market demand and supply from upstream stages of a supply chain. The next chapter studies the proposed problem with such dynamic and uncertain environment. To tackle the problem we build a Markov Decision Process (MDP) model which is, in turn, solved by a mathematical programming approach.

Moreover, several research perspectives can be mentioned as future research directions. First, the ergonomics aspect of the dynamic task assignment presents an important issue. Second, several workforce planning scenarios can be considered. For instance, temporary and utility workers can be employed besides regular walking workers. These scenarios are relevant because the number of workers required on the line fluctuates at each takt. Currently, we work on solving larger instances and providing complementary managerial insights. We are also improving f the proposed solution approaches to develop a heuristic that finds the worst order faster.

Only some initial results of this chapter have been published in Hashemi-Petroodi et al. [2020b].

Chapter 5

Markov decision process for dynamic task assignment in mixed-model assembly lines under uncertainty

5.1 Introduction

Customization increases product satisfaction and purchases likelihood [Kaiser et al., 2017, Moreau and Herd, 2010, Valenzuela et al., 2009]. In many industries, this strategy is no longer just an opportunity to create a competitive advantage but rather a condition of the company's long-term prosperity. The production system used by a company should be able to manufacture different product models without losing high productivity. Assembly lines, due to their flow shop nature, are the most common type of production systems. Dynamic changes in the market and increasing demand for customization push manufacturing companies to use mixed/multi-model assembly lines instead of single model lines [Kucukkoc and Zhang, 2014].

The production sequence has a large impact on the performance of a mixed-model assembly line because the items do not have the same processing time in each station. Typically, the sequence is selected to avoid successive items with long processing times at stations [Miltenburg, 1989] since the line can adjust to such situations with, for example, utility workers. For instance, Renault imposes a certain ratio on items with complex operations in the production sequence [Zufferey, 2016]. Unfortunately, there exist many situations where the production sequence on the assembly line may not be perfectly controllable. For instance, the production sequence may depend on the schedule of an upstream workshop. Items may be removed from the initial sequence because they fail quality test [Boctor et al., 2000]. Urgent orders may be added. Some orders may be removed due to part shortages [Liu et al., 2012]. Some companies face a highly variable demand in a make-to-order production environment [Bukchin et al., 2002].

To overcome this challenge, the line should have a high degree of reconfigurability [Koren et al., 1999]. In other words, an ability to quickly adapt the resources, human operators, and machines, to the incoming product models. In this Chapter, we investigate a mixed-model assembly line able to adjust to any order of entering items subject to given sequencing constraints.

This chapter addresses a multi-manned manual mixed-model assembly line balancing problem with walking workers (MALBP-W). Reconfigurations allow shifting the production capacity from one task to another and keeping the required level of productivity. The studied problem has two stages. The design stage determines the number of workers and assigns equipment to stations for any possible order of products entering the line. At the operational stage, the items entering the line are revealed takt by takt. Tasks and workers are re-assigned depending on the new product and sets of already performed tasks for other products. Taking into account the highly dynamic nature of the studied problem, the choice of the solution method fell on Markov Decision Process (MDP) model due to its ability to tackle uncertainties. In our case, the order of entering products to the line is unknown and subject to frequent changes.

Compared to the majority of studies in the literature, we consider an infinite unknown order of products entering the line, and we aim to design a line that can self-adjust to incoming models. To the best of our knowledge, this work is the first attempt to apply an MDP model to a manual mixed-model assembly line balancing problem. The problem's dynamic nature with possible task re-assignments at each takt, uncertainty regarding the product order represent the sources of motivation behind this study. Two problems with two different criteria are studied. The first one minimizes the expected cost of workers and equipment. The second one minimizes the workforce and equipment cost for the worst takt. As the complexity of the studied problem is extremely high, we enhance the performance of MDP model via several reduction rules.

Application of machine learning approaches to operations research problems is becoming more and more attractive [Karimi-Mamaghan et al., 2022, Kang et al., 2020, Morin et al., 2019, Bengio et al., 2020]. The Markov Decision Process (MDP) is a modelling framework which is commonly used in reinforcement learning as one of the main machine learning paradigms [Bengio et al., 2020]. An MDP represents a sequential decision-making problem under uncertainty. It describes the process of transformation of the system's current state to another state through actions. Recent researches on the applications of MDP and their integration with optimization methods are reviewed in Ahluwalia et al. [2021], Alagoz et al. [2015], Steimle et al. [2021]. Most applications concern transport problems [Yu et al., 2019, Kamrani et al., 2020, Li et al., 2021], and these studies show that mathematical programming methods (e.g., LP, MILP) perform efficiently to solve MDP problems case-dependently [Alagoz et al., 2015]. Moreover, to overcome the higher complexity of large size instances using mathematical programming, exact optimization approaches such as decomposition [Steimle et al., 2021] and branch-and-bound [Ahluwalia et al., 2021] algorithms can perform well to tackle MDPs.

This chapter compares the dynamic to model-dependent and fixed task assignments. In the fixed task assignment, tasks are performed at the same stations for all possible product models. In the model-dependent task assignment studied in Chapter 3, each product model has its own task assignment. Dynamic task assignment means that tasks can be re-assigned at the end of each takt depending on the new product entering the line, the position of other product models on the line and the sets of already performed tasks for existing products. The superiority of model-dependent task assignment over the fixed one has been already revealed in Chapter 3. The goal of this chapter consists not only in assessing the efficiency of dynamic task assignment represented by an MDP model. We also seek to compare dynamic task assignment to model-dependent and fixed task assignments, similarly to what we have proposed in Chapter 4.

This chapter is structured as follows: Section 5.2 describes the general problem and provides an illustrative example. Section 5.3 explains the MDP application and presents the stochastic and robust versions of the MDP model. Section 5.4 presents the restricted task assignment policies and the algorithmic improvements of the proposed MDP. Section 5.5 provides the computational results and managerial insights. The chapter ends with the conclusion and future research directions in Section 5.6.

5.2 Problem description

This section describes a multi-manned mixed-model assembly line balancing problem with walking workers ($MALBP-W$). The problem consists in designing a mixed-model manual assembly line. The line includes a set $\mathcal{S} = \{1 \dots S\}$ of sequential stations, and it assembles a set $\mathcal{I} = \{1 \dots I\}$ of product models. The line is paced, and at the end of each takt time C, all products move simultaneously towards the next station. Each product model i requires the set $\mathcal{O}_i$ of tasks, and we denote by $\mathcal{O}$ the unified set of all tasks. The processing time p_{io}^l of task o varies with product model i, and it depends on the number of workers l processing the task. In other words, the task processing time decreases when the number of workers at the station increases [Battaïa et al., 2015]. At most, l_{max} workers are allowed to work at the same station. The assignment of tasks to stations must respect the set A of precedence constraints. More precisely, A contains the pairs of tasks (o, o') if task o must precede task o'. To perform a task o at a station s, at least one equipment piece with the ability to perform o must be installed in s. The set of equipment is denoted by $\mathcal{E}$, and the requirements are represented with the parameter r_{oe}. If r_{oe} is equal to 1, equipment e has the ability to perform to task o, and r_{oe} is equal to 0 otherwise. Each equipment e has a cost c_{se} at each station s. The cost of a worker α is the same for all workers because we assume all workers are identical in terms of their ability to perform any tasks.

The objective is to design a line able to reconfigure to face any possible product model order. We call a reconfiguration the movements of workers and the task re-assignment. The design decisions include the number Y of workers to hire and the positions of fixed equipment. At the operational stage, problem $MALBP-W$ dynamically assigns the tasks and workers to the stations at each takt depending on the line's state. Workers can move from a station to another at the end of each takt to adapt the production capacity at stations before the arrival of an entering product. We assume that the workers' walking times compared to task processing times are negligible compared to the takt and processing times [Battaïa et al., 2015]. Tasks can be assigned to any stations with the required equipment, and equipment can be duplicated to increase the reconfigurability.

As products enter the line, they consecutively occupy workstations creating different "pictures" of the line. As mentioned in Chapter 3, by **picture**, we mean the sequence of pairs station-product model that changes (product items shift towards the last station) every takt. As opposed to the concept of the line's picture, a **product order** can be defined as a sequence of product models entering the line whose number is not limited by the number of stations.

A line able to reconfigure for any possible input order of products requires many workers, since the order may include the items with the largest process duration in all stations. To reduce the cost of the line, the decision-maker may impose constraints on the incoming product orders. In this chapter also, we consider two types of constraints which have been described in Section 4.2, Chapter 4. Note that the proposed solution approach is generally applicable to any constraints that respect the Markov property of the MDP. We recall the two types of constraints. The first one prevents the presence of more than u_i units of a product model i in a picture of the line. The second prevents having more than u'_i successive units of a product model i in the product order. Such restrictions can be caused by producing a relatively rare customized/luxury product model together with more popular mass-production models.

The studied problem $MALBP{-}W$ with dynamic task assignment is called $MALBP{-}W^{Dyn}$. To further clarify the studied problem in this chapter, we provide an illustrative example (the same as the one in Section 3.2.2, Chapter 3). In Chapter 3, Figure 3.1 and Table 3.1 provide the precedence graphs, processing times and the compatibility between equipment and tasks, and the cost of using the equipment at each station. Similarly, the cost of a worker is $\alpha = 500$, the takt time is $C = 25$, and at most $l_{max} = 3$ workers can work at the same station, simultaneously. The line consists of two stations, and assembles two models: A and B which can enter the line in any order as an infinite unknown sequence. A picture of the line is denoted as $(1 - i, ...S - i')$, $i, i' \in \mathcal{I}$. It determines the sequence of pairs station-product model in a certain takt. Thus, the only possible pictures of the line (Pic.) are $(1 - A, 2 - B)$, $(1 - B, 2 - B)$, $(1 - A, 2 - A)$, and $(1 - A, 2 - B)$. Notice that, we consider no restrictions on the picture of the line ($u_A = u_B = S$, where $S = 2$), but the restricted number of the same product model in the whole order of products ($u'_A = u'_B = S$, where $S = 2$). More precisely, we are not allowed to have more than two items from the same model in the whole order, which means that we cannot move from state possessing two of the same product model on the line to the state itself (from $(1 - A, 2 - A)$ to $(1 - A, 2 - A)$ or from $(1 - B, 2 - B)$ to $(1 - B, 2 - B)$).

Figure 5.1 demonstrates an example of the optimal solution for $MALBP{-}W^{Dyn}$ which is the objective of the current study. As one can see, the tasks are re-assigned in each takt considering four possible pictures of the line. Figure 5.1 shows the tasks, equipment and workers assigned to the stations for each picture through some possible takts. It also points out the worst picture(s) marked in red, number of workers, used equipment, and total cost. Since the order of products is unknown, to handle all possible pictures of the line, the optimal solution of $MALBP{-}W^{Dyn}$ requires 4 workers with total equipment and workforce cost of 2424. As expected, dynamic assignment of $MALBP - W^{Dyn}$ provides a solution with less total cost of 2424 compared to 2872 for $MALBP - W^{Md}$ and 3372 for $MALBP - W^{Fix}$ in Figure 3.2, Section 3.2.2, Chapter 3.

We consider two variants of the problem. Both variants seek to minimize the worker

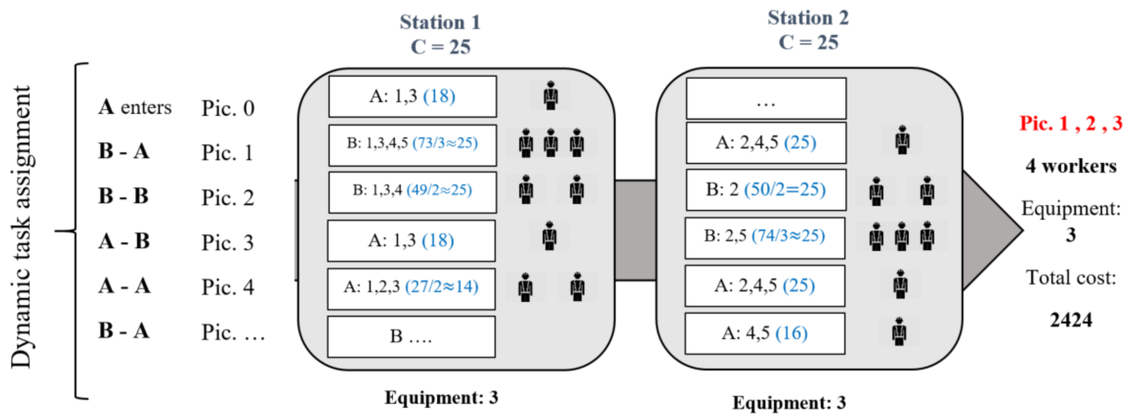

Figure 5.1 – The optimal solution of $MALBP - W^{Dyn}$ in the simple example.

equipment cost, but they compute the workforce cost differently. In the first variant, workers are hired specifically for the assembly line, and the objective is to minimize the number of workers to hire. In the second, the workers may work on other work cells, and the objective is to minimize the expected number of workers.

5.3 Markov Decision Process (MDP) application

This section presents the proposed MDP model application.

5.3.1 Markov Decision Process (MDP) model

Markov Decision Processes (MDPs) model sequential decision-making problems in dynamic and uncertain environements. An MDP includes states, actions, transition matrix, and transition rewards. In each decision step, the system is in a state $d \in \mathcal{D}$, and the agent selects an action a in the set A_d of possible actions in state d. The system switch from a state to another with a probability, and $Tr_a(d \mid d')$ gives the probability to transition from state d to state d' with action a. We present below the state, actions, transition probability matrix, and reward function for the considered problem.

State. Each state $d \in \mathcal{D}$, represents the picture of the line (positions of product models at stations) and sets of tasks that have already been performed for each product model located in the line. The information given by state d is described by two matrices. The first matrix contains values F_{isd} equal to 1 when model i is located at station s in state d, 0 otherwise. The second matrix contains values P_{osd} equal to 1 if task o has already been executed for the model in station s when the system passed to state d. Note that P_{osd} is equal to 0 for all $o \in \mathcal{O}_i$, when product i comes to the first station ($s = 1$) since there are no performed tasks for an item entering the line. The set of states D includes all combinations of all pictures of the lines respecting the restriction with all possible sets of performed tasks respecting the precedence constraints.

Action. At each state d, an action $a \in A_d$ determines a possible task assignment to stations for each product model on the line in the current takt/state. Each action associated with state d is represented by a binary matrix where each cell R_{osa} is equal to 1 if action a performs task o on the model in station s. To create the actions, we consider precedence constraints, takt time and the maximum number of workers per station l_{max}. Given a state d, we consider for each station s, the set Z_{sd} of tasks that are not already performed ($Z_{sd} = \left\{o | o \in \mathcal{O}_{i(s,d)}, P_{osd} = 0\right\}$), where $i(s, d)$ denotes the product model in station s in state d. The set of action corresponds to all combinations (over station 1 to $S - 1$) of subsets of Z_{sd} that respect the precedence constraints. For station S, the set of task to perform is exactly Z_{sd} since the product must be completely assembled in the last station.

Several properties of an action are needed in the solution method. The number of workers needed per station is given by an integer parameter q_s^a. The number q_s^a of workers needed in station s for action a is the smallest integer l such that the total processing time in station s respect the takt time. q_s^a is calculated for each station $s \in \mathcal{S}$ and action $a \in \mathcal{A}_d$, where $d \in \mathcal{D}$, as follows:

$$q_s^a = \min \left\{ l | \sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}_i} p_{io}^l R_{osa} F_{isd} \leq C \right\}, \tag{5.1}$$

Where d is the state associated with action a. If an action requires more than l_{max} workers, it is not considered. We also denote q'_{ad} as the number of workers needed in the line for action a in state d. We calculate q'_{ad} by summing up the number of workers needed at each station q_s^a, as $q'_{ad} = \sum_{s \in \mathcal{S}} q_s^a$ for each $a \in A_d$ and $d \in \mathcal{D}$. The task assignment to stations can be related with the item at the station with a binary parameter y_{soi}^a equal to 1 if task o of product model i is performed at station s within action a, 0 otherwise. y_{soi}^a can be generated considering R_{osa} and F_{isd} ($y_{soi}^a = R_{osa} F_{isd}$) where d is the state associated with action a.

Transition. Transition follows a procedure. Given the current state, the selected action gives the set of additional tasks performed on the items. The item in the last station of the current state leaves the line. All items in middle stations simultaneously move towards the next station, while a new item enters the line and passes to the first station. As several product models may enter the line, the sytem may transition to different states. The transition probability matrix corresponds to the probability of each product model to enter the line. This transition matrix can account for the restriction on the order of products if the restrictions respect the Markov property. In this case, the probability to move to a state that does not respect the maximum number of units constraint must be equal to 0 (such states can simply be removed). Similarly, the probability to move to a state that does not respect the successive number of item constraints must be equal to 0. However, the limit on the number of successive items in the line must be lower or equal to the number of stations to respect the Markov property.

Reward. In a mass production context, the line will run without interruption for a long period. Therefore, we consider an infinite horizon MDP, and we optimize for the long run use of the system.

Solution methods that find the optimal policies for MDPs include linear programming [Alagoz et al., 2015, Buchholz and Scheftelowitsch, 2019], policy iteration algorithm [Patek, 2004, Pavitsos and Kyriakidis, 2009], and value iteration algorithm [Zobel and Scherer, 2005]. While linear programming has been rarely applied to solve MDPs compared to other approaches [Alagoz et al., 2015], the LP solution methods are efficient and flexible enough to account for various constraints on the MDP [Buchholz and Scheftelowitsch, 2019]. The rest of this section provides the linear program to solve the two considered variants of the problem. MDP^{Sto} minimize the long run expected number of workers, and it corresponds to the classical application of MDPs, whereas MDP^{Ro} optimize for the worst possible reachable state in the long run.

5.3.2 Stochastic model MDP^{Sto}

The LP approach to solve the the MDP model relies on continuous variables $0 \leq X_{ad} \leq 1$ that determine the probability of taking an action a in state d. In addition, the variable W_{se} is equal to 1 if equipment e is chosen for station s, and 0 otherwise. The corresponding $MILP^{Sto}$ is represented by 5.2-5.6 as follows:

$$
\begin{aligned}
\min \quad & \sum_{d\in\mathcal{D}} \sum_{a\in A_d} \alpha\, q'_{ad}\, X_{ad} + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} c_{se} W_{se} && (5.2)\\
\text{s.t.} \quad & \\
& \sum_{d\in\mathcal{D}} \sum_{a\in A_d} Tr_a^{dd'} X_{ad} \;=\; \sum_{a'\in A_{d'}} X_{a'd'} && d' \in \mathcal{D} \quad (5.3)\\
& \sum_{d\in\mathcal{D}} \sum_{a\in A_d} X_{ad} \;=\; 1 && (5.4)\\
& y_{soi}^{a}\, X_{ad} \;\le\; \sum_{e\in\mathcal{E}} r_{oe} W_{se} && s\in\mathcal{S},\ o\in\mathcal{O},\ i\in\mathcal{I},\ d\in\mathcal{D},\ a\in A_d \quad (5.5)\\
& W_{se} \in \{0,1\},\ 0 \le X_{ad} \le 1 && (5.6)
\end{aligned}
$$

The objective function (5.2) is the equipment cost and the long run expected cost value of workers needed. Constraints (5.3) verify that total probability to come to a state (d') from any possible state (d) by any action a is equal to the total probability to get out of that state to any other following state. Constraint (5.4) forces the total probability to take actions in the system to be equal to 1. Constraints (5.5) locate necessary equipment at stations regarding the assigned tasks requirements. Constraints (5.6) provide the variables domains.

5.3.3 Robust model MDP^{Ro}

For the robust model MDP^{Ro}, the binary decision variable V_a is added to the model. It is equal to 1 if the action a is taken with non zero probability, and 0 otherwise. In addition, variable Y computes the worst case number of workers. Finally, $MILP^{Ro}$ is given as follows:

$$\min \quad \alpha\, Y + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} c_{se} W_{se} \tag{5.7}$$

s.t.

$$V_a \geq X_{ad} \qquad d \in \mathcal{D},\ a \in A_d \tag{5.8}$$

$$Y \;\geq\; \sum_{s\in\mathcal{S}} q_s^a\, V_a \qquad d \in \mathcal{D},\ a \in A_d \tag{5.9}$$

$$(5.3) - (5.6)$$

$$V_a \in \{0,1\},\ Y \geq 0 \tag{5.10}$$

The objective function (5.7) calculates the total cost of used equipment and the maximum number of workers needed for the worst possible action to move from any state. Constraints (5.8) set variable V_a to 1 if the probability to take action a is positive (non-zero). Constraints (5.9) calculate the maximum number of workers needed in the line for the worst case.

5.3.4 An example of an MDP graph.

Here, an MDP graph for the proposed example in Section 5.2 is given. Figure 5.2 demonstrates the results of the MDP approach for both stochastic and robust models. The information concerning the states and actions are highlighted on the right-side. The probabilities in red and green color give the values of X_{ad} and $Tr_a^{dd'}$ (is calculated by Equation (5.14)), respectively. Similarly, the restrictions on the product order are: $u_A = u_B = S$ and $u'_A = u'_B = S$; $S = 2$. Such restrictions mean that the system is not allowed to move from picture $(1-A, 2-A)$ to picture $(1-A, 2-A)$ or from picture $(1-B, 2-B)$ to picture $(1-B, 2-B)$). The main impact comes from the bottleneck product which has the maximum (with significant difference) tasks' processing time. In this example it is product B. Such restrictions on the whole order of products using u'_i can be justified for a highly customized or luxury product which requires higher processing times compared to the other product models. In this example, even if we only apply this restriction for product B and not for product A ($u_B = 2$), the solution is the same. The total cost of equipment and workers for the worst-case robust model and the expected costs of stochastic model are 2424 and 2122, respectively ($Cost^{Ro} = 2424$, and $Cost^{Sto} = 2122$).

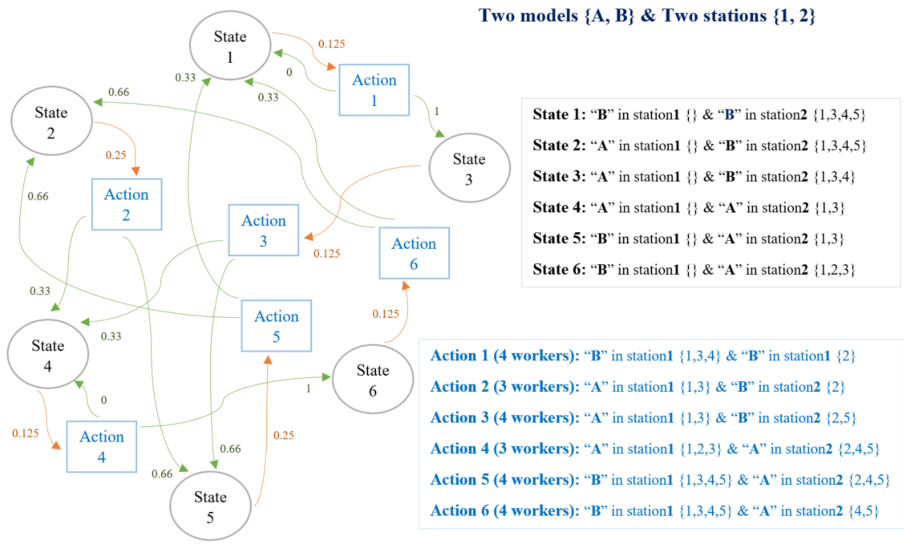

Figure 5.2 – The MDP graph for the presented illustrative example and the optimal solution of MDP^{Sto} and MDP^{Ro}.

5.4 Fixed task assignment policies and algorithmic improvements

This section first presents an extension of the models to yield restricted task assignment policy. In particular, we want to compare the performance of the dynamic task assignment, with policies where tasks are fixed to the station, or fixed for each model. Then, we introduce algorithmic improvements to speed up the computation. Solving the considered model is challenging because the number of states and, especially, actions, grows exponentially with the number of tasks and stations. To alleviate this issue, we propose several reduction rules that eliminate some not-optimal/feasible actions and states (Subsection 5.4.2). Building the transition matrix is the most time and memory consuming part of the pre-processing stage. A decomposition algorithm using sub-matrices related to different pictures of the line is proposed (Subsection 5.4.3). In addition, several actions and states which cannot be reached are eliminated by the algorithm.

5.4.1 Fixed task assignment policy

We aim to compare the solution quality of $MALBP - W^{Dyn}$ with model-dependent $MALBP - W^{Md}$ and fixed task assignment $MALBP - W^{Fix}$, studied in Chapter 3.

In $MALBP - W^{Md}$ task assignment varies for different product models, but for every product of a certain model it remains the same. In $MALBP-W^{Fix}$ task assignment is the same for all product models. To clarify these policies, we refer to the illustrative example in Section 3.2.2, Chapter 3, where the model-dependent and fixed task assignments are drawn compared to the dynamic policy through the same example as in Section 5.2. We provide below the constraint added to the MDP to yield such policies. Note that the resulting model is more generic than the one proposed in Chapter 3 since it allows modeling any constraint on the order of products that respect the Markov property, and it allows to minimize for the expected number of workers.

To restrict the policy to model-dependent ($MALBP - W^{Md}$) and fixed ($MALBP - W^{Fix}$) task assignment, we add a set of constraint on the selected actions. We identify the sets of all pairs of incompatible actions a and a'. These sets are denoted as $\mathcal{A}^{Md}$ and $\mathcal{A}^{Fix}$ for model-dependent and fixed task assignment, respectively. Two actions are incompatible if we do not observe the same task assignment to a station for a certain product model (resp. not observing the same task assignment for all products) in $MALBP-W^{Md}$ (resp. $MALBP-W^{Fix}$). Two sets of constraints (5.11) and (5.12) are added to $MILP^{Sto}$ (5.2) - (5.6) and $MILP^{Ro}$ (5.2) - (5.6) and (5.7) - (5.10) to ensure the resulting policy follows the model dependent and fixed task assignment policy, respectively. Note that, in this context, Constraints (5.8) are also added to $MILP^{Sto}$.

$$V_a \leq 1 - V_{a'} \qquad a, a' \in \mathcal{A}^{Md} \tag{5.11}$$

$$V_a \leq 1 - V_{a'} \qquad a, a' \in \mathcal{A}^{Fix} \tag{5.12}$$

5.4.2 Reduction rules for states and actions

The proposed approach consists of two stages: pre-processing and mathematical modelling, see Figure 5.3. At the pre-processing stage, we create all components of the MDP: states, actions, transition probabilities, and reward function. At the second stage, two $MILP$ models are built for stochastic and robust versions of the problem.

To improve the performance of the proposed MDP method, the number of generated states and, especially, actions must be reduced. Pre-processing accelerates the transition matrix creation, and, subsequently, enhances solving process by MILP. To eliminate states and actions that cannot be present in a feasible solution, we propose the following rules:

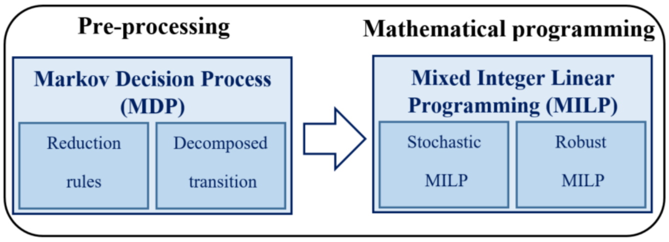

Figure 5.3 – The framework of the proposed methodology.

1. As the transition matrix does not allow "no product" to enter the line, the long run probability for such state is 0. Therefore, we eliminate states and the corresponding actions where only some stations are occupied.

2. Remove states where the total processing times of the performed tasks in previous takts for a product item requires more than $(s-1)$ l_{Max} workers, where s is the current station at which the item is located (l_{Max} per station). Such states are not reachable, since it would require actions with more than l_{Max} workers. This concerns all states d where there exist s such that :

$$\sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}_i} F_{isd} \cdot P_{osd} \cdot p_{io}^{l_{Max}} > (s-1)C.$$

3. Chapter 3 provides an efficient method to solve MDP^{Ro} for the special case of a model-dependent policy without restriction on the number of successive items on the line. The solution to this special case provides an upper bound for the generic version of MDP^{Ro} and MDP^{Sto}. This upper bound is valid for MDP^{Ro} because the solution of the model-dependent policy requires adding constraints (5.11). Therefore the solution to the special case of a model-dependent policy is a feasible but non necessarily optimal solution to MDP^{Ro}. In addition, the special case with less restriction on the order of products leads to larger costs than considering a restricted sequence, since it accounts for a larger uncertainty set. Obviously, MDP^{Ro} provides an upper bound to MDP^{Sto}. We remove the actions which need more than the upper limit on the number of workers in the line. Subsequently, we remove the states with no incoming transition.

5.4.3 Decomposed transition

Since creating the transition matrix demands a lot of computational effort, we propose a decomposition algorithm to make it more efficient. According to this algorithm, the whole transition matrix with all states and actions is decomposed in sub-matrices associated with each possible pictures of the line. Subsequently, the actions related to each subset of states are decomposed into actions corresponding to these stats.

A state d' can be reached from all states d where the items in position 1 to $S-1$ correspond to the items in position 2 to S in d', and where no task performed in d remain to be performed in d'. We denote by $\widehat{\mathcal{D}}_{d'}$ these states that may precede state d. For all the other states, $Tr_a^{dd'}$ is equal to 0. Therefore, Constraints (5.3) can be applied only to preceding states of state d' regarding the line pictures. Constraints 5.3 are reformulated as follows:

$$\sum_{d \in \widehat{\mathcal{D}}_{d'}} \sum_{a \in A_d} Tr_a^{dd'} x_{ad} \; = \sum_{a' \in A_{d'}} x_{a'd'} \qquad d' \in \mathcal{D} \tag{5.13}$$

Rather than generating a transition matrix with many 0, we generate a transition matrix for each set of starting states that correspond to identical sub-pictures from stations 1 to $S-1$. These states are the only ones that may lead to ending states with these sub-pictures in position 2 to S. When generating this transition matrix, we remove ending states that cannot be reached. For instance, reaching some states would require actions with more workers than the upper bound. The procedure iterates until no more state may be removed.

If there are no restrictions on the product model order, the total number of line pictures is equal to I^S. However, the restrictions may significantly reduce the number $Nr.Pic$ of line pictures. For each picture, we independently generate the set of states that correspond to different numbers of performed tasks. This process yields $Nr.Pic$ sets of states, denoted N_k, where $k = 1, \ldots, Nr.Pic$. Subsequently, the states existing in N_k determine $Nr.Pic$ subsets of actions, denoted as M_k where $k = 1, \ldots, Nr.Pic$. Finally, the set of possible resulting states d' characterized by line pictures, task, workforce and equipment assignments is denoted as N'_k where $k = 1, \ldots, Nr.Pic$, that contains the union aggregation of some sets of N_k ($N'_k = \bigcup_{k \in \{1,\ldots,Nr.Pic\}} N_k$). Note that if the picture remains the same from the current state to the resulting state, since the task assignment of the line changes, the system does not move to the same state and it must be removed from the set of resulting states ($N'_k = (\bigcup_{k\in\{1,\ldots,Nr.Pic\}} N_k) - d$).

Note that, we denote $|A0|$ and $|D0|$ as the number of all generated initial actions and states, respectively. Next, $|A|$ actions and $|D|$ states remain after applying reduction rules. Decomposed transition results in a less and final number of states ($|D1|$) and actions ($|A1|$) which enter the proposed $MILPs$.

Following the same simple example as the one current chapter (Section 5.2), we demonstrate the decomposition process through this example. Figure 5.4 demonstrates the performance of the proposed decomposition algorithm. The line is composed of 2 stations and produces 2 product models. Upon the completion of reduction rules, the number of $|D0| = 30$ states and $|A0| = 165$ actions has been reduced to $|D| = 26$ and $|A| = 102$, respectively. They are decomposed into $Nr.Pic = I^S = 2 \times 2 = 4$ subsets corresponding to each possible picture of the line with no restriction. To build the transition matrix the problem is divided into $Nr.Pic = 4$ sub-problems. Figure 5.4 shows all possible line pictures and demonstrates the decomposition process in the context of initial states, actions and resulting states. This example shows that 26 states and 102 actions remain after the reduction process. All 26 states classify into $Nr.Pic = 4$ sub-sets corresponding to each picture of the line, and as well as the same for 102 actions. Knowing that for the resulting states, products move to the following stations, the same correlated states can happen with possible pictures of the line as the resulting ones. For example, when product "A" is at the first station in an initial state ($k = 1, k = 2$), for each one of these two pictures product A goes to the second station for the following resulting state ($k = 2, k = 3$). The set N'_1 is equal to the union aggregation of two sets N_2 and N_3 ($N'_1 = N_2 \bigcup N_3$), whereas the set N'_2 is equal to the union aggregation of the sets N_1 and N_2 except the same state which the line is currently on for this picture since the task assignment of products to stations changes ($N'_2 = (N_2 \bigcup N_3) - d$). It is the same for pictures $k = 3$ and $k = 4$, where product "B" is in the first station ($N'_3 = N_1 \bigcup N_4$, $N'_4 = (N_1 \bigcup N_4) - d$). Finally, since we eliminate some states and actions which have never been reached, $|D1| = 22$ states and $|A1| = 91$ actions will be involved in the whole final transition matrix as the input of $MILPs$.

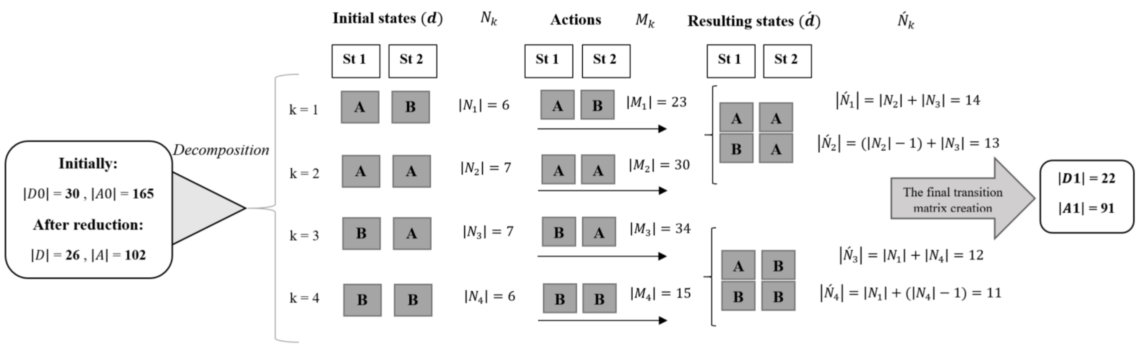

Figure 5.4 – Illustrative example of the decomposition process.

5.5 Computational experiments and results

This section first provides an adaptable approach to generate data for $MALBP-W$ using benchmark data generators from the literature. Second, to evaluate the performance of proposed MDP models and the algorithmic improvements, we conduct extensive computational experiments using the generated instances for MDP^{Ro} and MDP^{Sto}. models. Finally, we formulate managerial recommendations regarding the benefits of using the dynamic task assignment and compare it to model-dependent and fixed task assignments. The problems are solved with IBM ILOG CPLEX Optimization Studio V12.10. The experiments were run on an Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz - 2.11 GHz processor with 32 GB of RAM in MS Windows 10 Pro (64 bit) operational system.

5.5.1 Instances generation

To perform computational experiments, we adapt the data generator proposed by Otto et al. [2013] to the specificity of $MALBP-W$. Each of our instances merges $\mathcal{I}$ consecutive instances of Otto et al. [2013]. For example, our first generated instance contains the data of $\mathcal{I}$ first instances of Otto et al. [2013] and has $\mathcal{I}$ product models with different processing times and precedence graphs. The second instance contains the data of $\{2 \ldots \mathcal{I}{+}1\}$ first instances, and so on. We generate our instances using the set of benchmark instances with 20 tasks from Otto et al. [2013]. Note that product models have different task processing times and precedence relationships. Naturally, our approach can handle the case of products with the same precedence graphs. Equipment costs at each station are generated randomly using a uniform distribution in the range $[100, 300]$. Three different values for workers' salary are considered: less, within, and more than the range for equipment cost ($\alpha = \{50, 200, 500\}$). Two different numbers of stations ($S = \{2, 3\}$) and product models ($I = \{2, 3\}$) are defined. To consider smaller size instances, 8, 10, and 15 tasks among the 20 tasks of the instances from Otto et al. [2013] are selected randomly ($O = \{8, 10, 15\}$). The instances' sizes are determined by the 3-tuple $(\mathcal{I}, \mathcal{S}, \mathcal{O})$, where $\mathcal{I}$, S, and O represent the number of product models, stations, and tasks, respectively. To generate the compatibility matrix, R_{oe} is set to 1 with probability $\frac{\bar{c}_e}{\bar{c}}$ (and 0 otherwise), where $\bar{c}_e$ is the average cost of equipment e (over all stations), and $\bar{c}$ is the average equipment cost (over all equipment and stations). The takt time in the instances of Otto et al. [2013] is set to 1000. In $MALBP - W$, several workers may perform tasks in a station, and the processing time decreases with a higher number of workers. Therefore, we use a lower takt time than the one given in Otto et al. [2013]. Here, according to some initial tests, the takt time takes a value that provides "proper" results, i.e., feasible and with more than one worker per station.

To cover different production situations, several classes of instances are considered. These classes are mainly distinguished based on the set of tasks required for products, the ratio between products' task processing times, the restrictions on the product models order, and the type of transition matrix. The class characteristics are explained below.

Single/different set of tasks for product models: having the same set of tasks O for all product models (class $\#same$) as Otto et al. [2013] raises no issues. To have different sets of tasks O_i for product models (class $\#diff$), we initially set O_i to O , and we randomly eliminate 40% to 60% tasks in each item specific set of task O_i.

Ratio between products' task processing times: the total processing times per product are close to each other, see Otto et al. [2013] (class #1). To differentiate this ratio, we select the product model with maximum total processing time, the so-called "bottleneck product" and denote it as ($i = lux$). The bottleneck product can be a luxury product that is rarely produced in a given production period. In the other two classes, the total processing time for product lux remains fixed, but for all other products, it is either divided by 1.5 (class #1.5) or by 2 (class #2). We consider another class (class $\#diverse$) for instances with three product models ($I = 3$). The total processing time for product lux remains fixed, while it is divided by 1.5 for the product with the second-highest total processing time, and by 2 for the product with the third-highest total processing time.

Restrictions on the number of units of product models: As mentioned, the number of units of an item i in a picture of the line is restricted to u_i, as well as the limit on the number of consecutive units of an item i in the product order is restricted to u'_i. In the non-restricted case (denoted by $\#non - rest$), there is no restriction on the number of successive units of any product models in the order ($u'_i = \infty$ for all $i \in \mathcal{I}$), and there is no restriction on the pictures of the line ($u_i = S$ for all $i \in \mathcal{I}$). In the restricted class $\#rest - 1$, the order cannot have more than S (the number of stations) successive units of the same product model i ($u'_i = S$ for all $i \in \mathcal{I}$), and similarly, $u_i = S$ for all $i \in \mathcal{I}$. In the other restricted class $\#rest - 2$, the order cannot have more than S (the number of stations) successive units of the bottleneck product lux ($u'_{lux} = S$ and $u'_i = \infty$, $\forall$i $\neq lux$), and $u_i = S$ for all $i \in \mathcal{I}$. The class $\#rest - 2$ aims to analyze the impact of the bottleneck product lux compared to the class $\#rest - 2$ where all products are similarly restricted. In the last restricted class $\#rest - 3$, the order cannot have more than $S - 1$ successive units of the bottleneck product lux ($u'_{lux} = S - 1$) and more than S successive units of other products ($u'_i = S$ if $i \neq lux$), and a picture of the line cannot have more than $S - 1$ units of the bottleneck product lux and S units of others ($u_{lux} = S - 1$ and $u_i = S$ if $i \neq lux$).

Type of transition matrix: For all classes, all computational tests are performed over both formulated and random transition matrices which are denoted as $\#not - rand$ and $\#rand$, respectively. In the first case ($\#not-rand$), the transition probability depends on the the number of product model units already located in the line at the current state. Note that this number is limited by u_i which allows adjusting production to the demand. The probability $Tr_a^{dd'}$ of moving from state d to state d' by taking action a is calculated by formula (5.14).Note that the summation of all $Tr_a^{dd'}$ over all product models $i \in \mathcal{I}$ is equal to 1.

$$Tr_a^{dd'} = \{\frac{u_i - \sum_{s=1}^{S-1} F_{isd}}{\sum_{i \in \mathcal{I}} u_i - \sum_{i \in \mathcal{I}} \sum_{s=1}^{S-1} F_{isd}} \mid F_{i1d'} = 1\} \qquad d, d' \in \mathcal{D},\ a \in A_d,\ i \in \mathcal{I} \qquad (5.14)$$

In the second case ($\#rand$), each item i has a given probability r_i to enter the line, see formula (5.15). The values r_i are chosen randomly such that the summation of probabilities over $|I|$ products is equal to 1. In practice, the values r_i may correspond to the demand ratio of products.

$$Tr_a^{dd'} = \{r_i |\ F_{i1d'} = 1, \sum_{i=1}^{I} r_i = 1\} \qquad d, d' \in \mathcal{D},\ a \in A_d \qquad (5.15)$$

Notice that, when there are restrictions on successive units of product models in the order (classes $\#rest - 1$, $\#rest - 2$, and $\#rest - 3$), if the product model i with transition probability $Tr_a^{dd'} \neq 0$ cannot enter the line by moving from state d to state d' through the action a, then $Tr_a^{dd'} = 0$. Therefore, the transition probabilities of other product models $i' \neq i$ entering the line sum-up with $Tr_a^{dd'}/(|I|-1)$ to obtain the sum of probabilities equal to 1. This means that the probability of other product models entering the line increases with the same ratio.

For each size and each class, 5 instances from all instances of Otto et al. [2013] are randomly selected, and it leads to a total number of instances equal to almost 2500.

5.5.2 Analysis of the MDP models

This subsection evaluates the performance of the proposed stochastic and robust MDP models, with reduction rules and decomposition algorithm in terms of solution quality, the number of generated actions and states, and computational time.

Tables 5.1 and 5.2 show how reduction rules and the decomposition algorithm eliminate redundant actions and states. The computational times of pre-processing and MILP solving of $MALBP-W^{Dyn}$ (stochastic and robust) are also given in these tables. Solving $MILPs$ takes more time than for $MALBP-W^{Md}$ and $MALBP-W^{Fix}$, since more constraints and variables are generated which use a lot of memory. Table 5.1 is sorted by different sizes of instances. Table 5.2 is categorized by different classes of instances in terms of products' processing time variation. In these tables, as mentioned, $|A0|$ refers to the number of all generated initial actions, then $|A|$ actions remain after applying reduction rules, and finally $|A1|$ actions import to $MILPs$ for solving the problem after aggregation during the decomposition. Similarly, $|D0|$, $|D|$, and $|D1|$ refer to the number of all initial states, the reduced number of states, and the final number of states importing to $MILPs$, respectively. In both tables, we can see the reduction process and decomposition algorithm decreased the total number of actions and states, effectively. Decomposition becomes more effective when the number of stations and the number of decompositions/pictures ($Nr.Pic$) increase.

Table 5.1 – The number of actions and states, and the computational times based on the size of instances.

Size	**N° actions**			**N° states**			**CPU time (s)**		
(I,S,O)	|A1|	|A|	|A0|	|D1|	|D|	|D0|	Pre-processing	MILP^{Ro}	MILP^{Sto}
(3,2,10)	**856**	950	1852	**79**	102	579	**5.5**	**2.4**	**6.2**
(3,2,15)	**2541**	2603	4677	**125**	161	1317	**68.0**	**3.5**	**82.0**
(2,3,10)	**3753**	5371	6542	**218**	324	960	**754.2**	**319.5**	**457.4**

Table 5.1 shows that the number of states and actions grows exponentially with the

number O of tasks, and they are more sensitive to the number of stations (S) than to the number of product models (I). While the execution time increases with the size of the model, it remains manageable. The main issue with this approach is the time required to build the model and the memory consumption.

Table 5.2 shows that the numbers of actions and states increase with the diversity of the total processing time of products. This behavior is expected since more actions and states are generated with a smaller processing times and the takt time remains fixed.

Table 5.2 – The number of actions and states, and the computational times based on process time variety of products.

	N° actions			**N° states**			**CPU time (s)**		
	|A1|	|A|	|A0|	|D1|	|D|	|D0|	Pre-processing	MILP^{Ro}	MILP^{Sto}
#1	**1010**	1467	2867	**88**	154	850	**21.8**	**12.3**	**27.5**
#1.5	**2687**	3261	4674	**153**	201	956	**327.1**	**43.1**	**110.7**
$\#diverse$	**2770**	3427	4757	**157**	209	993	**357.7**	**50.4**	**118.3**
#2	**3066**	3745	5131	**164**	219	1009	**397.0**	**328.0**	**471.0**

Table 5.3 provides the percentage of cost-saving from the proposed dynamic task assignment compared to the proposed model-dependent and fixed task assignment, as well as the cost-saving from the model-dependent task assignment compared to the fixed one using the robust MDP model. Moreover, this table provides the expected cost saving of $MALBP - W^{Dyn}$ compared to $MALBP - W^{Md}$ using the stochastic MDP model. We report only the gap between dynamic and model-dependent task assignment in the stochastic model as $MALBP - W^{Fix}$ requires too many constraints and its time consuming to solve. The gaps between $MALBP - W^{Md}$ and $MALBP - W^{Fix}$ in MDP^{Sto} (2.54%, 1.68%, and 1.00%) are lower than the gaps found by MDP^{Rob} (4.50%, 3.64%, and 1.93%), since except the worst picture of the line (worst takt), in other pictures almost the same number of workers are required in the line. The analogy of $MALBP - W^{Md}$ and $MALBP - W^{Fix}$ for a given set of products order entering the line has been discussed in detail in Chapter 3.

Table 5.3 – Analogy of the solution quality of fixed, model dependent, and dynamic task assignment.

Size	**MDP^{Ro}**			**MDP^{Sto}**
(I,S,O)	Dyn/Md Gap (%)	Dyn/Fix Gap (%)	Md/Fix Gap (%)	Dyn/Md Gap (%)
(3,2,10)	4.50	5.79	1.58	2.54
(3,2,15)	3.64	6.98	3.60	1.68
(2,3,10)	1.93	2.74	1.01	1.00

5.5.3 Managerial insights

Herein, we evaluate the impact of the costs of workers (α) and the different classes of instances on the cost saving of $MALBP - W^{Dyn}$ compared to $MALBP - W^{Md}$ and $MALBP - W^{Fix}$, as well as on the number of workers and the equipment cost.

Table 5.4 evaluates the cost-saving of $MALBP-W^{Dyn}$ compared to $MALBP-W^{Md}$ and $MALBP - W^{Fix}$ based on the cost of workers (α). The gaps between the problems increase with the cost of workers. For instance, the average gap between dynamic task assignment and model-dependent increases from 3.12% to 5.94% when the cost of workers increases from 50 to 500. This observation was expected because a lower ratio between the cost of workers and equipment leads to a production system with more flexibility to re-assign tasks and workers. As the cost of workers is large compared to the equipment cost, it is desirable to duplicate equipment and re-assign the tasks and workers when needed.

Table 5.4 – The impact of the cost of workers on the solution quality of fixed, model dependent, and dynamic task assignment.

α	$\mathbf{MDP}^{Ro}$			$\mathbf{MDP}^{Sto}$
	Dyn/Md Gap (%)	Dyn/Fix Gap (%)	Md/Fix Gap (%)	Dyn/Md Gap (%)
50	3.12	4.46	1.78	1.68
200	4.89	6.79	2.29	2.40
500	5.94	8.02	2.60	3.13

Table 5.5 shows the influence of different classes of instances on the performance of the $MALBP-W^{Dyn}$ compared to $MALBP-W^{Md}$ and $MALBP-W^{Fix}$. The gap between the problems is large when: 1) products have the same set of tasks (i.e., for Dyn/Md, 3.77% versus 3.02%); 2) product models have processing times with large variance (i.e., for Dyn/Md, 4.97% versus 1.09%); 3) the processing time of a product (a luxury product) is significantly larger than the processing time of other products (i.e., for Dyn/Md, 4.56% versus 3.83%); 4) the user considers restrictions on the number of luxury product model units in the whole the order of products (u'_{lux}) (i.e., for Dyn/Md, 5.48% versus 4.99%); 5) the user considers restrictions on the number of all product models units only in the whole the order of products (u'_i) (i.e., for Dyn/Md, 4.99% versus 3.48%); 6) the user considers restrictions on the number of product models units in the picture of the line (u_i) (i.e., for Dyn/Md, 3.48% versus 0%).

Note that when there is no restriction on the order of products, $MALBP - W^{Dyn}$ provides the same result as $MALBP - W^{Md}$ for both robust and stochastic models since the worst takt corresponds to the situation where the bottleneck product (which has the longest total processing time) is in all stations. When the bottleneck product can be on all stations, the worst-case number of workers is the one computed by the model-dependent task assignment. For the classes of instances with restrictions and process time variations, dynamic task assignment leads to better solutions since the task assignment can be adapted to the product mix on the station at each takt (e.g. when a mix of bottleneck and non-bottleneck products are in stations), whereas model-dependent task assignment forces the user to keep model specified task assignment for all takts.

Table 5.5 – The impact of classes of instances on the solution quality of fixed, model dependent, and dynamic task assignment.

MALBP-W	**MDP^{Ro}**			**MDP^{Sto}**
	Dyn/Md Gap (%)	Dyn/Fix Gap (%)	Md/Fix Gap (%)	Dyn/Md Gap (%)
#same	3.77	6.40	2.59	2.10
#diff	3.02	4.39	1.72	1.96
#1	1.09	3.23	2.21	1.38
#1.5	3.83	5.50	1.93	1.73
#diverse	4.97	7.03	2.53	2.09
#2	4.56	6.35	2.08	3.00
#non-rest	0.00	2.30	2.30	0.00
#rest-1	4.99	6.99	2.30	2.51
#rest-2	5.48	7.01	2.30	3.28
#rest-3	3.48	5.27	2.01	2.33

Tables 5.6 and 5.7 show the impact of the cost of workers on the equipment cost and on the number of workers required for the worst-case (in MDP^{Ro}) and the expected value (in MDP^{Sto}), respectively. Table 5.6 shows that increasing α increases the cost of equipment by duplicating equipment pieces in stations, while the number of required workers decreases. In addition, the number of workers in the worst takt of dynamic task assignment is less than the one in model-dependent and fixed cases, whereas the cost of equipment increases from the fixed task assignment to model-dependent and dynamic cases. To reduce the number of workers, dynamic and model-dependent cases rely on more capable equipment pieces. While these equipment pieces are more expensive, more types of tasks can be re-assigned. A higher cost of workers $(200, 500)$ provides almost the same number of workers for the worst takt (in MDP^{Ro}) which was expected since we could observe almost similar results in our previous work related to the model-dependent and fixed cases proposed in Chapter 3. Note that such an observation might be true with larger size instances.

Table 5.7 shows that the impact of α on the expected number of workers and on the equipment cost is not that considerable. As mentioned earlier, except for the worst takt, the number of workers is almost the same in other takts when we minimize the expected

Table 5.6 – The impact of the cost of workers on the cost of equipment and the number of workers in the robust model MDP^{Ro}.

α	**MALBP-W^{Dyn}**		**MALBP-W^{Md}**		**MALBP-W^{Fix}**	
	Eq. Cost	N° Worker	Eq. Cost	N° Worker	Eq. Cost	N° Worker
50	407.6	5.36	406.8	5.74	407.8	6.01
200	416.0	5.32	413.8	5.68	411.9	5.90
500	416.0	5.32	413.8	5.68	411.9	5.90

total cost of workers and equipment over all the takts. Contrarily to $MILP^{Ro}$, $MILP^{Sto}$ provides lower equipment costs for dynamic task assignment than for the model-dependent one. $MILP^{Sto}$ focuses only on the worst takt, and it invests in expensive equipment to reduce the number of workers in the worst takt. On the contrary, $MILP^{Sto}$ provides almost the same expected number of workers for both dynamic and model dependent task assignments, but lower cost of equipment for the dynamic case.

Table 5.7 – The impact of the cost of workers on the cost of equipment and the number of workers in the stochastic model MDP^{Sto}.

α	**MALBP-W^{Dyn}**		**MALBP-W^{Md}**	
	Eq. Cost	N° Worker	Eq. Cost	N° Worker
50	413.0	4.57	420.4	4.73
200	418.0	4.55	424.8	4.72
500	418.0	4.53	424.8	4.69

Tables 5.8 and 5.9 draw the influence of the classes of instances on the equipment cost and on the number of workers. Manufacturing companies may use fewer workers but face slightly higher costs of equipment, when: 1) products require different sets of tasks, 2) they can impose restrictions on product orders. In addition, dynamic task assignment reduces the number of workers in all classes of instances except when there is no restriction on the number of product models in the line. However, dynamic task assignment leads to large equipment costs. The model-dependent task assignment requires fewer workers compared to the fixed one. In some cases, the equipment cost decreases (besides decreasing the number of workers) in the dynamic task assignments since forcing fixed or model-dependent task assignments may require equipment pieces with more capabilities (e.g., class #1.5). This point is also valid when we compare the model-dependent and fixed cases since the model-dependent case can also benefit from the flexibility to assign the same task to different stations.

Table 5.8 – The impact of the classes of instances on the cost of equipment and the number of workers in the robust model MDP^{Ro}.

	MALBP-W^{Dyn}		**MALBP-W^{Md}**		**MALBP-W^{Fix}**	
	Eq. Cost	N° Worker	Eq. Cost	N° Worker	Eq. Cost	N° Worker
#same	435.6	5.6	433.0	6.0	433.6	6.4
#diff	388.7	5.0	389.9	5.4	389.1	5.5
#1	396.3	6.0	415.6	6.2	423.3	6.4
#1.5	427.7	5.4	416.8	5.7	416.8	6.0
#diverse	386.5	4.6	389.0	4.9	372.0	5.2
#2	429.6	5.1	416.8	5.7	418.5	5.9
#non-rest	411.3	6.1	411.4	5.9	411.3	6.1
#rest-1	412.1	5.5	411.4	5.9	411.3	6.1
#rest-2	412.1	5.2	411.4	5.9	411.3	6.1
#rest-3	412.1	5.2	411.4	5.5	411.3	5.8

Table 5.9 – The impact of the classes of instances on the cost of equipment and the number of workers in the stochastic model MDP^{Sto}.

	MALBP-W^{Dyn}		**MALBP-W^{Md}**	
	Eq. Cost	N° Worker	Eq. Cost	N° Worker
#same	434.5	4.9	434.5	5.1
#diff	398.2	4.2	412.1	4.3
#1	422.8	5.5	424.5	5.6
#1.5	421.9	4.4	428.9	4.6
#diverse	386.5	3.7	391.5	3.9
#2	424.2	4.2	437.8	4.4
#non-rest	414.7	4.9	414.7	4.9
#rest-1	416.3	4.7	414.7	4.9
#rest-2	416.3	4.2	414.7	4.9
#rest-3	417.0	4.4	414.7	4.6

5.6 Conclusion

This study addresses a multi-manned mixed-model assembly line balancing problem with walking workers ($MALBP-W$). We evaluate the impact of dynamic task assignment to stations, where tasks can be re-assigned at the end of each takt depending on the product models located in the line and already performed tasks. Besides, we compare the proposed dynamic task assignment with model dependent and fixed task assignments. In the model dependent task assignment, each product model can have its own task assignment which remains fixed, while the fixed task assignment case assumes the same task assignment for all product models. We assume an infinite unknown sequence of product models entering the line. Two objectives are considered. The first objective is to minimize the expected total cost over all possible takts. The second goal consists in minimizing the total cost in the worst takt.

At the design stage, the number of workers to hire is determined and the required equipment is installed at stations where each equipment can be duplicated if it is needed at other stations. At the operational stage, in the end of each takt, workers can move from one station to another while tasks can be re-assigned to other stations.

In view of the problem's dynamic nature and uncertainty of the products order, a

Markov Decision Process (MDP) model is developed. To the best of our knowledge this approach has never been applied to the mixed-model assembly line balancing problem before. To accelerate the solution process, several reduction rules and a decomposition algorithm have been formulated to solve instances efficiently. After the pre-processing of MDP model, the problem is solved using the proposed stochastic and robust mixed-integer linear programming (MILP) model.

The computational results show that dynamic task assignment leads to larger cost savings compared to model-dependent and fixed task assignments studied in the literature. Extensive managerial insights are given and discussed for different classes of instances that can exist in the real industries. For example, dynamic task assignment performs better in a situation where one or some luxury product models are produced in a limited quantity compared to other common product models. An example can be one or several luxury car model(s) which is(are) produced at a certain time period. However, dynamic task assignment performs identically to the model dependent assignment in the case when total processing times of all products are almost the same and there is no restriction on the number of a product model's units produced uninterruptedly. As dynamic task assignment is efficient even for an unknown sequence of products and the worst takt, we conclude that it would also perform better than model-dependent and fixed assignments in the case of a given product order.

This study is our initial attempt to apply MDP using mathematical programming to a line balancing problem. Our study has the potential to be improved, especially on the methodological side. Recent studies in the literature which integrate MDP and combinatorial optimization approaches show a promising perspective for this research as well. A future work prospect consists in developing an approximate dynamic programming approach to solve large size instances efficiently. The main consideration should be decreasing the number of actions, which causes an out-of-memory problem during the solution. An important feature that can be taken into account in a future research concerns the ergonomic impact of dynamic task assignment on workers: side effects, stress, overload, de-routinization of work tasks etc.

The results of this chapter have been submitted as in Hashemi-Petroodi et al. [2021a].

CONCLUSION

Increased customization and frequent market changes force manufacturing companies to employ multi/mixed-model instead of simple assembly lines. A large number of studies is dedicated to assembly line design and balancing. This thesis compared the profitability of a multi-model line with multiple dedicated lines. Decreasing the setup time for each product model favors a multi-model line as opposed to several simple ones. At the same time, a multi-model line converts to a mixed-model one, if setup times are ignored and if products can enter the line in any order. Motivated by the growing attention of manufacturing companies to mixed-model lines, we studied mixed-model assembly line balancing problem under some challenging and realistic assumptions such as walking workers and task re-assignment policies. The production sequence has a large impact on the performance of a mixed-model assembly line because items do not have the same processing time in each station. Product sequences can be either given, based on the demand and historical data, or unknown. To well adjust the line's capacity to production requirements, the line can benefit from the concept of reconfigurability.

Herein, we considered walking workers who can move between stations when needed to adjust the production capacity for different product orders. This study dealt with three types of task assignment: fixed, model-dependent and dynamic. In the fixed task assignment, the task assignment to stations remains fixed for all product models regardless of the product order. In the model-dependent assignment, task allocating decisions depend on the product model, but for items of the same product model assignment remains the same. The dynamic assignment means that tasks can be re-assigned at each takt depending on the incoming product, existing products at stations and sets of already performed tasks. For dynamic task assignment we considered the cases with known and unknown product orders. Reconfigurability of the line has been achieved by workers' movement and equipment duplication at stations.

This thesis is organized as follows:

An extensive literature review on workforce planning and assembly line balancing problems has been provided in Chapter 1. We have defined different types of production and assembly systems, workforce planning problem, assembly line balancing problem, and relevant concepts used in the thesis like flexibility, reconfigurability, and workforce reconfiguration strategies. At the end of the chapter, we discussed the current state of knowledge in the related literature and discovered several potent future research possibilities. Some of them have been covered within this thesis.

In Chapter 2, we studied a configuration selection problem between a single multi-model line and multiple dedicated lines. The goal of employing any of the two configurations is to maximize the total profit, subject to the product demand and manufacturing time constraints. The selection problem has been reduced to two optimization problems, for one of which a polynomial time algorithm is developed, and NP-hardness is proved for the other. A dynamic programming algorithm, a constructive greedy heuristic, a randomized heuristic and a local search algorithm with steepest ascent hill climbing are presented for the NP-hard problem. Computer experiments with the heuristics, local search algorithm and the solution approach to the corresponding integer linear programming problem using a commercial solver are described. The results demonstrate appropriate quality of the heuristic and local search solutions. The proposed methodology and software can be used to evaluate different input data scenarios while making a selection decision between the two manufacturing configurations. The product demand and selling prices, setup and manufacturing times, demand and production cancellations are the parameters that affect the selection decision. Some future research directions were proposed.

Chapter 3 studied the impact of model-dependent task assignment, workforce reconfiguration, and equipment duplication in mixed-model assembly lines. The studied line is paced, and it can process different product models with different sets of tasks and precedence relations. Task and worker assignments to stations may change in each takt, and the goal is to design a line able to handle a predefined set of situations corresponding to different flows of products entering the line. We provided a new Mixed Integer Linear Programming (MILP) formulation to minimize the workforce and equipment costs in mixed-model assembly lines with model-dependent task assignment. We proposed an efficient reformulation of the MILP by relying on the dualization approach commonly used in robust optimization. In addition, we employed a constructive matheuristic (CM) and a fix-and-optimize heuristic (FOH) to deal with large-scale instances. Extensive computational experiments performed with well-known benchmarks from the literature showed that the suggested approaches perform well in terms of solution quality and computational time. In addition, the results revealed that model-dependent task assignment reduces significantly the equipment cost and the number of workers compared to the classical mixed-model assembly lines with fixed task assignment and walking workers.

Chapter 4 studied the impact of dynamic task assignment, workforce reconfiguration, and equipment duplication in mixed-model assembly lines. The studied line and the problem are the same as in Chapter 3. Task and worker assignments to stations may change in each takt depending on the product model order for the current period. The chapter provided a scenario-based Mixed Integer Linear Programming (MILP) formulation to minimize the workforce and equipment costs in the worst case with dynamic task assignment. Since generating all possible orders of products and solving the proposed MILP for them is time consuming, a sequence generator has been developed to create a set of possible orders of products. A simulation model has been developed to evaluate the level of robustness of the solution provided by MILP. A simulation-based optimization approach has been proposed to obtain the most robust solution compared to the one found by MILP. The approach separately makes design and operational decisions, in which MILP model designs the line while simulation verifies the operational decisions (i.e. workers and tasks assignments) for a given design. Moreover, we developed a local search algorithm to find the most robust solution faster. Several instances from Chapter 3 were solved. The results showed a better cost efficiency of the dynamic task assignment compared to the model-dependent and fixed ones. However, the results rely to some extent on input parameters.

Chapter 5 expanded the problem studied in Chapter 4 to the case with an infinite unknown sequence of products. In addition to the dynamic task assignment at each takt, there is no information on the product model entering the line which creates a highly dynamic and uncertain environment. Due to its ability to handle uncertainties, a Markov Decision Process (MDP) has been applied to model the system. The problem has been addressed using two criteria. The first one reflects a stochastic case in which the expected total cost of workers and equipment over all possible states (takts) is minimized. The second one minimizes the total cost of workers and equipment for the worst state (takt). Two corresponding MILP models have been developed to solve two stochastic and robust problems. The approach was able to solve middle size instances. The future work may concentrate on improving the computational time through an application of approximate methods.

This thesis provided useful results from both practical and theoretical points of view. It investigates the advantages and disadvantages of using either several dedicated or a single multi-model line. It studied the impact of fixed, model-dependent, and dynamic task assignments in different practical situations. The proposed solution approaches integrated the benefits of exact methods' solution quality and fast computational times of heuristics.

Several future research avenues rised from the studied problems. Our results showed that dynamic task assignment strategy and moving workers significantly decrease the cost of a mixed model assembly line, a future research direction is to design efficient optimization approaches (e.g. heuristics or an approximate dynamic programming) to handle large size instances and reduce the computational time. Besides moving workers, other workforce reconfiguration strategies can be used, such as the use of utility and temporary workers. A human-robot collaborative environment is also of great interest and presents a promising research prospective. The aspect of workers' well-being and safety is of highest importance. Therefore, future studies are invited to take ergonomic quality of the line's design and balancing into consideration. Finally, the future research may evaluate the impact of new technologies, such as smart devices, cameras, sensors, teleoperation, message exchange and augmented reality, on assembly line environments.

Publications

Published papers

1. Dolgui A., **Hashemi-Petroodi S. E.**, Kovalev S., & kovalyov M. **(2021)**. Profitability of a multi-model manufacturing line versus multiple dedicated lines. *International Journal of Production Economics, 236, 108113*.

2. **Hashemi-Petroodi S. E.**, Dolgui A., Kovalev S., Y. kovalyov M., & & Thevenin S. **(2020)**. Workforce reconfiguration strategies in manufacturing systems: a state of the art. *International Journal of Production Research, 1-24*.

3. **Hashemi-Petroodi S. E.**, Thevenin S., Kovalev S., & Dolgui A. **(2020)**. Operations management issues in design and control of hybrid human-robot collaborative manufacturing systems: a survey. *Annual Reviews in Control, 49, 264-276*.

Submitted papers

1. **Hashemi-Petroodi S. E.**, Thevenin S., Kovalev S., & Dolgui A. **(2021)**. Model dependent task assignment in multi-manned mixed-model assembly lines with moving workers. *Omega*, **submitted**.

2. **Hashemi-Petroodi S. E.**, Thevenin S., Kovalev S., & Dolgui A. **(2021)**. An application of markov decision process in multi-manned mixed-model assembly lines with walking worker and dynamic task assignment. *International Journal of Production Economics*, **submitted**.

Papers under preparation

1. **Hashemi-Petroodi S. E.**, Thevenin S., Kovalev S., & Dolgui A. **(2021)**. Dynamic task assignment in multi-manned mixed-model assembly lines with moving workers and given set of sequences. *?*, **Prepration**.

2. **Hashemi-Petroodi S. E.**, Thevenin S., Kovalev S., & Dolgui A. **(2021)**. Heuristics to solve markov decision process in mixed-model assembly lines with walking worker and dynamic task assignment. *?*, **Prepration**.

International conference papers

1. **Hashemi-Petroodi S. E.**, Thevenin S., Kovalev S., & Dolgui A. **(2020)**. The Impact of Dynamic Tasks Assignment in Paced Mixed-Model Assembly Line with Moving Workers. *IFIP International Conference on Advances in Production Management Systems, Novi Sad, Serbia, pp. 509-517. Springer, Cham.*

2. **Hashemi-Petroodi S. E.**, Gonnermann C., Paul M., Dolgui A., & Reinhart G. **(2019)**. Decision Support System for Joint Product Design and Reconfiguration of Production Systems. *IFIP International Conference on Advances in Production Management Systems, Austin, Texas, USA, pp. 231-238. Springer, Cham.*

3. Dolgui A., **Hashemi-Petroodi S. E.**, Kovalev S., & kovalyov M. **(2019)**. Workforce planning and assignment in mixed-model assembly lines as a factor of line reconfigurability: state of the art. *IFAC-PapersOnLine, Berlin, Germany, 52(13), 2746-2751.*

Conference Abstract

1. **Hashemi-Petroodi S. E.**, Thevenin S., Kovalev S., & Dolgui A. **(2021)**. Heuristics for fixed and model-dependent task assignments in manual mixed-model assembly lines with walking workers. *31th European Conference on Operational Research (EURO 2021), Athen, Greece.*

Bibliography

Vinayak S Ahluwalia, Lauren N Steimle, and Brian T Denton. Policy-based branch-and-bound for infinite-horizon multi-model markov decision processes. *Computers & Operations Research*, 126:105108, 2021.

Fumio Akagi, Hirokazu Osaki, and Susumu Kikuchi. A method for assembly line balancing with more than one worker in each station. *The International Journal of Production Research*, 21(5):755–770, 1983.

Sener AkpıNar, G Mirac Bayhan, and Adil Baykasoglu. Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks. *Applied Soft Computing*, 13(1):574–589, 2013.

Atiya Al-Zuheri, Lee Luong, and Ke Xing. Prediction and analysis impact of operational design of a manual assembly system with walking workers on performance. *International Journal of Computer Integrated Manufacturing*, 26(6):540–560, 2013.

Atiya Al-Zuheri, Lee Luong, and Ke Xing. Using simulation in verification of a mathematical model for predicting the performance of manual assembly line occupied with flexible workforce. *International Journal of Simulation and Process Modelling*, 9(4): 270–284, 2014.

Atiya Al-Zuheri, Lee Luong, and Ke Xing. Developing a multi-objective genetic optimisation approach for an operational design of a manual mixed-model assembly line with walking workers. *Journal of Intelligent Manufacturing*, 27(5):1049–1065, 2016.

Oguzhan Alagoz, Mehmet US Ayvaci, and Jeffrey T Linderoth. Optimally solving markov decision processes with total expected discounted reward function: Linear programming revisited. *Computers & Industrial Engineering*, 87:311–316, 2015.

Anas Alghazi and Mary E Kurz. Mixed model line balancing with parallel stations, zoning constraints, and ergonomics. *Constraints*, 23(1):123–153, 2018.

Ali Allahverdi, Chi To Ng, TC Edwin Cheng, and Mikhail Y Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032, 2008.

Achraf Ammar, Henri Pierreval, and Sabeur Elkosentini. Workers assignment problems in manufacturing systems: A literature analysis. In *In 5th International conference on industrial engineering and systems management (IESM)*, pages 1–7. IEEE, 2013.

Ann-Louise Andersen, Jesper Kranker Larsen, Thomas Ditlev Brunoe, Kjeld Nielsen, and Christopher Ketelsen. Critical enablers of changeable and reconfigurable manufacturing and their industrial implementation. *Journal of Manufacturing Technology Management*, 2018.

Michel J Anzanello and Flavio Sanson Fogliatto. Learning curve modelling of work assignment in mass customized assembly lines. *International Journal of Production Research*, 45(13):2919–2938, 2007.

Karim Aroui, Gülgün Alpan, and Yannick Frein. Minimising work overload in mixed-model assembly lines with different types of operators: a case study from the truck industry. *International Journal of Production Research*, 55(21):6305–6326, 2017.

Mir Bahador Aryanezhad, V Deljoo, and SMJ Mirzapour Al-e Hashem. Dynamic cell formation and the worker assignment problem: a new model. *The International Journal of Advanced Manufacturing Technology*, 41(3-4):329, 2009.

Sabina Asensio-Cuesta, José Antonio Diego-Mas, Lourdes Canós-Darós, and Carlos Andrés-Romano. A genetic algorithm for the design of job rotation schedules considering ergonomic and competence criteria. *The International Journal of Advanced Manufacturing Technology*, 60(9-12):1161–1174, 2012.

RG Askin and M Zhou. A parallel station heuristic for the mixed-model production line balancing problem. *International Journal of Production Research*, 35(11):3095–3106, 1997.

Ronald Askin and Yuanshu Huang. Employee training and assignment for facility reconfiguration. In *Proceedings of the 1997 6th Annual Industrial Engineering Research Conference, IERC*, pages 426–431. IIE, 1997.

Ronald G Askin. Contributions to the design and analysis of cellular manufacturing systems. *International Journal of Production Research*, 51(23-24):6778–6787, 2013.

Philippe Baecke, Shari De Baets, and Karlien Vanderheyden. Investigating the added value of integrating human judgement into statistical demand forecasting systems. *International Journal of Production Economics*, 191:85–96, 2017.

John J Bartholdi III and Donald D Eisenstein. A production line that balances itself. *Operations Research*, 44(1):21–34, 1996.

John J Bartholdi III, Donald D Eisenstein, and Yun Fong Lim. Deterministic chaos in a model of discrete manufacturing. *Naval Research Logistics*, 56(4):293–299, 2009.

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

Olga Battaïa, Xavier Delorme, Alexandre Dolgui, Johannes Hagemann, Anika Horlemann, Sergey Kovalev, and Sergey Malyutin. Workforce minimization for a mixed-model assembly line in the automotive industry. *International Journal of Production Economics*, 170:489–500, 2015.

Olga Battaïa, Alena Otto, Fabio Sgarbossa, and Erwin Pesch. Future trends in management and operation of assembly systems: from customized assembly systems to cyber-physical systems. *Omega*, 78:1–4, 2018.

D Battini, M Faccio, E Ferrari, A Persona, and F Sgarbossa. Design configuration for a mixed-model assembly system in case of low product demand. *The International Journal of Advanced Manufacturing Technology*, 34(1-2):188–200, 2007.

Joaquín Bautista and Alberto Cano. Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimisation and interruption rules. *European Journal of Operational Research*, 210(3):495–513, 2011.

Ilker 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, feb 2006. doi: 10.1016/j.ejor.2004.07.023.

Christian Becker and Armin Scholl. Balancing assembly lines with variable parallel workplaces: Problem definition and effective solution procedure. *European Journal of Operational Research*, 199(2):359–374, 2009.

Monica Bellgran and Eva Kristina Säfsten. *Production development: design and operation of production systems*. Springer Science & Business Media, 2009.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 2020.

Saif Benjaafar, Sunderesh S Heragu, and Shahrukh A Irani. Next generation factory layouts: research challenges and recent progress. *Interfaces*, 32(6):58–76, 2002.

Alexander Biele and Lars Mönch. Hybrid approaches to optimize mixed-model assembly lines in low-volume manufacturing. *Journal of Heuristics*, 24(1):49–81, 2018.

Diane P Bischak. Performance of a manufacturing module with moving workers. *IIE Transactions*, 28(9):723–733, 1996.

Stefan Bock, Otto Rosenberg, and Thomas van Brackel. Controlling mixed-model assembly lines in real-time by using distributed systems. *European Journal of Operational Research*, 168(3):880–904, 2006.

Fayez Boctor, François Danjou, Vincent Giard, et al. *Rank changes in production/assembly lines: impact and analysis*. Université Paris I, Panthéon-Sorbonne, Institut d'administration des entreprises, 2000.

J.A.C. Bokhorst, J. Slomp, and G.J.C. Gaalman. On the who-rule in dual resource constrained (drc) manufacturing systems. *International Journal of Production Research*, 42(23):5049–5074, 2004.

Marco Bortolini, Francesco Gabriele Galizia, and Cristina Mora. Reconfigurable manufacturing systems: Literature review and research trend. *Journal of Manufacturing Systems*, 49:93–106, 2018.

Marco Bortolini, Emilio Ferrari, Francesco Gabriele Galizia, Cristina Mora, and Francesco Pilati. Optimal redesign of cellular flexible and reconfigurable manufacturing systems. *Procedia CIRP*, 81:1435–1440, 2019.

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

Nils Boysen, Malte Fliedner, and Armin Scholl. Production planning of mixed-model assembly lines: overview and extensions. *Production Planning and Control*, 20(5):455–471, 2009.

Nils Boysen, Mirko Kiel, and Armin Scholl. Sequencing mixed-model assembly lines to minimise the number of work overload situations. *International Journal of Production Research*, 49(16):4735–4760, 2011.

Antoneta Iuliana Bratcu and Alexandre Dolgui. A survey of the self-balancing production lines ("bucket brigades"). *Journal of Intelligent Manufacturing*, 16(2):139–158, 2005.

Peter Buchholz and Dimitri Scheftelowitsch. Light robustness in the optimization of markov decision processes with uncertain parameters. *Computers & Operations Research*, 108:69–81, 2019.

Joseph Bukchin and Jacob Rubinovitz. A weighted approach for assembly line design with station paralleling and equipment selection. *IIE Transactions*, 35(1):73–85, 2003.

Joseph Bukchin and Michal Tzur. Design of flexible assembly line to minimize equipment cost. *IIE Transactions*, 32(7):585–598, 2000.

Joseph Bukchin, Ezey M Dar-El, and Jacob Rubinovitz. Mixed model assembly line design in a make-to-order environment. *Computers & Industrial Engineering*, 41(4):405–421, 2002.

Yossi Bukchin and Ithai Rabinowitch. A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. *European Journal of Operational Research*, 174(1):492–508, 2006.

Nasuh C Buyukkaramikli, J Will M Bertrand, and Henny PG van Ooijen. Flexible hiring in a make to order system with parallel processing units. *Annals of Operations Research*, 209(1):159–178, 2013.

Giovanni Celano, Antonio Costa, Sergio Fichera, and Giovanni Perrone. Human factor policy testing in the sequencing of manual mixed model assembly lines. *Computers & Operations Research*, 31(1):39–59, 2004.

Emre Cevikcan and M Bulent Durmusoglu. Minimising utility work and utility worker transfers for a mixed-model assembly line. *International Journal of Production Research*, 49(24):7293–7314, 2011.

André Chassein and Anika Kinscherff. Complexity of strict robust integer minimum cost flow problems: An overview and further results. *Computers & Operations Research*, 104:228–238, 2019.

Antonio A Chaves, Cristobal Miralles, and Luiz AN Lorena. Clustering search approach for the assembly line worker assignment and balancing problem. In *In 37th International conference on computers and industrial engineering, Alexandria, Egypt*, pages 1469–1478, 2007.

Haoxun Chen. Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, 56:25–36, 2015.

Qin Chen, ShiLong Liao, ZhongZhen Wu, and ShuPing Yi. Comparative analysis of the performance of a novel u-shaped 'chasing-overtaking'production line. *International Journal of Production Research*, 54(12):3677–3690, 2016.

Gyunghyun Choi. A goal programming mixed-model line balancing for processing time and physical workload. *Computers & Industrial Engineering*, 57(1):395–400, 2009.

Jui-Sheng Chou, Yian Tai, and Lian-Ji Chang. Predicting the development cost of tft-lcd manufacturing equipment with artificial intelligence models. *International Journal of Production Economics*, 128(1):339–350, 2010.

Xianghua Chu, Da Gao, Shi Cheng, Lang Wu, Jiansheng Chen, Yuhui Shi, and Quande Qin. Worker assignment with learning-forgetting effect in cellular manufacturing system using adaptive memetic differential search algorithm. *Computers & Industrial Engineering*, 136:381–396, 2019.

Thomas H Cormen, CE Leiserson, RL Rivest, and C Stein. Introduction to algorithms, sec. 16, 2001.

Albert Corominas, Rafael Pastor, and Joan Plans. Balancing assembly line with skilled and unskilled workers. *Omega*, 36(6):1126–1132, 2008.

Pâmela MC Cortez and Alysson M Costa. Sequencing mixed-model assembly lines operating with a heterogeneous workforce. *International Journal of Production Research*, 53(11):3419–3432, 2015.

Con Cronin, Andrew Conway, and Joseph Walsh. Flexible manufacturing systems using iiot in the automotive sector. *Procedia Manufacturing*, 38:1652–1659, 2019.

Michela Dalle Mura and Gino Dini. Worker skills and equipment optimization in assembly line balancing by a genetic approach. *Procedia CIRP*, 44:102–107, 2016.

Quang-Vinh Dang, Thijs van Diessen, Tugce Martagan, and Ivo Adan. A matheuristic for parallel machine scheduling with tool replacements. *European Journal of Operational Research*, 291(2):640–660, 2021.

Darwin J Davis, Hemant V Kher, and Bret J Wagner. Influence of workload imbalances on the need for worker flexibility. *Computers & Industrial Engineering*, 57(1):319–329, 2009.

Philippe De Bruecker, Jorne Van den Bergh, Jeroen Beliën, and Erik Demeulemeester. Workforce planning incorporating skills: State of the art. *European Journal of Operational Research*, 243(1):1–16, 2015.

Alberto De Toni and Stefano Tonchia. Manufacturing flexibility: a literature review. *International Journal of Production Research*, 36(6):1587–1617, 1998.

A Deepak, R Srivatsan, and V Samsingh. A case study on implementation of walking worker assembly line to improve productivity and utilisation of resources in a heavy duty manufacturing industry. *FME Transactions*, 45(4):496–502, 2017.

Xavier Delorme, Alexandre Dolgui, Sergey Kovalev, and Mikhail Y Kovalyov. Minimizing the number of workers in a paced mixed-model assembly line. *European Journal of Operational Research*, 272(1):188–194, 2019.

Andrea Díaz, Simón Fernández, Laura Guerra, and Eleazar Díaz. Manufacturing cost prediction through data mining. In *International Conference of Research Applied to Defense and Security*, pages 251–258. Springer, 2020.

Sotirios G Dimitriadis. Assembly line balancing and group working: A heuristic procedure for workers' groups operating on the same product and workstation. *Computers & Operations Research*, 33(9):2757–2774, 2006.

Alexandre Dolgui and I Ihnatsenka. Branch and bound algorithm for a transfer line design problem: Stations with sequentially activated multi-spindle heads. *European Journal of Operational Research*, 197(3):1119–1132, 2009.

Alexandre Dolgui and Jean-Marie Proth. *Supply chain engineering: useful methods and techniques*. Springer Science & Business Media, 2010.

Alexandre Dolgui, NN Guschinsky, and GM Levin. Graph approach for optimal design of transfer machine with rotary table. *International Journal of Production Research*, 47 (2):321–341, 2009.

Alexandre Dolgui, Sergey Kovalev, Mikhail Y Kovalyov, Sergey Malyutin, and Ameur Soukhal. Optimal workforce assignment to operations of a paced assembly line. *European Journal of Operational Research*, 264(1):200–211, 2018.

Alexandre Dolgui, S Ehsan Hashemi Petroodi, Sergey Kovalev, Mikhail Y Kovalyov, and Simon Thevenin. Workforce planning and assignment in mixed-model assembly lines as a factor of line reconfigurability: state of the art. *IFAC-PapersOnLine*, 52(13):2746–2751, 2019.

Alexandre Dolgui, S Ehsan Hashemi-Petroodi, Sergey Kovalev, and Mikhail Y Kovalyov. Profitability of a multi-model manufacturing line versus multiple dedicated lines. *International Journal of Production Economics*, 236:108113, 2021.

József Dombi, Tamás Jónás, and Zsuzsanna Eszter Tóth. Modeling and long-term forecasting demand in spare parts logistics businesses. *International Journal of Production Economics*, 201:1–17, 2018.

Alan Carl Ebeling and C-Y Lee. Cross-training effectiveness and profitability. *The International Journal of Production Research*, 32(12):2843–2859, 1994.

Stefan Edelkamp and Stefan Schrödl. *Heuristic search: theory and applications*. Elsevier, 2011.

Gokhan Egilmez, Bulent Erenay, and Gürsel A Süer. Stochastic skill-based manpower allocation in a cellular manufacturing system. *Journal of Manufacturing Systems*, 33 (4):578–588, 2014.

Shirine El Zaatari, Mohamed Marei, Weidong Li, and Zahid Usman. Cobot programming for collaborative industrial tasks: An overview. *Robotics and Autonomous Systems*, 116: 162–180, 2019.

Hoda ElMaraghy, Vishvas Patel, and Imed Ben Abdallah. Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. *Journal of Manufacturing Systems*, 19(3):186–201, 2000.

Hoda A ElMaraghy. Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems*, 17(4):261–276, 2005.

Shirley Adriane Elprama, Charlotte IC Jewell, An Jacobs, Ilias El Makrini, and Bram Vanderborght. Attitudes of factory workers towards industrial and collaborative robots. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 113–114, 2017.

Maurizio Faccio, Mauro Gamberi, and Marco Bortolini. Hierarchical approach for paced mixed-model assembly line balancing and sequencing with jolly operators. *International Journal of Production Research*, 54(3):761–777, 2016.

David Francas, Nils Löhndorf, and Stefan Minner. Machine and labor flexibility in manufacturing networks. *International Journal of Production Economics*, 131(1):165–174, 2011.

Elisa Gebennini, Luca Zeppetella, Andrea Grassi, and Bianca Rimini. Optimal job assignment considering operators' walking costs and ergonomic aspects. *International Journal of Production Research*, 56(3):1249–1268, 2018.

Michael A Greig, Judy Village, Filippo A Salustri, Saeed Zolfaghari, and W Patrick Neumann. A tool to predict physical workload and task times from workstation layout design data. *International Journal of Production Research*, 56(16):5306–5323, 2018.

Dávid Gyulai, Botond Kádár, and László Monostori. Scheduling and operator control in reconfigurable assembly systems. *Procedia CIRP*, 63:459–464, 2017.

Natalia Svensson Harari, Anders Fundin, and Anna-Lena Carlsson. Components of the design process of flexible and reconfigurable assembly systems. *Procedia Manufacturing*, 25:549–556, 2018.

S Ehsan Hashemi-Petroodi, Alexandre Dolgui, Sergey Kovalev, Mikhail Y Kovalyov, and Simon Thevenin. Workforce reconfiguration strategies in manufacturing systems: a state of the art. *International Journal of Production Research*, pages 1–24, 2020a.

S Ehsan Hashemi-Petroodi, Simon Thevenin, Sergey Kovalev, and Alexandre Dolgui. The impact of dynamic tasks assignment in paced mixed-model assembly line with moving workers. In *IFIP International Conference on Advances in Production Management Systems*, pages 509–517. Springer, 2020b.

S Ehsan Hashemi-Petroodi, Simon Thevenin, Sergey Kovalev, and Alexandre Dolgui. Operations management issues in design and control of hybrid human-robot collaborative manufacturing systems: a survey. *Annual Reviews in Control*, 2020c.

S Ehsan Hashemi-Petroodi, Simon Thevenin, Sergey Kovalev, and Alexandre Dolgui. An application of markov decision process in multi-manned mixed-model assembly lines with walking worker and dynamic task assignment. *International Journal of Production Economics*, submitted, 2021a.

S Ehsan Hashemi-Petroodi, Simon Thevenin, Sergey Kovalev, and Alexandre Dolgui. Model dependent task assignment in multi-manned mixed-model assembly lines with moving workers. *Omega*, submitted, 2021b.

Stefan Helber and Florian Sahling. A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2): 247–256, 2010.

Wallace J Hopp, Eylem Tekin, and Mark P Van Oyen. Benefits of skill chaining in serial production lines with cross-trained workers. *Management Science*, 50(1):83–98, 2004.

Michael P Hottenstein and Sherry A Bowman. Cross-training and worker flexibility: A review of drc system research. *The Journal of High Technology Management Research*, 9(2):157–174, 1998.

ReaKook Hwang and Hiroshi Katayama. Integrated procedure of balancing and sequencing for mixed-model assembly lines: a multi-objective evolutionary approach. *International Journal of Production Research*, 48(21):6417–6441, 2010.

Robert R Inman, William C Jordan, and Dennis E Blumenfeld. Chained cross-training of assembly line workers. *International Journal of Production Research*, 42(10):1899–1910, 2004.

Adam Janiak and Mikhail Y Kovalyov. Single machine scheduling subject to deadlines and resource dependent processing times. *European Journal of Operational Research*, 94(2):284–291, 1996.

Edward Gil Jones, Brett Browning, M Bernardine Dias, Brenna Argall, Manuela Veloso, and Anthony Stentz. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 570–575. IEEE, 2006.

Ulrike Kaiser, Martin Schreier, and Chris Janiszewski. The self-expressive customization of a product can improve performance. *Journal of Marketing Research*, 54(5):816–831, 2017.

Ikou Kaku, Yasuhiko Murase, and Yong Yin. A study on human-task-related performances in converting conveyor assembly line to cellular manufacturing. *European Journal of Industrial Engineering*, 2(1):17–34, 2008.

Mohsen Kamrani, Aravinda Ramakrishnan Srinivasan, Subhadeep Chakraborty, and Asad J Khattak. Applying markov decision process to understand driving decisions using basic safety messages data. *Transportation Research Part C: Emerging Technologies*, 115:102642, 2020.

Ziqiu Kang, Cagatay Catal, and Bedir Tekinerdogan. Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering*, 149:106773, 2020.

Maryam Karimi-Mamaghan, Mehrdad Mohammadi, Patrick Meyer, Amir Mohammad Karimi-Mamaghan, and El-Ghazali Talbi. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, 296(2):393–422, 2022.

Richard M Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

Talip Kellegöz. Assembly line balancing problems with multi-manned stations: a new mathematical formulation and gantt based heuristic method. *Annals of Operations Research*, 253(1):377–404, 2017.

Hans Kellerer, Ulrich Pferschy, and David Pisinger. Multidimensional knapsack problems. In *Knapsack Problems*, pages 235–283. Springer, 2004.

Daesoo Kim and Won J Lee. Optimal joint pricing and lot sizing with fixed and variable capacity. *European Journal of Operational Research*, 109(1):212–227, 1998.

Dongwook Kim, Dug Hee Moon, and Ilkyeong Moon. Balancing a mixed-model assembly line with unskilled temporary workers: algorithm and case study. *Assembly Automation*, 2018.

Sungsu Kim and David A Nembhard. Rule mining for scheduling cross training with a heterogeneous workforce. *International Journal of Production Research*, 51(8):2281–2300, 2013.

Markus Koppenborg, Peter Nickel, Birgit Naber, Andy Lungfiel, and Michael Huelke. Effects of movement speed and predictability in human–robot collaboration. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 27(4):197–209, 2017.

Yoram Koren, U Heisel, F Jovane, T Moriwaki, G Pritschow, G Ulsoy, and H Van Brussel. Reconfigurable manufacturing systems. *CIRP Annals*, 48:2, 1999.

J Krüger, G Schreck, and D Surdilovic. Dual arm robot for flexible and cooperative assembly. *CIRP Annals*, 60(1):5–8, 2011.

Jörg Krüger, Terje K Lien, and Alexander Verl. Cooperation of human and machines in assembly lines. *CIRP Annals*, 58(2):628–646, 2009.

Mirko Kück and Michael Freitag. Forecasting of customer demands for production planning by local k-nearest neighbor models. *International Journal of Production Economics*, 231:107837, 2021.

Ibrahim Kucukkoc and David Z Zhang. Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. *International Journal of Production Research*, 52(12):3665–3687, 2014.

Tsung-Chyan Lai, Yuri N Sotskov, and Alexandre Dolgui. The stability radius of an optimal line balance with maximum efficiency for a simple assembly line. *European Journal of Operational Research*, 274(2):466–481, 2019.

Jan Christian Lang and Zuo-Jun Max Shen. Fix-and-optimize heuristics for capacitated lot-sizing with sequence-dependent setups and substitutions. *European Journal of Operational Research*, 214(3):595–605, 2011.

Chung-Yee Lee and George L Vairaktarakis. Workforce planning in mixed model assembly systems. *Operations Research*, 45(4):553–567, 1997.

Dong-Kyu Lee, Jeong-Hoon Shin, and Dong-Ho Lee. Operations scheduling for an advanced flexible manufacturing system with multi-fixturing pallets. *Computers & Industrial Engineering*, 144:106496, 2020.

Jinlin Li and Jie Gao. Balancing manual mixed-model assembly lines using overtime work in a demand variation environment. *International Journal of Production Research*, 52 (12):3552–3567, 2014.

Mengyu Li, Alix Carter, Judah Goldstein, Terence Hawco, Jan Jensen, and Peter Vanberkel. Determining ambulance destinations when facing offload delays using a markov decision process. *Omega*, 101:102251, 2021.

Jie Lian, ChenGuang Liu, WenJuan Li, and Yong Yin. A multi-skilled worker assignment problem in seru production systems considering the worker heterogeneity. *Computers & Industrial Engineering*, 118:366–382, 2018.

Yun Fong Lim. Cellular bucket brigades. *Operations Research*, 59(6):1539–1545, 2011.

Yun Fong Lim. Order-picking by cellular bucket brigades: a case study. In *Warehousing in the global supply chain*, pages 71–85. Springer, 2012.

Yun Fong Lim. Performance of cellular bucket brigades with hand-off times. *Production and Operations Management*, 26(10):1915–1923, 2017.

Yun Fong Lim and Yue Wu. Cellular bucket brigades on u-lines with discrete work stations. *Production and Operations Management*, 23(7):1113–1128, 2014.

Shih-Wei Lin and Kuo-Ching Ying. Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics. *Omega*, 64:115–125, 2016.

Chunfeng Liu, Jufeng Wang, Joseph Y-T Leung, and Kai Li. Solving cell formation and task scheduling in cellular manufacturing system by discrete bacteria foraging algorithm. *International Journal of Production Research*, 54(3):923–944, 2016.

WN Liu, LJ Zheng, DH Sun, XY Liao, M Zhao, JM Su, and YX Liu. Rfid-enabled real-time production management system for loncin motorcycle assembly line. *International Journal of Computer Integrated Manufacturing*, 25(1):86–99, 2012.

Thiago Cantos Lopes, Giuliano Vidal Pastre, Adalberto Sato Michels, and Leandro Magatão. Flexible multi-manned assembly line balancing problem: Model, heuristic procedure, and lower bounds for line length minimization. *Omega*, 95:102063, 2020.

Iraj Mahdavi, Amin Aalaei, Mohammad Mahdi Paydar, and Maghsud Solimanpur. Designing a mathematical model for dynamic cellular manufacturing systems considering production planning and worker assignment. *Computers & Mathematics with Applications*, 60(4):1014–1025, 2010.

F Makssoud, O Battaïa, and Alexandre Dolgui. Multi-objective approach and model for transfer line reconfigurations. In *Reconfigurable Manufacturing Systems: From Design to Implementation*, pages 193–208. Springer, 2020.

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

Neda Manavizadeh, Nilufar-sadat Hosseini, Masoud Rabbani, and Fariborz Jolai. A simulated annealing algorithm for a mixed model assembly u-line balancing type-i problem considering human efficiency and just-in-time approach. *Computers & Industrial Engineering*, 64(2):669–685, 2013.

Massimo Manzini, Johannes Unglert, Dávid Gyulai, Marcello Colledani, Juan Manuel Jauregui-Becker, László Monostori, and Marcello Urgo. An integrated framework for design, management and operation of reconfigurable assembly systems. *Omega*, 78: 69–84, 2018.

Silvano Martello and Paolo Toth. Bin-packing problem. *Knapsack Problems: Algorithms and Computer Implementations*, pages 221–245, 1990.

John K McCreery and Lee J Krajewski. Improving performance using workforce flexibility in an assembly environment with learning and forgetting effects. *International Journal of Production Research*, 37(9):2031–2058, 1999.

Mostafa G Mehrabi, A Galip Ulsoy, and Yoram Koren. Reconfigurable manufacturing systems: Key to future manufacturing. *Journal of Intelligent Manufacturing*, 11(4): 403–419, 2000.

Mostafa G Mehrabi, A Galip Ulsoy, Yoram Koren, and Peter Heytler. Trends and perspectives in flexible and reconfigurable manufacturing systems. *Journal of Intelligent Manufacturing*, 13(2):135–146, 2002.

Yaileen M Méndez-Vázquez and David A Nembhard. Worker-cell assignment: The impact of organizational factors on performance in cellular manufacturing systems. *Computers & Industrial Engineering*, 127:1101–1114, 2019.

Adalberto Sato Michels, Thiago Cantos Lopes, Celso Gustavo Stall Sikora, and Leandro Magatão. A benders' decomposition algorithm with combinatorial cuts for the multi-manned assembly line balancing problem. *European Journal of Operational Research*, 278(3):796–808, 2019.

John Miltenburg. Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, 35(2):192–207, 1989.

Cristóbal Miralles, José P García-Sabater, Carlos Andrés, and Manuel Cardós. Branch and bound procedures for solving the assembly line worker assignment and balancing problem: Application to sheltered work centres for disabled. *Discrete Applied Mathematics*, 156(3):352–367, 2008.

C.Page Moreau and Kelly B. Herd. To each his own? how comparisons with others influence consumers' evaluations of their self-designed products. *Journal of Consumer Research*, 36(5):806–819, 2010.

Mayron César de O Moreira and Alysson M Costa. A minimalist yet efficient tabu search algorithm for balancing assembly lines with disabled workers. *Pesquisa Operacional na Gestao do Conhecimento, Porto Seguro*, pages 660–671, 2009.

Mayron César O Moreira, Jean-François Cordeau, Alysson M Costa, and Gilbert Laporte. Robust assembly line balancing with heterogeneous workers. *Computers & Industrial Engineering*, 88:254–263, 2015.

Michael Morin, Jonathan Gaudreault, Edith Brotherton, Frédérik Paradis, Amélie Rolland, Jean Wery, and François Laviolette. Machine learning-based models of sawmills for better wood allocation planning. *International Journal of Production Economics*, 222:107508, 2019.

Hadi Mosadegh, SMT Fatemi Ghomi, and Gürsel A Süer. Stochastic mixed-model assembly line sequencing problem: Mathematical modeling and q-learning based simulated annealing hyper-heuristics. *European Journal of Operational Research*, 282(2):530–544, 2020.

Bahman Naderi, Ahmed Azab, and Katayoun Borooshan. A realistic multi-manned five-sided mixed-model assembly line balancing and scheduling problem with moving workers and limited workspace. *International Journal of Production Research*, 57(3):643–661, 2019.

Koichi Nakade and Rei Nishiwaki. Optimal allocation of heterogeneous workers in a u-shaped production line. *Computers & Industrial Engineering*, 54(3):432–440, 2008.

Koichi Nakade and Katsuhisa Ohno. An optimal worker allocation problem for a u-shaped production line. *International Journal of Production Economics*, 60:353–358, 1999.

Tava Lennon Olsen and Brian Tomlin. Industry 4.0: Opportunities and challenges for operations management. *Manufacturing & Service Operations Management*, 22(1):113–122, 2020.

Alena Otto and Olga Battaïa. Reducing physical ergonomic risks at assembly lines by line balancing and job rotation: A survey. *Computers & Industrial Engineering*, 111:467–480, 2017.

Alena Otto, Christian Otto, and Armin 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.

Uğur Özcan, Hakan Çerçioğlu, Hadi Gökçen, and Bilal Toklu. Balancing and sequencing of parallel mixed-model assembly lines. *International Journal of Production Research*, 48(17):5089–5113, 2010.

Grammatoula Papaioannou and John M Wilson. The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research. *European Journal of Operational Research*, 206(3):509–521, 2010.

Stephen D Patek. Policy iteration type algorithms for recurrent state markov decision processes. *Computers & Operations Research*, 31(14):2333–2347, 2004.

A Pavitsos and EG Kyriakidis. Markov decision models for the optimal maintenance of a production unit with an upstream buffer. *Computers & Operations Research*, 36(6): 1993–2006, 2009.

Jordi Pereira and Eduardo Álvarez-Miranda. An exact approach for the robust assembly line balancing problem. *Omega*, 78:85–98, 2018.

Margherita Peruzzini and Marcello Pellicciari. A framework to design a human-centred adaptive manufacturing system for aging workers. *Advanced Engineering Informatics*, 33:330–349, 2017.

Chris N Potts and Mikhail Y Kovalyov. Scheduling with batching: A review. *European Journal of Operational Research*, 120(2):228–249, 2000.

Markus Pröpster, Lothar März, Gunther Reinhart, and Carsten Intra. Validation of line balancing by simulation of workforce flexibility. *Procedia CIRP*, 33:93–98, 2015.

SM Mizanoor Rahman and Yue Wang. Mutual trust-based subtask allocation for human–robot collaboration in flexible lightweight assembly in manufacturing. *Mechatronics*, 54: 94–109, 2018.

Divakar Rajamani, N Singh, and YP Aneja. Integrated design of cellular manufacturing systems in the presence of alternative process plans. *The International Journal of Production Research*, 28(8):1541–1554, 1990.

Brahim Rekiek, Alexandre Dolgui, Alain Delchambre, and Antoneta Bratcu. State of art of optimization methods for assembly line design. *Annual Reviews in control*, 26(2): 163–174, 2002.

Kaj Rosling. A capacitated single-item lot-size model. *International Journal of Production Economics*, 30:213–219, 1993.

Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

Nima Safaei, Mohammad Saidi-Mehrabad, and MS Jabal-Ameli. A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. *European Journal of Operational Research*, 185(2):563–592, 2008.

Florian Sahling, Lisbeth Buschkühl, Horst Tempelmeier, and Stefan Helber. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, 36(9):2546–2553, 2009.

Ullah Saif, Zailin Guan, Li Zhang, Fei Zhang, Baoxi Wang, and Jahanzaib Mirza. Multi-objective artificial bee colony algorithm for order oriented simultaneous sequencing and balancing of multi-mixed model assembly line. *Journal of Intelligent Manufacturing*, 30(3):1195–1220, 2019.

Parvaneh Samouei and Parviz Fattahi. An algorithm for integrated worker assignment, mixed-model two-sided assembly line balancing and bottleneck analysis. *Journal of Industrial and Systems Engineering*, 11(2):151–174, 2018.

Sule Itir Satoglu and Nallan C Suresh. A goal-programming approach for design of hybrid cellular manufacturing systems in dual resource constrained environments. *Computers & Industrial Engineering*, 56(2):560–575, 2009.

Serpil Sayın and Selçuk Karabatı. Assigning cross-trained workers to departments: A two-stage optimization model to maximize utility and skill improvement. *European Journal of Operational Research*, 176(3):1643–1658, 2007.

Armin Scholl and Christian 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.

George F Schrader and Ahmad K Elshennawy. *Manufacturing processes and materials*. Society of Manufacturing Engineers, 2000.

Scott M Shafer, David A Nembhard, and Mustafa V Uzumeri. The effects of worker learning, forgetting, and heterogeneity on assembly line productivity. *Management Science*, 47(12):1639–1653, 2001.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

Avraham Shttjb. The effect of incompletion cost on line balancing with multiple manning of work stations. *The International Journal of Production Research*, 22(2):235–245, 1984.

Celso Gustavo Stall Sikora, Thiago Cantos Lopes, and Leandro Magatão. Traveling worker assembly line (re) balancing problem: Model, reduction techniques, and real case studies. *European Journal of Operational Research*, 259(3):949–971, 2017.

N Singh. Design of cellular manufacturing systems: an invited review. *European Journal of Operational Research*, 69(3):284–291, 1993.

Jannes Slomp, Jos AC Bokhorst, and Eric Molleman. Cross-training in a cellular manufacturing environment. *Computers & Industrial Engineering*, 48(3):609–624, 2005.

Jing Song, Bi Xi Zhang, and Ying Ying Guan. Study on balance points of bucket brigade lines with return velocities. In *Advanced Materials Research*, volume 314, pages 2273–2277. Trans Tech Publ, 2011.

Majid Soolaki. A multi-objective integrated cellular manufacturing systems design with production planning, worker assignment and dynamic system reconfiguration. *International Journal of Industrial and Systems Engineering*, 12(3):280–300, 2012.

Srinath Sriram, Michael E Kuhl, Brian K Thorn, and Andres L Carrano. A novel worksharing protocol for u-shaped assembly lines. In *Proceedings of the Winter Simulation Conference 2014*, pages 2113–2123. IEEE, 2014.

Lauren N Steimle, Vinayak S Ahluwalia, Charmee Kamdar, and Brian T Denton. Decomposition methods for solving markov decision processes with multiple models of the parameters. *IISE Transactions*, pages 1–58, 2021.

Jeff K Stratman, Aleda V Roth, and Wendell G Gilland. The deployment of temporary production workers in assembly operations: a case study of the hidden costs of learning and forgetting. *Journal of Operations Management*, 21(6):689–707, 2004.

GA Süer and RR Tummaluri. Multi-period operator assignment considering skills, learning and forgetting in labour-intensive cells. *International Journal of Production Research*, 46(2):469–493, 2008.

Bin-qi Sun and Ling Wang. A decomposition-based matheuristic for supply chain network design with assembly line balancing. *Computers & Industrial Engineering*, 131:408–417, 2019.

Banu Sungur and Yasemin Yavuz. Assembly line balancing with hierarchical worker assignment. *Journal of Manufacturing Systems*, 37:290–298, 2015.

F Taube and Stefan Minner. Resequencing mixed-model assembly lines with restoration to customer orders. *Omega*, 78:99–111, 2018.

Atthawit Techawiboonwong, Pisal Yenradee, and Sanchoy K Das. A master scheduling model with skilled and unskilled temporary workers. *International Journal of Production Economics*, 103(2):798–809, 2006.

Eylem Tekin, Wallace J Hopp, and Mark P Van Oyen. Benefits of skill chaining in production lines with cross-trained workers. *Manufacturing & Service Operations Management*, 4(1):17–20, 2002.

Kanajana Thongsanit, Rein Boondisakulchok, and Wipawee Tharmmaphornphilas. Heuristic for task-worker assignment with varying learning slopes. *Engineering Journal*, 14(2):1–14, 2010.

Lorenzo Tiacci and Mario Mimmi. Integrating ergonomic risks evaluation through ocra index and balancing/sequencing decisions for mixed model stochastic asynchronous assembly lines. *Omega*, 78:112–138, 2018.

Panagiota Tsarouchi, Sotiris Makris, and George Chryssolouris. Human–robot interaction review and challenges on task planning and programming. *International Journal of Computer Integrated Manufacturing*, 29(8):916–931, 2016.

Ana Valenzuela, Ravi Dhar, and Florian Zettelmeyer. Contingent response to self-customization procedures: Implications for decision satisfaction and choice. *Journal of Marketing Research*, 46(6):754–763, 2009.

CPM Van Hoesel and Albert Peter Marie Wagelmans. An o (t 3) algorithm for the economic lot-sizing problem with constant capacities. *Management Science*, 42(1):142–150, 1996.

Harvey M Wagner and Thomson M Whitin. Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, 1958.

Junliang Wang, Jungang Yang, Jie Zhang, Xiaoxi Wang, and Wenjun Zhang. Big data driven cycle time parallel prediction for production planning in wafer manufacturing. *Enterprise Information Systems*, 12(6):714–732, 2018.

Qian Wang, Geraint Wyn Owen, and Antony Roy Mileham. Determining numbers of workstations and operators for a linear walking-worker assembly line. *International Journal of Computer Integrated Manufacturing*, 20(1):1–10, 2007.

Qian Wang, Martin Sowden, and Antony R Mileham. Modelling human performance within an automotive engine assembly line. *The International Journal of Advanced Manufacturing Technology*, 68(1-4):141–148, 2013.

Gede Agus Widyadana. Multi objective model for balancing u-type assembly line with permanent and temporary workers. *Jurnal Teknik Industri*, 11(1):33–42, 2009.

JM Wilson. Formulation of a problem involving assembly lines with multiple manning of work stations. *International Journal of Production Research*, 24(1):59–63, 1986.

Robert J Wittrock. Note—operator assignment and the parametric preflow algorithm. *Management Science*, 38(9):1354–1359, 1992.

Lang Wu, Fulin Cai, Li Li, and Xianghua Chu. Cross-trained worker assignment problem in cellular manufacturing system using swarm intelligence metaheuristics. *Mathematical Problems in Engineering*, 2018, 2018.

Zhitao Xu, XG Ming, Maokuan Zheng, Miao Li, Lina He, and Wenyan Song. Cross-trained workers scheduling for field service using improved nsga-ii. *International Journal of Production Research*, 53(4):1255–1272, 2015.

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

Jae Kyu Yoo, Yoshiaki Shimizu, and Rei Hino. A sequencing problem for mixed-model assembly line with the aid of relief-man. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 48(1):15–20, 2005.

Xinlian Yu, Song Gao, Xianbiao Hu, and Hyoshin Park. A markov decision process approach to vacant taxi routing with e-hailing. *Transportation Research Part B: Methodological*, 121:114–134, 2019.

H Yue, J Slomp, E Molleman, and DJ Van Der Zee. Worker flexibility in a parallel dual resource constrained job shop. *International Journal of Production Research*, 46(2): 451–467, 2008.

Beikun Zhang, Liyun Xu, and Jian Zhang. A multi-objective cellular genetic algorithm for energy-oriented balancing and sequencing problem of mixed-model assembly line. *Journal of Cleaner Production*, 244:118845, 2020.

Christopher W Zobel and William T Scherer. An empirical study of policy convergence in markov decision process value iteration. *Computers & Operations Research*, 32(1): 127–142, 2005.

Nicolas Zufferey. Tabu search approaches for two car sequencing problems with smoothing constraints. In *Metaheuristics for Production Systems*, pages 167–190. Springer, 2016.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

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

Titre : Optimisation combinatoire pour l'affectation des opérateurs et la configuration des équipements dans les lignes d'assemblage reconfigurables

Mot clés : Optimisation combinatoire, Équilibrage de lignes d'assemblage, Opérateurs mobiles, Reconfigurabilité, Optimisation robuste, Optimisation stochastique

Résumé : Une forte personnalisation de produits et des fréquentes fluctuations du marché poussent les entreprises industrielles à utiliser des lignes d'assemblage à modèles multiples/mixtes, flexibles et reconfigurables plutôt que des lignes dédiées. Cette thèse de doctorat est consacrée à cette problématique. Elle est portée principalement sur la conception et l'équilibrage de lignes d'assemblage à modèles mixtes. Les questions de l'efficacité de telles lignes, l'importance d'affectation optimale de tâches et d'utilisation des opérateurs mobiles sont posées et étudiées. Pour augmenter la flexibilité de la ligne, nous prenons en compte différents types d'affectation des tâches : fixe, dépendant du modèle et dynamique. Nous cherchons à concevoir une ligne qui peut gérer la variété possible de produits entrants. Nous utilisons des techniques d'optimisation combinatoire, et, en particulier, des approches d'optimisation robuste.

Nous présentons une revue extensive de la littérature sur l'équilibrage des lignes, la planification des opérateurs et les stratégies de reconfiguration de lignes via la réaffectation des opérateurs. Le premier problème de la thèse porte sur la sélection de configuration entre une ligne unique à multi-modèles et plusieurs lignes dédiées. Le second problème consiste à concevoir et à équilibrer une ligne d'assemblage à modèles mixtes avec des opérateurs mobiles. Nous proposons des affectations de tâches fixes et dépendantes du modèle pour une gamme donnée de produits entrants sur la ligne. L'objectif est de minimiser le coût total de la ligne composé des coûts d'opérateurs et d'équipement dans le pire des cas. Le troisième problème est une extension du deuxième, où les tâches peuvent être affectées de manière dynamique. Enfin, dans le dernier, quatrième problème, nous étendons le troisième problème au cas où la séquence de produits est découverte takt par takt. Dans ce contexte, nous minimisons à la fois l'espérance mathématique de coût total et le coût dans le pire des cas.

Afin de résoudre ces problèmes, nous développons plusieurs méthodes exactes et des heuristiques : des modèles de programmation linéaire en variables mixtes, un algorithme glouton, une recherche locale, une matheuristique et une heuristique de type " fixer et optimiser ", entre autres. Nous appliquons également un processus de décision markovien au dernier problème, ce qui représente la première étude qui utilise cette approche pour l'équilibrage de lignes dans la littérature. Des expériences numériques évaluent la performance des modèles proposés en termes de la qualité de solution et du temps de calcul. Nous tirons des conclusions managériales dans chaque chapitre. Nos résultats montrent la supériorité de l'affectation dynamique des tâches par rapport aux affectations fixes et dépendantes de modèle dans différentes situations de production.

Title: Combinatorial optimization for the configuration of workforce and equipment in reconfigurable assembly lines

Keywords: Combinatorial optimization, Assembly line balancing, Walking workers, Reconfigurability, Robust optimization, Stochastic optimization.

Abstract: Mass customization and frequent market fluctuations push industrial companies to employ flexible and reconfigurable multi/mixed-model assembly lines instead of dedicated ones. This thesis focuses on this problem. It concentrates mainly on mixed-model assembly line design and balancing problems. The questions concerning the efficiency of such lines, the importance of optimal task assignment and use of walking workers are asked and studied. To increase the flexibility of the line, we account for different types of task assignments: fixed, model-dependent, and dynamic. We aim to design a line that can handle various entering product models. We use combinatorial optimization methods, and, in particular, robust optimization approaches.

We present an extensive literature review on line balancing, workforce planning, and workforce reconfiguration strategies in different production systems. The first problem addresses a configuration selection problem between a single multi-model line and multiple dedicated lines. The second problem consists in designing and balancing a mixed-model assembly line with walking workers. We propose fixed and model-dependent task assignments for a given set of product mixes. The goal is to minimize the total cost of workers and equipment for the worst case. The third problem extends the second one. It considers the dynamic task assignment. In the last problem, we extend the third problem for the case where the sequence of products unfolds takt by takt. In this context, we minimize both the expected total cost and the worst-case cost.

In order to solve the considered problems, we develop several exact methods and heuristics: mixed-integer linear programming models, greedy algorithm, local search, mat-heuristic and fixed-and-optimize heuristics among others. We also apply a Markov Decision Process to the proposed line balancing problem in the last chapter. It is the first study applying this method to a line balancing problem. Computational experiments evaluate the performance of the proposed approaches in terms of solution quality and time consumption. We draw managerial insights in each chapter. Our results show the superiority of the dynamic task assignment compared to model-dependent and fixed ones in different production situations.