HAL
open science

Apport de l'optimisation combinatoire pour la reconfiguration des lignes de production

Fatme Makssoud

To cite this version:

Fatme Makssoud. Apport de l'optimisation combinatoire pour la reconfiguration des lignes de production. Autre. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2014. Français. NNT : 2014EMSE0742 . tel-01084135

HAL Id: tel-01084135

https://theses.hal.science/tel-01084135

Submitted on 18 Nov 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : XXXXX

T H È S E

présentée par

Fatme MAKSSOUD

pour obtenir le grade de

Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : Génie Industriel

Apport de l'optimisation combinatoire pour la reconfiguration des lignes de production

la soutenance est prévue le 20 mai 2014

Membres du jury

Rapporteur	Lyes BENYOUCEF	Professeur à l'Université Aix Marseille
Rapporteur	Farouk YALAOUI	Professeur à l'Université de Technologie de Troyes
Examinateur	Jean-Pierre CAMPAGNE	Professeur à l'INSA de Lyon
Examinateur	Michel GOURGAND	Professeur à l'Université Blaise Pascal
Examinateur	Ameur SOUKHAL	Professeur à l'université François Rabelais, Tours
Directeur	Alexandre DOLGUI	Professeur à l'École des Mines de St-Étienne
Co-encadrante	Olga BATTAIA	Chargé de recherche, École des Mines de St-Étienne

Spécialités doctorales :	Responsables :
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche
MECANIQUE ET INGENIERIE	S. Drapier, professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche
MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
INFORMATIQUE	O. Boissier, Professeur
IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
GENIE INDUSTRIEL	A. Dolgui, Professeur
MICROELECTRONIQUE	

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BENABEN	Patrick	PR1	Sciences et génie des matériaux	CMP
BERNACHE-ASSOLLANT	Didier	PR0	Génie des Procédés	CIS
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR1	Informatique	FAYOL
BORBELY	Andras	MR(DR2)	Sciences et génie de l'environnement	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BURLAT	Patrick	PR2	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DARRIEULAT	Michel	IGM	Sciences et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR1	Sciences et génie des matériaux	SMS
DESRAYAUD	Christophe	PR2	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FEILLET	Dominique	PR2	Génie Industriel	CMP
FOREST	Bernard	PR1	Sciences et génie des matériaux	CIS
FORMISYN	Pascal	PR0	Sciences et génie de l'environnement	DIR
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GIRARDOT	Jean-jacques	MR(DR2)	Informatique	FAYOL
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HAN	Woo-Suck	CR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
INAL	Karim	PR2	Microélectronique	CMP
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Franck	DR	Sciences et génie des matériaux	SMS
PERIER-CAMBY	Laurent	PR2	Génie des Procédés	DFG
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	CR	Génie des Procédés	CIS
ROUSTANT	Olivier	MA(MDC)		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
SZAFNICKI	Konrad	MR(DR2)	Sciences et génie de l'environnement	CMP
TRIA	Assia		Microélectronique	CMP
VALDIVIESO	François	MA(MDC)	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	MR(DR2)	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR0	Génie industriel	CIS

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	MCF	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	MCF	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

PR 0	Professeur classe exceptionnelle
PR 1	Professeur 1ère classe
PR 2	Professeur 2ème classe
PU	Professeur des Universités
MA (MDC)	Maître assistant
DR	Directeur de recherche

Ing.	Ingénieur
MCF	Maître de conférences
MR (DR2)	Maître de recherche
CR	Chargé de recherche
EC	Enseignant-chercheur
IGM	Ingénieur général des mines

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
FAYOL	Institut Henri Fayol
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

Mise à jour : 07/04/2013

Remerciements

Les remerciements constituent une partie très importante puisqu'elle est dédiée à toutes les personnes qui m'ont permis d'arriver où je suis et parce qu'elle symbolise également la fin du travail de thèse.

Pour ne remonter qu'à des temps raisonnables, un très grand merci à mes deux encadrants de thèse, Mme Olga Battaia et Mr Alexandre Dolgui, pour m'avoir fortement encouragée à poursuivre dans cette voie et choisie pour ce sujet de thèse.

Comme mes directeurs de thèse ont une grande place dans mon coeur, je tiens également à leur témoigner toute ma gratitude pour leur aide, leurs conseils, leur soutien, et ceci aussi bien sur le plan professionnel que personnel. Grâce à eux, j'ai pu rencontrer et travailler avec des chercheurs de différents pays, ce qui constitue un atout et une chance exceptionnelle. Je leur en suis particulièrement reconnaissante.

Je tiens bien évidemment à exprimer ma reconnaissance aux membres du jury : Monsieur Lyes Benyoucef et Monsieur Farouk Yalaoui mes rapporteurs, pour avoir pris le temps de lire ce mémoire dans les moindres détails et juger ce travail. Merci à Jean-Pierre Campagne, président du jury, pour sa confiance et son estime ; Merci également à Monsieur Michel Gourgand et Monsieur Ameur Soukhal, pour avoir accepter d'examiner mon mémoire et de faire partie de mon jury de thèse.

Petite dédicace aux thésards qui ont partagé mon quotidien ou de simples moments : mes amis Bissan et sa famille, Amro et sa famille, Lounes, Evgeny, Sonia, Malik, Oussama, Guillaume, John, Hassan, Hussein et l'équipe ISCOD.

Merci à mon fils Ali, pour avoir été l'ange parfait. Un chaleureux merci à mon mari, qui a toujours été là pour moi tout au long de mes études. Merci de m'avoir encouragée et d'avoir toujours cru en moi. Pour finir, je dis un grand merci à mes parents et à ma famille pour leur soutien, la fierté et la confiance qu'ils me portent.

Table des matières

Liste des figures vii
Liste des tableaux ix
Introduction générale 1

1 Systèmes d'usinage et d'assemblage **3**
1.1 Systèmes de production : notions de base 3
1.1.1 Systèmes d'usinage 5
1.1.2 Lignes d'assemblage 9
1.2 Conception préliminaire des lignes de production 12
1.3 Optimisation de la configuration des lignes de production 13
1.4 Équilibrage des lignes d'assemblage 14
1.5 Équilibrage des lignes d'usinage 16
1.5.1 Équilibrage avec choix des équipements 18
1.5.2 Équilibrage avec choix des modèles de machines 19
1.5.3 Équilibrage avec conception des boîtiers multibroches 20
1.5.4 Équilibrage avec conception de machines spéciales 21
1.6 Conclusion 24

2 Approches de l'optimisation combinatoire pour la configuration des systèmes de production **25**
2.1 Optimisation : notions de base 25
2.2 Méthodes exactes mono-objectif 26
2.2.1 Programmation linéaire en nombres entiers (PLNE) ou en variables mixtes (MIP) 27
2.2.2 Procédures par Séparation et Évaluation (PSE) 32
2.2.3 Programmation Dynamique (PD) 34
2.3 Méthodes d'optimisation approchées 35
2.3.1 Heuristiques 35
2.3.2 Méta-heuristiques 36
2.4 Optimisation multi-objectif 38
2.4.1 Approches a priori 40
2.4.2 Approches a posteriori 43
2.4.3 Approches progressives 44
2.5 Conclusion 45

3 Reconfiguration des lignes de production **47**
3.1 Reconfiguration dans le cycle de vie des lignes de production 47
3.2 Évaluation de la reconfiguration des systèmes de production 48
3.3 Reconfiguration des lignes de transfert 50
3.3.1 Réutilisabilité des boîtiers multibroches 51
3.3.2 Réutilisabilité des machines 52
3.4 Reconfiguration des lignes d'assemblage 52
3.5 Conclusion 54

4 Modèles MIP pour la reconfiguration des lignes de transfert **57**
4.1 Formulation du problème 58
4.1.1 Contraintes d'affectation des opérations 61
4.1.2 Contraintes de productivité 62
4.1.3 Fonction objectif 62
4.1.4 Modèle mathématique 63
4.1.5 Contraintes de reconfiguration 64
4.2 Exemple illustratif 65
4.2.1 Solution pour le modèle MIP1 71
4.2.2 Solution pour le modèle MIP2 71
4.3 Expérimentations numériques 73
4.4 Conclusion 82

5 Utilisation de Goal Programming pour la reconfiguration des lignes de transfert **83**
5.1 Méthode de goal programming : notions de base 83
5.2 Modélisation du problème de reconfiguration de lignes de transfert à boîtiers multibroches 84
5.2.1 Contraintes des buts à atteindre et fonction objectif 84
5.3 Exemple illustratif 85
5.4 Expérimentations numériques 89
5.5 Conclusion 104

6 Une méthode exacte pour le problème de rééquilibrage des lignes d'assemblage **107**
6.1 Définition du problème 108
6.2 Modèle mathématique 109
6.3 Linéarisation du modèle 110
6.4 Exemple illustratif 112
6.5 Expérimentations numériques 114
6.6 Conclusion 118

Conclusion générale **121**

Bibliographie **124**

Annexe **147**

Table des figures

1.1 Différents types de systèmes d'usinage 6
1.2 Stations de travail (PCI-SCEMM) 10
1.3 Boîtier multibroche 10
1.4 Ligne d'assemblage 11
1.5 Ligne en forme de U 12
1.6 Ligne d'usinage à machines parallèles 19
1.7 Ligne de transfert 20
1.8 Machine à table mobile 23
1.9 Machine à table rotative circulaire 23

2.1 Concepts de l'optimisation : modélisation et résolution 26
2.2 Optimisation mono-objectif : approches exactes 27
2.3 Optimisation mono-objectif : méthodes approchées 35
2.4 Optimisation multi-objectif 39

3.1 Processus de configuration/reconfiguration des lignes de production 48
3.2 Reconfiguration physique, logique et humaine 49
3.3 Coûts de production vs coûts de reconfiguration 53

4.1 Pièce initiale à usiner 66
4.2 Graphe initial de précédence et configuration initiale de la ligne . . 68
4.3 Pièce après modification 69
4.4 Graphe de précédence après modification 70
4.5 Solutions optimales de reconfiguration (modèle 1 & 2) 72

5.1 Graphe représentant les contraintes de précédence 85
5.2 Pièce initiale à usiner 86
5.3 Solution optimale pour la ligne initiale 87
5.4 Nouvelle pièce à usiner 87
5.5 Nouveau graphe de précédence 88
5.6 Solution optimale pour le nouveau problème 89
5.7 Comparaison au regard du critère du nombre de stations 101
5.8 Comparaison au regard du critère du nombre de blocs 102
5.9 Comparaison au regard du critère du nombre de blocs réutilisés . . 103

6.1 Graphe de précédence 112
6.2 Ligne initiale 112
6.3 Diagramme de précédence après modification 113
6.4 Ligne Rééquilibrée 114

Liste des tableaux

1.1 Questions analysées 7
1.2 Analyse des travaux sur RMS 8
1.3 Fonctions objectifs considérées conjointement 17

4.1 Opérations et leurs paramètres 67
4.2 Ensemble d'opérations N et leurs paramètres après modification . . 69
4.3 Configuration de la ligne initiale 74
4.4 Configuration optimale pour le nouveau produit 75
4.5 Résultats pour la reconfiguration (modèle MIP1) 76
4.6 Bénéfices de la reconfiguration (modèle MIP1) 77
4.7 Bénéfices de la reconfiguration (modèle MIP1) en tenant compte du coût de la ligne initiale 78
4.8 Résultats pour la reconfiguration (modèle MIP2) 79
4.9 Bénéfices de la reconfiguration (modèle MIP2) 80
4.10 Bénéfices de la reconfiguration (modèle MIP2) en tenant compte du coût de la ligne initiale 81

5.1 Paramètres d'opérations. 86
5.2 Les opérations et leurs paramètres 88
5.3 Résultats pour GP ($w_1 = 0{,}1$ et $w_2 = 0{,}9$) 90
5.4 Résultats pour GP ($w_1 = 0{,}2$ et $w_2 = 0{,}8$) 91
5.5 Résultats pour GP ($w_1 = 0{,}3$ et $w_2 = 0{,}7$) 92
5.6 Résultats pour GP ($w_1 = 0{,}4$ et $w_2 = 0{,}6$) 93
5.7 Résultats pour GP ($w_1 = 0{,}5$ et $w_2 = 0{,}5$) 94
5.8 Résultats pour GP ($w_1 = 0{,}6$ et $w_2 = 0{,}4$) 95
5.9 Résultats pour GP ($w_1 = 0{,}7$ et $w_2 = 0{,}3$) 96
5.10 Résultats pour GP ($w_1 = 0{,}8$ et $w_2 = 0{,}2$) 97
5.11 Résultats pour GP ($w_1 = 0{,}9$ et $w_2 = 0{,}1$) 98
5.12 Comparaison entre MIP et GP 99
5.13 Comparaison entre MIP et GP 100

6.1 Ensemble d'opérations initial et les temps d'exécution 113
6.2 Ensemble d'opérations V et les temps d'exécution 114
6.3 Résultats pour le rééquilibrage : 25% des opérations modifiées . . . 115
6.4 Résultats pour le rééquilibrage : 50% des opérations modifiées . . . 116
6.5 Résultats pour le rééquilibrage : 75% des opérations modifiées . . . 117

Introduction générale

Actuellement, les fabricants sont soumis à une pression économique importante et à une concurrence internationale accrue due à la globalisation des marchés. Pour rester compétitifs, ils sont contraints d'innover et d'améliorer le cycle d'élaboration de leurs produits de façon permanente. Le défi consiste non seulement de développer des produits selon les besoins des clients mais aussi d'être en mesure de les mettre sur le bon marché au bon moment. Pour une bonne réussite sur le marché, les fabricants doivent être capables de répondre rapidement aux changements de la demande en adaptant leurs systèmes de production. Cette adaptation aux changements peut être réalisée à travers multiples reconfigurations du système de production.

Les travaux présentés dans ce mémoire portent sur l'élaboration des méthodes de recherche opérationnelle permettant d'accompagner le décideur lors de la reconfiguration d'une ligne de transfert ou d'assemblage. Ce problème apparaît lorsqu'un nouveau produit doit être fabriqué par une ligne existante ou lorsqu'il y a eu des changements dans les caractéristiques du produit. Par conséquent, il devient nécessaire de modifier la configuration du système de production tout en minimisant les coûts induits. Ces coûts sont évalués différemment pour les systèmes automatisés ou manuels. Dans le premier cas, qui correspond au cas des lignes de transfert, pour limiter les investissements, il est souhaitable de réutiliser au maximum les équipements existants à condition que les contraintes techniques et technologiques soient respectées. Dans le cas des lignes manuelles qui sont représentées dans notre étude par les lignes d'assemblage, l'objectif est de minimiser les coûts liés à l'apprentissage des opérateurs causés par la réaffectation de leurs tâches.

Le mémoire est organisé en six chapitres qui décrivent les résultats de nos travaux de recherche.

Nous commençons ce mémoire par présenter dans le chapitre 1 le contexte de notre étude et les systèmes de production concernés. Les définitions, les concepts et les notions de base nécessaires pour une bonne compréhension des différentes parties de ce mémoire y sont également introduits. Nous décrivons également les différents problèmes d'optimisation liés à la configuration des lignes d'usinage et d'assemblage.

Le chapitre 2 présente un panorama des méthodes d'optimisation combinatoire utilisées pour résoudre les problèmes de conception des systèmes de production que nous classons en trois catégories : les méthodes exactes, les méthodes approchées et les méthodes d'aide à la décision multi-objectif. Dans chaque catégorie, les approches les plus répandues sont détaillées.

Le chapitre 3 décrit l'état de l'art dans le domaine de la reconfiguration des systèmes de production. Nous positionnons notamment le problème de la reconfiguration dans le cycle de vie d'un système de production et nous le comparons avec les problèmes d'optimisation qui se posent lors de la conception initiale du système. Les travaux existants dans la littérature sont analysés et le manque des méthodes exactes d'optimisation combinatoire pour la minimisation des coûts de la reconfiguration des lignes de transfert et des lignes d'assemblage est mis en évidence.

Dans le chapitre 4, nous présentons deux modèles de programmation linéaire en variables mixtes pour la minimisation des coûts de la reconfiguration des lignes de transfert. Nous illustrons les contraintes de chaque modèle sur un exemple industriel. Les résultats d'une étude expérimentale sont également analysés pour comparer leurs performances. Le chapitre se termine par une comparaison de l'approche proposée avec la conception d'un nouveau système de production sans réutilisation des équipements existants. Les bénéfices de la reconfiguration sont ainsi mis en évidence.

Le chapitre 5 propose une autre formulation du problème de la reconfiguration des lignes d'usinage s'appuyant sur la programmation par but (Goal Programming ou GP). Cette méthode est évaluée sur les mêmes instances du problème que dans le chapitre 4 et les résultats obtenus par les deux méthodes sont comparés.

Dans le chapitre 6, nous développons un modèle linéaire en variables mixtes pour la reconfiguration des lignes d'assemblage manuelles. Un exemple industriel est utilisé pour illustrer l'ensemble des données et des contraintes. Les résultats d'une étude expérimentale sont analysés pour évaluer l'approche proposée.

Finalement, nous terminons ce mémoire par la conclusion générale en présentant un bilan final de notre travail et en ouvrant quelques perspectives de recherche.

Chapitre 1

Systèmes d'usinage et d'assemblage

Sommaire

1.1 Systèmes de production : notions de base 3
1.1.1 Systèmes d'usinage 5
1.1.2 Lignes d'assemblage 9
1.2 Conception préliminaire des lignes de production 12
1.3 Optimisation de la configuration des lignes de production 13
1.4 Équilibrage des lignes d'assemblage 14
1.5 Équilibrage des lignes d'usinage 16
1.5.1 Équilibrage avec choix des équipements 18
1.5.2 Équilibrage avec choix des modèles de machines 19
1.5.3 Équilibrage avec conception des boîtiers multibroches 20
1.5.4 Équilibrage avec conception de machines spéciales 21
1.6 Conclusion 24

1.1 Systèmes de production : notions de base

Un système de production manufacturier transforme les objets en d'autres plus élaborés. Dans la première phase de la fabrication du produit, nous trouvons souvent le processus d'usinage. Comme dans l'automobile, il peut constituer une part importante du travail. Les systèmes d'usinage sont utilisés pour enlever progressivement de la matière afin d'obtenir la forme désirée avec une grande précision, à l'aide d'une machine-outil. Le mouvement de l'outil coupant ou celui de la pièce par rapport à l'outil, induit une déformation à un point précis et produit le copeau. Comme la formation du copeau cause une distorsion dans le métal, des procédés de finition (fraisage, alésage, perçage, etc.) sont nécessaires pour rendre la surface plus lisse. Les systèmes d'usinage seront présentés plus en détail dans la sous-section suivante.

Le processus d'assemblage quant à lui permet de passer d'un ensemble de pièces indépendantes les unes des autres à un ou plusieurs produits finis par agrégation. Le

processus d'assemblage est essentiellement composé d'opérations de manipulation et de liaison des pièces entre elles.

Malgré leurs différences, les notions suivantes sont définies de façon similaire pour les systèmes d'usinage ou d'assemblage.

Opération (ou tâche) : une opération est un processus de travail qui n'est pas fractionnable. Dans notre étude, elle est caractérisée par sa durée appelée temps opératoire. Ainsi le travail à réaliser pour fabriquer un produit est modélisé par un ensemble d'opérations à effectuer dans une ligne de production. La nature des temps opératoires est un paramètre important de la modélisation. Trois types de modèles peuvent être définis selon l'évolution du système en fonction du temps :

- **Modèle déterministe** : Les temps d'exécution des opérations déterministes sont justifiés lorsqu'une faible variabilité est observée. C'est ce type de modèle qui sera utilisé dans la suite de cette thèse.

- **Modèle stochastique** : Les temps probabilistes sont utilisés lorsque des variations significatives sont observées. Elles sont causées par des facteurs tels que : des produits présentant des défauts de conception, la motivation, les compétences ou la formation des opérateurs, etc.

- **Modèle dynamique** : les temps d'opération peuvent être réduits, notamment une fois l'apprentissage sur une nouvelle ligne achevé ou grâce aux améliorations successives du procédé de production.

Station de travail (poste de travail) : une station est un élément de la ligne de production qui réalise un ensemble d'opérations. La totalité du travail à effectuer sur une station est appelée la charge de station. Le temps nécessaire pour réaliser ce travail est appelé le temps de station.

Temps de cycle : pour les lignes de production cadencées, considérées dans ce mémoire, il s'agit de la durée qui sépare deux sorties successives de produit. Généralement, le temps de cycle est déterminé par le temps de station le plus long (station goulet) de la ligne. A la fin du travail sur cette station goulet, toutes les pièces se trouvant sur la ligne sont déplacées vers la station suivante. Le temps de cycle détermine la productivité de la ligne de production.

$$T = max\{T_k\}, k = 1, ..., m; \tag{1.1}$$

où T_k est le temps de la station k, m est le nombre de stations installées sur la ligne. Remarquons que le temps de cycle ne doit pas être confondu avec le temps de

réponse ou temps du système, qui lui correspond au temps de présence d'un produit dans la ligne et qui est égal à $T.m$.

Taux de production (cadence) : le taux de production est défini comme le nombre de produits fabriqués par unité de temps, ce taux est égal à la valeur inverse du temps de cycle de la ligne :

$$t_p = 1/T; \tag{1.2}$$

Temps mort : le temps mort d'une station équivaut à la différence entre son temps de travail (le temps de station) et le temps de cycle de la ligne. Le temps mort constitue un gaspillage que l'on cherche à minimiser pour réduire les coûts de production.

Contraintes de précédence : les contraintes de précédence définissent un ordre partiel entre les opérations, imposant qu'une opération ne peut pas être affectée à une station antérieure à celles de ses prédécesseurs. Les contraintes de précédence sont illustrées par un graphe appelé graphe de précédence, où chaque opération est représentée par un nœud et est généralement numérotée topologiquement selon sa position dans le graphe. Ainsi le nombre associé à une opération est toujours plus grand que celui de l'opération précédente. Un arc (i, j) indique la relation de précédence entre les opérations i et j. Si un arc (i, j) existe, l'opération i doit être accomplie pour que la j puisse commencer. L'opération i (resp. opération j) est alors un prédécesseur direct (resp. un successeur direct) de l'opération j (resp. i).

Contraintes d'inclusion : les contraintes d'inclusion reflètent la nécessité d'exécuter un ensemble d'opérations sur la même station [Deckro 1989; Dolgui *et al.* 2006c].

Contraintes d'exclusion : elles sont utilisées pour un ensemble d'opérations dont la réalisation sur la même station de travail est impossible. [Dolgui *et al.* 2008].

1.1.1 Systèmes d'usinage

Comme le montre la Figure 1.1, les systèmes d'usinage peuvent être classifiés en trois groupes selon leur productivité et leur flexibilité.

1.1.1.1 Systèmes d'usinage flexibles

Les systèmes d'usinage flexibles (Flexible Manufacturing Systems ou FMS) sont des systèmes de production fortement automatisés et adaptés à la fabrication en petites et moyennes séries de produits ayant des gammes différentes [MacCarthy &

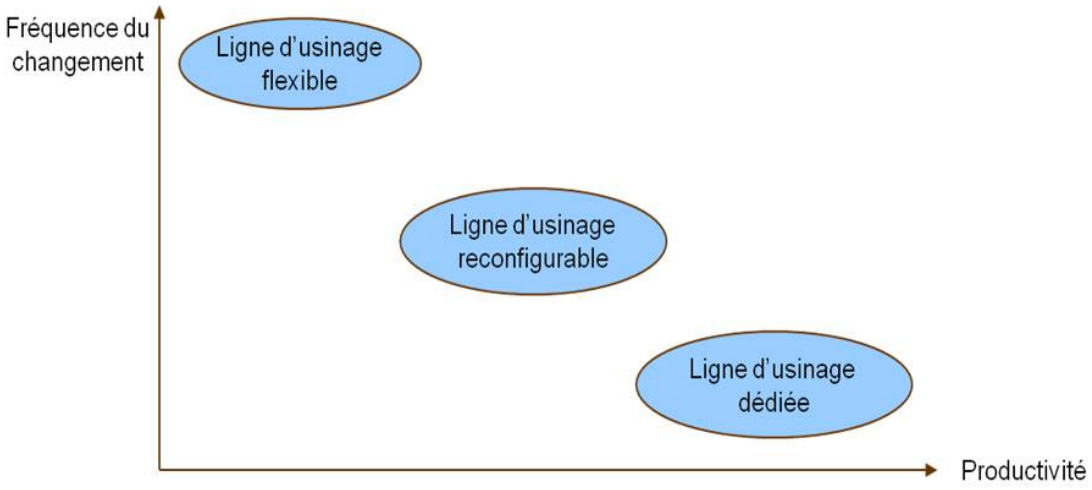


FIGURE 1.1 – Différents types de systèmes d'usinage

Liu 1993]. Malgré leur capacité à répondre à la fois aux changements internes et externes, les FMS présentent un certain nombre d'inconvénients :

- coût initial très élevé des équipements employés,
- leurs capacités de production sont souvent fixes et peuvent s'avérer très inférieures aux objectifs.

La fiabilité des FMS est aussi plus faible par rapport aux autres systèmes à cause de leur complexité.

1.1.1.2 Systèmes d'usinage reconfigurables

Le concept des systèmes d'usinage reconfigurables (RMS Reconfigurable Manufacturing Systems) a été défini par Y. Koren [Koren *et al.* 1999] : « Le système de production reconfigurable est un système qui est conçu au départ pour le changement rapide de structure, aussi bien des composants que des logiciels. Il peut être ajusté rapidement à la capacité et à la fonctionnalité de production, pour une famille de pièces, en réponse aux changements rapides du marché ou des conditions de normalisation ».

Le concept de la reconfigurabilité se base sur les six principes suivants [Koren & Ulsoy 2002; Spicer *et al.* 2002] :

- la « modularité », qui est la faculté de regroupement des fonctions assurées par des composants du système (logiciels et matériels) de façon qu'ils soient modulaires,

- l'« intégrabilité », qui est la possibilité d'intégration rapide des modules constitutifs du système par un ensemble d'interfaces,

- la « convertibilité », qui est la faculté de transformation de la fonctionnalité du système, des machines et de la commande,

- l'« extensibilité », qui est la faculté d'ajustement à la capacité de la production requise,

- la « personnalisation », qui est la faculté d'adaptation à une flexibilité personnalisée (juste nécessaire au bon moment) du système,

- la « diagnosticabilité », qui est la possibilité d'identifier automatiquement l'état courant du système et de la commande afin de détecter ou diagnostiquer les sources de pannes.

La modularité, l'interchangeabilité, et la diagnosticabilité réduisent le temps de la reconfiguration, alors que la personnalisation et la convertibilité diminuent son coût.

Dans le Tableau 1.1, nous résumons les questions posées dans la littérature concernant la reconfiguration des RMS. Puis, dans la Table 1.2, nous présentons une synthèse des travaux de recherche abordant ces questions publiés depuis l'année 2000. Les sujets traités sont indiqués à l'aide du symbole ✓dans les colonnes portant les noms des colonnes du Tableau 1.1.

Tableau 1.1 – Questions analysées

Liste des questions relatives aux RMS	
	Quelles perturbations imposent une reconfiguration ?
B	Indicateurs de performance pour la reconfiguration
C	Comment reconfigurer ?
D	Pourquoi le RMS ?
E	Évolutivité des RMS
F	Personnalisation des RMS
G	Convertibilité des RMS

Tableau 1.2 – Analyse des travaux sur RMS

	A	B	C	D	E	F	G		A	B	C	D	E	F	G
[Lian *et al.* 2000]		✓		✓				[Endsley *et al.* 2006]		✓	✓				
[Mehrabi *et al.* 2000]		✓						[Xing *et al.* 2006]			✓			✓	
[S.E. Chick 2000]		✓		✓				[Azab & ElMaraghy 2007]	✓						
[Xiaobo *et al.* 2000a]		✓						[Bi *et al.* 2007]		✓		✓			
[Xiaobo *et al.* 2000b]			✓	✓				[Hon & Xu 2007]		✓		✓			
[Zhong *et al.* 2000]		✓		✓	✓		✓	[Ren *et al.* 2007]		✓		✓			
[Chandra & Grabis 2001]			✓			✓		[Wiendahl *et al.* 2007]				✓			
[Ku 2001]		✓		✓				[Youssef & ElMaraghy 2008]	✓						
[Son *et al.* 2001]					✓		✓	[Bi *et al.* 2008]		✓		✓			
[Xiaobo *et al.* 2001]		✓	✓	✓				[Elmaraghy *et al.* 2008]	✓	✓					
[Saad *et al.* 2002]		✓						[Abdi 2009b]		✓	✓	✓			
[Abdi & Labib 2003]			✓					[Abdi 2009a]		✓	✓	✓			
[Bruccoleri *et al.* 2003]	✓		✓	✓				[Fong & Sturges 2009]		✓	✓		✓		✓
[Yu *et al.* 2003]	✓	✓						[Chandra & Grabis 2009]		✓	✓				
[Nourelfath *et al.* 2003]		✓		✓		✓		[Li *et al.* 2009]		✓					
[Yigit & Allahverdi 2003]	✓			✓				[Salerno 2009]		✓					
[Abdi & Labib 2004]			✓					[Abbasi & Houshmand 2009]		✓		✓	✓		
[Deif & ElMaraghy 2005]				✓	✓			[Safaei & Tavakkoli 2009]				✓			
[Barhak *et al.* 2005]		✓	✓					[Covanich & McFarlane 2009]		✓		✓			
[Kumar *et al.* 2005]	✓					✓		[Dou *et al.* 2010]		✓	✓	✓			
[Anosike & Zhang 2006]	✓			✓				[Alsafi & Vyatkin 2010]	✓	✓					
[Youssef & ElMaraghy 2006]		✓	✓					[Zhang *et al.* 2010]		✓		✓			
[Bruccoleri *et al.* 2006]	✓							[Abdi & Labib 2011]	✓			✓			
[Galan *et al.* 2007]		✓		✓				[Gumasta *et al.* 2011]				✓	✓		✓
[Hirani *et al.* 2006]		✓						[Lee & Banerjee 2011]		✓		✓			
[Kelepouris *et al.* 2006]		✓													

1.1.1.3 Systèmes d'usinage dédiés

Les systèmes d'usinage dédiés (Dedicated Manufacturing Systems ou DMS) encore appelés **lignes de transfert** ont été développés dans le secteur de l'automobile (production de masse) où la rationalisation de la fabrication répond à un objectif de productivité [Dashchenko 2003; Dolgui & Proth 2006; Rigamonti & Tolio 2007].

Ces lignes sont dédiées, comme leur nom l'indique, à produire un seul produit. Une ligne de transfert est constituée d'une chaîne de stations de travail reliées par un dispositif de transfert de pièces. Chargée sur la première station, la pièce brute traverse la ligne et subit au fur et à mesure des opérations d'usinage pour être transformée en un produit fini qui sera déchargé de la dernière station de la ligne.

À intervalles réguliers, égaux au temps de cycle, toutes les pièces se trouvant sur la ligne sont simultanément déplacées vers la station de travail suivante. Au même moment, un nouveau brut est chargé sur la première station de travail et une pièce finie est déchargée sur la dernière. Afin d'assurer le déplacement synchrone des pièces et d'éviter le recours aux stocks tampons, le temps d'usinage sur chaque station de travail doit être inférieur au temps de cycle objectif, désigné par T_0 ($T \leq T_0$).

Chaque station de travail est équipée d'au moins une unité d'usinage (boîtier), mais peut en disposer plusieurs comme le montre la Figure 1.2. L'activation des boîtiers peut être réalisée de façon parallèle, séquentielle ou mixte.

Les boîtiers comportent un ou plusieurs outils. Les boîtiers multibroches (Figure 1.3) sont utilisés pour effectuer simultanément un ensemble d'opérations.

Dans certains cas, les stations de travail peuvent être équipées avec des machines dites spéciales, à savoir : machines à table mobile (Figure 1.8) et machines à table rotative circulaire (Figure 1.9). Leur fonctionnement sera expliqué dans la section 1.5.

Les lignes de transfert ne sont pas conçues pour répondre rapidement aux changements des gammes d'usinage, par conséquent, le problème de leur reconfiguration devient complexe et très coûteux. Ainsi l'utilisation des outils d'aide à la décision est nécessaire pour résoudre efficacement ce type de problème.

1.1.2 Lignes d'assemblage

Henry Ford est le premier en 1913 à utiliser, dans une usine du Michigan, une ligne d'assemblage, appelée aussi chaîne de montage pour la fabrication de véhicules

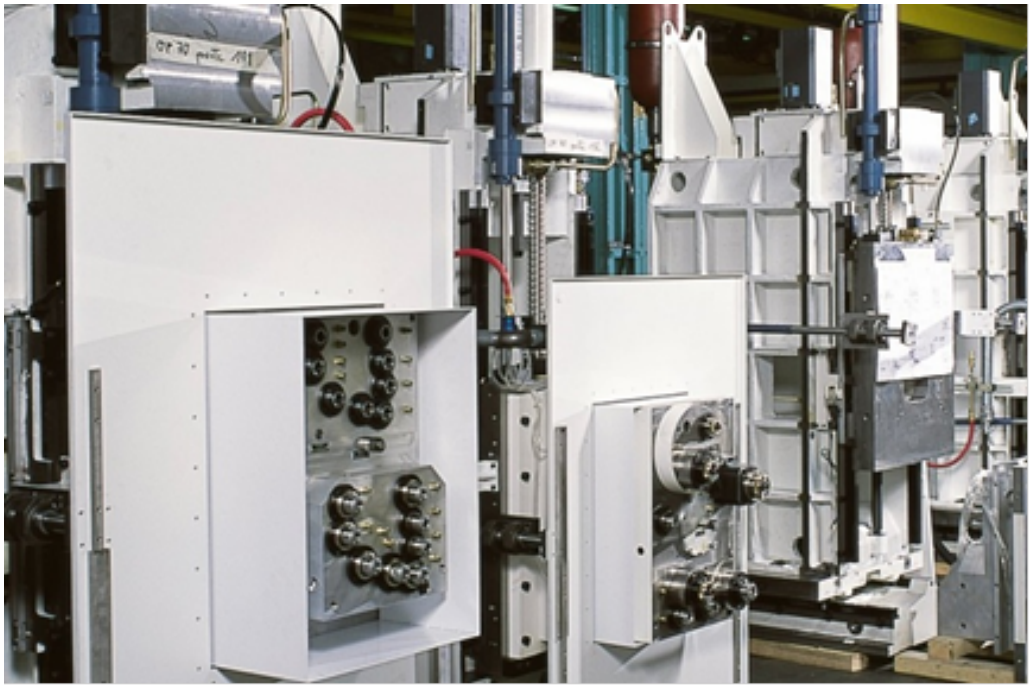


FIGURE 1.2 – Stations de travail (PCI-SCEMM)

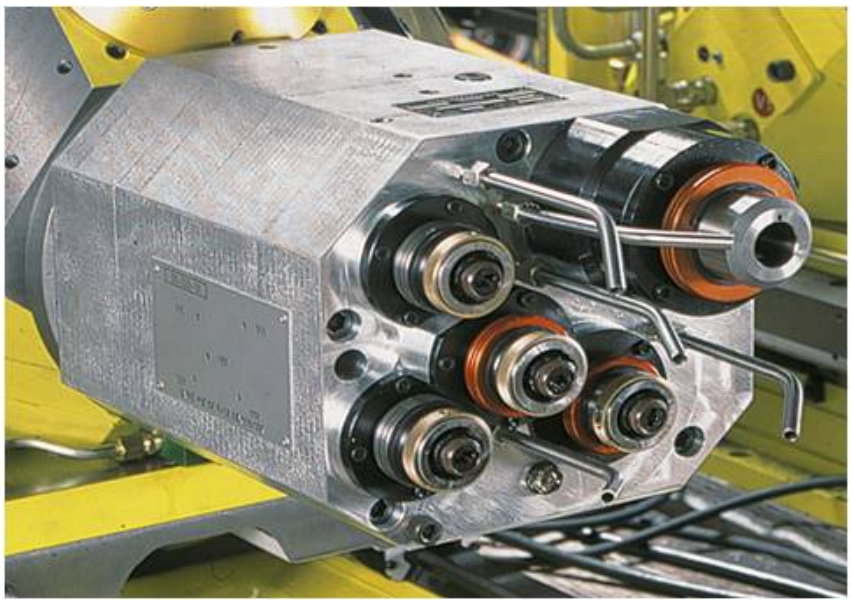


FIGURE 1.3 – Boîtier multibroche

de modèle T. Celle-ci a permis de réduire largement les coûts de fabrication, et par répercussion de multiplier la production et la rentabilité par dix. Depuis, les chaînes de montage ont été largement employées dans l'industrie automobile ou pharmaceutique, l'aéronautique, les télécommunications, etc. pour l'assemblage rapide d'une grande variété de produits.

Une **ligne d'assemblage** consiste en un ensemble de stations de travail où un ensemble d'opérations spécifiques sont exécutées pour assembler un produit. Ces stations peuvent être agencées de différentes manières :

- **Ligne droite** comme le montre la Figure 1.4.

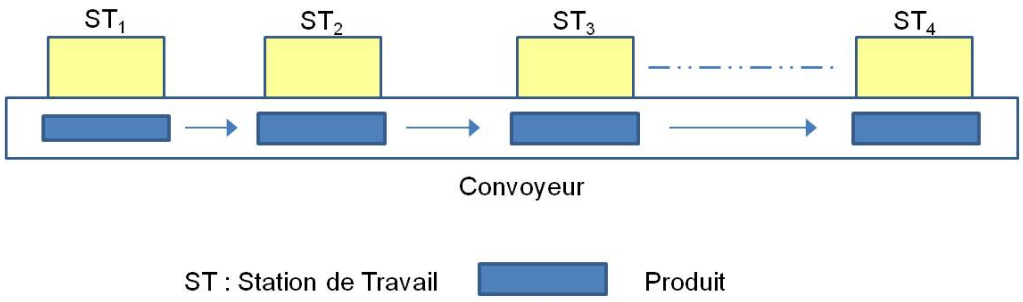


FIGURE 1.4 – Ligne d'assemblage

- **Ligne à machines parallèles** : les machines sont installées en parallèle sur certaines stations de la ligne pour fabriquer un ou plusieurs produits. Cet agencement augmente la flexibilité et diminue l'impact des pannes de machines. Une telle ligne permet de mieux s'adapter à la demande, car le nombre de machines utilisées peut varier en fonction de l'évolution de celle-ci [McMullen & Tarasewich 2006; Gökçen *et al.* 2006].

- **Ligne en forme de U** : les deux extrémités de cette ligne forment un U (Figure 1.5). Les opérateurs peuvent se déplacer sur la ligne pour exécuter des opérations sur des stations différentes. Par rapport à la ligne en série, une ligne en forme de U permet d'augmenter le taux de production, parfois de diminuer le nombre de stations ou de réaliser un meilleur équilibre du travail entre les opérateurs [Ajenblit & Wainwright 1998; Urban 1998; Aase *et al.* 2004].

- **Ligne à deux côtés** : les opérateurs peuvent être positionnés sur les deux côtés d'une ligne et plusieurs d'entre eux peuvent travailler simultanément sur le même produit en exécutant des opérations différentes. Ce type de ligne est notamment employé quand le produit à assembler est très grand et difficile à

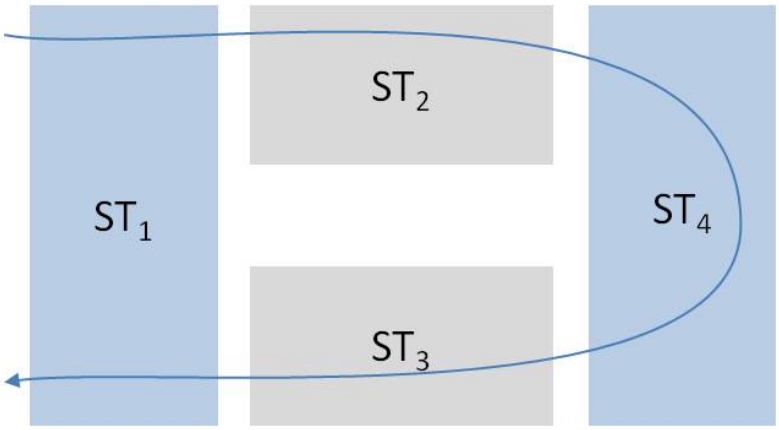


FIGURE 1.5 – Ligne en forme de U

déplacer.

Quel que soit leur agencement, les stations sont liées entre elles à l'aide d'un système de transfert mécanisé (comme un convoyeur, un tapis roulant, etc.) qui est utilisé pour déplacer les produits d'une station à une autre. Dans ce mémoire, nous ne considérons que les lignes d'assemblage cadencées pour lesquelles le temps de séjour de chaque produit à chaque station de travail est le même et il ne doit pas dépasser le temps de cycle. A la fin de chaque cycle, un produit assemblé sort à la fin de la ligne.

1.2 Conception préliminaire des lignes de production

De façon générale et selon [Pahl & Beitz 1961], le processus de conception d'un système de production suit une séquence de quatre étapes : l'analyse du besoin, la recherche des concepts, la conception architecturale et la conception détaillée. La conception préliminaire se limite aux trois premières étapes. Son enjeu principal est de converger le plus tôt possible vers les meilleures solutions au regard des exigences techniques, économiques et environnementales. La conception détaillée se base sur les résultats préliminaires tout en approfondissant l'étude de la solution adoptée. S'il devient nécessaire de revenir sur les décisions de la conception préliminaire pour non-respect de certaines contraintes [Lee & Stecke 1996], la réalisation du projet à temps peut être compromise.

La conception préliminaire est une phase primordiale dans le cycle de vie des

lignes de production, puisqu'une erreur à cette étape peut entraîner des coûts supplémentaires importants. Les processus décisionnels suivants y sont impliqués [Battaïa *et al.* 2012a; Dolgui & Proth 2006] :

1. Analyse du besoin (Produit à fabriquer) : cette phase consiste à caractériser le produit à fabriquer répondant au besoin du marché.

2. Planification des processus (Choix de gammes) : cette étape couvre la sélection des équipements ainsi que des séquences d'opérations nécessaires pour fabriquer le produit. A cette étape, les contraintes technologiques sont analysées, par exemple, les contraintes de compatibilité entre les opérations.

3. Configuration de la ligne (Équilibrage de la ligne) : l'objectif de cette étape est de répartir les opérations qui sont nécessaires pour la fabrication du produit à des stations de travail tout en respectant l'ensemble des contraintes exprimées.

4. Évaluation et simulation : une évaluation des mesures de performance du système telles que qualité, coût, temps et flexibilité doit être faite à ce stade [Chryssolouris 2006]. La simulation est utilisée pour étudier le flux dynamique des produits en tenant compte des aléas.

1.3 Optimisation de la configuration des lignes de production

L'optimisation de la configuration est une étape importante lors de la conception d'une ligne de production. Pour les lignes de production cadencées, considérées dans ce mémoire, ce problème est étroitement lié au problème d'équilibrage des charges ou de la minimisation du temps mort. Ce problème d'optimisation combinatoire a été formulé pour la première fois par Salveson [Salveson 1955] pour les lignes d'assemblage. Elle sera présentée dans la section 1.4. Cette formulation a été nommée SALBP (Simple Assembly Line Balancing Problem) par Baybars [Baybars 1986]. Plusieurs généralisations de cette formulation basique ont été proposées dans la littérature [Battaïa & Dolgui 2013], il est d'usage de les référencer comme GALBP (Generalized Assembly Line Balancing Problem).

Au sein de la classe GALBP, le problème d'équilibrage des lignes d'usinage ou de transfert (Transfert Line Balancing Problem, TLBP) est un cas particulier, car il tient compte des particularités du processus d'usinage par rapport au contexte d'assemblage considéré au départ pour le SALBP et le GALBP.

Le problème TLBP a été initialement défini pour les lignes de transfert dédiées qui comportaient une séquence de stations de travail munies des boîtiers multi-broches exécutant des blocs d'opérations de façon séquentielle [Dolgui *et al.* 1999]. Lors de la conception en avant-projet, toutes les opérations doivent être affectées à des boîtiers et des stations de travail avec l'objectif de minimiser le coût total de la ligne composé des coûts liés aux stations et aux boîtiers utilisés. Plus récemment, ce problème d'optimisation a été également étudié pour les lignes de transfert à configuration plus complexe [Battaïa *et al.* 2014b] et des lignes de transfert reconfigurables [Mohamed 2010]. Ces problèmes d'optimisation seront discutées dans la section 1.5.

Dans ce qui suit, nous présentons les hypothèses utilisées pour la modélisation des problèmes évoqués en commençant par le problème originel du SALBP. Néanmoins, nous montrons dans la suite de ce mémoire que même si les deux problèmes d'optimisation ont un certain nombre d'éléments en commun, les méthodes de résolution et les modèles mathématiques développés pour l'optimisation de la configuration de la ligne initiale ne peuvent pas être réutilisés directement dans le cas d'une reconfiguration.

1.4 Équilibrage des lignes d'assemblage

Le problème d'équilibrage des lignes d'assemblage dit "simple" ou SALBP concerne les lignes mono-produit avec les temps opératoires déterministes pour lequel l'ensemble des hypothèses suivantes est habituellement admis [Baybars 1986; Dolgui & Proth 2010].

- les stations sont agencées en une ligne droite ;
- tous les paramètres relatifs à la ligne sont connus avec certitude ;
- une opération est indivisible et elle ne peut pas donc être partagée entre plusieurs stations ;
- toutes les opérations doivent être exécutées ;
- n'importe quelle station peut exécuter n'importe quelle opération ;
- les temps opératoires sont fixés et indépendants de la station ;

- les opérations doivent être exécutées en fonction de leurs relations de précédence qui sont représentées par un graphe acyclique orienté ;

- la charge d'une station est égale à la somme des temps opératoires des tâches qui y sont affectées.

Selon les variantes du SALBP, une des hypothèses suivantes peut être utilisée :

- le temps de cycle T_0 est donné et fixe ;

- le nombre de stations m est donné et fixe.

Selon la fonction objectif employée, 3 variantes du SALBP sont généralement distinguées :

- SALBP-1 minimise le nombre de postes de travail en considérant un temps cycle fixé,

- SALBP-2 minimise le temps cycle en considérant un nombre de postes de travail fixé,

- SALBP-E minimise la somme des temps morts (écart entre le temps cycle et le temps de traitement sur le poste) sans fixer ni le nombre de postes ni le temps cycle en trouvant le meilleur compromis entre ces deux paramètres.

Les critères d'optimisation utilisés dans les généralisations du SALBP peuvent être classés de la façon suivante :

- **économiques** ou **d'efficacité** (minimisation des coûts de production, maximisation de la charge de travail des opérateurs sous la contrainte d'un seuil),

- **d'équité** (équilibrer la charge du travail des opérateurs pour différents modèles du produit ou entre les différentes stations de travail),

- **de stabilité** (maximiser la probabilité de compléter le travail en respectant le temps de cycle si les temps opératoires ont des durées stochastiques, minimiser les modifications dans la charge du travail lors d'un rééquilibrage).

Ces aspects sont pris en compte individuellement ou conjointement. Dans le dernier cas, on parle des problèmes d'optimisation multi-objectif).

La Table 1.3 présente un aperçu des problèmes d'équilibrage multi-objectifs étudiés dans la littérature. Les numéros donnent l'ordre de priorité des objectifs considérés. Par exemple, Chutima et Chimklai [Chutima & Chimklai 2012] ont utilisé quatre objectifs. Les deux premiers appartenaient à la catégorie "Efficacité", à savoir : la longueur de la ligne (avec la plus haute priorité) et le nombre de stations (avec une priorité moindre que le premier objectif mais plus haute que les deux derniers). Les deux autres objectifs appartenaient à la catégorie "Équité", à savoir l'indice d'équilibrage et l'indice de proximité technologique. Ces deux objectifs avaient la même priorité qui était moindre que celle de l'objectif 2.

Si tous les objectifs considérés ont la même priorité, le caractère "+" est utilisé pour le signaler dans le Tableau 1.3.

1.5 Équilibrage des lignes d'usinage

A la différence des lignes d'assemblage qui sont souvent manuelles, les lignes d'usinage sont hautement automatisées. Pour cette raison, les coûts liés aux équipements sont prépondérants dans le coût total de la ligne. A l'étape de la configuration de la ligne ou de son équilibrage, l'objectif primordial est donc la minimisation de ces coûts. Les différents problèmes d'équilibrage étudiés pour les lignes d'usinage peuvent être classés dans les trois catégories suivantes selon l'information disponible sur les équipements pouvant être utilisés dans la ligne :

- **Équilibrage avec choix des équipements**. Dans ce cas, les unités d'usinage (boîtiers multibroches) pouvant être utilisées dans la ligne sont connues. Chaque boîtier est décrit par l'ensemble des opérations qu'il peut exécuter en parallèle. Plusieurs boîtiers pouvant réaliser la même opération, il est nécessaire d'en choisir ceux qui seront utilisés et de les affecter aux stations de travail afin d'exécuter toutes les opérations en minimisant le coût total de la ligne.

- **Équilibrage avec choix des modèles de machines**. Ce problème a été étudié pour les lignes d'usinage constituées de machines à commande numérique (CNC) ou centres d'usinage. Plusieurs types de machines sont disponibles ayant des caractéristiques techniques et coûts différents. Dans ce cas, le problème d'optimisation concerne à la fois le choix du type de chaque machine utilisée et leur nombre, mais aussi l'affectation des opérations à chaque machine (ce qui ne faisait pas partie du problème précédent) tout en minimisant le coût total de la ligne.

Tableau 1.3 – Fonctions objectifs considérées conjointement

Source	Efficacité	Équité	Stabilité
[Özcan & Toklu 2009a]	1, 2		
[Chutima & Chimklai 2012]	1,2	3,3	
[Kara *et al.* 2009]	1,2	3	
[Otto & Scholl 2011; Simaria & Vilarinho 2009; Vilarinho & Simaria 2006; Yu & Yin 2010]	1	2	
[Ozbakir *et al.* 2011]	2	1	
[Fattahi & Roshani 2011]	2,1		
[Özcan & Toklu 2009b]	1,2	3	
[Baykasoglu 2006]		2	1
[Purnomo *et al.* 2011]	1	2	
[Mcgovern & Gupta 2007; McGovern & Gupta 2006; McGovern & Gupta 2011]	2, 3, 4	1	
[Toklu & Özcan 2008]	+,+	+	
[Yagmahan 2011]	+	+,+	
[Brudaru *et al.* 2010]	+	+	
[Chutima & Olanviwatchai 2010]	+	+,+	
[Ding *et al.* 2011]	+,+	+	
[Bagher *et al.* 2011]	+	+	+
[Pekin & Azizoglu 2008; Chica *et al.* 2010a; Chica *et al.* 2010b; Chica *et al.* 2011a; Chica *et al.* 2011b; Manuel *et al.* 2011]	+,+		
[McMullen & Tarasewich 2006]	+,+,+		+
[Nearchou 2008; Nearchou 2011; Zacharia & Nearchou 2012; Nourmohammadi & Zandieh 2011]	+	+	
[Hamta *et al.* 2011]	+,+		
[Yoosefelahi *et al.* 2012]	+,+,+		
[Zhang & Gen 2011]	+,+	+	
[Hwang *et al.* 2008; Hwang & Katayama 2009; Özcan & Toklu 2009c]	+,+		
[Cakir *et al.* 2011]	+	+	
[Tasan & Tunali 2008]	+, +	+	
[Gamberini *et al.* 2004; Gamberini *et al.* 2009]	+		+
[McMullen & Frazier 1998]	+		+

- **Équilibrage avec conception des boîtiers multibroches**. Les lignes d'usinage concernées sont équipées de stations de travail du même type mais pouvant accueillir plusieurs boîtiers multibroches. Ces boîtiers sont à concevoir en fonction de l'ensemble des opérations qui leur sont affectées.

- **Équilibrage avec conception des machines spéciales**. Dans ce cas, les lignes d'usinage peuvent contenir plusieurs machines spéciales (machines à table circulaire pivotante, machines à table mobile, etc.). Ces machines peuvent avoir plusieurs positions de travail équipées de boîtiers multibroches activés en parallèle. La solution doit déterminer le type et la configuration de chaque machine utilisée notamment le nombre de positions de travail ainsi que la configuration de chaque boîtier multibroche à installer.

Dans ce qui suit, nous présentons les hypothèses utilisées pour chacun de ces problèmes sans évoquer les méthodes d'optimisation associées qui feront l'objet d'étude dans le chapitre 2. Il est à noter que le problème de reconfiguration défini dans la suite du mémoire concerne à la fois la réutilisation des anciens équipements, donc nécessite un choix parmi les équipements disponibles, et la conception des nouveaux équipements. De ce fait, il appartient à la catégorie des problèmes non couverts par les études antérieures.

1.5.1 Équilibrage avec choix des équipements

Chaque boîtier est défini par l'ensemble d'opérations qu'il peut exécuter et son coût. Dans le cas général, plusieurs boîtiers sont capables d'exécuter la même opération, mais un seul sera retenu à l'étape de la configuration. Le problème d'optimisation consiste à déterminer le nombre de stations et à sélectionner un sous-ensemble de boîtiers à partir d'un ensemble donné au préalable pour les affecter aux stations. Les coûts des stations et de chaque boîtier sont connus. Le critère à optimiser est le coût de la ligne. Les contraintes de compatibilité entre les boîtiers, les contraintes de précédence entre les opérations et la contrainte du temps de cycle sont considérées.

Lorsque plusieurs boîtiers sont affectés à la même station de travail, il est important de savoir dans quel ordre ils seront activés pour usiner la pièce. Les modes suivants ont été étudiés : l'activation parallèle [Belmokhtar *et al.* 2006; Delorme *et al.* 2012; Dolgui *et al.* 2012; Dolgui *et al.* 2006c], l'activation séquentielle [Dolgui & Ihnatsenka 2009b] et mixte [Dolgui & Ihnatsenka 2009a]. Chaque mode implique un calcul différent de la charge des stations de travail et, par conséquent, un modèle mathématique différent pour la contrainte de temps de cycle. Il est intéressant de noter que pour le cas de l'activation parallèle, la contrainte de temps de cycle

peut être vérifiée avant la phase d'optimisation en éliminant tous les boîtiers dont le temps d'exécution ne la respecte pas.

1.5.2 Équilibrage avec choix des modèles de machines

Ce problème a été formulé pour les lignes d'usinage à machines parallèles [Mohamed 2010]. Comme précédemment, les stations de travail sont disposées en séquence, mais chacune entre elles peut être munie de plusieurs machines identiques. Les mêmes opérations sont dédoublées sur toutes les machines de la même station. Ainsi plusieurs pièces sont traitées en même temps sur la même station de travail. Les machines parallèles possèdent un temps de cycle local égal au nombre de machines multiplié par le temps de cycle objectif de la ligne. Ainsi à chaque temps de cycle une seule pièce quitte la station et une seule pièce rejoint la station. Cette configuration permet de réaliser les opérations ayant un temps opératoire qui dépasse le temps de cycle de la ligne.

Les machines pouvant être installées en parallèle n'utilisent pas de boîtiers multibroches et leur conception n'est pas personnalisée. Au contraire, chaque machine est caractérisée par son type (modèle) qui détermine ses capacités techniques (par exemple, le nombre d'axes d'usinage). Un coût est associé à chaque modèle de machines en fonction des fonctionnalités fournies. Un exemple de ligne à machines parallèles de deux types (rouges et vertes) est donné dans la Figure 1.6.

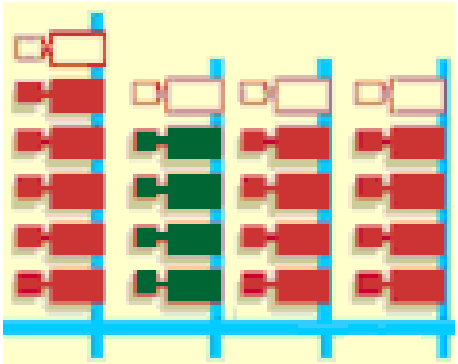


FIGURE 1.6 – Ligne d'usinage à machines parallèles

Ces machines sont monobroches, c'est-à-dire un seul outil peut être actif à la fois. Pour installer l'outil requis pour chaque opération, un temps de changement d'outil est nécessaire. Ainsi le temps de réalisation de deux opérations n'est pas égal à la somme des temps opératoires de ces opérations mais dépend de la séquence dans

laquelle elles se font, car les temps de déplacement d'outil, de changement d'outil et de rotation de la pièce ne sont pas négligeables. Le positionnement de la pièce sur la table de travail sur chaque station est une autre variable de décision. Il définit les surfaces accessibles pour l'usinage.

Alors, le problème d'optimisation consiste à repartir les opérations sur les stations de travail et déterminer le type et le nombre de machines parallèles installées sur chaque station ainsi que le positionnement de la pièce sur chaque station tout en respectant les contraintes données et en minimisant le coût total de la ligne.

1.5.3 Équilibrage avec conception des boîtiers multibroches

Une ligne de transfert cadencée à boîtiers multibroches où les boîtiers sont activés en séquence est présentée dans la Figure 1.7. C'est pour ce type de système d'usinage que nous formulerons le problème de reconfiguration dans la suite de ce mémoire.

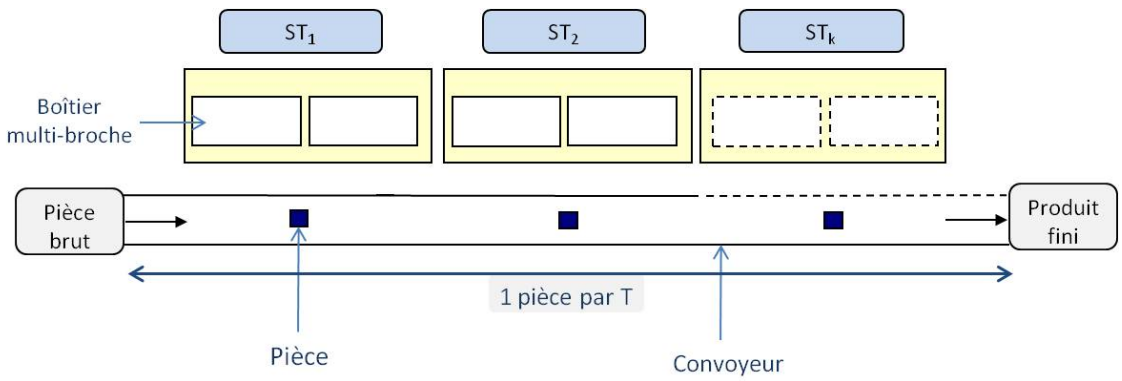


FIGURE 1.7 – Ligne de transfert

Dans cette section, nous décrivons le problème de la configuration initiale considérée pour ces lignes dans la littérature [Dolgui *et al.* 2006b; Finel 2004]. Il vise à déterminer le nombre de stations de la ligne, le nombre de boîtiers à chaque station et la composition de chaque boîtier. A cette étape, le critère prépondérant est le coût du système d'usinage qui doit être minimisé. Comme les valeurs exactes de certains coûts ne peuvent pas être connues à cette étape de la conception, par conséquent, l'optimisation s'appuie sur les coûts *structurants* définis par le nombre de stations de travail et de boîtiers multibroches utilisés.

Les deux paramètres suivants introduisent les coûts d'équipement dans le modèle mathématique : C_1 désigne le coût relatif d'une station de travail et C_2 le coût

relatif d'un boîtier. La valeur C_2 comprend tous les coûts liés aux parties porteuses, parties opératives, actionneurs, dispositifs d'injection de liquide, etc. Quant à la valeur de C_1, elle tient compte des coûts qui ne dépendent pas du nombre de boîtiers installés sur une station de travail, comme par exemple le coût des dispositifs de transfert de pièce, d'activation des boîtiers, etc.

Dans ce qui suit, sont récapitulées les hypothèses utilisées pour la formulation du problème de la configuration. Toutes ces hypothèses sont également valables pour le problème de reconfiguration que nous considérons dans les chapitres 4 et 5.

1. Toutes les stations de travail sont identiques en termes de mode d'activation des boîtiers et de nombre maximum autorisé de boîtiers.
2. Le nombre maximum de stations de travail est limité, il dépend de l'architecture du système d'usinage.
3. Le coût de base d'une station de travail ne dépend pas du nombre de boîtiers qui y sont installés, il est le même pour toutes les stations de travail de la ligne.
4. Tous les boîtiers ont le même coût quelles qui soient leur affectation ou composition.
5. Chaque opération doit être effectuée une seule fois, soit par un et un seul boîtier d'une et d'une seule station de travail.
6. La ligne est cadencée et la contrainte de temps de cycle objectif doit être respectée.
7. Il y a des contraintes de précédence entre les opérations et des contraintes d'inclusion et d'exclusion aussi bien au niveau des stations de travail qu'au niveau des boîtiers.

1.5.4 Équilibrage avec conception de machines spéciales

Dans la catégorie des systèmes d'usinage dédiés se trouvent aussi les machines spéciales telles que les machines à table mobile (Figure 1.8) et les machines à table circulaire pivotante (Figure 1.9). Ces machines peuvent être utilisées séparément ou groupées en une ligne d'usinage cadencée où le déplacement de la pièce est organisé d'une machine à l'autre d'une façon similaire aux lignes considérées dans la section précédente [Battaïa *et al.* 2014b]. Lors de la conception de ce type de ligne, le problème d'équilibrage concerne non seulement le nombre de machines et le type de chaque machine spéciale utilisée, mais aussi sa configuration détaillée en termes de position de travail et de boîtiers multibroches. Dans ce qui suit, nous décrivons le mode de fonctionnement des machines de ces deux types.

Sur une **machine à table mobile**, chaque pièce est usinée en passant par m positions de travail successifs. L'avancement de la pièce d'une position à la position suivante se fait par le déplacement de la table mobile sur laquelle la pièce est fixée. Ainsi une seule pièce est présente sur la machine tout au long du temps de cycle. Donc, seulement la position où se trouve la pièce traitée peut être active à la fois. Les boîtiers multibroches de la position active s'exécutent simultanément. A la fin de cette exécution, la pièce est avancée soit vers la position suivante soit si elle est finie vers la première position pour déchargement. Un nouveau cycle commence pour une nouvelle pièce et ainsi de suite.

Pour ne pas gêner le déplacement de la table mobile, seulement deux unités d'usinage peuvent être installées sur chaque position de travail : une à gauche par rapport à la table et une autre sur sa droite. Une troisième unité d'usinage peut être rajoutée devant la table uniquement sur la dernière position de travail. Ainsi, sur chaque position de travail (sauf la dernière), il est possible d'installer soit deux boîtiers horizontaux, soit un boîtier horizontal et un vertical, et sur le dernier poste on peut avoir au plus soit deux boîtiers horizontaux et un vertical, soit trois boîtiers horizontaux. Quant au nombre de positions de travail, ce type de machine en possède rarement plus de 3, pour que l'espace occupé par la machine et le temps d'usinage soient acceptables.

Lors de la conception d'une machine à table mobile, l'objectif d'optimisation peut être la minimisation soit du coût de la machine [Battaïa *et al.* 2012b; Guschinskaya 2010] soit du temps de cycle [Guschinskaya *et al.* 2009], car la configuration avec un seul poste de travail actif rend l'usinage de la pièce plus lent.

Une **machine à table circulaire pivotante** est également constituée de plusieurs positions de travail, mais elle peut traiter plusieurs pièces à la fois : une par position. Une position de travail correspond à une position de la table circulaire pivotante. La position 0 est toujours utilisée pour les opérations de chargement\déchargement des pièces et aucune opération d'usinage ne lui est attribuée. En revanche, à toute autre position de travail, nous pouvons affecter au plus deux unités d'usinage, dont une horizontale et une verticale activées en même temps pour traiter deux faces différentes de la pièce.

Tous les boîtiers de toutes les positions de travail sont déclenchés en même temps : ils effectuent les opérations d'usinage sur la pièce présente sur la position de travail où ils sont installés. Même si les temps d'usinage peuvent varier d'une position de travail à un autre, le déplacement de toutes les pièces se trouvant sur la machine se fait en même temps par la rotation de la table d'un pas. Ainsi, chaque pièce avance à la position de travail suivante.

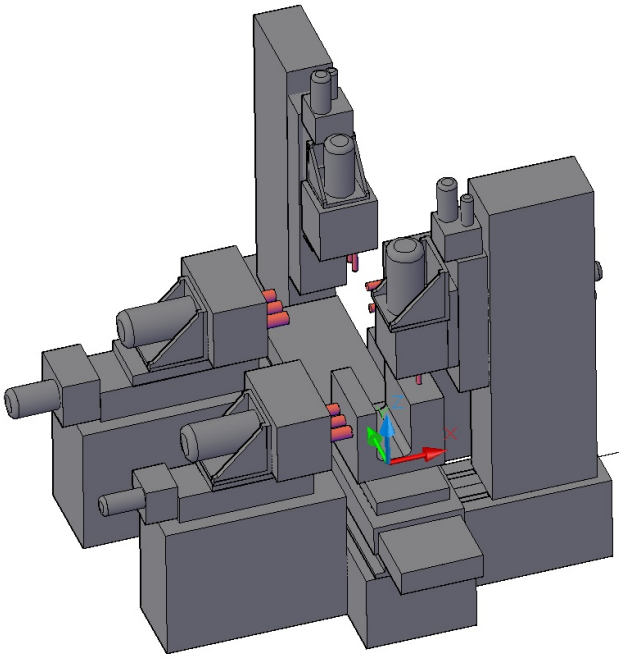


FIGURE 1.8 – Machine à table mobile

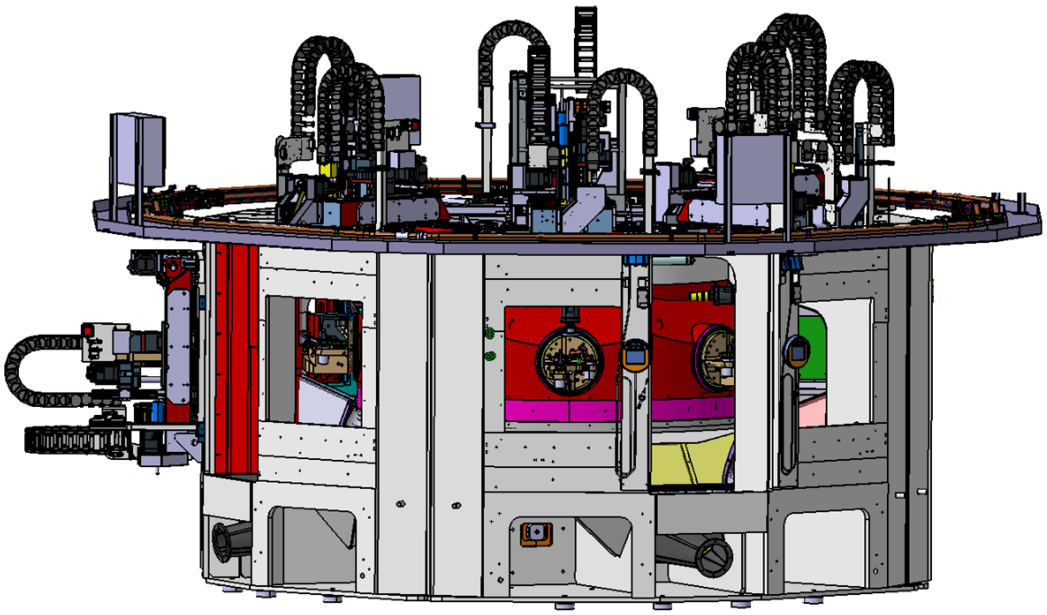


FIGURE 1.9 – Machine à table rotative circulaire

Le problème d'optimisation de la configuration des machines à table circulaire pivotante consiste à minimiser le coût total de la machine en déterminant le nombre de positions de travail et de boîtiers multibroches à installer sur chaque position de travail pour effectuer toutes les opérations d'usinage et en respectant les contraintes technologiques et techniques données [Dolgui *et al.* 2008; Dolgui *et al.* 2009; Battaïa *et al.* 2014a].

1.6 Conclusion

Dans ce chapitre, nous avons introduit les notions de base des systèmes de production en mettant l'accent sur les systèmes qui seront étudiés dans la suite du mémoire, à savoir : les lignes d'assemblage manuelles et les lignes d'usinage automatisées à boîtiers mutlibroches. Les problèmes d'optimisation qui se posent à l'étape de leur configuration ont été également présentés.

Comme les solutions retenues à cette étape ont des conséquences importantes sur toute la durée de vie du système de production, le développement des outils d'optimisation et d'aide à la décision efficaces et adaptés a fait l'objet des nombreux travaux dans le domaine de la recherche opérationnelle. Dans le chapitre suivant, nous passons en revue les méthodes d'optimisation mono- et multi-objectif qui ont été développées pour l'optimisation de la configuration des lignes de production et qui pourraient s'adapter aux problèmes de reconfiguration que nous formulerons dans la suite de ce mémoire.

Chapitre 2

Approches de l'optimisation combinatoire pour la configuration des systèmes de production

Sommaire

2.1 Optimisation : notions de base **25**
2.2 Méthodes exactes mono-objectif **26**
2.2.1 Programmation linéaire en nombres entiers (PLNE) ou en variables mixtes (MIP) 27
2.2.2 Procédures par Séparation et Évaluation (PSE) 32
2.2.3 Programmation Dynamique (PD) 34
2.3 Méthodes d'optimisation approchées **35**
2.3.1 Heuristiques 35
2.3.2 Méta-heuristiques 36
2.4 Optimisation multi-objectif **38**
2.4.1 Approches a priori 40
2.4.2 Approches a posteriori 43
2.4.3 Approches progressives 44
2.5 Conclusion **45**

2.1 Optimisation : notions de base

L'optimisation mono-objectif se définit comme la recherche de la meilleure solution (appelé optimum) parmi un ensemble de solutions faisables (appelé espace de recherche), en fonction d'un critère de comparaison. Si plusieurs objectifs doivent être considérés, il s'agit de l'optimisation multi-objectif [Celano *et al.* 1999; Graves & Redfield 1988].

Le processus d'optimisation comprend deux phases (Figure 3.1). La première est la modélisation qui définit les variables principales d'optimisation, les contraintes

et la ou les fonction(s) objectif. La deuxième phase est la résolution qui cherche les meilleures valeurs des variables de décision en faisant appel à un algorithme d'optimisation [Bonte *et al.* 2006].

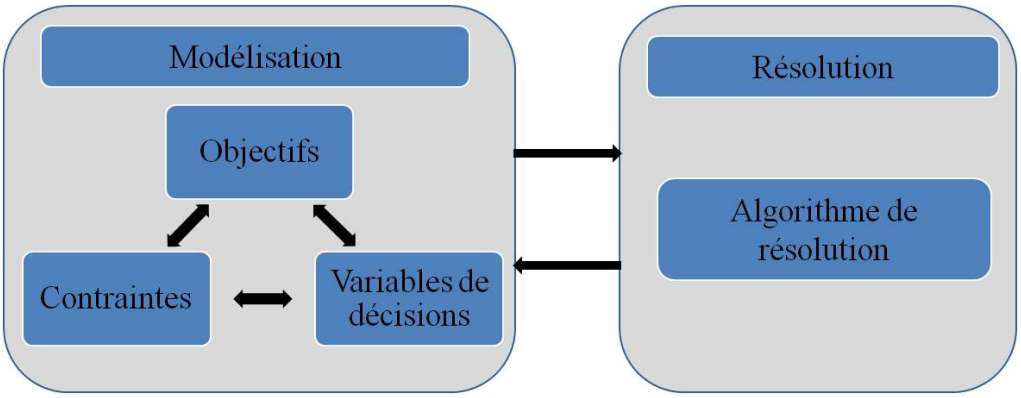


FIGURE 2.1 – Concepts de l'optimisation : modélisation et résolution

Le problème de configuration des systèmes de production est souvent associé dans la littérature au problème d'équilibrage de type SALBP. Plusieurs méthodes ont été développées pour résoudre de tels problèmes d'optimisation. Elles sont généralement réparties en deux grandes catégories :

- Les méthodes d'optimisation exactes, dont l'objectif est de déterminer l'optimum exact d'une fonction objectif ;

- Les méthodes d'optimisation approchées, qui dans le cas général fournissent une solution plus ou moins proche de l'optimum, mais grâce au temps de résolution réduit, elles sont particulièrement adaptées à la résolution des problèmes fortement combinatoires et de grande taille [Gary & Johnson 1979].

Dans ce qui suit, nous présentons des méthodes de ces deux catégories d'abord de manière générale et puis en donnant des exemples concernant la résolution des problèmes de configuration ou d'équilibrage des lignes de production.

2.2 Méthodes exactes mono-objectif

Ces méthodes sont basées sur une résolution algorithmique ou analytique. Elles garantissent l'obtention de la solution optimale et se limitent à la résolution des

problèmes de taille réduite : définis par un petit nombre de variables de décision discrètes (Figure 3.2). Parmi ces méthodes, nous développons ici la programmation linéaire en nombres entiers (PLNE), la méthode par séparation et évaluation ou Branch and Bound (B&B) et la programmation dynamique (PD).

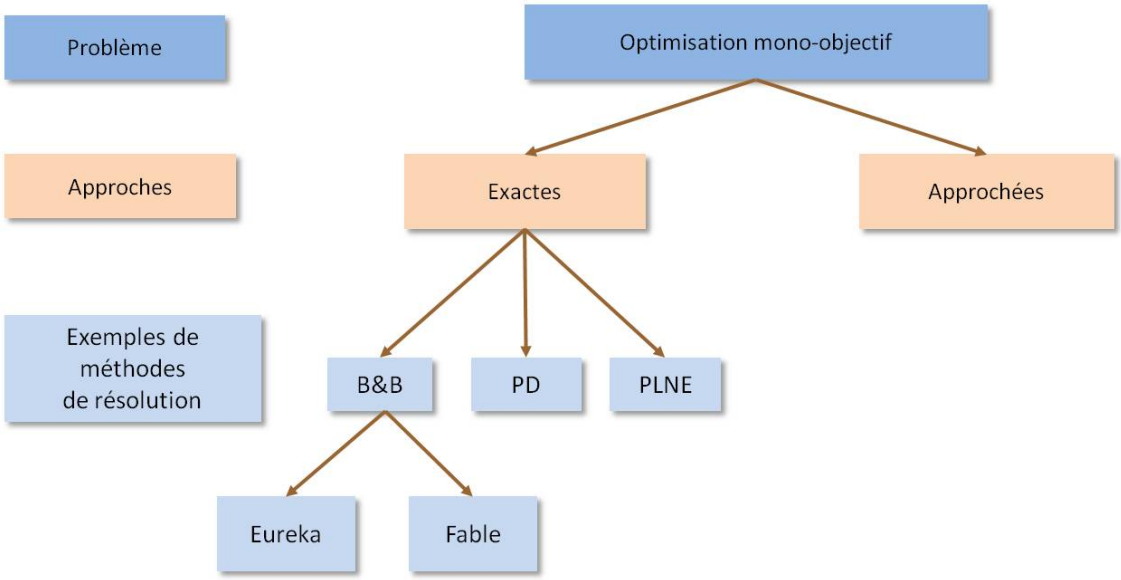


FIGURE 2.2 – Optimisation mono-objectif : approches exactes

2.2.1 Programmation linéaire en nombres entiers (PLNE) ou en variables mixtes (MIP)

La programmation linéaire permet de déterminer l'optimum d'une fonction linéaire tout en respectant des contraintes linéaires [Sakarovitch 1984]. L'algorithme du simplexe a été développé en 1947 par G.B Dantzig pour résoudre ce type de problème. Par la suite, cet algorithme a été perfectionné et de nombreuses recherches ont été menées et sont encore menées dans l'application de la programmation linéaire dans différents domaines.

PL traite des problèmes à variables réelles. Si les variables de décision sont entières, il s'agit alors de la programmation linéaire en nombres entiers (PLNE). S'il y a des variables entières et réelles, il s'agit d'un modèle mixte (Mixed Integer Programming ou MIP).

Un certain nombre de solveurs sont actuellement disponibles pour la résolution des modèles PLNE. Certains ont été conçus sous licence libre de droit, d'autres sous licence propriétaire. Les logiciels commerciaux présentent souvent des performances supérieures en proposant des mises à jour régulières. Les deux principaux logiciels existants sont CPLEX d'IBM ILOG et XPRESS-MP de Dash Optimization. Un ensemble d'outils libres pour l'optimisation de programmes linéaires est disponible sous le nom de GLPK (GNU linear programming kit). Il s'agit d'une librairie pouvant être utilisée par des programmes en langage C/C++. Les composants suivants peuvent ainsi être utilisés :

- la méthode du Simplexe révisé,
- la méthode des points intérieurs primal-dual,
- la méthode par Séparation et Évaluation,
- un solveur de programmes linéaires et programmes mixtes en nombres entiers.

Il est nécessaire de citer l'outil Bob++ développé par des chercheurs de l'Université de Versailles. Il inclut des bibliothèques facilitant la mise en œuvre des approches comme Branch and Bound ou la programmation dynamique présentées ci-dessous.

2.2.1.1 PLNE pour le problème d'équilibrage de lignes d'assemblage

Un modèle linéaire en nombres entiers pour le problème SALBP-1 a été proposé par [Patterson & Albracht 1975]. Soit V représente l'ensemble des opérations à affecter, m le nombre maximal de stations, T_0 le temps de cycle objectif et A l'ensemble de relations de précédences.

Dans ce modèle, les variables binaires de décision sont :

$$y_k = \begin{cases} 1 & \text{si la station } k \text{ est ouverte}, \forall k \in \{1, 2, \ldots, m\}, \\ 0 & \text{sinon.} \end{cases}$$

$$x_{ik} = \begin{cases} 1 & \text{si l'opération } i \text{ est affectée à la station } k, \forall i \in V, \forall k \in \{1, 2, \ldots, m\}, \\ 0 & \text{sinon.} \end{cases}$$

La fonction objectif qui vise à minimiser le nombre de stations ouvertes est définie de la façon suivante :

$$Minimiser \sum_{k=1}^{m} y_k \tag{2.1}$$

La contrainte 2.2 impose que chaque opération n'est affectée qu'une seule fois :

$$\sum_{k=1}^{m} x_{ik} = 1; i \in V; \tag{2.2}$$

Les contraintes 2.3 imposent les contraintes de précédence :

$$\sum_{k=1}^{m} kx_{ik} \leq \sum_{k=1}^{m} kx_{jk}, \forall (i,j) \in A; \tag{2.3}$$

Les contraintes 2.4 représentent la contrainte du temps de cycle :

$$\sum_{i \in V} t_i x_{ik} \leq T_0 y_k, \forall k \in \{1, 2, \ldots, m\}; \tag{2.4}$$

Dans ce qui suit, nous présentons d'autres modèles PLNE développés pour le problème d'équilibrage de lignes de transfert.

2.2.1.2 PLNE pour le problème d'équilibrage de lignes de transfert avec choix des équipements

Nous présentons ici le modèle PLNE pour le problème de configuration d'une ligne d'usinage dans le cas où les boîtiers multibroches sont connus et sont activés de manière parallèle [Belmokhtar *et al.* 2006]. Il est facile de constater que ce problème nécessite davantage des données techniques et que le modèle mathématique intègre de plus nombreuses contraintes technologiques par rapport au problème SALBP présenté dans la section précédente.

Tout d'abord, les notations suivantes doivent être introduites. Soit **N** est l'ensemble des opérations qui doivent être exécutées sur la ligne pour fabriquer chaque produit.

Les données relatives aux blocs :

- $\mathbf{B} = \{B_r \mid r \in \mathbf{R}\}$ est l'ensemble des boîtiers multibroches (blocs) possibles pour réaliser les opérations de $\mathbf{N}$, où $\mathbf{R}$ est l'ensemble des indices de blocs ; Cet ensemble ne contient que les blocs ayant un temps d'exécution inférieur ou égal au temps de cycle.

- q_r est le coût du bloc ;

- $K(r)$ est l'intervalle des indices de stations auxquelles le bloc $B_r \in \mathbf{B}$ peut être affecté ;

- $F = \{F_1, ..., F_v\}$ est une collection des ensembles F_s représentant les indices des blocs en conflit pour l'opération s (il n'y aura qu'un seul bloc de F_s qui sera retenu lors d'une prise de décision) ;

- $F_0 = \mathbf{R} \backslash \bigcup_{s \in V} F_s$ est l'ensemble des blocs contenant des opérations pour lesquelles il n'y a pas de conflits, autrement dit les blocs qui seront pris lors d'une décision ;

- w_{rt} nombre d'opérations concernées par la contrainte d'inclusion t et devant être affectées à la même station que le bloc r ;

- W_t est l'ensemble des blocs qui ont des opérations liées par la même contrainte d'inclusion t ;

- $U_t = \{r \in \mathbf{R} \mid i_t \in B_r\}$, pour chaque i_t de chaque contrainte d'inclusion t. Ainsi, à chaque opération i_t de D_t correspond un ensemble U_t contenant tous les blocs exécutant i_t ;

- $M(r)$ est l'ensemble des opérations précédant les opérations du bloc r ;

- H est l'ensemble contenant tous les blocs qui ont des prédécesseurs ;

- $H(r)$ est l'ensemble contenant toutes les opérations prédécesseurs du bloc r ;

- h_{tr} représente le nombre d'opérations prédécesseurs du bloc r contenu dans le bloc t ;

- R^* est un sous-ensemble de blocs de $\mathbf{R}$, ce sont les blocs qui peuvent être affectés à la dernière station de la ligne.

Les données relatives aux stations et à la ligne de transfert :

- q_0 est le coût d'une station ;

- m_0 et n_0 sont respectivement, les valeurs du nombre maximal autorisé de stations et du nombre maximal autorisé de blocs par station.

- m^* est une borne inférieure pour le nombre de stations ;

Les variables de décision sont :

- les variables binaires x_{rk}, où

$$x_{rk} = \begin{cases} 1 & \text{si le bloc } B_r \in \mathbf{B} \text{ est affecté à la station } k, k = 1, \ldots, m_0; \\ 0 & \text{sinon.} \end{cases}$$

- La variable y représente le nombre de stations, notons que y prend des valeurs entières.

La fonction objectif minimise le coût de la ligne composé des coûts des stations de travail et des boîtiers multibroches utilisés :

$$q_0 y + \sum_{r=1}^{|\mathbf{R}|} \sum_{k=1}^{m_0} (q_r + \varepsilon_r k) x_{rk} \rightarrow min, \tag{2.5}$$

où ε_r est une valeur non négative pour chacun des blocs B_r.

Les contraintes suivantes assurent l'exécution de chacune des opérations de $\mathbf{N}$ une et une seule fois. La première contrainte concerne l'exécution des opérations dans les blocs qui n'ont pas d'intersection avec les autres blocs (pas d'opération en commun). La seconde, quant a elle, concerne les opérations qui peuvent être exécutées dans plusieurs blocs, ainsi la contrainte assure qu'il n'y aura qu'un seul bloc choisi parmi les blocs en conflit :

$$\sum_{k \in K(r)} x_{rk} \leq 1, \qquad r \in F_0, \tag{2.6}$$

$$\sum_{r \in F_s} \sum_{k \in K(r)} x_{rk} \leq 1, \qquad s \in V, \tag{2.7}$$

L'exécution de toutes les opérations de l'ensemble $\mathbf{N}$ est forcée par :

$$\sum_{r \in \mathbf{R}} \sum_{k \in K(r)} |B_r| x_{rk} = |\mathbf{N}|, \tag{2.8}$$

Les contraintes de précédence :

$$\sum_{t \in H(r)} \sum_{s \in K(t), s<k} h_{tr} x_{ts} \geq |M(r)| x_{rk}, \qquad r \in H, k \in K(r), \tag{2.9}$$

Les contraintes d'inclusion imposent l'affectation à la même station de toutes les opérations de chaque ensemble e appartenant à la collection IS :

$$\sum_{r \in W_t} w_{rt} x_{rk} = |D_t| \sum_{s \in U_t} x_{sk}, \qquad i_t \in e, \forall e \in IS, k \in \bigcup_{s \in U_t} K(s), \tag{2.10}$$

Les contraintes d'exclusion interdissent l'allocation de certains blocs à la même station. La collection ES de sous-ensembles e tels que les blocs d'un e ne peuvent pas être tous affectés à la même station.

$$\sum_{r \in e} x_{rk} \leq |e| - 1, \qquad e \in ES, k \in \bigcap_{s \in D_t} K(s), \tag{2.11}$$

Chacune des stations est contrainte d'exécuter au plus n_0 blocs, ceci est exprimé par :

$$\sum_{r \in \{t \in \mathbf{R} | k \in K(t)\}} x_{rk} \leq n_0, \qquad k = 1, \ldots, m_0, \tag{2.12}$$

Les contraintes sur le nombre de stations sont données par :

$$y \geq k x_{rk}, \qquad r \in R^*, k \in K(r), k \geq m^*. \tag{2.13}$$

Le modèle présenté illustre de façon immédiate la complexité des contraintes technologiques à prendre en compte lors de la configuration des systèmes d'usinage.

2.2.2 Procédures par Séparation et Évaluation (PSE)

L'approche par Séparation et Évaluation (Branch and Bound ou B&B) [Land & Doig 1960] est l'une des méthodes les plus utilisées pour la résolution exacte

des nombreux problèmes d'optimisation combinatoire. Elle met en place une énumération intelligente de l'espace de solutions, soit en découpant l'espace initial de recherche en domaines de plus en plus restreints afin d'isoler l'optimum global (principe de séparation). Elle peut être efficace pour des problèmes de taille moyenne, mais pour des problèmes de grande taille, le temps de résolution peut devenir prohibitif. Son efficacité dépend essentiellement du choix des critères de séparation et d'évaluation et des bornes. Un bon choix permettra à l'algorithme de réduire l'arbre de recherche en évitant la construction de certaines branches. Dans le pire des cas, un mauvais choix peut conduire à l'exploration de tout l'espace de recherche.

Parmi les procédures de ce type proposées pour résoudre le problème d'équilibrage de lignes de production, les plus connues sont : EUREKA, FABLE, SALOME et OptPack. Elles sont brièvement présentées ci-dessous.

La méthode EUREKA [Hoffmann 1992; Dolgui & Pashkevich 2001] calcule une borne inférieure du nombre de stations. A chaque branchement, elle choisit un ensemble d'opérations qui formera une nouvelle station. Le temps mort pour la station courante est calculé. Pour savoir s'il faut ou non continuer le branchement, le temps mort cumulé (la somme des temps morts de toutes les stations déjà créées) est comparé avec le temps mort théorique. Ce dernier est obtenu en utilisant la borne inférieure du nombre de stations. L'algorithme essaie de trouver une solution avec un nombre de stations égal à la borne inférieure, si cela n'est plus possible, alors la borne est incrémentée. Cet algorithme est orienté « station » car à chaque branchement une station est créée.

La procédure FABLE (Fast Algorithm for Balancing Lines Effectively) a été proposée par [Johnson 1988]. C'est une procédure par séparation et évaluation où les branchements se font par l'ajout d'une nouvelle opération parmi les candidats à la station courante. S'il n'existe pas de telle opération, alors la charge de la station courante ne peut être augmentée et une nouvelle station est créée qui devient la station courante. Une fois toutes les opérations affectées, une solution faisable est obtenue. Contrairement à Eureka, l'algorithme Fable est orienté " opération ", puisqu'à chaque branchement dans l'arbre de décision une opération est choisie.

SALOME (Simple Assembly Line Balancing Optimization Method) est un algorithme proposé par Scholl et Klein (1997), on le retrouve également dans Scholl (1999), il combine les approches utilisées dans FABLE et EUREKA. À chaque nœud i de l'arbre de décision, une borne inférieure locale du nombre de stations est calculée et on obtient le temps mort à ne pas dépasser pour les nœuds suivants. Ce temps indique si le nombre de stations est suffisant ou non. Car s'il est négatif, on ajoute une station (comme dans FABLE). SALOME utilise également d'autres

règles comme les préfixes dynamiques (Dynamic Prefixing), voir (Scholl, 1999). Le préfixe dynamique est une règle de dominance et de réduction basée sur le calcul des stations au plus tôt et au plus tard auxquelles on peut affecter une opération. Grâce à cette règle, les opérations qui ne peuvent pas être assignées à la station suivante sont automatiquement affectées à la station courante avant la mise en place de la procédure d'énumération.

OptPack (Nourie et Venta, 1991) est une autre PSE qui est plus rapide que les autres pour des instances simples. Notons que selon Scholl et Klein (1999), les instances simples sont celles résolues par la majorité des PSE avec un temps de calcul de quelques secondes. Comme FABLE, OptPack assigne une opération à la fois (branchement est orienté opération). FABLE n'utilise pas d'heuristiques pour améliorer la recherche dans l'arbre d'énumération, alors qu'OptPack utilise l'heuristique de Hoffmann (1992).

D'autres méthodes PSE avancées ont été récemment proposées, notamment ABSALOM [Scholl *et al.* 2010] et "A Branch, Bound, and Remember Algorithm" [Sewell & Jacobson 2012].

2.2.3 Programmation Dynamique (PD)

C'est une méthode classique de résolution exacte qui peut-être utilisée pour un grand nombre de problèmes d'optimisation. Inventée à la fin des années 1940 par le professeur Richard Bellman, elle s'applique à des problèmes d'optimisation dont la fonction objectif est additive. Un exemple de ce type de problème est le plus court chemin entre deux sommets d'un graphe.

L'idée de base est d'éviter de calculer deux fois la même chose, généralement en utilisant une table de résultats déjà calculés, remplie au fur et à mesure qu'on résout les sous-problèmes. On commence par résoudre les petits sous-problèmes et en combinant leurs solutions, on obtient les solutions des sous-problèmes les englobant.

Plusieurs travaux ont été publiés concernant les problèmes d'équilibrage des lignes de production utilisant la programmation dynamique, souvent basée sur la recherche du plus court chemin sous contraintes, par exemple, [Jackson 1956; Held *et al.* 1963; Bard 1989; Gutjahr & Nemhauser 1964; Easton *et al.* 1989; Erel & Gokcen 1999; Dolgui *et al.* 2006a; Dolgui *et al.* 2008].

2.3 Méthodes d'optimisation approchées

Une méthode approchée est un algorithme d'optimisation qui a pour but de trouver une solution réalisable de la fonction objectif, mais sans garantie d'optimalité. Le principal avantage de ces méthodes est qu'elles peuvent s'appliquer à n'importe quelle classe de problèmes.

La Figure 3.3 présente les différentes approches employées pour la résolution mono-objectif des problèmes de configuration ou d'équilibrage des lignes de production. La liste des méthodes évoquées n'est pas exhaustive mais elle regroupe les méthodes les plus répandues.

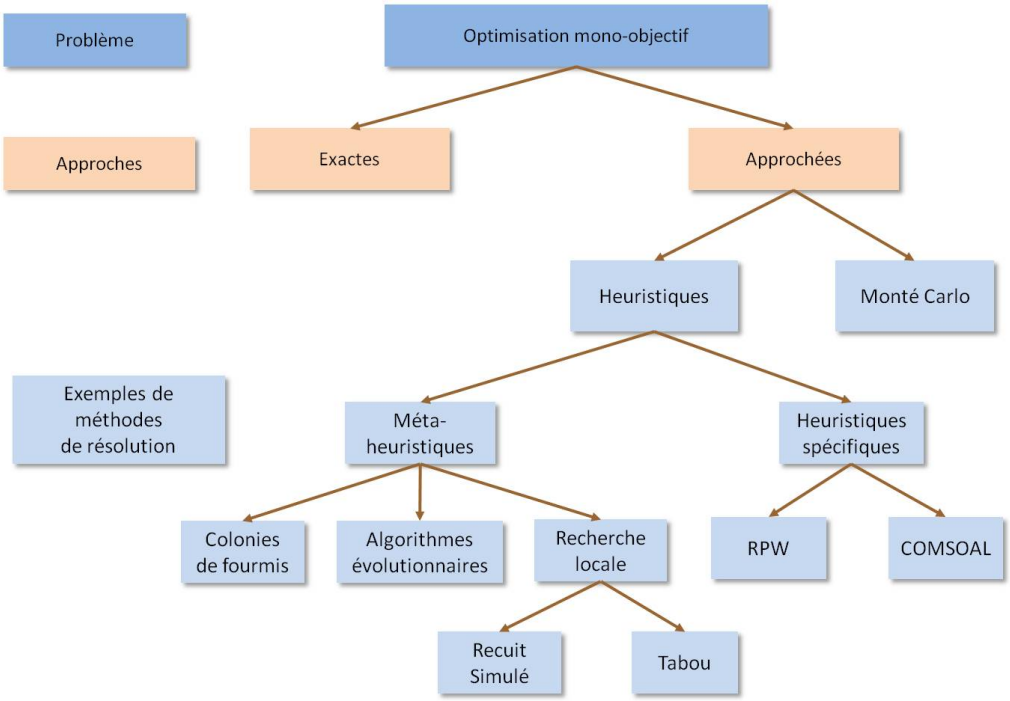


FIGURE 2.3 – Optimisation mono-objectif : méthodes approchées

2.3.1 Heuristiques

Une heuristique est un algorithme intuitif qui permet de déterminer une bonne solution à un problème difficile assez rapidement mais sans pouvoir garantir son optimalité. Quand cet algorithme est applicable sur une famille de problèmes différents, mais pouvant être formalisés de manière semblable, il est appelé méta-heuristique.

Les méthodes heuristiques les plus connues dans le domaine d'équilibrage des lignes de production sont RPW (Ranked Positional Weight) [Helgeson & Birnie 1961] et COMSOAL (COmputer Method for Sequencing Operations for Assembly Lines) [Arcus 1965].

L'idée de l'heuristique RPW est d'affecter les opérations en fonction de leur poids. Le poids d'une opération est calculé comme la somme des temps de traitement de cette opération et de celles qui la succèdent. Les opérations sont classées dans un ordre décroissant de ces pondérations et sont considérées dans cet ordre pour l'affectation aux stations. Donc les opérations qui ont le plus de poids sont prioritaires et sont affectés aux premières stations de la ligne.

La méthode COMSOAL (COmputer Method for Sequencing Operations for Assembly Lines) a été proposée par [Arcus 1965]. Cette heuristique intègre des choix aléatoires des opérations pour leur affectation aux stations. A chaque étape de l'algorithme COMSOAL, on détermine l'ensemble des opérations qui peuvent être affectées. Tous les prédécesseurs de telles opérations doivent être déjà affectés et le temps d'exécution de chacune de ces opérations doit être inférieur au temps restant de la station courante. Une opération de cet ensemble est sélectionnée au hasard et affectée à la station courante. Si l'ensemble des opérations candidates est vide, on crée une nouvelle station qui devient la station courante. Une itération est terminée s'il n'y a plus d'opérations à affecter. Si la solution obtenue à la fin de l'itération est meilleure que la meilleure solution connue jusque là, alors elle devient la meilleure solution et on passe à l'itération suivante en recommençant dès le début l'affectation des opérations. Cet algorithme est répété jusqu'à une condition d'arrêt et la meilleure solution trouvée est retenue.

Des études comparatives des performances des différentes heuristiques pour l'équilibrage des lignes d'assemblage ont été effectuées dans [Capacho *et al.* 2009; Talbot *et al.* 1986]. Pour les lignes d'usinage, des heuristiques de type COMSOAL ont été proposées dans [Dolgui *et al.* 2005a].

2.3.2 Méta-heuristiques

Les méta-heuristiques forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation complexes qui ne peuvent pas être abordés efficacement par une autre méthode. Dans le domaine de l'équilibrage des lignes d'assemblage, plusieurs méta-heuristiques ont montré leurs avantages en termes d'efficacité et de la qualité des solutions fournies, les plus connues sont les algorithmes évolutionnaires, la colonie de fourmis, le recuit simulé et la méthode Tabou.

Les Algorithmes Evolutionnaires (AE) sont des algorithmes d'optimisation inspirés du paradigme darwinien de l'évolution naturelle. Selon ce paradigme, les individus les plus aptes survivent à la sélection naturelle et se reproduisent d'une génération à l'autre. En termes d'optimisation, l'évolution se traduit par un processus itératif de recherche de l'optimum dans l'espace de recherche. Il existe plusieurs familles historiques d'algorithmes évolutionnaires qui se sont développées de façon indépendante : La Programmation Evolutionnaire (EP) introduite par L.J. Fogel [Fogel *et al.* 1966] dans les années 60, les algorithmes génétiques (GA) proposés par J.Holland [Holland 1975] aux USA, et popularisés un peu plus de dix ans plus tard par son élève D.E. Goldberg [Goldberg & Holland 1988], les stratégies d'évolution (ES) inventées par I.Rechenberg [Rechenberg 1973] dans les années 70 à Berlin, et enfin la programmation génétique (GP pour Genetic Programming) proposée par J.Koza [Koza 1992; Koza 1994]. Dans le cadre de l'équilibrage de lignes de d'assemblage, les méthodes évolutionnistes comme les méthodes évolutionnaires [Nearchou 2007; Nearchou 2008] ou les algorithmes génétiques sont très souvent employés ([Anderson & Ferris 1994; Leu *et al.* 1994; Rubinovitz & Levitin 1995; Kim & Kim 1996; Kim *et al.* 1998; Kim *et al.* 2000; Ponnambalam *et al.* 2000; Simaria & Vilarinho 2004; Sabuncuoglu *et al.* 2000; Baykasoğlu & Özbakır 2007; Levitin *et al.* 2006; Haq *et al.* 2006; Kulak *et al.* 2008; Gen *et al.* 2009; Hwang & Katayama 2009; Kim *et al.* 2009]. Une comparaison des algorithmes génétiques proposés pour les différentes formulations du problème d'équilibrage des lignes d'assemblage a été présentée dans [Tasan & Tunali 2008].

Les algorithmes de colonies de fourmis font partie des métaheuristiques de type distribué. Leur principe repose sur le phénomène d'auto-organisation d'un système par la coopération, directe ou indirecte, des composants élémentaires du système. De ce fait, ils peuvent être considérés comme un système multi-agents à part entière, où chaque fourmi constitue un agent autonome communicant. Ce genre de système est inspiré de l'étude du comportement des fourmis et constitue une nouvelle approche multi-agents efficace pour la résolution de problème d'optimisation combinatoire [Dorigo 1992]. Les travaux suivants ont employé l'algorithme de colonies de fourmis pour résoudre les problèmes d'équilibrage des lignes d'assemblage : [McMullen & Tarasewich 2003; McMullen & Tarasewich 2006; Bautista & Pereira 2007; Baykasoglu & Dereli 2008; Sabuncuoglu *et al.* 2009; Simaria & Vilarinho 2009].

Le recuit simulé s'appuie sur l'algorithme de Metropolis [Metropolis *et al.* 1953], qui permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie du système et la température sera un paramètre important de l'algorithme. Le principe général du recuit simulé est de parcourir de manière itérative l'espace des solutions. Une solution décrivant l'état initial sera modifiée afin d'en obtenir une seconde qui sera soit

acceptée et servira de base pour calculer la prochaine itération, soit refusée avec une probabilité. Si cette nouvelle solution améliore la fonction objectif, elle aura permis de diminuer l'énergie du système ; si celle-ci la dégrade, l'énergie du système augmentera (recuit). La probabilité d'accepter une solution moins bonne dépend de la température : plus la température est basse plus la probabilité d'accepter une solution moins bonne est petite. Le même procédé est répété itérativement jusqu'à ce qu'une condition d'arrêt soit vérifiée. L'algorithme du recuit simulé a été abordé dans plusieurs travaux traitant les problèmes d'équilibrage des lignes d'assemblage comme par exemple, ([McMullen & Frazier 1998; Vilarinho & Simaria 2002; Erel *et al.* 2001; Mendes *et al.* 2005; Baykasoglu 2006; Kara *et al.* 2007].

La méthode de Recherche Tabou a été la première à avoir porté le nom de "métaheuristique". Elle a été développée par Fred Glover en 1986 [Glover 1986; Glover 1989; Glover 1990]. C'est une méthode de recherche locale à mémoire adaptative. Sa caractéristique principale est de mémoriser des solutions, ou des caractéristiques de solutions, visitées durant la recherche, afin d'éviter de rester piégé dans des optimums locaux. Si aucune solution voisine n'améliore la solution courante, alors on peut choisir une solution détériorant la fonction objectif pour sortir de l'optimum local. La mémoire, appelée liste tabou, contient la liste des solutions récemment visitées, et permet d'éviter de retomber en permanence dans l'optimum local duquel on vient de sortir. Ce procédé simple permet de se diriger vers d'autres régions de l'espace des solutions, car à chaque itération, on choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction objectif. Plusieurs auteurs se sont intéressés à cette méthode pour résoudre le problème d'équilibrage des lignes d'assemblage [Chiang 1998; Lapierre *et al.* 2006; Özcan & Toklu 2009c].

2.4 Optimisation multi-objectif

L'optimisation multi-objectif trouve ses origines au cours du XIXème siècle dans les travaux en économie d'Edgeworth [Edgeworth 1889] et puis en XX siècle dans ceux de Pareto [Pareto 1964]). On peut formaliser un problème d'optimisation multi-objectif par la recherche d'une solution $x = (x_1, ..., x_n)$ optimisant m fonctions objectifs :

$$f(x) = (f_1(x), f_2(x), \ldots, f_m(x));$$

On dit alors que $f(x)$ est le vecteur de fonctions objectif et f_i les objectifs à optimiser avec $m \geq 2$, le nombre d'objectifs. Ces objectifs sont concurrents, c'est-à-dire l'amélioration de l'un entraîne la détérioration de l'autre et vice-versa. Pour les problèmes multi-objectifs, on ne parle pas de solutions optimales, mais de so-

lutions non dominées ou Pareto optimales. Une solution domine une autre si elle est meilleure sur au moins un critère et au moins aussi bonne sur les autres. Les solutions qui ne sont dominées par aucune autre, forment l'ensemble des solutions non dominées (front Pareto).

Les méthodes utilisées dans la littérature pour répondre à un problème d'optimisation multi-objectif peuvent être classées dans les trois catégories suivantes :

- méthodes a priori : l'utilisateur intervient en amont de la méthode d'optimisation en affectant des poids ou des priorités à chaque objectif, dans le but de transformer un problème multi-objectif en un problème à un seul objectif (mono-objectif).

- méthodes a posteriori : l'utilisateur intervient en aval de l'optimisation, il dispose d'un ensemble des solutions non dominées et peut choisir celle qui lui semble la plus intéressante,

- méthodes progressives : l'utilisateur affine son choix au fur et à mesure du déroulement de l'optimisation en utilisant des méthodes interactives.

La Figure 2.4 montre des exemples de méthodes de chaque catégorie utilisées dans le contexte de la configuration des lignes de production. Les notations suivantes sont utilisées : SP pour Somme Pondérée, GP pour Goal Programming, NSGA pour Non-Dominated Sorting Genetic Algorithm.

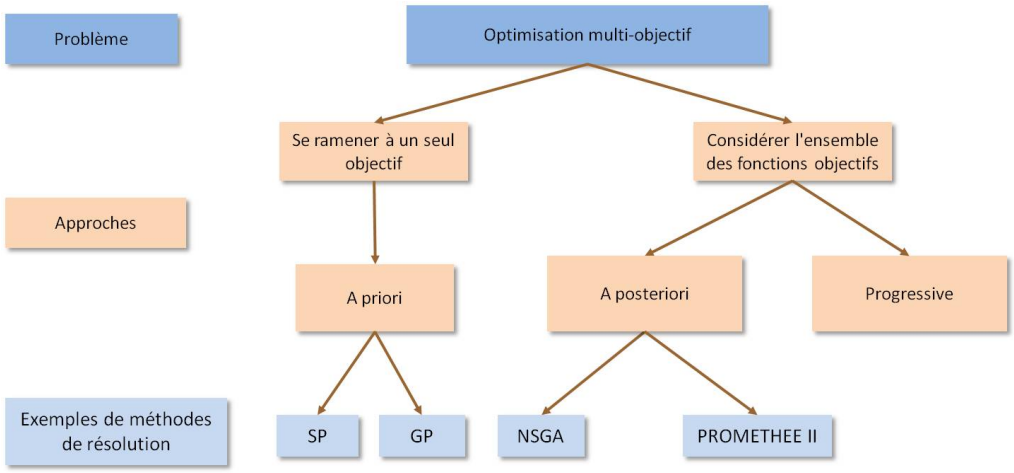


FIGURE 2.4 – Optimisation multi-objectif

2.4.1 Approches a priori

Ces approches sont souvent utilisées pour leur simplicité de mise en œuvre. Elles sont qualifiées d'approches naïves de l'optimisation multi-objectif, car elles transforment un problème multi-objectif en un problème mono-objectif [Collette & Siarry 2002]. Dans ce cas, le décideur détermine a priori l'importance de chaque critère afin de construire une fonction unique. Le processus d'optimisation mono-objectif est ensuite lancé afin d'obtenir la solution optimale. Il est possible aussi d'obtenir des solutions de l'ensemble Pareto si cette approche est appliquée avec différentes valeurs des poids mais il ne faut pas oublier le problème des solutions non supportées du front Pareto qui ne peuvent pas être retrouvées avec cette approche. Par ailleurs, la formulation des préférences utilisateur est délicate et nécessite une connaissance approfondie du problème de départ. Nous donnons ci-dessous des méthodes a priori ou méthodes scalaires les plus connues, d'autres méthodes sont exposées dans [Miettinen 1999].

2.4.1.1 Méthode des sommes pondérées

La méthode des sommes pondérées combine les différentes fonctions objectif dans une fonction linéaire. Ainsi le problème initialement multi-objectif peut être résolu à l'aide des méthodes d'optimisation mono-objectif. Pour combiner les objectifs, il s'agit d'associer à chaque fonction objectif un coefficient de pondération et de calculer ensuite la somme des fonctions objectifs pondérées pour obtenir une nouvelle et unique fonction objectif. La transformation que l'on effectue est la suivante :

$$Minimiser \vec{f}(\vec{x}) \qquad minimiser \sum_{i=1}^{k} w_i \ f_i(\vec{x}) \tag{2.14}$$

$$\longrightarrow$$

$$Avec \ \vec{x} \in S_r \qquad avec \ \vec{x} \in S_r, w_i \geq 0 \ et \sum_{i=1}^{k} w_i = 1 \tag{2.15}$$

Les résultats obtenus avec une telle méthode dépendent des poids choisis.

Plusieurs travaux de recherche ont été effectués pour la résolution du problème d'équilibrage des lignes d'assemblage en utilisant cette approche [McMullen & Tarasewich 2003; McMullen & Tarasewich 2006; Gamberini *et al.* 2009; Hwang & Katayama 2009; Simaria & Vilarinho 2009; Özcan & Toklu 2009c].

Une autre façon de ramener la résolution du problème multi-objectif à celle d'un problème mono-objectif est la méthode ε- contrainte qui consiste à optimiser

un seul objectif prioritaire et jugé important par le décideur et à transformer les autres objectifs en contraintes d'inégalités par rapport à un vecteur seuil. Le problème reformulé s'écrit alors de la façon suivante :

$$minimiser \;\; f_i(x) \;\; tel \;\; que \;\; \forall i \neq j, \;\; f_i(x) \leq \varepsilon_i \tag{2.16}$$

Cependant, les contraintes rajoutées compliquent la résolution du problème.

2.4.1.2 Goal Programming

Une autre méthode très souvent employée est la Méthode de programmation par but (Goal Programming). Initialement conçue par [Charnes & Cooper 1961], elle impose au décideur de définir les buts ou références qu'il désire à atteindre pour chaque objectif. Généralement, le point qui satisfait tous les buts n'est pas réalisable, la solution préférée serait donc celle qui se rapproche le plus possible de ces buts. La méthodologie mise en œuvre repose sur les points suivants :

- fixer les valeurs des cibles que l'on désire atteindre sur chaque critère ;
- définir des déviations positives et négatives relativement à ces buts ;
- minimiser la somme pondérée de ces déviations relativement à une norme définie ; les déviations sont respectivement appelés les sur-réalisations (d^+) et les sous-réalisations (d^-) des critères.

La méthode de Goal Programming (GP) résout donc un problème de minimisation d'une seule fonction. Dans le modèle GP, on distingue les contraintes du système introduites de la même façon que dans les modèles de PLNE et les contraintes dites "goal" qui portent sur les valeurs cibles. La méthode est facile à implémenter mais la définition des buts à atteindre est souvent difficile. L'objectif de la méthode est que la solution qui en résulte satisfasse le décideur au mieux. C'est réalisable lorsque les buts et les priorités de chaque but à atteindre sont bien définis. Sinon, la théorie des nombres flous est utilisée pour les modèles GP dits "flous". Pour plus de détails, voir [Miettinen 1999; Coello 1998].

Dans la littérature, l'approche Goal Programming a montré son efficacité pour certains problèmes de configuration des lignes de production que nous citons ci-dessous.

Le premier modèle de GP pour le problème d'équilibrage des lignes d'assemblage a été proposé dans [Deckro & Rangachari 1990]. Les buts considérés étaient de limiter le nombre d'opérations affectées à une station, respecter au maximum

les contraintes et respecter le temps de cycle (ou le minimiser). Différents ordres de priorité ont été testés pour ces buts. Gökçen et Erel [Gökçen & Erel 1997] ont étendu le modèle proposé au cas où plusieurs modèles de produit sont assemblés sur la même ligne.

Le modèle GP pour l'équilibrage des lignes d'assemblage en forme de U a été proposé dans [Gökçen & Agpak 2006]. Les objectifs considérés étaient le nombre de stations, la charge de chaque station et le nombre d'opérations affectées à chaque station. Un modèle de GP flou pour les mêmes critères a été développé par Toklu and Özcan [Toklu & Özcan 2008]. Le problème de l'équilibrage des lignes d'assemblage en forme de U a été également étudié par Kara et al. [Kara *et al.* 2009] qui ont proposé un modèle GP avec deux critères différents, à savoir : le nombre de stations et le temps de cycle.

Ağpak et Gökçen [Ağpak & Gökçen 2007] ont utilisé la méthode GP pour résoudre le problème d'équilibrage des lignes d'assemblage (en ligne droite et en forme de U) avec les temps stochastiques. L'objectif était d'équilibrer la charge des stations en fonction des leurs valeurs moyennes et leurs variances.

Özcan et al.[Özcan & Toklu 2009b] ont proposé des modèles GP pour le problème d'équilibrage des lignes à deux côtés. Comme critères d'optimisation, ils ont considéré le nombre de stations, le nombre de postes de travail (à droite et à gauche de chaque station), le temps de cycle et le nombre d'opérations affectées à chaque station.

Choi [Choi 2009] a développé un modèle GP pour l'équilibrer la charge des opérateurs en termes de temps de travail, mais aussi en considérant les paramètres ergonomiques comme la pénibilité du travail, l'effort physique, etc.

Kara et al. [Kara *et al.* 2010] ont proposé des modèles GP et GP flou pour l'équilibrage des lignes d'assemblage parallèles. Les trois critères d'optimisation ont été considérés : le nombre de stations, le temps de cycle et le nombre d'opérations affectées à chaque station.

Kara et al. [Kara *et al.* 2011] ont développé des modèles GP et GP flou pour l'équilibrage des lignes d'assemblage fabriquant différents modèles de produit. En considérant les conflits de précédence entre les différents modèles et les coûts liés à la duplication des opérations communes, les critères suivants ont été définis comme buts : le nombre de stations, le temps de cycle et le coût total induit par la duplication des opérations.

2.4.2 Approches a posteriori

Dans les méthodes a posteriori, les décisions sont prises d'après un ensemble de solutions connues. Dans ces approches, l'expression des préférences et la pondération des objectifs se font a posteriori, ce qui signifie que le problème d'optimisation détermine un ensemble de solutions candidates et la sélection de la meilleure ne se fait qu'à la fin.

Une des méthodes de cette classe NSGA-II a été appliquée avec succès pour l'équilibrage des lignes de transfert avec choix d'équipements dans [Makdessian *et al.* 2008].

L'algorithme NSGA-II est constitué de plusieurs étapes dont quelques-unes sont communes à celles de l'algorithme génétique classique comme par exemple, l'initialisation de la population et les opérateurs de la production génétique (croisement et mutation). La structure générale de NSGA-II consiste à : initialiser une nouvelle population ; trier les solutions de cette population et les placer dans des fronts.

Pour un problème d'optimisation bi-critère, où f_1 et f_2 sont les critères à optimiser, si $i \in front(k)$, ceci veut dire que la solution $pop(i)$ est assignée au front de niveau k. Le rang est un vecteur où $rang(i)$ représente l'indice du front dans lequel la solution $pop(i)$ est classée [Lacomme *et al.* 2006].

Le tri se fait de telle sorte que les meilleures solutions occupent toujours le premier front et ainsi de suite pour les autres solutions. Donc, le rang numéro 1 est attribué aux meilleures solutions. On commence à remplir le premier front. Les solutions sont comparées deux à deux pour vérifier la dominance. Soit x la solution considérée. Si x domine y, on ajoute y à l'ensemble des solutions qui sont pires que x. Sinon, si y domine x alors le nombre des solutions meilleures que x augmente de 1. Cette procédure se répète jusqu'à épuiser toute la population. Si aucune solution n'est meilleure que x, alors x appartient au premier front. La procédure continue pour les autres solutions et fournit une répartition de toutes les solutions dans des fronts en attribuant un rang à chaque solution. Ensuite, pour toutes les solutions appartenant au même front, la distance d'encombrement est calculée pour chaque paire des solutions (Crowding Distance Assignment en anglais).

De nouvelles solutions sont générées par une sélection des parents et l'application des opérateurs génétiques (croisement, mutation, etc.). Les parents sont choisis par la méthode de tournoi binaire. Dans l'algorithme multicritère, le fitness indique le rang. Entre deux individus, celui qui appartient au rang le plus petit (ou si les deux sont du même rang, celui qui a la distance d'encombrement la plus grande)

est choisi. On préfère donc les points les plus isolés dans chaque front, pour mieux couvrir l'ensemble Pareto optimal.

Il y a aussi les méthodes de surclassement : ces méthodes sont utilisées pour comparer un ensemble de solutions admissibles, souvent obtenues par une méthode de résolution approchée. Par exemple, dans ([Rekiek *et al.* 2001; Rekiek & Delchambre 2006]) la méthode PROMETHEE II [Brans & Mareschal 1994] a été appliquée pour comparer les individus d'une génération créée par un algorithme génétique. Cette méthode établit un processus de comparaison numérique de chaque solution par rapport à toutes les autres solutions. Ainsi il est possible de calculer le plus (mérite) ou le moins (démérite) de chaque solution par rapport à toutes les autres. Le résultat de cette comparaison permet un classement ordonné des solutions. Un autre exemple est la méthode TOPSIS (Total Order Preference by Similarity to the Ideal Solution) [Gamberini *et al.* 2006], où toutes les solutions sont comparées à la solution idéale.

2.4.3 Approches progressives

Dans les méthodes progressives, les processus de décision et d'optimisation sont alternés, permettant à l'utilisateur d'orienter la recherche de façon interactive en fonction de ses préférences et des solutions déjà obtenues. Ainsi l'espace des solutions peut être réduit successivement en recentrant les intervalles d'acceptation ainsi que les poids des objectifs. Ce type de démarche exige une expertise approfondie du problème traité ainsi que des méthodes employées.

Les méthodes les plus connues de cette classe sont les méthode STEM et celle de Steuer ([Coelho & Breitkopf 2009; Jones *et al.* 2002]). La première met en œuvre une suite successive de formulations Min-Max en minimisant la distance entre une solution candidate et une solution idéale $S*$. A chaque itération, si le décideur n'est pas satisfait par la solution obtenue, il peut relaxer certaines contraintes du problème, modifier les poids ou les objectifs eux-mêmes, voire rajouter des contraintes supplémentaires au problème.

Dans la méthode de Steuer, à chaque itération, un ensemble de solutions est proposé au décideur qui en choisit une ou bien établit un classement par ordre de préférence. Ceci permet de recalculer automatiquement les poids des objectifs. Cette méthode représente une alternative intéressante mais elle s'avère coûteuse en temps de calcul et se repose sur des outils très spécifiques.

Notre revue de la littérature scientifique ne nous permet pas de citer d'exemple

d'utilisation des méthodes de cette classe pour les problèmes d'équilibrage de lignes de production.

2.5 Conclusion

Dans ce chapitre, nous avons présenté les méthodes classiques d'optimisation mono- et multi-objectif. Des exemples de leur utilisation pour la résolution des problèmes de configuration et d'équilibrage des lignes de production ont également été donnés.

Dans le chapitre suivant, nous formulerons le problème de reconfiguration des lignes de production et nous le comparerons au problème de la configuration initiale afin de conclure si les méthodes de résolution développées pour le dernier problème pourraient être réutilisées pour la résolution efficace du premier.

Chapitre 3

Reconfiguration des lignes de production

Sommaire

3.1 Reconfiguration dans le cycle de vie des lignes de production 47
3.2 Évaluation de la reconfiguration des systèmes de production 48
3.3 Reconfiguration des lignes de transfert 50
3.3.1 Réutilisabilité des boîtiers multibroches 51
3.3.2 Réutilisabilité des machines 52
3.4 Reconfiguration des lignes d'assemblage 52
3.5 Conclusion 54

3.1 Reconfiguration dans le cycle de vie des lignes de production

Afin de réussir sur le marché mondial, les fabricants doivent être capables de répondre rapidement aux changements de la demande en adaptant leurs systèmes de production [Carlsson *et al.* 2010]. Ainsi la compétition n'exige pas seulement la haute productivité des systèmes de production mais aussi une réactivité aisée aux changements imposés dans les caractéristiques, volumes ou variantes du produit [Schuh *et al.* 2005]. Tout au long du cycle de vie d'un système de production, une nouvelle conception complète ou partielle des ses composantes et/ou de sa structure pourrait être nécessaire afin de répondre aux changements dans son environnement interne et/ou externe. Par exemple, après l'implémentation d'une ligne de production, sa reconfiguration peut devenir nécessaire si le produit ou les contraintes de production changent (Figure 3.1).

La reconfiguration comprend les actions d'ajout, de suppression et/ou de réorganisation de façon rentable des composants et des fonctions d'un système [Farid 2008]. Selon Deif et ElMaraghy [Deif & ElMaraghy 2007], la reconfiguration peut être physique, logique ou humaine.

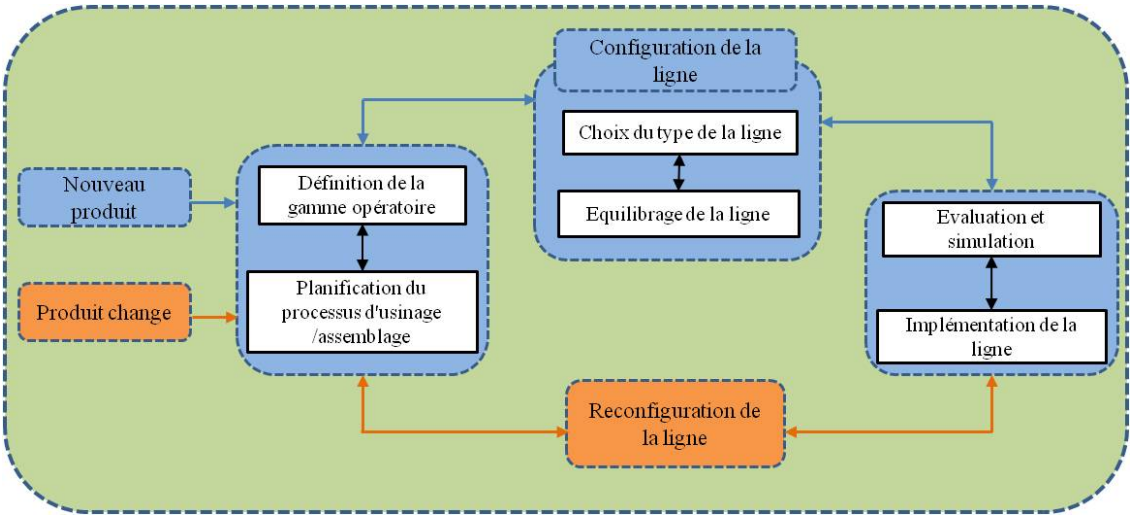


FIGURE 3.1 – Processus de configuration/reconfiguration des lignes de production

La **reconfiguration physique** comprend à la fois les systèmes techniques et de manutention.

La **reconfiguration logique** concerne les systèmes d'information (la reprogrammation des machines, re-planification, etc).

Une **reconfiguration humaine** implique la réallocation des ressources humaines (administrateurs, opérateurs, contrôleurs) ou la réaffectation des tâches de travail.

Les trois types de reconfiguration sont interconnectés entre eux comme le montre la Figure 3.2 [Rösiö 2012].

3.2 Évaluation de la reconfiguration des systèmes de production

Plusieurs métriques ont été identifiées pendant les dernières décennies dans la littérature pour évaluer l'efficacité de la reconfiguration d'un système de production. Dans ce qui suit, nous présentons les modèles principaux.

Ku et al. [Ku 2001] ont présenté un modèle pour la gestion des changements et ont pris pour critère d'efficacité le temps de la reconfiguration.

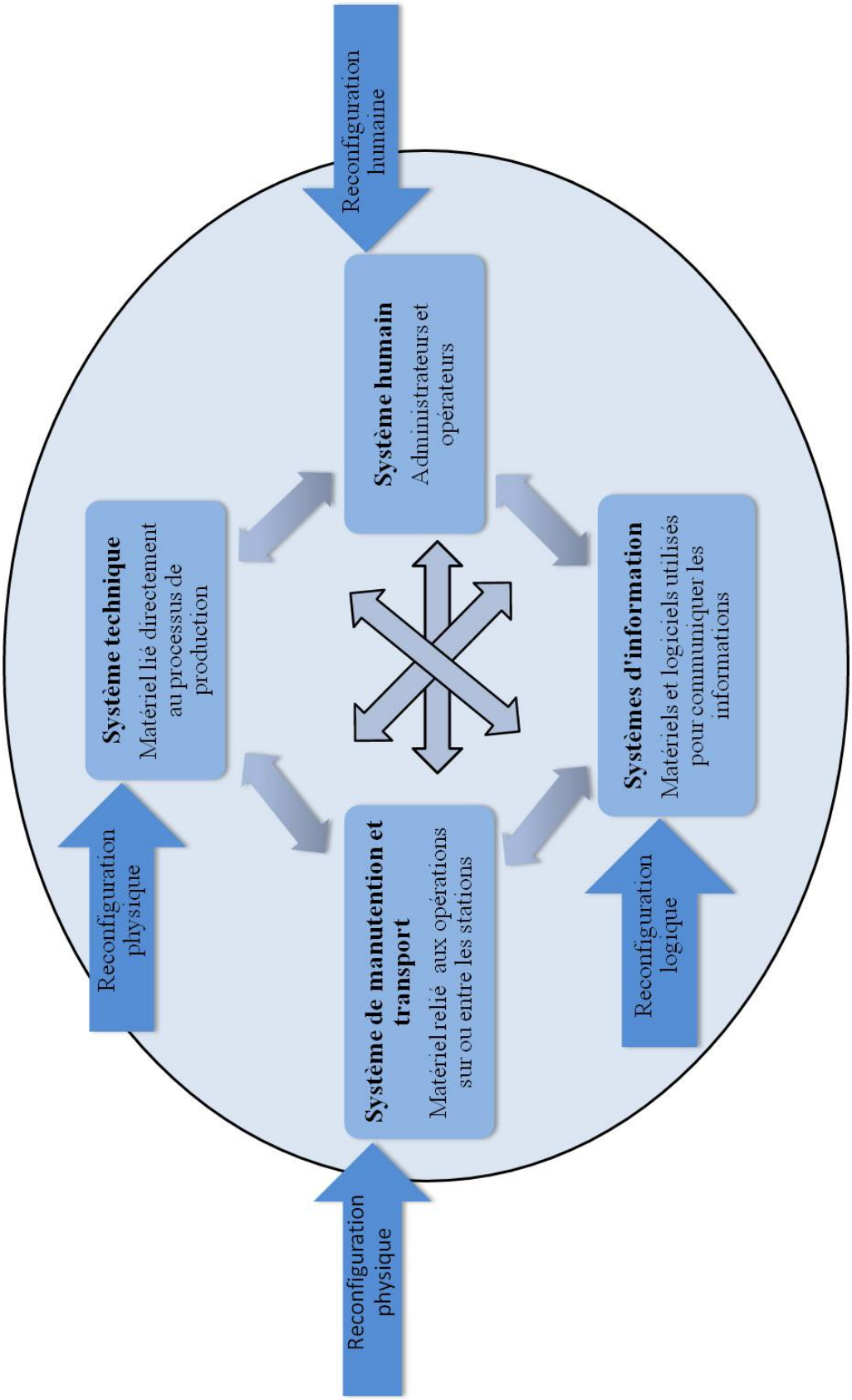


FIGURE 3.2 – Reconfiguration physique, logique et humaine

Youssef et ElMaraghy [Youssef & ElMaraghy 2006] ont proposé une métrique relative des coûts, du temps et des efforts nécessaires pour passer d'une configuration à une autre. Cette métrique se compose de trois éléments représentants les différents niveaux de la reconfiguration, à savoir : la reconfiguration au niveau du marché, au niveau du système et au niveau des machines.

Youssef et ElMaraghy [Youssef & ElMaraghy 2008] ont suggéré un modèle pour évaluer la disponibilité et les taux de production attendus pour les systèmes de production qui sont composés d'unités de production modulaires. Ils ont traité ces unités comme des modules indépendants, où l'arrêt d'une unité ne nécessite pas l'arrêt de l'ensemble de machines ni de sa production.

Hon et Xu [Hon & Xu 2007] ont étudié à travers la reconfiguration la relation entre le cycle de vie des produits d'une famille et les performances du système de production. Ils ont présenté la stratégie d'optimisation qui comprend les trois étapes suivantes : la simulation de la configuration originale, l'optimisation du système reconfiguré et l'optimisation du portefeuille des produits.

Abbasi et Houshm [Abbasi & Houshmand 2009] ont proposé une méthodologie pour ajuster rapidement et efficacement les capacités de production et la fonctionnalité du système à la demande du marché. Un modèle de programmation linéaire en variables mixtes a été développé pour déterminer la séquence optimale des tâches de production, les configurations correspondantes et les tailles des lots. Une procédure basée sur la recherche tabou a été utilisée pour résoudre ce problème d'optimisation.

Les particularités de chaque système de production exigent une modification de ces métriques afin de mieux refléter l'état du système et donc permettre une prise de décision plus efficace. Dans ce qui suit, nous considérons de façon plus détaillée les problèmes décisionnels liés à la reconfiguration des lignes de transfert (Section 3.3) et des lignes d'assemblage (Section 3.4).

3.3 Reconfiguration des lignes de transfert

Une ligne de transfert doit être reconfigurée si un nouveau produit doit être fabriqué par une ligne existante ou il y a eu des changements dans les caractéristiques du produit. Dans de tels cas, les nouvelles opérations doivent être intégrées aux équipements existants tandis que certaines opérations précédemment exécutées sont supprimées. Ce type de problème de reconfiguration est un problème difficile à cause des nombreuses contraintes de compatibilité entre les anciennes et les nouvelles opérations à prendre en considération. Or, l'analyse de la littérature a montré

que le problème de la reconfiguration des lignes de transfert n'a pas été abordé dans la littérature pour le moment.

La prise en compte des équipements existants en même temps que la conception de nouveaux équipements rend ce problème original par rapport au problème d'équilibrage des lignes de transfert (Transfert Line Balancing Problem, TLBP) [Dolgui *et al.* 1999] où tous les équipements sont à concevoir.

L'objectif considéré dans ce mémoire pour la reconfiguration des lignes de transfert est de minimiser les coûts nécessaires pour modifier la ligne initiale afin de suivre le nouveau processus de production. Ces coûts sont dus à la modification de certains boîtiers et à l'installation de nouvelles machines et boîtiers. Par conséquent, il est souhaitable de réutiliser le plus possible les équipements existants pour minimiser ces coûts. Toutefois, les conditions de réutilisation des boîtiers multibroches et des machines doivent être prises en compte.

En étudiant le processus de reconfiguration chez nos partenaires industriels, nous avons pu voir que différentes hypothèses sur la réutilisabité des équipements existants peuvent être valides techniquement dans la pratique. Le choix des hypothèses à utiliser se fait au cas par cas.

3.3.1 Réutilisabilité des boîtiers multibroches

Ici nous présentons les hypothèses recensées de la plus restrictive à la plus permissive :

Hypothèse B1. Si au moins une opération est supprimée d'un boîtier multibroche alors il devient obsolète et ne peut plus être réutilisé.

Hypothèse B2. Des opérations peuvent être supprimées d'un boîtier multibroche mais aucune opération ne peut être ajoutée à un boîtier multibroche existant.

Hypothèse B3. Des opérations peuvent être supprimées et de nouvelles opérations peuvent être ajoutées à un boîtier multibroche existant si toutes les contraintes d'exclusion sont respectées.

Hypothèse B4. Des opérations peuvent être supprimées et de nouvelles opérations peuvent être ajoutées à un boîtier multibroche existant mais l'utilisation de chaque nouvel outil exige un coût supplémentaire.

3.3.2 Réutilisabilité des machines

Ici également, nous commençons par le cas le plus restreint, puis augmentons la flexibilité de la réutilisabilité :

Hypothèse M1. Aucun boîtier multibroche ne peut être ajouté à des machines existantes mais d'anciens boîtiers multibroches peuvent être supprimés.

Hypothèse M2. De nouveaux boîtiers multibroches peuvent être installés sur des machines existantes mais ils ne peuvent être activés qu'après les anciens boîtiers multibroches.

Hypothèse M3. Des boîtiers multibroches existants peuvent être supprimés de leur machine initiale et peuvent être installés sur une autre ancienne machine. Seuls de nouveaux boîtiers multibroches peuvent être installés sur une nouvelle machine.

Hypothèse M4. Des boîtiers multibroches existants peuvent être supprimés de leur machine initiale et peuvent être installés sur n'importe quelle machine de la ligne. Un nouveau boîtier multibroche peut être intercalé entre deux anciens.

Dans les chapitres 4 et 5, nous étudierons et comparerons les hypothèses B2 et B3 conjointement avec l'hypothèse M4.

3.4 Reconfiguration des lignes d'assemblage

Falkenauer [Falkenauer 2005] était parmi les premiers à souligner l'importance du problème de la reconfiguration des lignes d'assemblage. En effet, comme dans la pratique la production de la plupart de nouveaux produits se base sur des lignes d'assemblage existantes, le problème de reconfiguration se pose même plus souvent que le problème de conception d'une nouvelle ligne. Comme ces lignes sont souvent manuelles, les coûts liés à la réaffectation du personnel et leur apprentissage doivent être étudiés avec attention, puisqu'ils peuvent constituer une part importante dans le coût d'un système d'assemblage comme le montre la Figure 3.3.

Parmi les premières formulations du problème du rééquilibrage des lignes d'assemblage, on trouve la contribution de Gamberini et al. [Gamberini *et al.* 2006] qui ont étudié la reconfiguration d'une ligne manuelle avec des temps opératoires stochastiques. Un facteur de similitude a été défini pour mesurer la similitude entre la ligne existante et la ligne rééquilibrée. Les objectifs étaient de minimiser les coûts de la main-d'oeuvre et les coûts liés au dépassement du temps de cycle ainsi que de maximiser ce facteur de similitude de la nouvelle solution avec la solution existante.

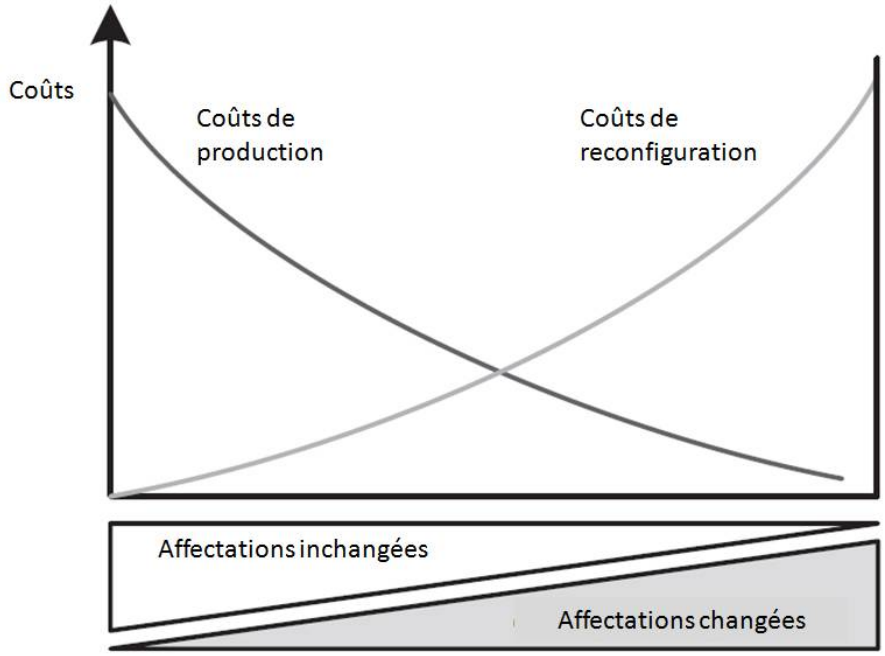


FIGURE 3.3 – Coûts de production vs coûts de reconfiguration

Pour résoudre ce problème multi-objectif, une heuristique de Kottas et Lau [Kottas & Lau 1973] a été couplée avec la méthode "Technique for Order Preference by Similarity to Ideal Solution" (TOPSIS) pour classer les solutions obtenues en variant les poids donnés aux objectifs. Pour le même problème, un algorithme génétique multi-objectif et une heuristique à passages multiples ont été présentés et comparés [Gamberini *et al.* 2009].

Ko et al. [Ko *et al.* 2005] ont présenté une étude détaillée de la réutilisabilité des ressources d'une ligne d'assemblage en utilisant un exemple d'une ligne pour l'assemblage d'un châssis de côté droit d'un véhicule. Dans ce travail, une expression du critère de réutilisabilité est calculée comme le rapport entre les ressources utilisées dans la ligne et les ressources disponibles.

Altemier et al. [Altemeier *et al.* 2010] ont considéré le problème de reconfiguration des lignes d'assemblage dans le contexte de production à la demande d'une grande variété de produits. Partant d'une solution de l'équilibrage de la ligne existante, le but était de minimiser les coûts de la reconfiguration pour adapter la ligne à la production d'un autre type de produit ainsi que les coûts liés au dépassement du temps de cycle. Une approche de la reconfiguration semi-automatique a été développée.

Le problème du rééquilibrage d'une ligne d'assemblage dans le contexte d'emploi des opérateurs saisonniers a été traité dans [Corominas *et al.* 2008]. Les opérateurs temporaires non seulement nécessitaient plus de temps pour exécuter les tâches que les travailleurs permanents, mais ils devaient être accompagnés par des opérateurs permanents pour encadrer leur travail. Le problème du rééquilibrage concernait la réaffectation des tâches entre les opérateurs de deux catégories avec l'objectif de minimiser le nombre d'opérateurs temporaires.

[Grangeon *et al.* 2011] ont considéré un problème du rééquilibrage des lignes de montage dans l'industrie automobile. Trois heuristiques ont été proposées afin de minimiser le nombre de stations, puis améliorer la répartition des charges entre les stations de travail, et ensuite de minimiser le nombre d'opérations transférées. Ces objectifs ont été considérés de façon indépendante afin de fournir au décideur un ensemble de solutions non-dominées.

Une heuristique de type COMSOAL (A Computer Method of Sequencing Operations for Assembly Lines) a été proposée par Agpak [Agpak 2010] pour le problème du rééquilibrage des lignes d'assemblage avec l'objectif de déterminer une affectation des opérations qui peut rester invariable pour différentes valeurs du temps de cycle.

Le problème du rééquilibrage des lignes d'assemblage mixtes avec des demandes saisonnières a été étudié par Yang et al. [Yang *et al.* 2012]. Pour chaque nouveau produit, les opérations devaient être réaffectées en minimisant le nombre de stations utilisées, mais aussi la différence dans la charge d'une station d'un cycle à l'autre et le coût du rééquilibrage dépendant des durées des opérations qui ont été réaffectées. Un algorithme génétique multi-objectif (moGA) a été proposé pour résoudre ce problème.

L'analyse des travaux existants dans la littérature fait apparaître le manque de modèles exacts pour la minimisation des coûts de la reconfiguration des lignes d'assemblage, car le problème du rééquilibrage est souvent résolu à l'aide de méthodes approchées. Pour ces raisons, nous présenterons un tel modèle dans le Chapitre 6.

3.5 Conclusion

Ce chapitre a été consacré à l'état de l'art concernant la problématique de reconfiguration des systèmes de production. Nous avons constaté que dans le contexte actuel de l'instabilité des marchés, les systèmes de production peuvent subir des ajustements et des reconfigurations fréquents. Cependant dans la littérature la re-

configurabilité est principalement traitée dans les travaux sur les systèmes de production reconfigurables (RMS). En effet, RMS constituent une nouvelle classe de systèmes caractérisés par une structure ajustable. Ils sont conçus pour permettre des changements simples dans leur structure physique afin de répondre à la fluctuation dans la demande en termes de volume et de type de produit.

Cependant, les RMS ne sont pas encore bien répandus dans l'industrie. La question se pose néanmoins dans la pratique, comment adapter les systèmes non conçus pour être reconfigurables à une nouvelle réalité du marché ? Le remplacement complet demande des investissements trop importants et bien souvent une reconfiguration du système existant est préférable. Le problème de minimisation des coûts liées à la reconfiguration d'un système de production est donc fréquent dans l'industrie. Or, ce problème, différent de celui de la configuration initiale du fait de l'organisation existante, n'a été que très peu aborde dans la littérature. Pour les lignes d'assemblage, il n'existe que des méthodes approchées et pour les lignes de transfert, aucune étude n'a pas été faite. En partant de ce constat, nous dédions la suite de ce mémoire à l'étude de ces deux problèmes d'optimisation.

Les problèmes décisionnels liés à une reconfiguration des lignes de production dédiées n'ont été que très peu abordés dans la littérature. En partant de ce constant, nous consacrons ce mémoire à l'étude de ces problèmes et au développement des méthodes de résolution adaptées. Dans le chapitre suivant, ce problème est étudié pour les lignes d'usinage à boîtiers multibroches.

Chapitre 4

Modèles MIP pour la reconfiguration des lignes de transfert

Sommaire

4.1 Formulation du problème 58
4.1.1 Contraintes d'affectation des opérations 61
4.1.2 Contraintes de productivité 62
4.1.3 Fonction objectif 62
4.1.4 Modèle mathématique 63
4.1.5 Contraintes de reconfiguration 64
4.2 Exemple illustratif 65
4.2.1 Solution pour le modèle MIP1 71
4.2.2 Solution pour le modèle MIP2 71
4.3 Expérimentations numériques 73
4.4 Conclusion 82

Dans ce chapitre, nous considérons le problème de reconfiguration pour les lignes de transfert à boîtiers multibroches présentées dans la section 1.5.3 du chapitre 1 et illustrées par la Figure 1.7.

Le problème de configuration optimale de ces lignes a déjà été étudié dans la littérature. Trois méthodes exactes ont été développées pour sa résolution. La première consiste à transformer le problème initial en un problème de recherche du plus court chemin sous contraintes dans un graphe spécialement construit [Dolgui *et al.* 1999]. La deuxième approche utilise la modélisation du problème sous forme d'un programme linéaire en variables mixtes (Mixed Integer Programming) [Dolgui *et al.* 2006b]. Enfin, une procédure par séparation et évaluation a été proposée [Dolgui & Ihnatsenka 2009b]. Pour faciliter la résolution exacte, des procédures de pré-traitement et de calcul des bornes inférieures efficaces ont été développées [Battaïa & Dolgui 2012; Battaïa *et al.* 2013].

Comme ce problème TLBP est NP-difficile, des méthodes approchées ont également été proposées pour la résolution des instances de grande taille. Deux méthodes

heuristiques RAP (Recursive Assignment of Predecessors) et FSIC (First Satisfying Inclusion Constraints) ont été développées pour la résolution de telles instances [Dolgui *et al.* 2005b]. Elles se basent sur des règles de priorité. Ces deux heuristiques construisent à chaque itération une seule branche de l'arbre d'énumération qui représente tous les choix possibles. Une fois l'itération terminée, les deux algorithmes reviennent à la racine. Ces méthodes se différencient principalement par la manière de sélectionner une opération dans la liste des opérations candidates quand il y a un choix à faire. L'algorithme FSIC a été amélioré et testé sur un large échantillon de problèmes réels [Guschinskaya & Dolgui 2007; Guschinskaya & Dolgui 2008]. L'approche de type GRASP et l'algorithme génétique ont également été conçus et testés sur des problèmes réels dans [Battaïa *et al.* 2012].

La stabilité des solutions trouvées pour le TLBP a été étudiée dans [Gurevsky *et al.* 2013]. Les auteurs ont proposé des mesures pour évaluer l'aptitude d'une solution à supporter les changements dans les temps opératoires. Une approche heuristique a été développée pour construire une approximation de l'ensemble Pareto des solutions les plus robustes qui minimisent en même temps le coût de l'investissement initial.

Cependant, comme nous l'avons déjà souligné, le problème de la reconfiguration des lignes de transfert n'a pas été abordé dans la littérature et fera donc l'objet de notre étude présentée dans ce chapitre.

4.1 Formulation du problème

Nous supposons que le produit fabriqué par une ligne de transfert existante change pour répondre aux nouvelles exigences normatives ou du marché. Les décideurs disposent donc des données concernant le nouveau produit et les équipements tels que machines et boîtiers multibroches utilisés dans la ligne initiale. Leur objectif est de réaliser une reconfiguration de la ligne existante pour répondre aux nouveaux besoins de production en minimisant les investissements requis par ce changement.

Comme nous l'avons évoqué dans le chapitre 3, plusieurs hypothèses sur la réutilisabilité des boîtiers multibroches peuvent être formulées. Dans ce chapitre, nous utilisons les hypothèses B2 et B3 pour créer et évaluer deux modèles MIP différents :

B2. D'anciennes opérations peuvent être supprimées d'un boîtier multibroche existant, mais aucune nouvelle opération ne peut être ajoutée à un boîtier multibroche existant.

B3. D'anciennes opérations peuvent être supprimées d'un boîtier multibroche existant et de nouvelles opérations peuvent être ajoutées si toutes les contraintes d'exclusion sont respectées. On suppose que le coût d'ajout d'un outil à un boîtier multibroche existant est négligeable par rapport au coût du boîtier.

Les hypothèses B2 et B3 sont contradictoires et elles ne peuvent pas être considérées simultanément. Nous les testons l'une et l'autre séparément.

En ce qui concerne la réutilisabilité des machines, l'hypothèse M4 est appliquée, c'est-à-dire, il est considéré que tous les boîtiers multibroches sont modulaires et peuvent être facilement installés sur n'importe quelle, ancienne ou nouvelle, machine de la ligne.

Les notations utilisées pour la formulation du problème sont résumées comme suit :

- Station de travail : Ensemble des opérations affectées à une machine

  Bloc : Ensemble des opérations affectées à un boîtier multibroche

  n_0 : Nombre maximal de blocs par station de travail

  m_0 : Nombre maximal autorisé de stations dans la ligne

  m^* : Nombre de stations dans la ligne initiale,

- Indices :

  k : Indice d'une station de travail, $k = 1, \ldots, m$

  q : Indice d'un bloc, $q = 1, \ldots, q_0 = m_0 n_0$

  l : Indice de blocs dans la ligne initiale, $l = 1, \ldots, l_0$

- Ensembles :

  N : Ensemble des opérations qui sont nécessaires pour l'usinage de la nouvelle pièce

  $N' \in N$: Ensemble des opérations nécessaires pour l'usinage de la nouvelle

pièce qui existaient déjà pour la pièce initiale

$PredD(j)$: Ensemble de prédécesseurs directs de $j \in N$

$B(k)$: Ensemble d'indices de blocs pour la station $k, B(k) = \{(k-1)n_0 + 1, \ldots, kn_0\}$

$Q(j)$: Ensemble d'indices de blocs q où l'opération j peut être affectée

$K(j)$: Ensemble d'indices des stations k où l'opération j peut être affectée

e : Ensemble des opérations pour représenter un élément des contraintes IS, ES ou EB

$N_q \in N'$: Ensemble des opérations affectées au bloc q dans la ligne existante

- Temps :

  t_j : Temps d'exécution de l'opération $j \in N$

  T_0 : Temps de cycle objectif de la ligne (définit la productivité de la ligne)

  τ^b : Temps auxiliaire nécessaire pour l'activation d'un boîtier multibroche

  τ^s : Temps auxiliaire nécessaire pour le chargement/dé chargement de la pièce sur une machine

- Coûts :

  C_1 : Coût d'une nouvelle machine

  C_2 : Coût d'un nouveau boîtier multibroche

  C_3 : Bénéfice de la réutilisation d'un ancien boîtier multibroche dans la ligne reconfigurée

- Variables de décision :

$$X_{jq} = \begin{cases} 1 & \text{si l'opération } i \text{ est affectée au bloc } q \text{ dans la nouvelle ligne,} \\ 0 & \text{sinon.} \end{cases}$$

$$B_{lq} = \begin{cases} 1 & \text{si le bloc } l \text{ est réutilisé comme bloc } q \text{ dans la nouvelle ligne,} \\ 0 & \text{sinon.} \end{cases}$$

Y_q : Variable auxiliaire qui indique si le bloc q existe dans la nouvelle configuration de la ligne.

Z_k : Variable auxiliaire qui indique si la station de travail k existe dans la nouvelle configuration de la ligne.

Il est à noter que les ensembles N_l contiennent seulement les opérations qui doivent être assignées lors de la reconfiguration : si une opération n'est plus nécessaire pour l'usinage d'un nouveau produit, elle est supprimée de l'ensemble correspondant.

4.1.1 Contraintes d'affectation des opérations

L'affectation des opérations doit respecter les contraintes technologiques liées au processus de production de la nouvelle pièce :

- Les contraintes de précédence entre les opérations. Elles peuvent être exprimées par un graphe orienté $G = (N,D)$. L'arc $(i,j) \in N \times N$ appartient à l'ensemble D si et seulement si l'opération j ne peut pas précéder l'opération i et donc l'opération j doit être attribuée soit au même bloc que l'opération i ou à un bloc qui succède le bloc où l'opération i est assignée.

- Les contraintes d'inclusion qui forcent l'affectation de certaines opérations à la même station. Ces contraintes sont représentées par la famille IS de sous-ensembles de N tels que toutes les opérations du même sous-ensemble $e \in IS$ doivent être affectées à la même station.

- Les contraintes d'exclusion au niveau des stations définissent les groupes d'opérations ne pouvant pas être assignées à la même station. Ces contraintes sont représentées par la famille ES de sous-ensembles de N tels que chaque sous-ensemble $e \in ES$ ne peut pas être assigné à la même station.

- Les contraintes d'exclusion au niveau des blocs définissent les groupes d'opérations qui ne peuvent pas être assignées au même bloc. Ces contraintes sont représentées par une famille EB de sous-ensembles de N tels que chaque sous-ensemble $e \in EB$ ne peut pas appartenir au même bloc.

4.1.2 Contraintes de productivité

La contrainte de productivité est nécessaire pour assurer le volume de production voulu. Ainsi il est nécessaire que le temps de chaque station soit inférieur à la valeur du temps de cycle objectif. Pour les lignes considérées, les blocs de la même station sont activés séquentiellement et les opérations du même bloc sont exécutées en parallèle. Donc, le temps de la station $t^s(N_k)$ est égal à la somme des temps d'exécution de ses blocs + τ^s, une constante modélisant le temps de chargement/déchargement de la pièce. Le temps d'exécution du bloc q est modélisé par la variable F_q et déterminé comme le temps de l'opération la plus longue incluse dans le bloc + τ^b, une constante modélisant le temps nécessaire pour l'activation d'un boîtier multibroche.

4.1.3 Fonction objectif

L'objectif de la reconfiguration est de minimiser le coût de la ligne finale, soit minimiser le coût des nouvelles pièces d'équipements à installer sur la ligne afin de répondre au nouveau processus de production. Ceci peut être exprimé par la fonction objectif suivante :

$$Minimize\ \ C(p) = C_1 \sum_{k=1}^{m_0} Z_k + C_2 \sum_{q=1}^{q_0} Y_q - C_3 \sum_{l=1}^{l_0} \sum_{q=1}^{q_0} B_{lq} \tag{4.1}$$

Les variables Z_k and Y_q sont utilisées pour calculer le nombre de stations de travail et de blocs respectivement dans la ligne et la variable B_{lq} est utilisée pour calculer le nombre de blocs réutilisés. Pour rappel, C_1 est le coût d'une nouvelle station, C_2 représente le coût d'un nouveau bloc et C_3 est le bénéfice de la réutilisation d'un ancien bloc dans la ligne modifiée. Généralement, $C_3 < C_2 \leq C_1$.

Il est à noter que comme les blocs sont modulaires et peuvent être installés sur les nouvelles comme sur les anciennes machines, nous ne faisons pas de distinction entre les nouvelles et anciennes machines. Ainsi nous pouvons considérer que m^* machines existantes dans la ligne initiale sont réutilisées dans la nouvelle ligne si $\sum_{k=1}^{m_0} Z_k > m^*$, sinon seulement $\sum_{k=1}^{m_0} Z_k$ machines sont réutilisées. Comme m^* est une constante, sa présence dans la fonction objectif n'influence pas le processus d'optimisation. Par conséquent, elle peut être omise. Pour obtenir le coût de reconfiguration, il suffit de soustraire le coût des machines réutilisées de la valeur $C(p)$.

La section suivante présente le modèle mathématique pour ce problème.

4.1.4 Modèle mathématique

Le but du problème d'optimisation est de minimiser la fonction objectif (4.1) sous les contraintes (4.2) - (4.15) ou (4.2) - (4.15').

Chaque opération de l'ensemble N doit être affectée à un seul bloc exactement, l'indice du bloc doit appartenir à l'ensemble $Q(j)$:

$$\sum_{q \in Q(j)} X_{jq} = 1; j \in N; \tag{4.2}$$

Les contraintes de précédence doivent être respectées :

$$\sum_{q' \in Q(i)} q' X_{iq'} \geq \sum_{q \in Q(i)} q X_{iq}; i \in PredD(j); j \in N; \tag{4.3}$$

Les contraintes d'inclusion sont satisfaites si :

$$\sum_{q \in Q(i) \cap B(k)} X_{iq} = \sum_{q \in Q(j) \cap B(k)} X_{jq}; i, j \in e; e \in IS; k \in K(i); \tag{4.4}$$

Les contraintes d'exclusion pour les blocs sont respectées si :

$$\sum_{j \in e} X_{jq} \leq \mid e \mid -1; e \in EB; q \in \bigcap_{j \in e} Q(j); \tag{4.5}$$

Les contraintes d'exclusion pour les stations :

$$\sum_{j \in e} \sum_{B(k) \bigcap Q(j)} X_{jq} \leq \mid e \mid -1; e \in ES; k \in \bigcap_{j \in e} K(j); \tag{4.6}$$

Pour calculer les temps d'exécution des blocs, des variables auxiliaires $F_q \in [0, T_0 - \tau^s]$, $q = 1, 2, \ldots, q_0$ sont utilisées. Pour un bloc q, la valeur de F_q ne peut pas être inférieure au temps d'exécution de l'opération la plus longue plus la constante τ^b :

$$F_q \geq (t_j + \tau^b) X_{jq}; j \in N; q \in Q(i); \tag{4.7}$$

La somme des temps d'exécution des blocs assignés à la même station ne peut pas excéder une valeur donnée T_0 - τ^s où T_0 est le temps de cycle objectif. C'est ce qu'on appelle la contrainte du temps de cycle :

$$\sum_{q \in B(k)} F_q \geq T_0 - \tau^s; k = 1, 2, \ldots, m_0; \tag{4.8}$$

Un bloc q est considéré comme créé, s'il y a au moins une opération affectée à ce bloc, c'est-à-dire si et seulement si $X_{jq} = 1$ au moins pour une opération j :

$$Y_q \geq X_{jq}; j \in N; q \in Q(j); \tag{4.9}$$

Une station k est considérée comme créée, s'il y a au moins un bloc assigné à cette station, c'est-à-dire si et seulement si $Y_q = 1$ au moins pour un $q \in S(k)$:

$$Z_k \geq Y_q; k = 1, 2, \ldots, m_0; q = (k-1)n_0 + 1; \tag{4.10}$$

Les blocs sont créés séquentiellement dans une station de travail, c'est-à-dire le bloc q est créé dans une station k si et seulement si le bloc q - 1 existe pour cette station :

$$Y_{q-1} - Y_q \geq 0; q \in B(k) \backslash \{(k-1)n_0 + 1\}; k = 1, 2, \ldots, m_0; \tag{4.11}$$

Une station k peut être créée si la station (k -1) existe déjà :

$$Z_{k-1} - Z_k \geq 0; k = 2, 3, \ldots, m_0; \tag{4.12}$$

où $j \in N$; $i \in PredD(i)$; q= 1,2,..., $m_0 n_0$; k=1,..., m_0 ;

4.1.5 Contraintes de reconfiguration

Comme les blocs sont modulaires et peuvent être installés sur les nouvelles comme sur les anciennes machines, nous ne faisons pas de distinction entre les nouvelles et anciennes machines. Il suffit juste de s'assurer qu'un bloc existant est réutilisé au maximum une fois :

$$\sum_{q=1}^{q_0} B_{lq} \leq 1; l = 1, ..., l_0; \tag{4.13}$$

Un nouveau bloc peut recevoir un et seulement un ancien bloc :

$$\sum_{l=1}^{l_0} B_{lq} \leq 1; q = 1, ..., q_0; \tag{4.14}$$

Comme nous l'avons déjà annoncé, nous utilisons les hypothèses B2 (Modèle 1) et B3 (Modèle 2) sur la réutilisabilité des boîtiers multibroches ou blocs. Nous les modélisons comme suit.

4.1.5.1 Modèle 1

Des opérations peuvent être supprimées d'un bloc mais aucune nouvelle opération ne peut être ajoutée à un bloc existant (hypothèse B2) :

$$X_{iq} + B_{lq} \leq 1; \forall i \notin N_{kl}, l = 1, ..., l_0; q = 1, ..., q_0; \tag{4.15}$$

4.1.5.2 Modèle 2

Il y a au moins une ancienne opération du bloc l assignée au bloc q qui justifie la réutilisation du bloc l (hypothèse B3) :

$$\sum_{i \in N_l} X_{iq} \geq B_{lq}, l = 1, \ldots, l_0, q = 1, ..., q_0; \tag{4.15'}$$

Il est à noter que les contraintes (15) et (15') sont contradictoires et ne peuvent pas être utilisées simultanément.

Les deux modèles présentés peuvent être implémentés comme des programmes linéaires en variables mixtes et résolus par un solveur.

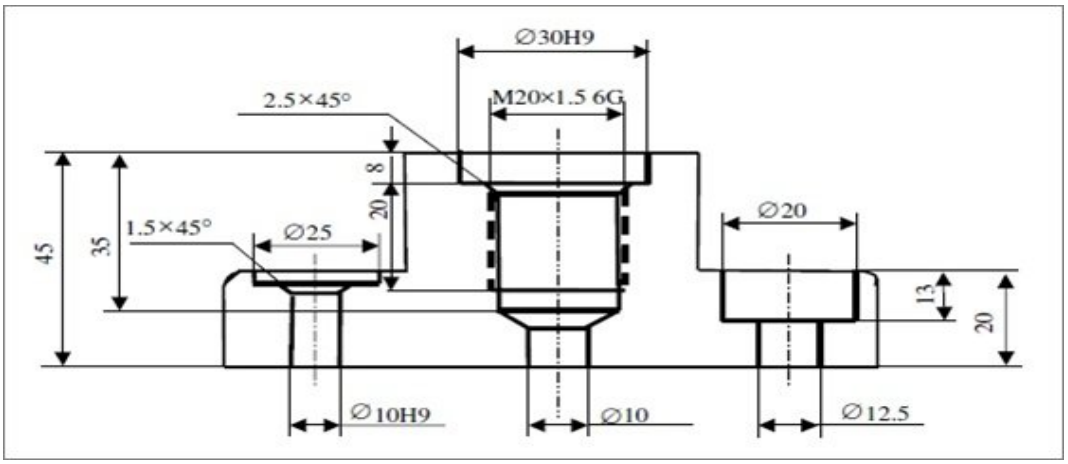
4.2 Exemple illustratif

Le problème de reconfiguration d'une ligne de transfert à boîtiers multibroches peut être illustré par l'exemple industriel suivant.

La pièce initiale avec 24 opérations est présentée dans la Figure 4.1. Les opérations et paramètres d'usinage sont décrits dans le Tableau 4.1. Le graphe de précédence est donné dans la Figure 4.2. Le temps de cycle de la ligne désirée $T_0 =$ 100 secondes, $\tau^s = 0$, $\tau^b = 0$, le nombre maximal de stations dans la ligne, $m_0 =$ 15 et le nombre maximal de blocs par station, n_0 =4.

La ligne initiale présentée dans la Figure 4.2 comportait quatre stations de travail : la première et la deuxième avaient deux blocs chacune, la troisième trois blocs et la quatrième quatre blocs. Le contenu en opérations de chaque bloc est également affiché dans la Figure 4.2.

La nouvelle pièce à fabriquer est présentée dans la Figure 4.3. On peut constater que les éléments 3-4 ont été supprimés et les nouveaux éléments 7-10 ont été ajoutés. Leurs opérations et les paramètres d'usinage sont donnés dans le Tableau 4.2.



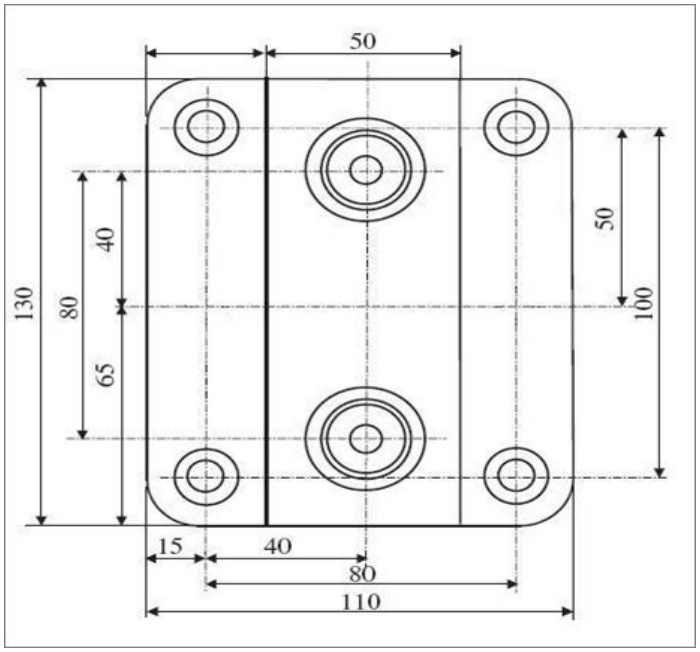


FIGURE 4.1 – Pièce initiale à usiner

Tableau 4.1 – Opérations et leurs paramètres

Élément usiné	Type d'opération	Opération	Temps t_j (min)
Plan F	Fraisage ébauche	1	0,93
	Fraisage finition	2	1,06
Élément H1	Perçage ∅12,5	3	0,68
	Laminage ∅20	4	0,16
Élément H2	Perçage ∅12,5	5	0,68
	Laminage ∅20	6	0,16
Élément H3	Perçage ∅12,5	7	0,68
	Laminage ∅20	8	0,16
Élément H4	Perçage ∅12,5	9	0,68
	Laminage ∅20	10	0,16
Élément H5	Perçage ∅10	11	1
	Élargissement ∅18	12	0,78
	Perçage ∅17	13	0,64
	Perçage ∅22	14	0,09
	Taraudage	15	0,17
	Alésage ébauche ∅29,5	16	0,09
	Alésage finition ∅30	17	0,12
Élément H6	Perçage ∅10	18	1
	Élargissement ∅18	19	0,78
	Perçage ∅17	20	0,64
	Perçage ∅22	21	0,09
	Taraudage	22	0,17
	Alésage ébauche ∅29,5	23	0,09
	Alésage finition ∅30	24	0,12

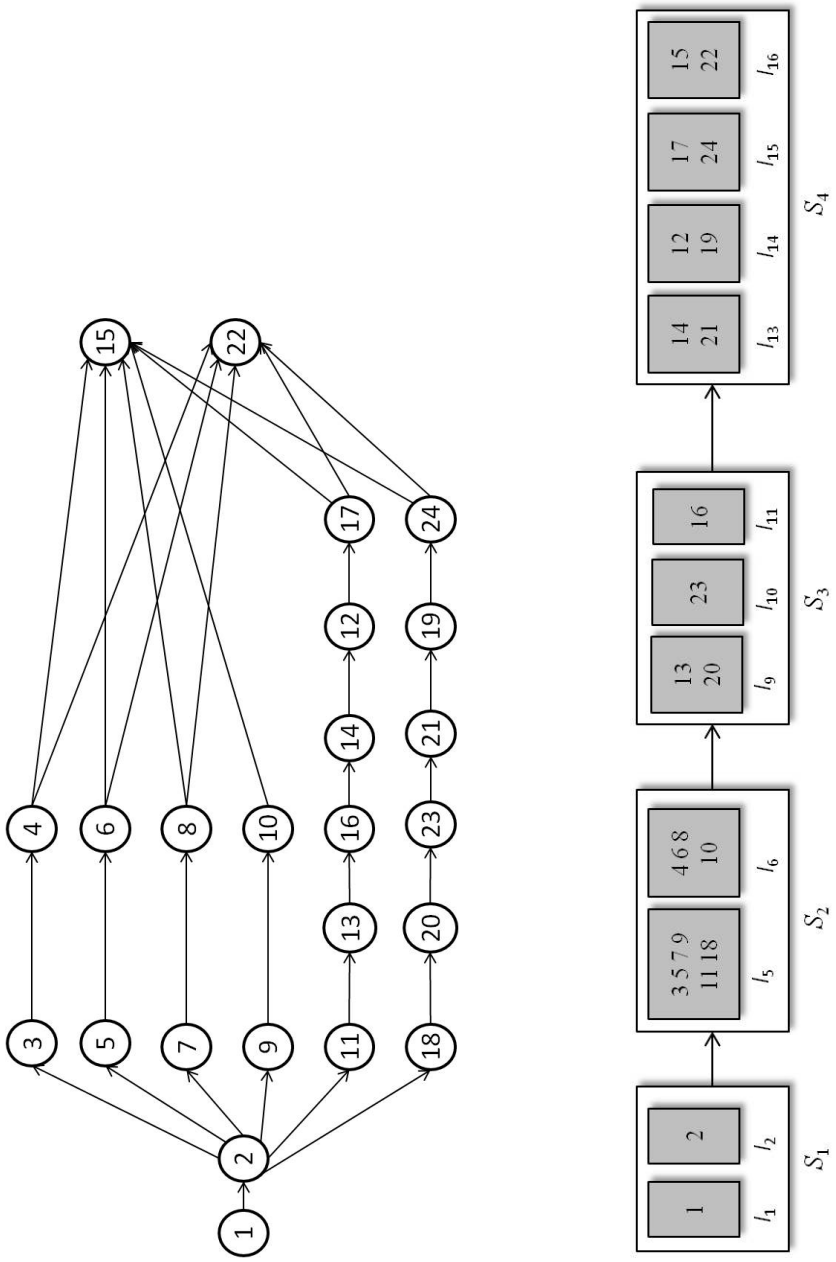


FIGURE 4.2 – Graphe initial de précédence et configuration initiale de la ligne

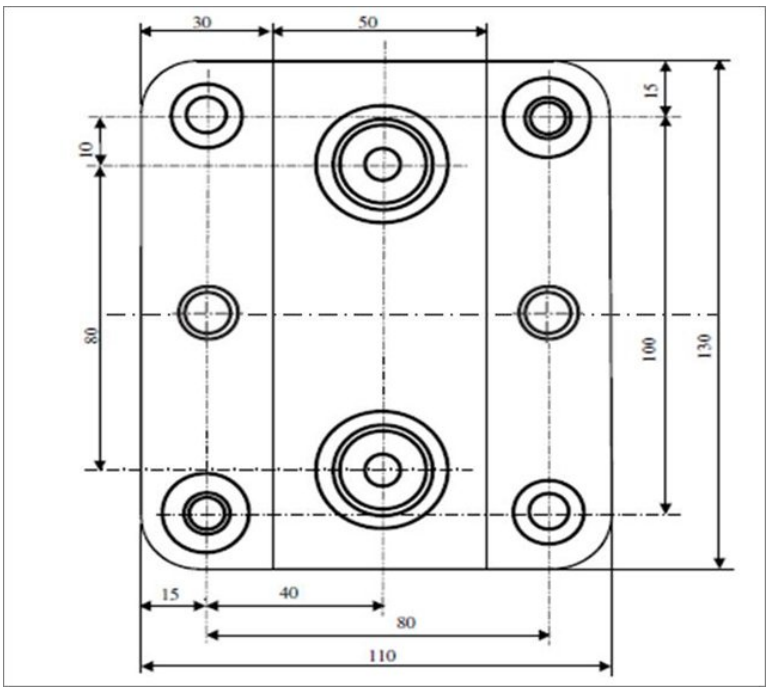


FIGURE 4.3 – Pièce après modification

Tableau 4.2 – Ensemble d'opérations N et leurs paramètres après modification

Élément usiné	Type d'opération	Opération	Temps t_j (min)
Élément H7	Perçage ∅9,2	25	0,86
	Laminage ∅25	26	0,02
	Fraisage ∅10	27	0,44
Élément H8	Perçage ∅9,2	28	0,86
	Laminage ∅25	29	0,02
	Fraisage ∅10	30	0,44
Élément H9	Perçage ∅9,8	31	0,52
	Perçage ∅12	32	0,12
	Fraisage ∅10	33	0,17
Élément H10	Perçage ∅9,8	34	0,52
	Perçage ∅12	35	0,12
	Fraisage ∅10	36	0,17

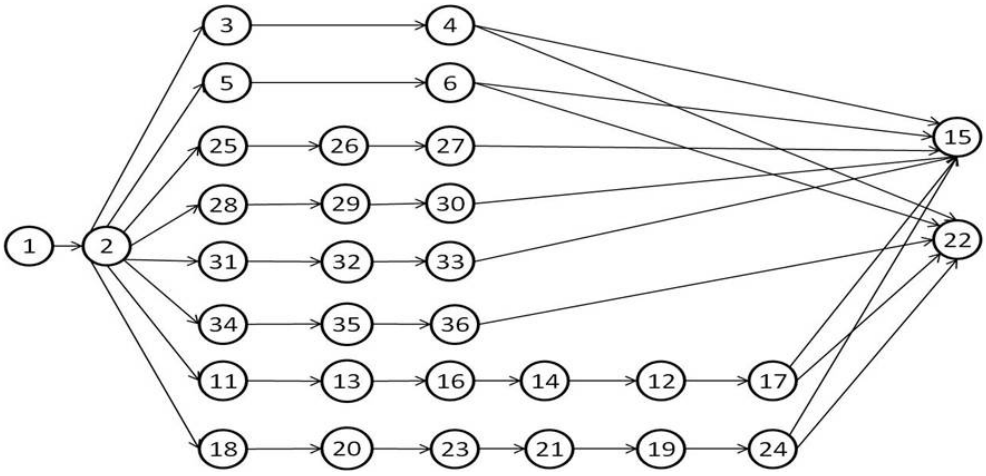


FIGURE 4.4 – Graphe de précédence après modification

On peut constater que les opérations $\{7, 8, 9, 10\}$ ont été supprimées de N et les opérations $\{25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36\}$ ont été introduites dans N.

Les contraintes de précédence pour la nouvelle pièce sont données dans la Figure 4.4.

Les nouvelles contraintes d'exclusion pour les blocs sont données par : $EB = \{\{2, 1\}, \{3, 2\}, \{4, 3\}, \{5, 2\}, \{6, 5\}, \{25, 2\}, \{25, 26\}, \{28, 2\}, \{28,29\}, \{11, 2\}, \{12, 20\}, \{13, 11\}, \{14, 16\}, \{15, 4\}, \{15, 6\}, \{15, 27\}, \{15,30\}, \{15, 17\}, \{15, 24\}, \{16, 13\}, \{17, 12\}, \{18, 2\}, \{19, 21\}, \{20, 18\}, \{21, 23\}, \{22, 4\}, \{22, 6\}, \{22, 27\}, \{22, 17\}, \{22, 24\}, \{23, 20\}, \{24, 19\}, \{11, 4\}, \{11, 6\}, \{12, 4\}, \{12, 6\}, \{13, 4\}, \{13, 6\}, \{13, 11\}, \{13, 12\}, \{14, 3\}, \{14, 4\}, \{14, 5\}, \{14, 6\}, \{14, 11\}, \{14, 12\}, \{15, 3\}, \{15, 4\}, \{15, 5\}, \{15, 6\}, \{15, 11\}, \{15, 12\}, \{16, 3\}, \{16, 4\}, \{16, 5\}, \{16, 6\}, \{16, 25\}, \{16, 26\}, \{16, 28\}, \{16, 29\}, \{16, 30\}, \{16, 11\}, \{16, 12\}, \{16, 13\}, \{16, 14\}, \{16, 15\}, \{17, 3\}, \{17, 4\}, \{17, 5\}, \{17, 6\}, \{17, 11\}, \{17, 12\}, \{17, 13\}, \{17, 14\}, \{17, 15\}, \{18, 26\}, \{18, 29\}, \{18, 16\}, \{19, 26\}, \{19, 29\}, \{19, 16\}, \{20, 26\}, \{20, 29\}, \{20, 16\}, \{20, 18\}, \{20, 19\}, \{21, 25\}, \{21, 26\}, \{21, 28\}, \{21, 29\}, \{21, 16\}, \{21, 18\}, \{21, 19\}, \{22, 25\}, \{22, 26\}, \{22, 28\}, \{22, 29\}, \{22, 16\}, \{22, 18\}, \{22, 19\}, \{23, 3\}, \{23, 4\}, \{23, 5\}, \{23, 6\}, \{23, 25\}, \{23, 26\}, \{23, 28\}, \{23, 29\}, \{23, 11\}, \{23, 12\}, \{23, 13\}, \{23, 14\}, \{23, 15\}, \{23, 16\}, \{23, 17\}, \{23, 18\}, \{23, 19\}, \{23, 20\}, \{23, 21\}, \{23, 22\}, \{24, 25\}, \{24, 26\}, \{24, 28\}, \{24, 29\}, \{24, 16\}, \{24, 18\}, \{24, 19\}, \{24, 20\}, \{24, 21\}, \{24, 22\}, \{25, 26\}, \{28, 29\}, \{31, 32\}, \{34, 35\}, \{33, 15\}, \{36, 22\}\}$.

Les nouvelles contraintes de compatibilité pour les stations sont : IS = {{12, 19, 17, 24}, {33, 36}}, ES = {{2,3}, {2,5}, {2,25}, {2,28}, {2,31}, {2,34}, {2,11}, {2,18}}.

Les paramètres du problème de reconfiguration sont comme suit : le coût d'une station C_1=10, le coût d'un bloc C_2=2, le bénéfice de réutilisation d'un ancien bloc : C_3=1, T_0 = 2,8 min ; τ^b = 0,2 ; τ^s = 0,4 min, m_0 = 5, n_0 = 4.

Ce problème a été résolu en utilisant le modèle (1) - (15) et le modèle (1) - (15') à l'aide du solveur ILOG CPLEX 12.4 sur PC Intel(R), 2,20 GHz, avec 8 Go de RAM. Les temps de résolution sont respectivement 29,42 et 84,42 secondes.

4.2.1 Solution pour le modèle MIP1

La solution obtenue pour le problème étudié avec le modèle (1) - (15) est donnée dans la Figure 4.5. Neuf 9 blocs ont été réutilisés et deux nouveaux blocs ont été introduits afin d'adapter la ligne initiale à la nouvelle pièce. La correspondance entre les nouveaux et les anciens blocs est comme suit : {(q_1, l_1), (q_2 , l_2), (q_6, l_9), (q_9, l_{10}), (q_7, l_{11}), (q_{10}, l_{13}), (q_{11}, l_{14}), (q_{12}, l_{15}), (q_{14}, l_{16})}.

La valeur de la fonction objectif donnant le coût total de la ligne est égal à 53. Comme nous ne faisons pas de distinction entre les anciennes et les nouvelles machines et nous considérons que la réutilisation d'une ancienne machine n'induit aucun coût supplémentaire, nous pouvons soustraire de 53 le coût correspondant aux quatre machines réutilisées soit $4 * C_1$=$4 * 10$=40. Ainsi que le coût de reconfiguration est égal à $53 - 30$=23.

4.2.2 Solution pour le modèle MIP2

L'hypothèse B3 utilisée par le modèle MIP2 facilite la réutilisation des blocs existants. Ainsi dans la solution obtenue avec ce modèle tous les anciens blocs ont été réutilisés comme suit : {(q_1, l_1), (q_2 , l_2), (q_5, l_5), (q_3, l_6), (q_6, l_9), (q_9, l_{10}), (q_7, l_{11}), (q_{10}, l_{13}), (q_{11}, l_{14}), (q_{12}, l_{15}), (q_{14}, l_{16})}.

Cette solution contient également le même nombre de stations que la ligne initiale, alors nous pouvons considérer que toutes les machines ont été aussi réutilisées. La valeur de la fonction objectif donnant le coût total de la ligne est égal à 51. Si nous en soustrayons le coût correspondant aux machines réutilisées, alors le coût de reconfiguration devient $51 - 40$=11.

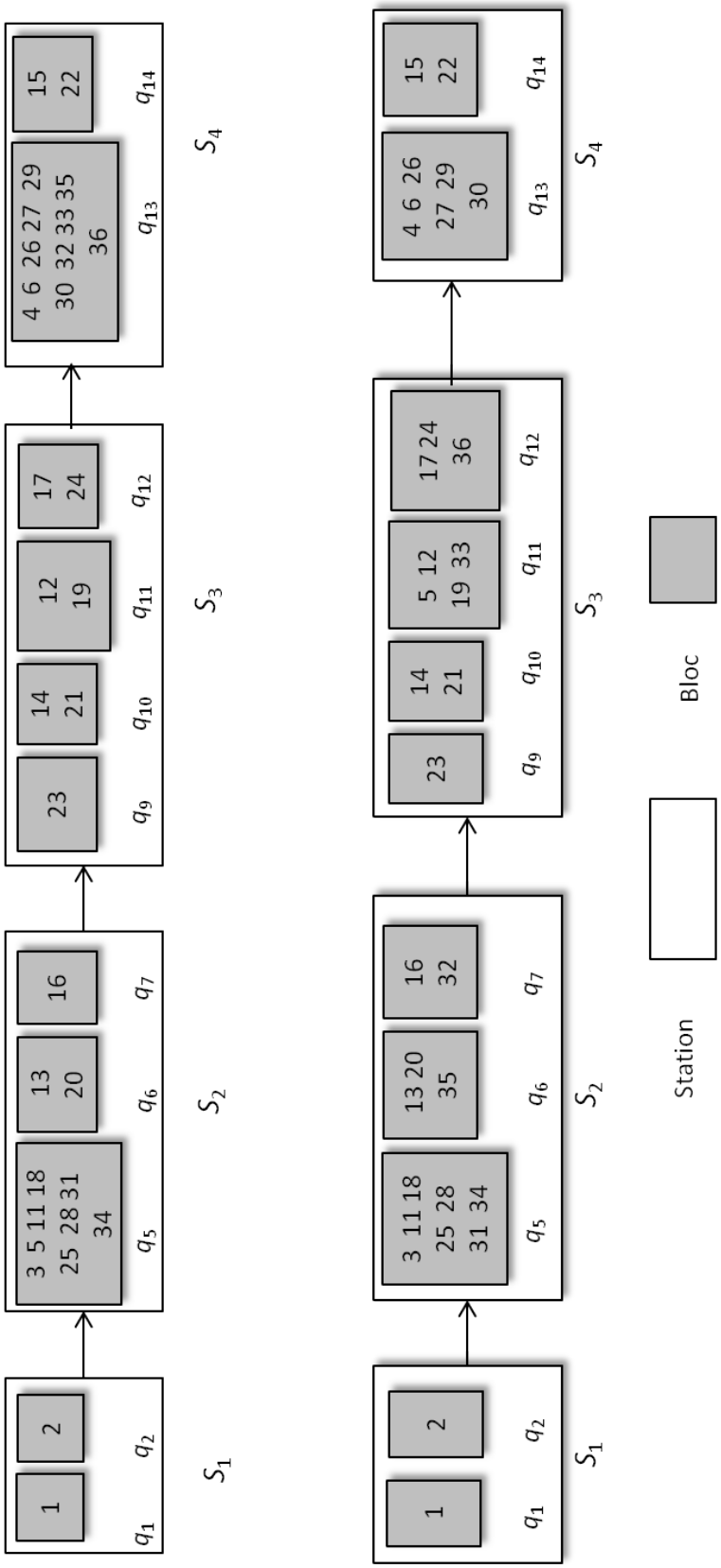


FIGURE 4.5 – Solutions optimales de reconfiguration (modèle 1 & 2)

4.3 Expérimentations numériques

L'objectif de notre étude expérimentale est d'analyser l'efficacité des modèles proposés pour la résolution des instances proches des cas industriels. Pour le problème de la configuration optimale des lignes de transfert à boîtiers multibroches, un tel échantillon d'instances a été présenté dans [Battaïa *et al.* 2012]. Nous reprenons 41 instances disponibles dans la littérature et les modifions afin de simuler le cas de la reconfiguration. Les instances testées sont caractérisées par le nombre d'opérations $|N|$, les valeurs m_0 et n_0, les contraintes de précédence, d'exclusion et d'inclusion IS, ES et EB ainsi que par l'ensemble des blocs existants. Ces données pour les 41 instances utilisées sont répertoriées dans l'Annexe de ce mémoire.

Les tests ont été réalisés sur un PC Intel(R), 2,20 GHz, avec 8 Go de RAM. Les modèles ont été codés en C++ et résolus avec le solveur ILOG CPLEX 12.4.

Pour étudier l'apport de la réutilisation des équipements, nous avons également utilisé le modèle de [Dolgui *et al.* 2006b] pour obtenir les configurations optimales pour le produit initial (les résultats sont donnés dans le Tableau 4.3) et le produit modifié sans réutiliser les équipements existants (Tableau 4.4). Ces solutions seront comparées avec les solutions fournies par nos modèles et présentées dans les Tableaux 4.5 et 4.8 pour les modèles MIP1 et MIP2, respectivement. Ces tableaux reportent le nombre de stations (colonne "#Sts") et de blocs (colonne "#Blocs"), le coût et le temps d'exécution pour chaque instance (colonne "N"). Le coût de la reconfiguration a été obtenu par soustraction du coût des machines réutilisées de la valeur de la fonction objectif fournie par modèles MIP1 et MIP2.

Pour mettre en évidence les bénéfices de la reconfiguration, les coûts suivants sont reportés dans les Tableaux 4.6 et 4.9 ainsi que le nombre de blocs réutilisés dans les colonnes correspondantes :

- C_{new} le coût des investissements nécessaires si la ligne pour le produit modifié est conçue sans tenir compte des équipements existants (le coût de la ligne dans le Tableau 4.4)

- C_{rec} le coût de la reconfiguration pour le modèle MIP1 (Tableau 4.5) ou pour le modèle MIP2 (Tableau 4.8),

- Le pourcentage du gain obtenu par la réutilisation des équipements existants a été calculé comme suit :

$$\texttt{Gain} = \left(\frac{C_{new} - C_{rec}}{C_{new}}\right) * 100\%$$

Tableau 4.3 – Configuration de la ligne initiale

N °	#Sts	#Blocs	Coût	Temps	N °	#Sts	#Blocs	Coût	Temps
1	2	4	28	2,46	22	2	4	28	9,13
2	3	3	36	9,55	23	2	4	28	8,22
3	4	4	48	10,09	24	4	4	48	29,53
4	3	3	36	5,44	25	3	3	36	12,23
5	3	4	38	3,39	26	3	3	36	12,50
6	3	3	36	9,25	27	3	3	36	6,02
7	3	4	38	15,49	28	4	4	48	12,76
8	3	4	38	11,20	29	3	5	40	15,51
9	3	5	40	12,53	30	3	3	36	11,72
10	3	4	38	6,91	31	3	4	38	18,92
11	2	5	30	12,36	32	2	4	28	7,38
12	3	4	38	3,34	33	3	3	36	9,11
13	3	4	38	6,69	34	3	4	38	15,44
14	2	4	28	5,49	35	2	4	28	8,83
15	4	4	48	21,70	36	3	3	36	11,82
16	3	4	38	17,77	37	3	4	38	6,16
17	3	3	36	8,70	38	3	4	38	9,66
18	3	4	38	14,68	39	3	4	38	9,66
19	3	4	38	11,26	40	3	4	38	4,84
20	4	4	48	39,86	41	3	3	38	13,72
21	2	4	28	9,33	-	-	-	-	-

Tableau 4.4 – Configuration optimale pour le nouveau produit

N °	#Sts	#Blocs	Coût C_{new}	Temps	N °	#Sts	#Blocs	Coût C_{new}	Temps
1	2	5	30	7,33	22	2	4	28	14,07
2	3	4	38	5,52	23	3	4	38	23,3
3	4	5	50	6,75	24	4	4	48	36,55
4	2	4	38	11,63	25	3	3	36	14,33
5	4	4	48	13,47	26	3	4	38	2,83
6	3	3	36	7,12	27	3	3	36	11,12
7	4	5	50	23,24	28	4	4	48	20,91
8	3	5	40	26,62	29	4	5	50	20,23
9	3	6	42	45,06	30	3	4	38	21,43
10	3	4	38	14,57	31	3	4	38	21,85
11	3	4	38	30,43	32	3	3	36	16,67
12	4	4	48	20,21	33	3	5	40	22,3
13	3	4	48	20,04	34	3	5	40	57,81
14	3	4	38	5,95	35	3	3	36	13,43
15	4	5	50	48,53	36	2	4	28	4,60
16	3	4	38	14,38	37	3	4	38	4,75
17	2	4	28	7,65	38	3	4	38	23,11
18	3	4	38	9,99	39	3	4	38	25,52
19	3	4	38	21,43	40	3	5	40	8,93
20	4	4	48	20,81	41	3	4	38	13,72
21	3	5	40	23,8	-	-	-	-	-

Tableau 4.5 – Résultats pour la reconfiguration (modèle MIP1)

N °	#Sts	#Blocs	Coût C_{rec}	Temps	N °	#Sts	#Blocs	Coût C_{rec}	Temps
1	2	5	7	20	22	2	4	6	34,7
2	3	4	5	23	23	3	5	17	45,5
3	4	5	8	46,4	24	4	4	6	134,3
4	2	4	7	19,2	25	3	3	6	49,1
5	4	4	15	48,2	26	3	4	6	37,2
6	3	4	6	78,1	27	3	3	5	35,4
7	4	5	17	117,6	28	4	5	6	51,4
8	3	5	8	61,5	29	4	5	18	155,5
9	3	6	8	94,7	30	3	4	7	108
10	3	4	7	47,4	31	3	4	6	145
11	3	5	18	75,9	32	3	3	16	35,8
12	4	4	15	69,9	33	3	5	9	66,2
13	3	5	7	60,2	34	3	5	8	152,9
14	3	4	17	388,7	35	3	4	15	77,7
15	4	6	9	152,2	36	2	4	5	8,1
16	3	4	5	256,7	37	3	4	6	30,9
17	2	4	6	29,7	38	3	4	6	88,5
18	3	4	7	80,6	39	3	4	6	83,9
19	3	4	6	72	40	3	6	8	26,1
20	4	4	6	232,1	41	3	4	7	26,7
21	3	5	18	148,8	-	-	-	-	-

Nous pouvons constater que le temps de résolution du problème de résolution est généralement supérieur à celui nécessaire pour l'optimisation de la configuration de la ligne sans tenir compte des équipements existants. Au total, la résolution de 41 instances a duré 3515,8 seconds pour le modèle MIP1 (la moyenne est de 33,97 seconds par instance), les instances du problème de la configuration initiale pour le même produit ont été résolues en 761,99 seconds (la moyenne est de 18,59 par instance).

Tableau 4.6 – Bénéfices de la reconfiguration (modèle MIP1)

N °	C_{new}	C_{rec}	Gain %	Blocs réutilisés	N °	C_{new}	C_{rec}	Gain %	Blocs réutilisés
1	30	7	76,67	3	22	28	6	78,57	2
2	38	5	86,84	3	23	38	17	55,26	3
3	50	8	84,00	2	24	48	6	87,50	2
4	38	7	81,58	1	25	36	36	83,33	0
5	48	15	68,75	3	26	38	6	84,21	2
6	36	6	83,33	2	27	36	5	86,11	1
7	50	17	66,00	3	28	48	6	87,50	4
8	40	8	80,00	2	29	50	18	64,00	2
9	42	8	80,95	4	30	38	7	81,58	1
10	38	7	81,58	1	31	38	6	84,21	2
11	38	18	52,63	2	32	36	16	55,56	0
12	48	15	68,75	3	33	40	9	77,50	1
13	48	7	85,42	3	34	40	8	80,00	2
14	38	17	55,26	1	35	36	15	58,33	3
15	50	9	82,00	3	36	28	5	82,14	3
16	38	5	86,84	3	37	38	6	84,21	2
17	28	6	78,57	2	38	38	6	84,21	2
18	38	7	81,58	1	39	38	6	84,21	2
19	38	6	84,21	2	40	40	8	80,00	4
20	48	6	87,50	2	41	38	7	81,58	1
21	40	18	55,00	2	-	-	-	-	-

Les gains réalisés montrent l'importance de l'optimisation du processus de la reconfiguration : avec 87 blocs réutilisés au total, entre 52,63% et 87,5% des investissements ont pu être économisés (avec 77,25% en moyenne).

Tableau 4.7 – Bénéfices de la reconfiguration (modèle MIP1) en tenant compte du coût de la ligne initiale

N °	C_{new}	C_{rec}	Gain %	Blocs réutilisés	N °	C_{new}	C_{rec}	Gain %	Blocs réutilisés
1	58	35	39,66	3	22	56	34	39,29	2
2	74	41	44,59	3	23	66	45	31,82	3
3	98	56	42,86	2	24	96	54	43,75	2
4	74	43	41,89	1	25	72	42	41,67	0
5	86	53	38,37	3	26	74	42	43,24	2
6	72	42	41,67	2	27	72	41	43,06	1
7	88	55	37,50	3	28	96	54	43,75	4
8	78	46	41,03	2	29	90	58	35,56	2
9	82	48	41,46	4	30	74	43	41,89	1
10	76	45	40,79	1	31	76	44	42,11	2
11	68	48	29,41	2	32	64	44	31,25	0
12	86	53	38,37	3	33	76	45	40,79	1
13	86	45	47,67	3	34	78	46	41,03	2
14	66	45	31,82	1	35	64	43	32,81	3
15	98	57	41,84	3	36	64	41	35,94	3
16	76	43	43,42	3	37	76	44	42,11	2
17	64	42	34,38	2	38	76	44	42,11	2
18	76	45	40,79	1	39	76	44	42,11	2
19	76	44	42,11	2	40	78	46	41,03	4
20	96	54	43,75	2	41	76	45	40,79	1
21	68	46	32,35	2	-	-	-	-	-

Comme la reconfiguration n'est pas possible sans la réutilisation des équipements de la ligne initiale, dans le Tableau 4.7 nous calculons les gains réalisés grâce à la reconfiguration en tenant compte du coût de la ligne initiale. Avec ce calcul, entre 29,41% et 47,67% des investissements ont pu être économisés (avec 39,8% en moyenne).

Tableau 4.8 – Résultats pour la reconfiguration (modèle MIP2)

N °	#Sts	#Blocs	Coût C_{rec}	Temps	N °	#Sts	#Blocs	Coût C_{rec}	Temps
1	2	5	6	8,45	22	2	4	5	13,27
2	3	4	5	9,89	23	3	4	14	22,21
3	4	5	6	18,45	24	4	4	4	43,14
4	2	4	5	4,86	25	3	3	3	6,24
5	4	4	14	26,3	26	3	4	5	26,91
6	3	3	3	6,16	27	3	3	3	23,55
7	4	5	16	38,06	28	4	4	4	16,48
8	3	5	6	29,35	29	4	5	15	17,17
9	3	6	7	58,98	30	3	4	5	35,72
10	3	4	4	24,94	31	3	4	4	19,7
11	3	4	14	38,84	32	3	3	13	28,5
12	4	4	14	48,36	33	3	5	7	25,67
13	3	4	4	23,91	34	3	5	6	69,6
14	3	4	14	7,73	35	3	3	13	20,2
15	4	5	6	177,38	36	2	3	4	7,69
16	3	4	4	20,24	37	3	4	5	7,37
17	2	4	5	10,1	38	3	4	4	30,81
18	3	4	4	30	39	3	4	4	26,56
19	3	4	4	19,35	40	3	5	6	22,51
20	4	4	4	81,22	41	3	4	5	27,26
21	3	5	16	46,33	-	-	-	-	-

Avec le temps de résolution total égal à 1219,46 seconds pour le modèle MIP2 (la moyenne est de 29,74 seconds par instance), le problème correspondant au modèle MIP2 semble être plus facile à résoudre que le modèle MIP1, mais il reste plus difficile à résoudre que le problème de la configuration initiale pour le même produit.

Tableau 4.9 – Bénéfices de la reconfiguration (modèle MIP2)

N °	C_{new}	C_{rec}	Gain %	Blocs réutilisés	N °	C_{new}	C_{rec}	Gain %	Blocs réutilisés
1	30	6	80,00	4	22	28	5	82,14	3
2	38	5	86,84	3	23	38	14	63,16	4
3	50	6	88,00	4	24	48	4	91,67	4
4	38	5	86,84	3	25	36	3	91,67	3
5	48	14	70,83	4	26	38	5	86,84	3
6	36	3	91,67	3	27	36	3	91,67	3
7	50	16	68,00	4	28	48	4	91,67	4
8	40	6	85,00	4	29	50	15	70,00	5
9	42	7	83,33	5	30	38	5	86,84	3
10	38	4	89,47	4	31	38	4	89,47	4
11	38	14	60,15	4	32	36	13	63,88	3
12	48	14	70,83	4	33	40	7	82,50	3
13	48	4	91,67	4	34	40	6	85,00	4
14	38	14	63,16	4	35	36	13	63,88	3
15	50	6	88,00	4	36	28	4	85,71	2
16	38	4	89,47	4	37	38	5	86,84	3
17	28	5	82,14	3	38	38	4	89,47	4
18	38	4	89,47	4	39	38	4	89,47	4
19	38	4	89,47	4	40	40	6	85,00	4
20	48	4	91,67	4	41	38	5	86,84	3
21	40	16	60,00	4	-	-	-	-	-

Comme l'hypothèse B3 utilisée par le modèle MIP2 facilite la réutilisation des blocs existants, les gains réalisés en utilisant le modèle MIP2 sont plus importants que pour le modèle MIP1 : avec 150 blocs réutilisés au total, entre 60% et 91,67% des investissements ont pu être économisés (avec 82,26% en moyenne). .

Tableau 4.10 – Bénéfices de la reconfiguration (modèle MIP2) en tenant compte du coût de la ligne initiale

N °	C_{new}+C_{in}	C_{rec}+C_{in}	Gain %	Blocs réutilisés	N °	C_{new}	C_{rec}	Gain %	Blocs réutilisés
1	58	34	41,38	4	22	56	33	41,07	3
2	74	41	44,59	3	23	66	42	36,36	4
3	98	54	44,90	4	24	96	52	45,83	4
4	74	41	44,59	3	25	72	39	45,83	3
5	86	52	39,53	4	26	74	41	44,59	3
6	72	39	45,83	3	27	72	39	45,83	3
7	88	54	38,64	4	28	96	52	45,83	4
8	78	44	43,59	4	29	90	55	38,89	5
9	82	47	42,68	5	30	74	41	44,59	3
10	76	42	44,74	4	31	76	42	44,74	4
11	68	44	35,29	4	32	64	41	35,94	3
12	86	52	39,53	4	33	76	43	43,42	3
13	86	42	51,16	4	34	78	44	43,59	4
14	66	42	36,36	4	35	64	41	35,94	3
15	98	54	44,90	4	36	64	40	37,50	2
16	76	42	44,74	4	37	76	43	43,42	3
17	64	41	35,94	3	38	76	42	44,74	4
18	76	42	44,74	4	39	76	42	44,74	4
19	76	42	44,74	4	40	78	44	43,59	4
20	96	52	45,83	4	41	76	43	43,42	3
21	68	44	35,29	4					

Comme la reconfiguration n'est pas possible sans la réutilisation des équipements de la ligne initiale, dans le Tableau 4.10 nous calculons les gains réalisés grâce à la reconfiguration en tenant compte du coût de la ligne initiale. Avec ce calcul, entre 35,29% et 51,16% des investissements ont pu être économisés (avec 42,41% en moyenne).

4.4 Conclusion

Le problème de la reconfiguration des lignes de transfert a été considéré. Ce problème se pose lorsqu'une ligne existante doit être mise à jour afin de répondre aux nouvelles exigences de la production imposées par les nouvelles caractéristiques du produit à fabriquer. L'objectif était de réduire les nouveaux investissements en réutilisant les stations et les blocs existants autant que possible.

Deux modèles mathématiques avec les hypothèses de réutilisation différentes ont été proposés : (1) des opérations peuvent être supprimées d'un boîtier multibroche mais aucune opération ne peut être ajoutée à un boîtier multibroche existant, (2) des opérations peuvent être supprimées et des nouvelles opérations peuvent être ajoutées à un boîtier multibroche existant si toutes les contraintes d'exclusion des blocs sont respectées.

Pour ces nouveaux problèmes d'optimisation, des méthodes exactes ont été développées et évaluées sur un cas industriel et sur un échantillon des instances disponibles dans la littérature. Les résultats obtenus montrent que la reconfiguration d'une ligne existante permet des économies d'investissement jusqu'à 91% par rapport à l'installation d'une nouvelle ligne.

Néanmoins, les modèles proposés se basent sur l'hypothèse qu'il est possible d'estimer le bénéfice financier obtenu par la réutilisation d'un boîtier multibroche. Malheureusement, en pratique, la valeur exacte de ce bénéfice est parfois non-mesurable. Pour pallier à cette situation, nous proposerons, dans le chapitre suivant, l'approche goal programming pour la reconfiguration de lignes de transfert qui ne requiert pas cette information du décideur. A la fin du chapitre suivant, nous comparerons cette nouvelle méthode à l'utilisation du modèle MIP que nous venons de présenter.

Les résultats de nos travaux sur le développement de modèles MIP pour la reconfiguration des lignes de transfert ont été présentés à la ROADEF'2013 [Makssoud *et al.* 2013e] et ont donné lieu à 2 articles publiés dans les actes des conférences internationales : INCOM'2012 [Makssoud *et al.* 2012b] et IESM'2013 [Makssoud *et al.* 2013c] à un chapitre de livre [Makssoud *et al.* 2013f] ainsi qu'à 2 publications dans les revues internationales : International Journal of Advanced Manufacturing Technology [Makssoud *et al.* 2014] et Journal of Mathematical Modelling and Algorithms in Operations Research [Battaïa *et al.* 2013].

Chapitre 5

Utilisation de Goal Programming pour la reconfiguration des lignes de transfert

Sommaire

5.1 Méthode de goal programming : notions de base **83**
5.2 Modélisation du problème de reconfiguration de lignes de transfert à boîtiers multibroches **84**
5.2.1 Contraintes des buts à atteindre et fonction objectif 84
5.3 Exemple illustratif **85**
5.4 Expérimentations numériques **89**
5.5 Conclusion **104**

5.1 Méthode de goal programming : notions de base

Dans ce chapitre, nous développons une approche de programmation par objectif (Goal Programming) pour optimiser la reconfiguration de lignes de transfert. C'est une technique souvent utilisée dans la littérature pour la résolution de problèmes multi-objectifs. Cette technique vise à optimiser plusieurs fonctions objectif selon des poids donnés à chaque objectif.

Le principe de cette technique consiste à introduire des variables auxiliaires nommées déviations par rapport aux objectifs et de minimiser la somme pondérée des déviations. Ces dernières reprśentent la distance entre les niveaux d'aspiration des objectifs et les résultats réalisés. Deux genres de déviations sont pris en compte, la sur-réalisation de l'objectif qui est représentée par une déviation positive d^+ et la sous-réalisation de l'objectif qui est représentée par une déviation négative d^-.

Le modèle Goal Programming (GP) comporte deux ensembles de contraintes, les contraintes système et les contraintes but. Les contraintes système sont formulées comme dans la programmation linéaire, tandis que les contraintes but sont

des contraintes auxiliaires qui déterminent la meilleure solution possible respectant l'ensemble d'objectifs désirés.

Dans ce chapitre, cette approche sera appliquée pour la reconfiguration des lignes de transfert. Nous commençons la présentation par le modèle mathématique développé. Ce modèle sera illustré sur un exemple, puis évalué sur le même échantillon de problèmes que dans le chapitre 4.

5.2 Modélisation du problème de reconfiguration de lignes de transfert à boîtiers multibroches

Nous reprenons les hypothèses suivantes déjà utilisées dans le chapitre 4 pour le modèle MIP2 :

- Des opérations peuvent être supprimées d'un boîtier multibroche mais aucune opération ne peut être ajoutée à un boîtier existant (hypothèse B2).

- Tous les boîtiers multibroches sont conçus suivant les principes de modularité : des boîtiers multibroches peuvent être supprimés de leur station initiale et installées à n'importe quelle autre station de la ligne. Un nouveau boîtier peut être intercalé entre deux anciennes (hypothèse M4).

Par rapport au modèle (4.2 - 4.15') du chapitre 4, nous introduisons les notions supplémentaires suivantes :

- $(d_1^-,\ d_1^+)$ et $(d_2^-,\ d_2^+)$ sont respectivement appelées la sous - réalisation et la sur - réalisation des objectifs 1 et 2 du modèle,

- w_1, w_2 sont les poids des objectifs où $w_1 + w_2 = 1$.

5.2.1 Contraintes des buts à atteindre et fonction objectif

L'objectif de ce modèle est de maximiser la réutilisation des équipements de la ligne existante pour minimiser les coûts complémentaires de reconfiguration. Nous avons donc deux buts à atteindre, le premier "But 1" est de réutiliser toutes les machines de la ligne initiale :

$$\sum_{k=1}^{m_0} Z_k - m^* - d_1^+ + d_1^- = 0; \tag{5.1}$$

Le second "But 2" est de réutiliser tous les blocs de la ligne initiale :

$$\sum_{q=1}^{q_0} Y_q - \sum_{l=1}^{l_0} \sum_{q=1}^{q_0} B_{lq} - d_2^+ + d_2^- = 0; \tag{5.2}$$

La fonction objectif vise la réutilisation de tous les équipements existants dans la ligne initiale :

$$Minimize \quad (w_1(d_1^-, d_1^+) + w_2(d_2^-, d_2^+)); \tag{5.3}$$

En résumé, le modèle "GP" comprend les contraintes (4.2 - 4.14, 4.15') reprises du modèle 2 du chapitre 4, contraintes (5.1 - 5.2) mentionnées ci-dessus et la fonction objectif 5.3.

5.3 Exemple illustratif

La pièce initiale à usiner présentée dans la Figure 5.2 requiert 16 opérations. Les opérations et leurs paramètres sont donnés dans le Tableau 5.1.

Les contraintes de compatibilité concernant les opérations deux à deux sont :

- ES= {{2,5},{2,9},{2,13},{3,4},{3,5},{3,9},{3,13}} ;
- EB = {{1,2}, {2,3}, {6,7},{7,8}, {10,11},{11,12}, {13,14}, {14,l5}, {8, 16}} ;
- IS = {{7, 11}}.

Les contraintes de précédence sont représentées par un graphe orienté dans la Figure 5.1.

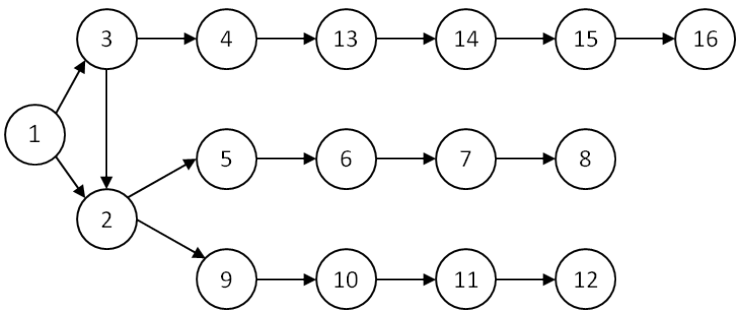


FIGURE 5.1 – Graphe représentant les contraintes de précédence

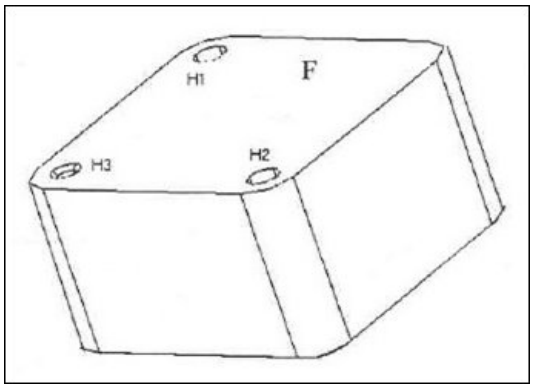


FIGURE 5.2 – Pièce initiale à usiner

Tableau 5.1 – Paramètres d'opérations.

Elément usiné	Opération	Numéro d'opération	Temps d'exécution t_j
Plan F	Laminage	1	0,54
	Laminage finition	2	0,62
Rainure	Laminage	3	0,50
	Laminage finition	4	0,58
Élément H_1	Perçage	5	0,95
	Fraisage en bout	6	0,21
	Fraisage	7	0,29
	Taraudage	8	0,09
Élément H_2	Perçage	9	0,95
	Fraisage en bout	10	0,21
	Fraisage	11	0,29
	Taraudage	12	0,09
Élément H_3	Perçage	13	1,18
	Lamage	14	0,29
	Fraisage	15	0,43
	Alésage	16	0,80

Les autres paramètres du problème sont : le coût d'une station C_1=10, le coût d'un bloc C_2=2, T_0=2,75 ; τ^b=0,3 ; τ^s=0,2 ; n_0=4 ; m_0=4 ;

Une solution optimale pour la configuration de cette ligne de transfert a été obtenue par un programme linéaire en nombres mixtes présenté dans [Guschinskaya & Dolgui 2009]. Elle est donnée dans la Figure 5.3. Cette solution comprend trois stations : la première a deux blocs et trois opérations affectées, la seconde station contient un bloc et neuf opérations affectées et la troisième est composée de deux blocs et quatre opérations. Le coût de la ligne totale est égal à 42.

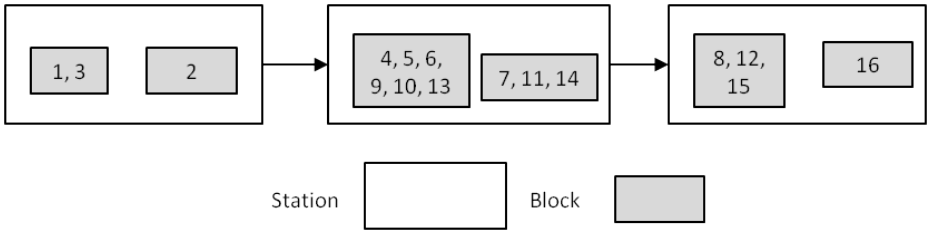


FIGURE 5.3 – Solution optimale pour la ligne initiale

Supposons maintenant que le produit a été changé par le client et une nouvelle pièce (présentée dans la Figure 5.4) devrait être produite. La nouvelle pièce est obtenue en supprimant l'élément H_2 et en ajoutant l'élément H_4 (les détails concernant cet élément sont donnés dans le Tableau 5.2). Par conséquent, les opérations {9, 10, 11, 12} ont été supprimées de N et les opérations {17, 18, 19, 20} ont été ajoutées.

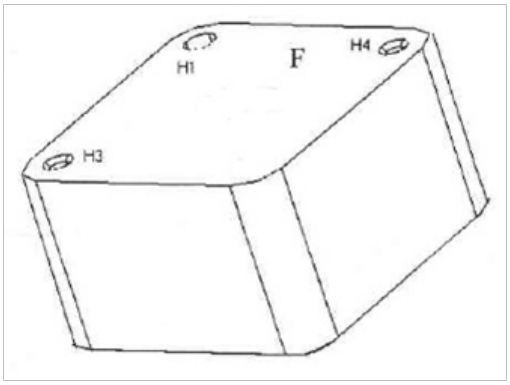


FIGURE 5.4 – Nouvelle pièce à usiner

Les nouvelles contraintes de compatibilité sont :

Tableau 5.2 – Les opérations et leurs paramètres

Elément usiné	Opération	Numéro d'opération	Temps d'exécution t_j
Trou H_4	Perccage	17	1,18
	Lamage	18	0,29
	Fraisage	19	0,43
	Alésage	20	0,80

- Ensemble des contraintes d'exclusion au niveau des stations : ES = {{2,5}, { 2,17},{ 3,4},{ 3,5}, { 3,13}, { 3,17}} ;

- Ensemble des contraintes d'exclusion au niveau des blocs EB = {{1, 2}, {2, 3}, {6, 7}, {7, 8}, {13, 14}, {13, 18}, {13, 19}, {14, l5}, {14, 17}, {14, 18}, {14, 19}, {15, 16}, {15, 17}, {15, 18}, {15, 19}, {17, 18}, {18, 19}, {19, 20}} ;

- Il n'existe pas des contraintes d'inclusion au niveau des stations.

Les contraintes de précédence pour la nouvelle pièce sont données dans la Figure 5.5.

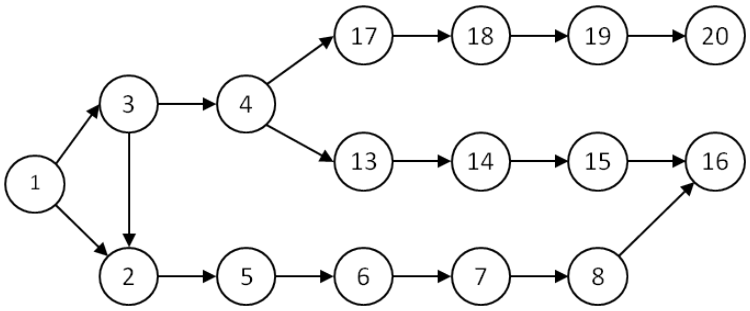


FIGURE 5.5 – Nouveau graphe de précédence

La valeur de la fonction objectif pour la solution est égale à 1,5 ; $d_1^+ = 0$; $d_1^- = 0$; $d_2^+ = 2$; $d_2^- = 0$; $w_1 = w_2 = 0{,}5$.

Toutes les stations et tous les blocs ont été réutilisés, deux nouveaux blocs et une station ont été nécessaires pour adapter la ligne initiale pour produire la nouvelle pièce.

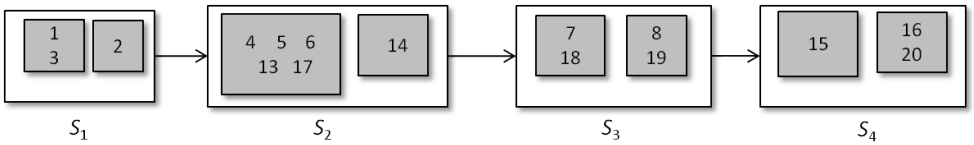


FIGURE 5.6 – Solution optimale pour le nouveau problème

5.4 Expérimentations numériques

Pour analyser la performance du modèle proposé, nous utilisons les mêmes instances que dans le chapitre 4, puis nous comparons les solutions fournies par le modèle GP avec les solutions trouvées dans le chapitre 4.

Dans les Tableaux 5.3-5.11 de même structure, nous présentons les résultats obtenus en tenant compte des deux objectifs de réutilisation des équipements et en attribuant à chaque fois un poids différent à chacun des deux objectifs. Dans ces tableaux, la deuxième colonne donne le nombre de stations, les troisième et quatrième colonnes présentent le nombre de blocs et le nombre de blocs réutilisés, la cinquième collone reporte le coût de la reconfiguration de la ligne et la sixième le temps de résolution de chaque instance.

L'avantage de la méthode GP réside dans le fait qu'elle ne requiert pas du décideur des informations sur les bénéfices de la réutilisation de chaque équipement. Le décideur peut indiquer tout simplement qu'il souhaite que tous les équipements soient réutilisés. Néanmoins, il devra préciser les poids qu'il donne aux deux objectifs utilisés : la proportion entre ces poids peut correspondre à la proportion des coûts d'une nouvelle machine et un nouveau bloc, par exemple.

Du fait que le modèle GP n'utilise pas les coûts C_1, C_2, C_3, la comparaison entre ses résultats et ceux obtenus en utilisant le modèle MIP2 n'est pas vraiment justifiée. Néanmoins, nous avons calculé les coûts de la reconfiguration en utilisant les mêmes valeurs des coûts que dans le chapitre 4, à savoir : C_1=10, C_2=2, C_3=1. Ensuite, dans les Tableaux 5.8 et 5.9, nous indiquons pour chaque instance, le coût de la reconfiguration obtenu avec le modèle MIP2 et celui du modèle GP.

Dans les Figures 5.7, 5.8 et 5.9, nous comparons les taux de réutilisation des équipements obtenus avec les modèles MIP2 et GP pour $w1 = 0,8$ et $w_2 = 0,2$. Le temps de résolution pour GP est légèrement supérieur à celui de la méthode MIP2 tout en restant du même ordre de grandeur : 1632,85 secondes au total pour GP (39,83 secondes en moyenne) et 1219,46 secondes au total pour MIP2 (29,74

secondes en moyenne).

Tableau 5.3 – Résultats pour GP ($w_1 = 0{,}1$ et $w_2 = 0{,}9$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	3	4	4	14	10	22	3	3	3	13	19,48
2	3	4	3	5	55	23	3	4	4	14	14,5
3	4	5	4	6	51	24	4	4	4	4	6
4	3	4	3	5	35	25	3	3	3	3	24,6
5	4	4	4	14	89,9	26	3	4	3	5	47,4
6	3	3	3	3	3,44	27	3	3	3	3	8,9
7	4	5	4	16	18,6	28	4	4	4	4	76,6
8	4	4	4	14	238,4	29	4	5	4	16	38
9	4	5	5	15	259,1	30	3	4	3	5	162,8
10	3	4	4	4	75,1	31	3	4	4	4	4,8
11	3	5	5	15	24,8	32	3	4	4	14	12,9
12	4	4	4	14	141,6	33	3	5	3	7	125,4
13	3	4	4	4	45,5	34	4	4	4	14	312,6
14	3	4	4	14	17,3	35	3	4	4	14	23,3
15	4	5	4	6	287,4	36	3	4	3	5	53
16	4	5	4	16	4,3	37	3	4	3	5	39,3
17	3	4	3	5	274	38	3	4	4	4	3,7
18	3	4	4	4	61,7	39	3	4	4	4	7,5
19	3	4	4	4	2,5	40	4	4	4	14	229,7
20	4	4	4	4	4,4	41	3	3	3	3	633,2
21	3	5	4	16	50,6	-	-	-	-	-	-

Tableau 5.4 – Résultats pour GP ($w_1 = 0{,}2$ et $w_2 = 0{,}8$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	3	4	4	14	15,44	22	3	3	3	13	19,21
2	3	4	3	5	166,79	23	3	4	4	14	19,14
3	4	5	4	6	160	24	4	4	4	4	5,69
4	3	4	3	5	114,63	25	3	3	3	3	101,55
5	4	5	3	17	79,32	26	3	4	3	5	124,3
6	3	3	3	3	3,29	27	3	3	3	3	9,28
7	4	5	4	16	218,71	28	4	4	4	4	35,42
8	4	4	4	14	204,64	29	4	5	5	15	89,1
9	4	5	5	15	221,91	30	3	4	3	5	159,87
10	3	4	4	4	57,26	31	3	4	4	4	5,25
11	3	5	5	15	173,19	32	3	4	4	14	16,62
12	4	4	4	14	224,14	33	3	5	3	7	434,32
13	3	4	4	4	31,79	34	4	4	4	14	296,46
14	3	4	4	14	5,69	35	3	4	4	14	14,25
15	4	5	2	8	576,43	36	3	4	3	5	207,66
16	3	4	4	4	4,71	37	3	4	3	5	12,97
17	3	4	3	5	25,03	38	3	4	4	4	12,69
18	3	4	4	4	26,28	39	3	4	4	4	7,94
19	3	4	4	4	2,58	40	4	4	4	14	85,06
20	4	4	4	4	4,74	41	3	4	3	5	115,86
21	3	5	4	16	469,85	-	-	-	-	-	-

Tableau 5.5 – Résultats pour GP ($w_1 = 0{,}3$ et $w_2 = 0{,}7$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	3	4	4	14	13,29	22	3	3	3	13	22,3
2	3	4	3	5	122,88	23	3	4	4	14	44,39
3	4	5	4	6	201,1	24	4	4	4	4	5,44
4	3	4	3	5	34,52	25	3	3	3	3	64,91
5	4	4	4	14	34,35	26	3	4	3	5	87,56
6	3	3	3	3	3,24	27	3	3	3	3	8,98
7	4	5	4	16	85,59	28	4	4	4	4	49,28
8	4	4	4	14	97,14	29	4	5	5	15	81,82
9	4	5	5	15	155	30	3	4	3	5	145,98
10	3	4	4	4	56,25	31	3	4	4	4	5,39
11	3	5	5	15	173,77	32	3	4	4	14	15,63
12	4	4	4	14	36,84	33	3	5	3	7	221,95
13	3	4	4	4	23,41	34	4	4	4	14	394,24
14	3	4	4	14	16,06	35	3	3	3	13	17,47
15	4	5	4	6	561,58	36	3	4	3	5	76,33
16	3	4	4	4	4,47	37	3	4	3	5	3,3
17	3	4	3	5	66,61	38	3	4	4	4	16,41
18	3	4	4	4	34,16	39	3	4	4	4	7,39
19	3	4	4	4	2,48	40	4	3	2	14	142,44
20	4	4	4	4	4,43	41	3	4	3	5	39,23
21	3	5	3	17	508,28	-	-	-	-	-	-

Tableau 5.6 – Résultats pour GP ($w_1 = 0{,}4$ et $w_2 = 0{,}6$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	3	4	4	14	11,37	22	3	3	3	13	23,29
2	3	4	3	5	90,94	23	3	4	4	14	15,56
3	4	5	4	6	98,45	24	4	4	4	4	6,22
4	3	4	3	5	18,73	25	3	3	3	3	34,05
5	4	4	4	14	62,44	26	3	4	3	5	105,84
6	3	3	3	3	3,51	27	3	3	3	3	10,62
7	4	5	4	16	72,82	28	4	4	4	4	31,16
8	4	4	4	14	81,02	29	4	5	5	15	11,96
9	4	5	5	15	241,53	30	3	4	3	5	98,49
10	3	4	4	4	79,63	31	3	4	4	4	6,02
11	3	5	5	15	29,82	32	3	3	3	13	20,77
12	4	4	4	14	19,18	33	3	5	3	7	130,4
13	3	4	4	4	36,22	34	4	4	4	14	234,34
14	3	4	4	14	53,6	35	3	4	4	14	13,08
15	4	5	4	6	563,7	36	3	3	2	4	55,05
16	3	4	4	4	4,96	37	3	4	3	5	6,22
17	3	4	3	5	30,18	38	3	4	4	4	4,27
18	3	4	4	4	27,89	39	3	4	4	4	7,87
19	3	4	4	4	2,85	40	4	4	4	14	179,3
20	4	4	4	4	5,39	41	3	4	3	5	23,55
21	3	5	4	16	476,77	-	-	-	-	-	-

Tableau 5.7 – Résultats pour GP ($w_1 = 0{,}5$ et $w_2 = 0{,}5$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	2	5	4	6	11,2	22	2	4	3	5	10,51
2	3	4	3	5	24,67	23	3	4	4	14	20,37
3	4	5	4	6	78,84	24	4	4	4	4	6,28
4	3	4	3	5	23,21	25	3	3	3	3	167,56
5	4	4	4	14	28,45	26	3	4	3	5	97,43
6	3	3	3	3	3,99	27	3	3	3	3	11,27
7	4	5	4	16	40,24	28	4	4	4	4	19,82
8	3	5	4	6	26,11	29	4	5	5	15	36,06
9	4	5	4	16	360,67	30	3	4	3	5	55,16
10	3	4	4	4	78,14	31	3	4	4	4	5,14
11	3	5	5	15	34,14	32	3	4	4	14	12,09
12	4	4	4	14	13,94	33	3	5	3	7	120,35
13	3	4	4	4	17,7	34	4	4	4	6	85,44
14	4	4	3	14	33,25	35	3	4	4	14	17,61
15	4	5	4	6	510,65	36	3	4	3	5	31,07
16	3	4	4	3	4,85	37	3	4	3	5	5,41
17	3	4	3	5	22,55	38	3	4	4	4	5
18	3	4	4	4	15,06	39	3	4	4	4	6,69
19	3	4	4	4	2,91	40	4	4	4	14	136,17
20	4	4	4	4	5,66	41	3	4	3	5	47,26
21	3	5	4	16	306,15	-	-	-	-	-	-

Tableau 5.8 – Résultats pour GP ($w_1 = 0{,}6$ et $w_2 = 0{,}4$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	2	5	4	6	5,39	22	2	4	3	5	16,27
2	3	4	3	5	18,86	23	3	4	4	14	18,92
3	4	5	4	6	36,56	24	4	4	4	4	5,35
4	3	4	3	5	11,16	25	3	3	3	3	58,95
5	4	4	4	14	31,65	26	3	4	3	5	67,9
6	3	3	3	3	3,29	27	3	3	3	3	8,65
7	4	5	4	16	18,18	28	4	4	4	4	22,63
8	3	5	4	6	21,07	29	4	5	5	15	17,61
9	3	6	5	7	315,82	30	3	4	3	5	48,85
10	3	4	4	4	22,97	31	3	4	4	4	5,05
11	3	5	5	15	20,29	32	3	4	4	14	18,84
12	4	4	4	14	6,45	33	3	5	3	7	44,94
13	3	4	4	4	70,13	34	3	5	4	6	87,31
14	3	4	4	14	40,48	35	3	4	4	14	18,01
15	4	5	4	6	59,6	36	3	4	3	5	21,6
16	3	4	4	4	4,83	37	3	4	3	5	3,29
17	3	4	3	5	23,71	38	3	4	4	4	17,45
18	3	4	4	4	42,04	39	3	4	4	4	7,47
19	3	4	4	4	2,66	40	3	5	4	6	55,52
20	4	4	4	4	5,35	41	3	4	3	5	46,7
21	3	5	4	16	147,85	-	-	-	-	-	-

Tableau 5.9 – Résultats pour GP ($w_1 = 0{,}7$ et $w_2 = 0{,}3$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	2	5	4	6	7,42	22	2	4	3	5	13,05
2	3	4	3	5	25,22	23	3	4	4	14	24,91
3	4	5	4	6	21,63	24	4	4	4	4	5,25
4	3	4	3	5	15,69	25	3	3	3	3	50,84
5	4	4	4	14	26,91	26	3	4	3	5	27,79
6	3	3	3	3	3,32	27	3	3	3	3	8,68
7	4	5	4	16	37,45	28	4	4	4	4	32,66
8	3	5	4	6	4,89	29	3	3	3	3	29,43
9	3	6	5	7	183,8	30	3	4	3	5	23,08
10	3	4	4	4	19,15	31	3	4	4	4	5,19
11	3	5	5	15	361,76	32	3	4	4	14	23,01
12	4	4	4	14	4,63	33	3	5	3	7	58,5
13	3	4	4	4	74,22	34	3	5	4	6	68,21
14	3	4	4	14	6,58	35	3	4	4	14	15,45
15	4	5	4	6	460,28	36	3	4	3	5	25,86
16	3	4	4	4	4,38	37	3	4	3	5	3,22
17	3	4	3	5	38,59	38	3	4	4	4	5,41
18	3	4	4	4	26,37	39	3	4	4	4	7,65
19	3	4	4	4	2,35	40	3	5	4	6	46,22
20	4	4	4	4	4,46	41	3	4	3	5	35,78
21	3	5	4	16	120,41	-	-	-	-	-	-

Tableau 5.10 – Résultats pour GP ($w_1 = 0{,}8$ et $w_2 = 0{,}2$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	2	5	4	6	6,19	22	2	4	3	5	9,85
2	3	4	3	5	64,58	23	3	4	4	14	13,43
3	4	5	4	6	51,52	24	4	4	4	4	5,42
4	3	4	3	5	18,95	25	3	3	3	3	45,56
5	4	4	4	14	69,01	26	3	4	3	5	27,73
6	3	3	3	3	3,29	27	3	3	3	3	5,52
7	4	5	4	16	25,44	28	4	4	4	4	35,28
8	3	5	4	6	5,55	29	4	5	5	15	43,49
9	3	6	5	7	203,55	30	3	4	3	5	46,95
10	3	4	4	4	17,25	31	3	4	4	4	4,97
11	3	4	4	14	199,16	32	3	3	3	13	16,36
12	4	4	4	14	4,57	33	3	5	3	7	67,68
13	3	4	4	4	13,61	34	3	5	4	6	47,31
14	3	4	4	14	20,54	35	3	4	4	14	21,65
15	4	5	4	6	178,59	36	3	4	3	5	20,32
16	3	4	4	4	4,85	37	3	4	3	5	4,19
17	3	4	3	5	11,35	38	3	4	4	4	5,91
18	3	4	4	4	31,4	39	3	4	4	4	8,17
19	3	4	4	4	2,43	40	3	5	4	6	156,93
20	4	4	4	4	4,6	41	3	4	3	5	31,6
21	3	5	4	16	78,1	-	-	-	-	-	-

Tableau 5.11 – Résultats pour GP ($w_1 = 0{,}9$ et $w_2 = 0{,}1$)

N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps	N °	#Sts	#Blocs	#Blocs réutilisés	Coût	Temps
1	2	4	3	5	4,97	22	2	4	3	5	8,2
2	3	4	3	5	28,14	23	3	4	4	14	14,53
3	4	5	4	6	54,91	24	4	4	4	4	5,86
4	3	4	3	5	39,29	25	3	3	3	3	56,16
5	4	4	4	14	63,32	26	3	4	3	5	35,73
6	3	3	3	3	3,44	27	3	3	3	3	5,94
7	4	5	4	16	46,95	28	4	4	4	4	91,29
8	3	5	4	6	5,31	29	4	4	4	14	28,92
9	3	6	5	7	95,02	30	3	4	3	5	58,37
10	3	4	4	4	21,48	31	3	4	4	4	5,08
11	3	5	5	15	20,03	32	3	4	4	14	17,2
12	4	4	4	14	4,29	33	3	5	3	7	43,3
13	3	4	4	4	12,1	34	3	5	4	6	38,2
14	3	4	4	14	12,83	35	3	4	3	14	21,79
15	4	5	4	6	135,05	36	3	4	3	5	21,55
16	3	4	4	4	4,47	37	3	4	3	5	4,29
17	3	4	3	5	9,06	38	3	4	4	4	5,24
18	3	4	4	4	17,51	39	3	4	4	4	8,19
19	3	4	4	4	2,43	40	3	5	4	6	33,3
20	4	4	4	4	4,58	41	3	4	3	5	16,58
21	3	5	4	16	50,48	-	-	-	-	-	-

Tableau 5.12 – Comparaison entre MIP et GP

N °	MIP	w_1=0,1	w_1=0,2	w_1=0,3	w_1=0,4	w_1=0,5	w_1=0,6	w_1=0,7	w_1=0,8	w_1=0,9
		w_2=0,9	w_2=0,8	w_2=0,7	w_2=0,6	w_2=0,5	w_2=0,4	w_2=0,3	w_2=0,2	w_2=0,1
1	6	14	14	14	14	6	6	6	6	6
2	5	5	5	5	5	5	5	5	5	5
3	6	6	6	6	6	6	6	6	6	6
4	5	5	5	5	5	5	5	5	5	5
5	14	14	17	14	14	14	14	14	14	14
6	3	3	3	3	3	3	3	3	3	3
7	16	16	16	16	16	16	16	16	16	16
8	6	14	14	14	14	6	6	6	6	6
9	7	15	15	15	15	16	7	7	7	7
10	4	4	4	4	4	4	4	4	4	4
11	14	15	15	15	15	15	15	15	14	15
12	14	14	14	14	14	14	14	14	14	14
13	4	4	4	4	4	4	4	4	4	4
14	14	14	14	14	14	14	14	14	14	14
15	6	6	8	6	6	6	6	6	6	6
16	4	4	4	4	4	4	4	4	4	4
17	5	5	5	5	5	5	5	5	5	5
18	4	4	4	4	4	4	4	4	4	4
19	4	4	4	4	4	4	4	4	4	4
20	4	4	4	4	4	4	4	4	4	4
21	16	16	16	17	16	16	16	16	16	16

Tableau 5.13 – Comparaison entre MIP et GP

N°	MIP	w_1=0,1	w_1=0,2	w_1=0,3	w_1=0,4	w_1=0,5	w_1=0,6	w_1=0,7	w_1=0,8	w_1=0,9
		w_2=0,9	w_2=0,8	w_2=0,7	w_2=0,6	w_2=0,5	w_2=0,4	w_2=0,3	w_2=0,2	w_2=0,1
22	5	13	13	13	13	5	5	5	5	5
23	14	14	14	14	14	14	14	14	14	14
24	4	4	4	4	4	4	4	4	4	4
25	3	3	3	3	3	3	3	3	3	3
26	5	5	5	5	5	5	5	5	5	5
27	3	3	3	3	3	3	3	3	3	3
28	4	4	4	4	4	4	4	4	4	4
29	15	16	15	15	15	15	15	16	15	16
30	5	5	5	5	5	5	5	5	5	5
31	4	4	4	4	4	4	4	4	4	4
32	13	14	14	14	13	14	14	14	13	14
33	7	7	7	7	7	7	7	7	7	7
34	6	14	14	14	14	6	6	6	6	6
35	13	14	14	13	14	14	14	14	14	14
36	4	5	5	5	4	5	5	5	5	5
37	5	5	5	5	5	5	5	5	5	5
38	4	4	4	4	4	4	4	4	4	4
39	4	4	4	4	4	4	4	4	4	4
40	6	14	14	14	14	14	6	6	6	6
41	5	5	5	5	5	5	5	5	5	5

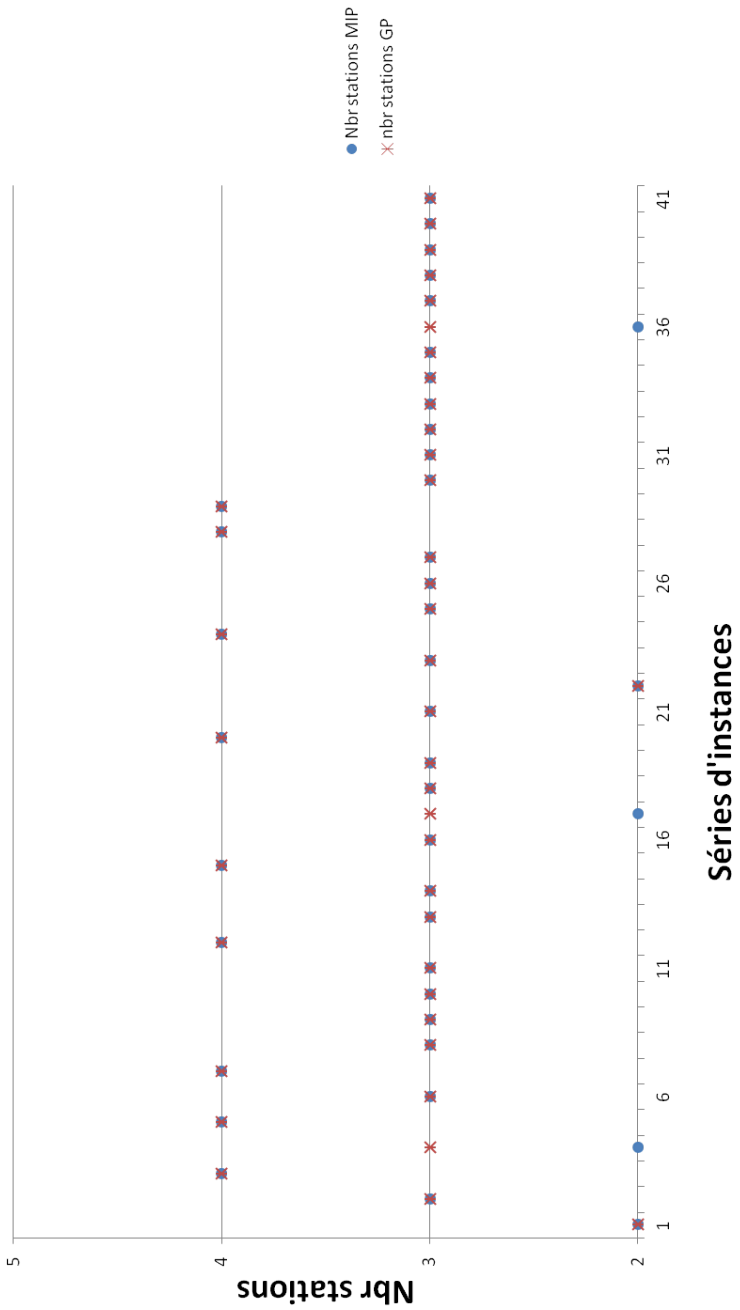


FIGURE 5.7 – Comparaison au regard du critère du nombre de stations

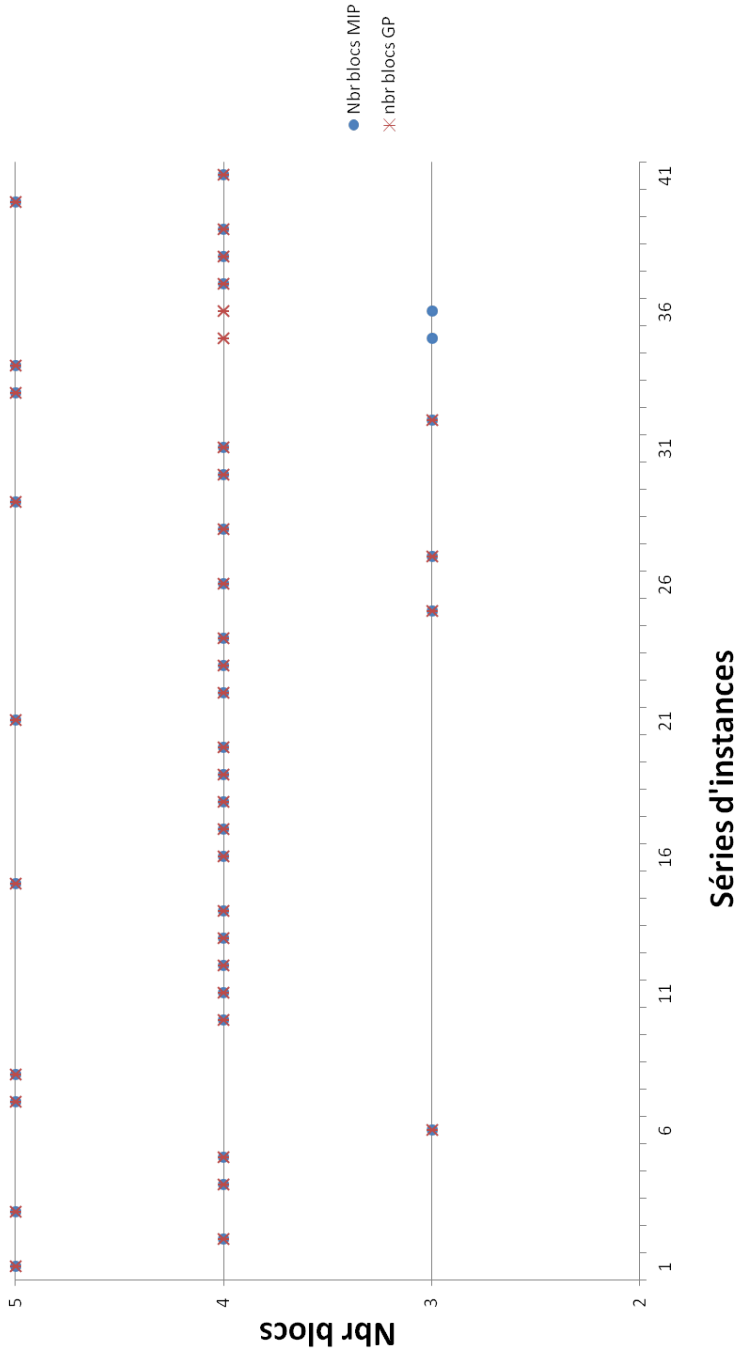


FIGURE 5.8 – Comparaison au regard du critère du nombre de blocs

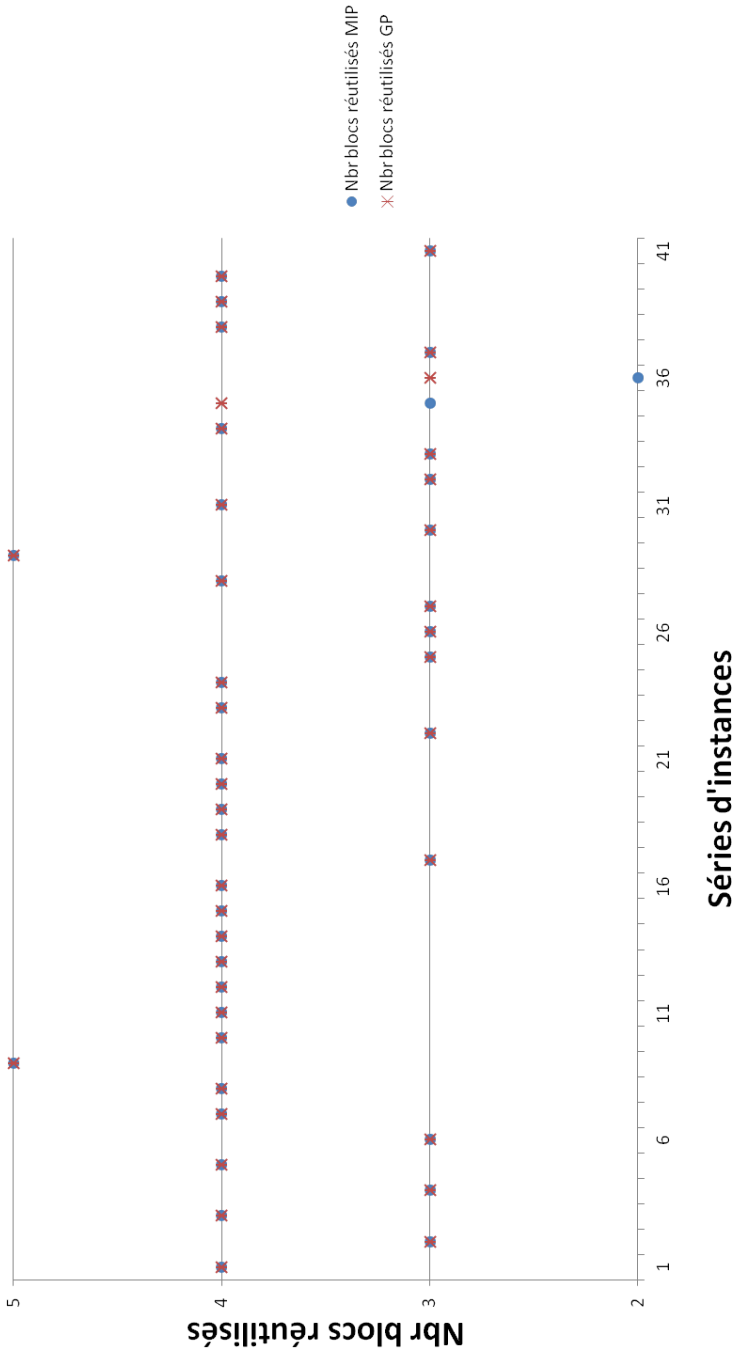


FIGURE 5.9 – Comparaison au regard du critère du nombre de blocs réutilisés

Dans un premier temps, nous comparons la réutilisation des stations réalisée par les deux méthodes. La Figure 5.5 montre que les résultats obtenus sont très proches. Néanmoins, pour 3 instances, le modèle GP fournit des solutions qui réutilisent toutes les stations de la ligne initiale, alors que le modèle MIP2 élimine une station à chaque fois.

Nous pouvons aussi constater que les blocs sont mieux réutilisés grâce à la méthode GP pour les instances 35 et 36. Ainsi nous pouvons conclure que la méthode GP atteint mieux l'objectif de maximisation des équipements réutilisés. Néanmoins, pour certaines valeurs des poids attribués aux objectifs, les coût de la reconfiguration sont plus élevés que ceux pour les solutions trouvées en utilisant le modèle MIP2.

Comme nous avons déjà énoncé, la comparaison des coûts de reconfiguration entre les deux méthodes n'est pas vraiment possible, car la méthode GP devrait être utilisée si la valeur C_3 représentant le bénéfice de la réutilisation d'un bloc ne peut pas être estimée. Cependant, si nous utilisons les mêmes valeurs des coûts que dans le chapitre 4, à savoir : C_1=10, C_2=2, C_3=1, les coûts de reconfiguration fournis par la méthode GP avec $w_1 = 0,8$ $w_2 = 0,2$ sont identiques à ceux obtenus avec le modèle MIP2 sauf pour les instances 35 et 36. Ainsi la réutilisation de tous les blocs de la ligne initiale résulte en une augmentation des coûts de reconfiguration par rapport au résultat obtenu par la méthode MIP2. Il est à noter que cette observation ne serait plus valable, si les valeurs suivantes avaient été utilisées : C_1=10, C_2=2, C_3=2. Bien évidemment, la qualité des solutions fournies par la méthode GP en termes de nombre de stations et de blocs réutilisés dépend fortement des poids w_1 et w_2 attribués aux objectifs du programme GP.

5.5 Conclusion

Dans ce chapitre, nous avons développé la méthode goal programming pour résoudre le problème de la reconfiguration des lignes de transfert. Les deux objectifs suivants étaient visés : réutiliser toutes les machines et tous les blocs existants dans la ligne initiale. Cette méthode a été appliquée sur le même échantillon de problèmes que dans le chapitre 4.

La comparaison des deux méthodes nous permet de conclure que si le bénéfice de la réutilisation des anciens équipements peut être estimé avec précision acceptable, les modèles MIP sont plus efficaces pour résoudre le problème de reconfiguration

des lignes de transfert. Cependant, si ce bénéfice n'est pas mesurable, la méthode GP est assez efficace pour maximiser le nombre d'équipements réutilisés.

Nos travaux sur le développement de la méthode Goal Programming pour la reconfiguration des lignes de transfert ont abouti à un article publié dans les actes de la conférence internationale ICINCO'2012 [Makssoud *et al.* 2012a]. Un article a également été soumis à la revue internationale RAIRO - Operations Research.

Chapitre 6

Une méthode exacte pour le problème de rééquilibrage des lignes d'assemblage

Sommaire

6.1 Définition du problème 108
6.2 Modèle mathématique 109
6.3 Linéarisation du modèle 110
6.4 Exemple illustratif 112
6.5 Expérimentations numériques 114
6.6 Conclusion 118

La formulation la plus connue du problème d'équilibrage des lignes d'assemblage vise à trouver un nombre minimal de stations de travail nécessaires pour affecter un ensemble donné d'opérations V en tenant compte des contraintes de précédence et du temps de cycle. Les contraintes de précédence sont données par un graphe orienté acyclique $G = (V, E)$ sur cet ensemble d'opérations, où chaque arc $(i, j) \in E$ indique que l'opération i est un prédécesseur de l'opération j et donc elle doit être affectée à une station précédente ou à la même station que l'opération j.

Chaque opération $j \in V$ est aussi caractérisée par son temps d'exćution t_j. La somme des temps d'opérations affectées à la même station de travail ne doit pas dépasser un temps de cycle donné noté par T_0. Ce problème est connu NP-difficile. Même s'il a été introduit dans la littérature en 1955 [Salveson 1955], de nombreuses études récentes abordent ce probléme [Battaïa & Dolgui 2013; Bautista & Pereira 2009; Kilincci 2011; Pastor & Ferrer 2009; Sewell & Jacobson 2012].

Cependant, en raison des changements fréquents dans les caractéristiques du produit et de la demande, le problème qui se pose souvent est de savoir comment réaffecter l'ensemble des opérations afin de répondre aux nouvelles exigences de production en minimisant le nombre de modifications à faire dans la ligne existante.

Nous appelons ce problème : Simple Assembly Line Rebalancing Problem (SALReBP), nous proposons une méthode exacte pour le résoudre.

6.1 Définition du problème

Dans cette section, nous présentons une définition formelle du SALPeBP. Comme mentionné précédemment, nous désignons par T_0 le temps de de cycle et par m le nombre de stations de travail.

Les notations suivantes sont utilisées dans notre modèle mathématique :

- Indices :
  - i, j pour les opérations,
  - k pour les stations de travail.
- Paramètres :
  - V^* l'ensemble initial d'opérations , $i \in V^*$,
  - V le nouvel ensemble d'opérations, $j \in V$,
  - t_j le temps d'exécution d'une opération j, $j \in V$,
  - $M = \{1, 2, ..., m\}$ est l'ensemble de stations dans la ligne existante,
  - $L = \{1, 2, ..., l\}$ est l'ensemble de stations dans la nouvelle ligne, où l est une borne supérieure sur le nombre de stations pour la nouvelle ligne,
  - $Q(j)$ est l'intervalle de stations dans la nouvelle ligne, où l'opération $j \in V$ peut être affectée. Cet intervalle est calculé en utilisant les contraintes de précédence,
  - $G = (V, A)$ est un graphe orienté acyclique direct qui représente les contraintes de précédence,
  - ES est un ensemble de pairs d'opérations qui ne doivent pas être affectées ensemble à la même station.

– Variables de décision :

$$x^*_{jk} = \begin{cases} 1, \text{si l'opération } j \in V^* \subset V \text{ est affectée à la station} \\ \text{de travail } k \text{ dans la configuration initiale,} \\ 0, \quad \texttt{sinon} \end{cases}$$

$$y_{jk} = \begin{cases} 1, \text{si l'opération } j \in V \text{ est affectée à la station} \\ \text{de travail } k \text{ dans la nouvelle solution,} \\ 0, \text{sinon} \end{cases}$$

6.2 Modèle mathématique

Nous proposons le modèle suivant pour le problème de rééquilibrage des lignes d'assemblage.

La fonction objectif 6.1 consiste à minimiser les changements dans l'affectation d'opérations existantes.

$$Minimize \sum_{j \in V^*} \sum_{k \in L} \mid x^*_{jk} - y_{jk} \mid \tag{6.1}$$

La contrainte 6.2 garantie que chaque opération j est assignée à une et une seule station de travail.

$$\sum_{k \in Q(j)} y_{jk} = 1, \forall j \in V \tag{6.2}$$

La contrainte 6.3 impose les contraintes de précédence.

$$\sum_{k \in L} k y_{ik} \leq \sum_{k \in L} k y_{jk}, \forall (i,j) \in A \tag{6.3}$$

La contrainte 6.4 assure que la durée totale des opérations affectées à la station de travail k ne dépasse pas le temps de cycle.

$$\sum_{j \in V} t_j y_{jk} \leq T_0, \forall k \in L \tag{6.4}$$

La contrainte 6.5 porte sur l'impossibilité d'exécution de certaines opérations à la même station.

$$y_{ik} + y_{jk} \leq 1, \forall \{i,j\} \in E, \forall k \in L \tag{6.5}$$

La contrainte 6.6 assure que les variables y_{jk} hors les intervalles $Q(j)$ sont mises à 0.

$$y_{jk} = 0, \forall j \in V, \forall k \notin Q(j) \tag{6.6}$$

La contrainte 6.7 montre que les variables x^*_{jk} ne sont définies que pour m stations existant dans la ligne initiale.

$$x^*_{jk} = 0, \forall j \in V, \forall k > m \tag{6.7}$$

6.3 Linéarisation du modèle

Dans la suite, nous proposons une linéarisation du modèle (6.1 - 6.7).

Lemme 1 Soient $x, y, z \in \{0,1\}$. Alors l'expression logique suivante

$$\text{si } x = 1 \text{ et } y = 1, \text{ alors } z = 1$$

peut être modélisée comme suit :

$$x + y \leq z + 1.$$

Lemme 2 Soient $x, y \in \{0,1\}$. Alors l'expression non-linéaire suivante

$$z := \mid x - y \mid$$

peut être linéarisée en utilisant les inégalités suivantes

$$x + y \leq (1 - z) + 1,$$
$$x + (1 - y) \leq z + 1,$$
$$(1 - x) + y \leq z + 1,$$
$$(1 - x) + (1 - y) \leq (1 - z) + 1,$$

Il est évident que $z \in \{0,1\}$. Par ailleurs, seulement les quatre cas suivants sont possibles :

$$\texttt{si } x = 1 \texttt{ et } y = 1; \texttt{ alors } z = 0,$$

$$\texttt{si } x = 1 \texttt{ et } y = 0; \texttt{ alors } z = 1,$$

$$\texttt{si } x = 0 \texttt{ et } y = 1; \texttt{ alors } z = 1,$$

$$\texttt{si } x = 0 \texttt{ et } y = 0; \texttt{ alors } z = 0.$$

En appliquant le Lemme 1 à ces quatre cas, nous obtenons les inégalités nécessaires.

Pour linéariser le problème (6.1 - 6.7), nous introduisons donc une nouvelle variable $z_{jk} :=\mid x^*_{jk} - y_{jk} \mid$.

En utilisant le Lemme 2, nous obtenons :

$$Minimize \sum_{j \in V^*} \sum_{k \in L} z_{jk} \tag{6.8}$$

$$\sum_{k \in Q(j)} y_{jk} = 1, \forall j \in V \tag{6.9}$$

$$\sum_{k \in L} k y_{ik} \leq \sum_{k \in L} k y_{jk}, \forall (i,j) \in A \tag{6.10}$$

$$\sum_{j \in V} t_j y_{jk} \leq T_0, \forall k \in L \tag{6.11}$$

$$y_{ik} + y_{jk} \leq 1, \forall \{i,j\} \in E, \forall k \in L \tag{6.12}$$

$$y_{jk} = 0, \forall j \in V, \forall k \notin Q(j) \tag{6.13}$$

$$x^*_{jk} = 0, \forall j \in V, \forall k > m \tag{6.14}$$

$$x^*_{jk} + y_{jk} \leq (1 - z_{jk}) + 1, \forall j \in V, \forall k \in L \tag{6.15}$$

$$x^*_{jk} + (1 - y_{jk}) \leq z_{jk} + 1, \forall j \in V, \forall k \in L \tag{6.16}$$

$$(1 - x^*_{jk}) + y_{jk} \leq z_{jk} + 1, \forall j \in V, \forall k \in L \tag{6.17}$$

$$(1 - x^*_{jk}) + (1 - y_{jk}) \leq (1 - z_{jk}) + 1, \forall j \in V, \forall k \in L \tag{6.18}$$

6.4 Exemple illustratif

Dans cette section, un exemple est présenté afin d'illustrer le problème étudié. La Figure 6.1 montre le graphe de précédence. Les temps d'exécution des opérations sont reportés dans le Tableau 6.1 et le temps de cycle $T_0 = 100$.

La configuration optimale de la ligne d'assemblage initiale est donnée dans la Figure 6.2 qui comporte quatre stations de travail. Sur la première station, les opérations {1,2,3,4,5,6,7,8,9,11,15,19,26} sont exécutées. La deuxième station effectue les opérations {10,12,13,14}. La troisième réalise l'ensemble d'opérations {17,18,20,21,11}. Enfin à la quatrième, l'ensemble d'opérations suivant est affecté : {23,24,25,26,28,29,30}.

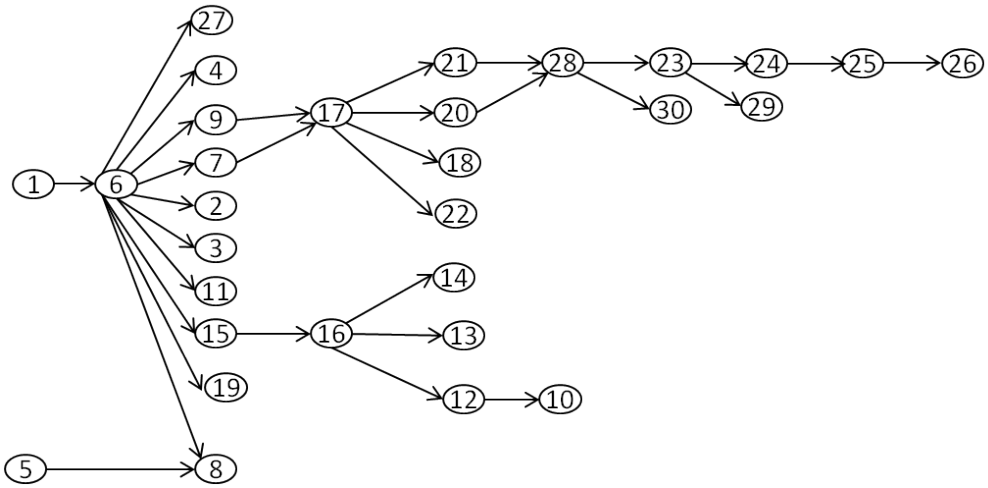


FIGURE 6.1 – Graphe de précédence

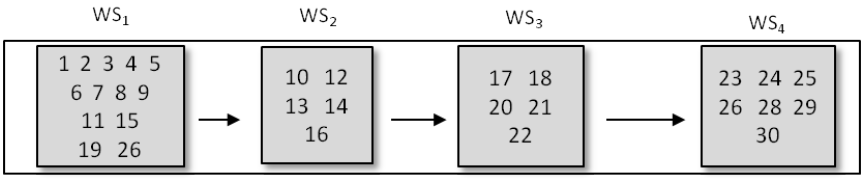


FIGURE 6.2 – Ligne initiale

La ligne initiale doit être rééquilibrée pour un produit modifié où les opérations {14, 19, 23, 28, 29} ont été supprimées et les opérations {31, 32, 33, 34, 35} ont été introduites. Le nouveau diagramme de précédence est présenté dans la Figure 6.3.

Tableau 6.1 – Ensemble d'opérations initial et les temps d'exécution

Opération	Temps (s)	Opération	Temps (s)	Opération	Temps (s)
1	0,93	11	1	21	0,09
2	1,06	12	0,78	22	0,17
3	0,68	13	0,64	23	0,09
4	0,16	14	0,09	24	0,12
5	0,68	15	0,17	25	0,91
6	0,16	16	0,09	26	0,72
7	0,68	17	0,12	27	0,15
8	0,16	18	1	28	0,19
9	0,68	19	0,78	29	0,33
10	0,16	20	0,6	30	0,97

Les nouvelles contraintes d'exclusion pour cet exemple au niveau des stations de travail sont : {{1, 4}, {1, 17}, {1, 20}, {2, 11}, {3, 24}, {3, 7}, {4, 15}, {5, 22}, {6, 24}, {8, 21}, {9, 22}, {10, 15}, {11, 31}, {12, 13}, {12, 20}, {13, 28}, {15, 17}, {16, 17}, {22, 26}, {30, 33}, {31, 32}, {33, 35}}.

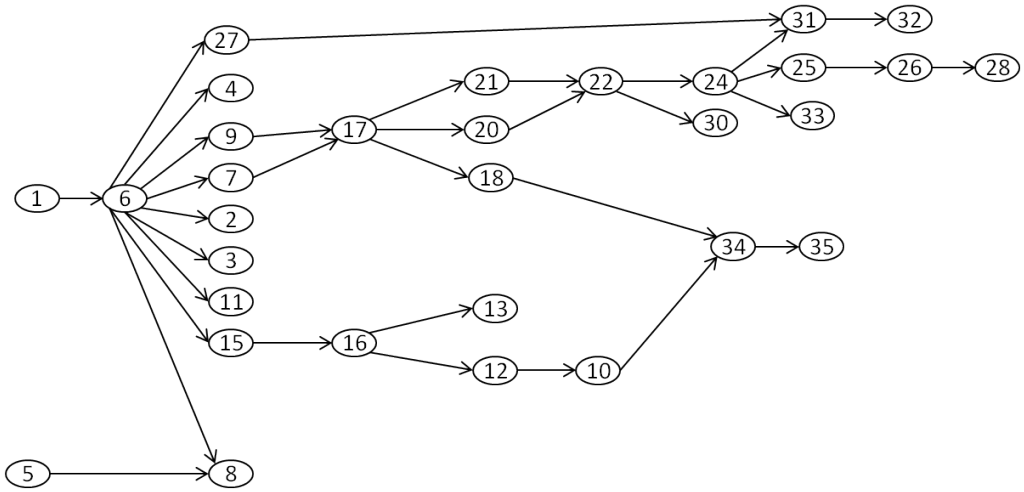


FIGURE 6.3 – Diagramme de précédence après modification

Pour résoudre la problème (6.8 - 6.18) pour cet exemple, le solveur ILOG CPLEX a été utilisé sur un PC Intel(R), 2,20 GHz, avec 8 Go de RAM. La solution optimale a été obtenue en 0.36 seconde. Elle comporte cinq stations où les opérations sui-

Tableau 6.2 – Ensemble d'opérations V et les temps d'exécution

Opération	Temps (s)	Opération	Temps (s)	Opération	Temps (s)
1	0,93	11	1	24	0,09
2	1,06	12	0,78	25	0,17
3	0,68	13	0,64	26	0,09
4	0,16	15	0,09	27	0,12
5	0,68	16	0,17	30	0,91
6	0,16	17	0,09	31	0,72
7	0,68	18	0,12	32	0,15
8	0,16	20	1	33	0,19
9	0,68	21	0,78	34	0,33
10	0,16	22	0,6	35	0,97

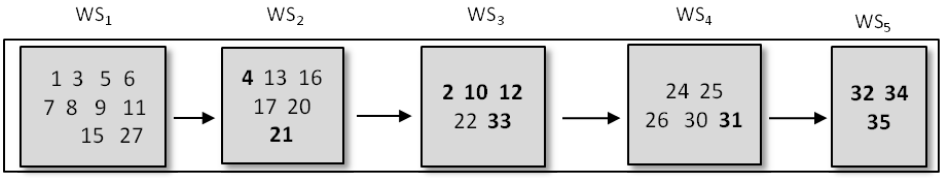


FIGURE 6.4 – Ligne Rééquilibrée

vantes : {2, 4, 10, 12, 21, 31, 32, 33, 34, 35} ont été réaffectées, elles sont montrées en gras dans la Figure 6.4 qui illustre la ligne rééquilibrée.

Ainsi par exemple pour l'opération 2, x^*_{21} =1 et $y_{23} = 1$, c'est-à-dire l'opération 2 était exécutée à la station 1 dans la ligne initiale, mais elle a été affectée à la station 3 lors de la reconfigration. Il est à noter qu'une nouvelle station de travail a été introduite dans la ligne modifiée.

6.5 Expérimentations numériques

La méthode proposée a été évaluée sur un ensemble de données qui se compose de 42 problèmes de 25 opérations chacun. Pour chaque problème initial, 3 versions du produit modifié ont été générées :

- en chageant 25% des opérations (cinq opérations supprimées et cinq ajoutées) ;
- en changeant 50% des opérations (cinq opérations supprimées et huit ajoutées) ;
- en changeant 75% des opérations (cinq opérations supprimées et 14 ajoutées).

Les résultats de calcul sont respectivement présentés dans les Tables 6.3, 6.4 et 6.5 où les valeurs de la fonction objectif indiquant le nombre d'opérations qui ont été réaffectées et le temps de la solution sont donnés. Les expérimentations ont été effectuées sur un ordinateur ayant un PC Intel(R), 2,20 GHz, avec 8 Go de RAM. Le modèle a été codé en utilisant le langage C++ avec le solveur ILOG CPLEX 12.4.

Tableau 6.3 – Résultats pour le rééquilibrage : 25% des opérations modifiées

Num ex	Ops réaffectées	Temps (s)	Num ex	Ops réaffectées	Temps (s)	Num ex	Ops réaffectées	Temps (s)
1	22	0,45	15	4	0,07	29	17	1,6
2	22	0,42	16	14	0,92	30	13	1,62
3	8	0,12	17	6	0,51	31	10	0,98
4	12	0,04	18	11	0,1	32	21	0,12
5	13	0,96	19	12	0,2	33	13	0,31
6	10	0,15	20	5	0,74	34	13	0,07
7	10	0,09	21	23	0,09	35	24	0,99
8	14	1,49	22	21	0,12	36	11	0,32
9	16	0,93	23	19	0,12	37	14	0,10
10	9	0,09	24	4	0,68	38	12	1,23
11	24	0,07	25	13	0,43	39	17	0,18
12	13	0,12	26	10	0,14	40	10	0,17
13	11	0,6	27	12	0,1	41	12	0,17
14	22	0,04	28	4	0,06	42	15	0,87

Tableau 6.4 – Résultats pour le rééquilibrage : 50% des opérations modifiées

Num ex	Ops réaffectées	Temps (s)	Num ex	Ops réaffectées	Temps (s)	Num ex	Ops réaffectées	Temps (s)
1	23	0,2	15	6	0,45	29	21	0,51
2	27	0,54	16	14	0,18	30	13	0,18
3	14	1,04	17	5	0,29	31	10	1,07
4	13	0,4	18	15	0,2	32	20	0,24
5	21	0,29	19	12	0,07	33	15	0,14
6	12	0,51	20	5	0,18	34	16	0,12
7	11	0,12	21	23	1,31	35	24	0,29
8	14	0,37	22	20	0,14	36	15	0,14
9	18	0,67	23	20	0,23	37	15	0,18
10	12	0,92	24	4	0,21	38	15	0,31
11	24	0,1	25	12	0,1	39	18	0,2
12	14	0,2	26	11	0,09	40	10	0,17
13	14	0,59	27	13	0,17	41	13	0,6
14	22	0,18	28	10	0,98	42	17	0,53

Tableau 6.5 – Résultats pour le rééquilibrage : 75% des opérations modifiées

Num ex	Ops réaffectées	Temps (s)	Num ex	Ops réaffectées	Temps (s)	Num ex	Ops réaffectées	Temps (s)
1	23	1,07	15	12	0,29	29	27	0,45
2	23	0,85	16	20	0,23	30	18	0,14
3	16	0,96	17	5	0,39	31	16	0,23
4	19	1,09	18	20	0,21	32	26	0,18
5	27	1,35	19	18	0,14	33	21	0,93
6	18	0,63	20	11	0,2	34	22	0,26
7	15	0,18	21	29	0,24	35	29	0,45
8	20	0,45	22	26	0,12	36	21	0,43
9	24	0,59	23	26	0,18	37	21	0,35
10	17	0,15	24	9	1,06	38	21	0,74
11	30	0,1	25	18	0,1	39	23	0,28
12	20	0,32	26	17	0,15	40	16	0,17
13	19	0,99	27	18	0,34	41	19	0,23
14	27	0,12	28	16	1,13	42	22	1,02

6.6 Conclusion

Le problème de rééquilibrage des lignes d'assemblages a été abordé. Nous avons présenté un modèle d'optimisation mathématique qui vise à minimiser le nombre de changements dans la ligne initiale. Sur la base d'une étude de cas, nous avons illustré l'application du modèle. Nous avons également testé notre approche sur 42 autres exemples, les tests ont révélé que le modèle est capable de résoudre des problèmes ayant au maximum 34 opérations dans le diagramme de précédence dans un temps raisonnable (1,07 seconde au plus).

Nous avons présenté les résultats de nos travaux sur le développement de modèles MIP pour la reconfiguration des lignes d'assemblage dans deux articles publiés dans les actes des conférences internationales : ECCO'2013 [Makssoud *et al.* 2013a] et APMS'2013 [Makssoud *et al.* 2013b] ainsi que dans un chapitre de livre [Makssoud *et al.* 2013d]. Un article pour revue internationale est actuellement en préparation.

Conclusion générale

Ce mémoire est une des premières études sur la reconfiguration des lignes de production, notamment des lignes d'usinage à boîtiers multibroches. Pour les systèmes qui ne n'ont pas été conçus pour être flexibles ou reconfigurables, le décideur doit s'appuyer sur des méthodes d'optimisation efficaces pour minimiser les coûts nécessaires pour leur adaptation à un nouveau contexte de production. Le développement de telles méthodes était le but principal de nos travaux.

Nous avons entamé cette thèse par l'étude présentée dans le chapitre 1 des particularités des systèmes de production tels que les lignes de transfert et les lignes d'assemblage manuelles ainsi que de la problématique de leur conception. Un intérêt particulier a été porté sur le problème d'équilibrage de ces systèmes connu dans la littérature sous le nom générique « Line Balancing Problem ». Une étude bibliographique des méthodes d'optimisation combinatoire exactes et approchées utilisées pour la résolution de ce problème a été présentée dans le chapitre 2.

Dans le chapitre 3, nous nous sommes intéressés à l'étude des concepts de reconfigurabilité des systèmes de production. Nous avons notamment étudié le rôle de la reconfiguration dans le cycle de vie d'un système de production. Une revue des approches existantes dans la littérature a été faite pour les lignes de transfert et les lignes d'assemblage manuelles. Ensuite, nous avons présenté une analyse des études concernant la reconfiguration et reconfigurabilité des systèmes existants dans la littérature. Cette analyse nous a permis de conclure que la plupart des travaux ont été dédiés à des systèmes reconfigurables par définition et peu de méthodes permettaient aux décideurs de piloter la reconfiguration d'une ligne de transfert ou d'une ligne d'assemblage. Par conséquent, nous avons décidé d'axer nos travaux de recherche sur le développement de telles méthodes. Des méthodes d'optimisation multi-objectif ont également été présentées.

Dans le chapitre 4, nous avons formulé le problème d'optimisation de la reconfiguration des lignes de transfert et nous avons proposé deux programmes linéaires en variables mixtes (MIP). Les lignes étudiées appartiennent à la classe des systèmes d'usinage à boîtiers multibroches ayant un mode d'activation séquentiel des boîtiers. L'objectif de la reconfiguration de tels systèmes est de minimiser les nouveaux investissement en réutilisant au maximum les équipements existants. Néanmoins, les contraintes techniques et technologiques entre les anciens boîtiers d'usinage et les nouvelles opérations à effectuer sur le produit doivent être respectées. L'évaluation de ces deux modèles a été réalisée à l'aide du solveur ILOG Cplex, sur un échantillon des cas d'étude. Les résultats obtenus sont encourageants. Nous avons notamment

montré que la reconfiguration d'une ligne existante permet des économies d'investissement jusqu'à 91% par rapport à l'installation d'une nouvelle ligne. Comme il s'agit ici d'une première formulation pour un problème de grande importance dans l'industrie, nombreuses sont les pistes possibles pour les recherches futures. D'abord, il convient de formaliser et étudier d'autres hypothèses de reutilisabilité des équipements, notamment celles évoquées dans le chapitre 3, mais qui n'ont pas été considérées dans ce mémoire. Ensuite, une approche méta-heuristique serait nécessaire pour traiter des problèmes de plus grande taille. Une autre piste intéressante concerne l'étude de stabilité des solutions proposées et notamment l'intégration de ce critère lors de la reconfiguration afin de mettre en place des solutions de reconfiguration qui non seulement minimisent les investissement nécessaires, mais aussi seront plus robustes aux futurs changements dans le contexte de production. Puis, le moment de la reconfiguration peut s'avérer opportun pour évaluer et améliorer les performances environnementales d'un système de production. Ainsi on pourrait envisager de considérer à la fois la performance économique (le coût de la mise en place et de fonctionnement), mais aussi la performance environnementale (consommation énergétique, recyclage des équipements obsolètes) des solutions proposées aux décideurs.

En se basant sur les résultats obtenus dans le chapitre 4, nous avons développé dans le chapitre 5, une méthode de programmation par but (Goal Programming) pour traiter le problème de reconfiguration de mêmes lignes d'usinage mais en formulant différemment l'objectif d'optimisation. En effet, le premier but était d'utiliser le même nombre de machines comme dans la configuration initiale et le deuxième objectif était de réutiliser tous les boîtiers d'usinage employés dans ligne initiale. Ici aussi les pistes d'amélioration sont multiples. L'approche de goal programming étant assez flexible pour l'introduction de nouveaux critères d'optimisation, il pourrait être intéressant d'étudier les solutions obtenues en considérant d'autres objectifs de reconfiguration comme ceux déjà évoqués ci-dessus, à savoir : la robustesse, les performances environnementales, etc.

Dans le chapitre 6, nous avons considéré le problème de rééquilibrage des lignes d'assemblage manuelles. Une définition formelle du SALReBP (Simple Assembly Line Rebalancing Problem) est présentée qui a comme objectif de minimiser le nombre de changements dans l'affectation des opérations existantes. En effet, les changements des opérations imposent un coût d'apprentissage pour les opérateurs. Afin de minimiser ce coût, il convient donc de minimiser le nombre de changements. Le modèle initial étant non linéaire, sa linéarisation a été proposée en utilisant des variables de décision complémentaires. Le modèle PLNE ainsi obtenu a été résolu à l'aide du solveur ILOG Cplex. Des études de ses performances ont été réalisées sur un échantillon de problèmes de référence dans la littérature. Les résultats obtenus sont

satisfaisants, car toutes les instances ont été résolues en un temps raisonnable. Cette première étude étant un succès, elle ouvre la voie pour des nombreuses recherches futures. En effet, seule la ligne dite “simple” a été concernée par la modélisation réalisée. Ainsi les mêmes idées pourront être reprises afin de modéliser et étudier le problème de reconfiguration des lignes d'assemblage ayant des architectures différentes, comme par exemple, les lignes en forme de U, les lignes à deux côtés, les lignes à machines parallèles, voire les lignes parallèles. Le même problème pourrait être considéré en tenant compte de la nature stochastique des temps d'exécution des opérations manuelles. Le modèle pourrait également être enrichi par l'introduction des critères d'optimisation complémentaires tels que : l'équilibre de la charge entre les opérateurs, la considération des paramètres ergonomiques, etc.

Nous devons aussi souligner que l'état de l'art présenté dans les chapitres 2 et 3 montre que des méthodes d'optimisation mutli-objectif dites a posteriori et progressives n'ont pas été mises en œuvre pour la résolution des problèmes de reconfiguration des systèmes d'usinage ou des systèmes d'assemblage. Ainsi le développement de telles méthodes présente une piste intéressante dans le domaine de la recherche opérationnelle.

Bibliographie

G.R. Aase, J.R. Olson et M.J. Schniederjans. *U-shaped assembly line layouts and their impact on labor productivity : An experimental study.* European Journal of Operational Research, vol. 156, no. 3, pages 698–711, 2004. (Cité en page 11.)

M. Abbasi et M. Houshmand. *Production planning of reconfigurable manufacturing systems with stochastic demands using Tabu search.* International Journal of Manufacturing Technology and Management, vol. 17, no. 1, pages 125–148, 2009. (Cité en pages 8 et 50.)

M.R. Abdi et A.W. Labib. *A design strategy for reconfigurable manufacturing systems (RMSs) using analytical hierarchical process (AHP) : a case study.* International Journal of Production Research, vol. 41, no. 10, pages 2273–2299, 2003. (Cité en page 8.)

M.R. Abdi et A.W. Labib. *Grouping and selecting products : the design key of Reconfigurable Manufacturing Systems (RMSs).* International journal of production research, vol. 42, no. 3, pages 521–546, 2004. (Cité en page 8.)

M.R. Abdi et A.W. Labib. *Performance evaluation of reconfigurable manufacturing systems via holonic architecture and the analytic network process.* International Journal of Production Research, vol. 49, no. 5, pages 1319–1335, 2011. (Cité en page 8.)

M.R. Abdi. *Fuzzy multi-criteria decision model for evaluating reconfigurable machines.* International Journal of Production Economics, vol. 117, no. 1, pages 1–15, 2009. (Cité en page 8.)

M.R. Abdi. *Layout configuration selection for reconfigurable manufacturing systems using the fuzzy AHP.* International Journal of Manufacturing Technology and Management, vol. 17, no. 1, pages 149–165, 2009. (Cité en page 8.)

K. Ağpak et H. Gökçen. *A chance-constrained approach to stochastic line balancing problem.* European Journal of Operational Research, vol. 180, no. 3, pages 1098–1115, 2007. (Cité en page 42.)

K. Agpak. *An approach to find task sequence for re-balancing of assembly lines.* Assembly Automation, vol. 30, no. 4, pages 378–387, 2010. (Cité en page 54.)

D.A. Ajenblit et R.L Wainwright. *Applying genetic algorithms to the U-shaped assembly line balancing problem.* Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., pages 96–101, 1998. (Cité en page 11.)

Y. Alsafi et V. Vyatkin. *Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing.* Robotics and Computer-Integrated Manufacturing, vol. 26, no. 4, pages 381–391, 2010. (Cité en page 8.)

S. Altemeier, M. Helmdach, A. Koberstein et W. Dangelmaier. *Reconfiguration of assembly lines under the influence of high product variety in the automotive industry–a decision support system.* International Journal of Production Research, vol. 48, no. 21, pages 6235–6256, 2010. (Cité en page 53.)

E.J. Anderson et M.C. Ferris. *Genetic algorithms for combinatorial optimization : The assemble line balancing problem.* ORSA Journal on Computing, vol. 6, no. 2, pages 161–173, 1994. (Cité en page 37.)

A.I. Anosike et D.Z. Zhang. *Dynamic reconfiguration and simulation of manufacturing systems using agents.* Journal of Manufacturing Technology Management, vol. 17, no. 4, pages 435–447, 2006. (Cité en page 8.)

A.L. Arcus. *A computer method of sequencing operations for assembly lines.* International Journal of Production Research, vol. 4, no. 4, pages 259–277, 1965. (Cité en page 36.)

A. Azab et H.A. ElMaraghy. *Mathematical modeling for reconfigurable process planning.* CIRP Annals-Manufacturing Technology, vol. 56, no. 1, pages 467–472, 2007. (Cité en page 8.)

M. Bagher, M. Zandieh et H. Farsijani. *Balancing of stochastic U-type assembly lines : an imperialist competitive algorithm.* The International Journal of Advanced Manufacturing Technology, vol. 54, no. 1-4, pages 271–285, 2011. (Cité en page 17.)

J.F. Bard. *Assembly line balancing with parallel workstations and dead time.* International Journal of Production Research, vol. 27, no. 6, pages 1005–1018, 1989. (Cité en page 34.)

J. Barhak, D. Djurdjanovic, P. Spicer et R.Katz. *Integration of reconfigurable inspection with stream of variations methodology.* International Journal of Machine Tools and Manufacture, vol. 45, no. 4, pages 407–419, 2005. (Cité en page 8.)

O. Battaïa, A. Dolgui, N. Guschinsky et G. Levin. *Optimal design of machines processing pipeline parts.* The International Journal of Advanced Manufacturing Technology, vol. 63, no. 9-12, pages 963–973, 2012. (Cité en pages 58 et 73.)

O. Battaïa, E. Gurevsky, F. Makssoud et A. Dolgui. *Equipment Location in Machining Transfer Lines with Multi-spindle Heads.* Journal of Mathematical Modelling and Algorithms in Operations Research, vol. 12, no. 2, pages 117–133, 2013. (Cité en pages 57 et 82.)

O. Battaïa et A. Dolgui. *Reduction approaches for a generalized line balancing problem.* Computers & Operations Research, vol. 39, no. 10, pages 2337–2345, 2012. (Cité en page 57.)

O. Battaïa et A. Dolgui. *A taxonomy of line balancing problems and their solution approaches.* International Journal of Production Economics, vol. 142, pages 259–277, 2013. (Cité en pages 13 et 107.)

O. Battaïa, A. Dolgui, N. Guschinsky et G. Levin. *A decision support system for design of mass production machining lines composed of stations with rotary or mobile table.* Robotics and Computer Integrated Manufacturing, vol. 28, pages 672–680, 2012. (Cité en page 13.)

O. Battaïa, A. Dolgui, N. Guschinsky et G. Levin. *Optimal design of machines processing pipeline parts.* International Journal of Advanced Manufacturing Technology, vol. 63, pages 963–973, 2012. (Cité en page 22.)

O. Battaïa, A. Dolgui, N. Guschinsky et G. Levin. *Combinatorial techniques to optimally customize an automated production line with rotary transfer and turrets.* IIE Transactions, page In Press, 2014. (Cité en page 24.)

O. Battaïa, A. Dolgui, N. Guschinsky et G. Levin. *Integrated configurable equipment selection and line balancing for mass production with serial-parallel machining systems.* Engineering Optimisation, page In Press, 2014. (Cité en pages 14 et 21.)

J. Bautista et J. Pereira. *Ant algorithms for a time and space constrained assembly line balancing problem.* European Journal of Operational Research, vol. 177, no. 3, pages 2016–2032, 2007. (Cité en page 37.)

J. Bautista et J. Pereira. *A dynamic programming based heuristic for the assembly line balancing problem.* European Journal of Operational Research, vol. 194, no. 3, pages 787–794, 2009. (Cité en page 107.)

I. Baybars. *A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem.* Management Science, vol. 32, no. 8, pages 909–932, 1986. (Cité en pages 13 et 14.)

A. Baykasoglu et T. Dereli. *Two-sided assembly line balancing using an ant-colony-based heuristic.* The International Journal of Advanced Manufacturing Technology, vol. 36, no. 5-6, pages 582–588, 2008. (Cité en page 37.)

A. Baykasoğlu et L. Özbakır. *Stochastic U-line balancing using genetic algorithms.* The International Journal of Advanced Manufacturing Technology, vol. 32, no. 1-2, pages 139–147, 2007. (Cité en page 37.)

A. Baykasoglu. *Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems.* Journal of Intelligent Manufacturing, vol. 17, no. 2, pages 217–232, 2006. (Cité en pages 17 et 38.)

S. Belmokhtar, A. Dolgui, N. Guschinsky et G. Levin. *Integer programming models for logical layout design of modular machining lines.* Computers & Industrial Engineering, vol. 51, no. 3, pages 502–518, 2006. (Cité en pages 18 et 29.)

Z. Bi, Y.T. Lang et L. Wang. *Design of reconfigurable manufacturing systems with strongly coupled nature.* Chinese Journal of Mechanical Engineering, vol. 1, page 020, 2007. (Cité en page 8.)

Z.M. Bi, SYT. Lang, W. Shen et L. Wang. *Reconfigurable manufacturing systems : the state of the art.* International Journal of Production Research, vol. 46, no. 4, pages 967–992, 2008. (Cité en page 8.)

M. Bonte, L.Fourment, T.Do, AH.van den Boogaard et J. Huétink. *Optimisation of metal forming processes using Finite Element simulations : A sequential approximate optimisation algorithm and its comparison to other algorithms by application to forging.* Structural and Multidisciplinary Optimization, 2006. (Cité en page 26.)

J.P. Brans et B. Mareschal. *The PROMCALC & GAIA decision support system for multicriteria decision aid.* Decision support systems, vol. 12, no. 4, pages 297–310, 1994. (Cité en page 44.)

M. Bruccoleri, M. Amico et G. Perrone. *Distributed intelligent control of exceptions in reconfigurable manufacturing systems.* International Journal of Production Research, vol. 41, no. 7, pages 1393–1412, 2003. (Cité en page 8.)

M. Bruccoleri, Z.J. Pasek et Y. Koren. *Operation management in reconfigurable manufacturing systems : Reconfiguration for error handling.* International Journal of Production Economics, vol. 100, no. 1, pages 87–100, 2006. (Cité en page 8.)

O. Brudaru, D. Popovici et C. Copăceanu. *Cellular genetic algorithm with communicating grids for assembly line balancing problems.* Advances in Electrical and Computer Engineering, vol. 10, no. 2, pages 87–93, 2010. (Cité en page 17.)

B. Cakir, F. Altiparmak et B. Dengiz. *Multi-objective optimization of a stochastic assembly line balancing : A hybrid simulated annealing algorithm.* Computers & Industrial Engineering, vol. 60, no. 3, pages 376–384, 2011. (Cité en page 17.)

L. Capacho, R. Pastor, A. Dolgui et O. Guschinskaya. *An evaluation of constructive heuristic methods for solving the alternative subgraphs assembly line balancing problem.* Journal of Heuristics, vol. 15, no. 2, pages 109–132, 2009. (Cité en page 36.)

A. Carlsson, S. Hjelm, B. Holmgren, L. Malmsköld, M. Granström, J. Svenningstorp et A. Wik. *Sustainable Manufacturing Systems Capable of Producing Innovative Environmentally Friendly and Safe Products.* R&D program proposal to secure competetive vehicle and powertrain production in Sweden, 2010. (Cité en page 47.)

G. Celano, S. Fichera, V. Grasso, U. La Commare et G. Perrone. *An evolutionary approach to multi-objective scheduling of mixed model assembly lines.* Computers & industrial engineering, vol. 37, no. 1, pages 69–73, 1999. (Cité en page 25.)

C. Chandra et J. Grabis. *Reconfiguration of multi-stage production systems to support product customization using generic simulation models.* In Proceedings

of the 6th International Conference of Industrial Engineering Theory, Applications and Practice, San-Francisco, California, November 18-20. Dept. of Computer Engineering, Chalmers University of Technology, 2001. (Cité en page 8.)

C. Chandra et J. Grabis. *Reconfigurable manufacturing systems : meeting the challenges of a dynamic business paradigm.* International Journal of Manufacturing Technology and Management, vol. 17, no. 1, pages 1–4, 2009. (Cité en page 8.)

A. Charnes et W. Cooper. Management models and industrial applications of linear programming. John Wiley, New York, 1961. (Cité en page 41.)

W.C. Chiang. *The application of a tabu search metaheuristic to the assembly line balancing problem.* Annals of Operations Research, vol. 77, pages 209–227, 1998. (Cité en page 38.)

M. Chica, O. Cordón, S. Damas et J. Bautista. *A multiobjective GRASP for the 1/3 variant of the time and space assembly line balancing problem.* Trends in Applied Intelligent Systems, vol. 6098, pages 656–665, 2010. (Cité en page 17.)

M. Chica, O. Cordón, S. Damas, J. Bautista et J. Pereira. *Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem : ACO and random greedy search.* Information Sciences, vol. 180, pages 3465–3487, 2010. (Cité en page 17.)

M. Chica, O. Cordón, S. Damas et J. Bautista. *Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different Nissan scenarios.* Expert Systems with Applications, vol. 38, no. 1, pages 709–720, 2011. (Cité en page 17.)

M. Chica, O. Cordón, S. Damas et J. Bautista. *A new diversity induction mechanism for a multi-objective ant colony algorithm to solve a real-world time and space assembly line balancing problem.* Memetic Computing, vol. 3, pages 15–24, 2011. (Cité en page 17.)

G. Choi. *A goal programming mixed-model line balancing for processing time and physical workload.* Computers & Industrial Engineering, vol. 57, no. 1, pages 395–400, 2009. (Cité en page 42.)

G. Chryssolouris. Manufacturing systems : theory and practice. Springer, 2006. (Cité en page 13.)

P. Chutima et P. Chimklai. *Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge.* Computers & Industrial Engineering, vol. 62, no. 1, pages 39–55, 2012. (Cité en pages 16 et 17.)

P. Chutima et P. Olanviwatchai. *Mixed-Model U-Shaped Assembly Line Balancing Problems with Coincidence Memetic Algorithm.* Journal of Software Engineering & Applications, vol. 3, pages 347–363, 2010. (Cité en page 17.)

R.F. Coelho et P. Breitkopf. Optimisation multidisciplinaire en mécanique : Démarche de conception, stratégies collaboratives et concourantes, multiniveaux de modèles et de paramètres. Hermès science publications, 2009. (Cité en page 44.)

C.A. Coello. *An updated survey og GA based multiobjective optimization technics, Rapport technique Lania-RD-98-08, Xalapa Veracruz, Mexico*, décembre 1998. (Cité en page 41.)

Y. Collette et P. Siarry. Optimisation multiobjectif. Ed. Techniques Ingénieur, 2002. (Cité en page 40.)

A. Corominas, R. Pastor et J. Plans. *Balancing assembly line with skilled and unskilled workers.* Omega, vol. 36, no. 6, pages 1126–1132, 2008. (Cité en page 54.)

W. Covanich et D. McFarlane. *Assessing ease of reconfiguration of conventional and Holonic manufacturing systems : Approach and case study.* Engineering Applications of Artificial Intelligence, vol. 22, no. 7, pages 1015–1024, 2009. (Cité en page 8.)

A. Dashchenko. Manufacturing technologies for machines of the future : 21st century technologies ; with 41 tables, volume 1. Springer, 2003. (Cité en page 9.)

R.F. Deckro et S. Rangachari. *A goal approach to assembly line balancing.* Computers & Operations Research, vol. 17, no. 5, pages 509–521, 1990. (Cité en page 41.)

R.F. Deckro. *Balancing Cycle Time and Workstations.* IIE Transactions, vol. 21, no. 2, pages 106–111, 1989. (Cité en page 5.)

A.M. Deif et W. ElMaraghy. *Investigating optimal capacity scalability scheduling in reconfigurable manufacturing system.* International Journal of Advance Manufacturing Technology, vol. 52, no. 1, pages 174–193, 2005. (Cité en page 8.)

A.M. Deif et W. ElMaraghy. *Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system.* The International Journal of Advanced Manufacturing Technology, vol. 32, no. 5-6, pages 557–562, 2007. (Cité en page 47.)

X. Delorme, A. Dolgui et M.Y. Kovalyov. *Combinatorial design of a minimum cost transfer line.* Omega, vol. 40, no. 1, pages 31–41, 2012. (Cité en page 18.)

L.P. Ding, Y.X. Feng, J.R. Tan et Y.C. Gao. *A new multi-objective ant colony algorithm for solving the disassembly line balancing problem.* The International Journal of Advanced Manufacturing Technology, vol. 48, no. 5–8, pages 761–771, 2011. (Cité en page 17.)

A. Dolgui et I. Ihnatsenka. *Balancing modular transfer lines with serial-parallel activation of spindle heads at stations.* Discrete Applied Mathematics, vol. 157, no. 1, pages 68–89, 2009. (Cité en page 18.)

A. Dolgui et I. Ihnatsenka. *Branch and bound algorithm for a transfer line design problem : Stations with sequentially activated multi-spindle heads.* European Journal of Operational Research, vol. 197, no. 3, pages 1119–1132, 2009. (Cité en pages 18 et 57.)

A. Dolgui et A. Pashkevich. Manufacturing systems design. Université de Technologie de Troyes, 2001. (Cité en page 33.)

A. Dolgui et J.M. Proth. Les systèmes de production modernes. Hermès/Lavoisier, 2006. (Cité en pages 9 et 13.)

A. Dolgui et J.M. Proth. Supply chain engineering. Springer, 2010. (Cité en page 14.)

A. Dolgui, N. Guschinbsky et G. Levin. *On problem of optimal design of transfer lines with parallel and sequential operations.* In Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA'99. 1999 7th IEEE International Conference on, volume 1, pages 329–334. IEEE, 1999. (Cité en pages 14, 51 et 57.)

A. Dolgui, B. Finel, F. Vernadat, N. Guschinsky et G. Levin. *A heuristic approach for transfer lines balancing.* Journal of Intelligent Manufacturing, vol. 16, no. 2, pages 159–172, 2005. (Cité en page 36.)

A. Dolgui, B. Finel, F. Vernadat, N. Guschinsky et G. Levin. *A heuristic approach for transfer lines balancing.* Journal of Intelligent Manufacturing, vol. 16, no. 2, pages 159–172, 2005. (Cité en page 58.)

A. Dolgui, B. Finel, O. Guschinskaya, N. Guschinsky, G. Levin et F. Vernadat. *Balancing large-scale machining lines with multi-spindle heads using decomposition.* International Journal of Production Research, vol. 44, no. 18–19, pages 4105–4120, 2006. (Cité en page 34.)

A. Dolgui, B. Finel, N. Guschinsky, G. Levin et F. Vernadat. *MIP approach to balancing transfer lines with blocks of parallel operations.* IIE transactions, vol. 38, no. 10, pages 869–882, 2006. (Cité en pages 20, 57 et 73.)

A. Dolgui, N. Guschinbsky et G. Levin. *A special case of transfer lines balancing by graph approach.* European Journal of Operational Research, vol. 168, no. 3, pages 732–746, 2006. (Cité en pages 5 et 18.)

A. Dolgui, N. Guschinbsky, G. Levin et J.M. Proth. *Optimisation of multi-position machines and transfer lines.* European Journal of Operational Research, vol. 185, no. 3, pages 1375–1389, 2008. (Cité en pages 5, 24 et 34.)

A. Dolgui, N. Guschinbsky et G. Levin. *Graph approach for optimal design of transfer machine with rotary table.* International Journal of Production Research, vol. 47, no. 2, pages 321–341, 2009. (Cité en page 24.)

A. Dolgui, N. Guschinbsky et G. Levin. *Enhanced mixed integer programming model for a transfer line design problem.* Computers and Industrial Engineering, vol. 62, no. 2, pages 570–578, 2012. (Cité en page 18.)

M. Dorigo. *Optimization, learning and natural algorithms.* Ph. D. Thesis, Politecnico di Milano, Italy, 1992. (Cité en page 37.)

J. Dou, X. Dai et Z. Meng. *Optimisation for multi-part flow-line configuration of reconfigurable manufacturing system using GA.* International Journal of Production Research, vol. 48, no. 14, pages 4071–4100, 2010. (Cité en page 8.)

F. Easton, B. Faaland, T.D. Klastorn et T. Schmitt. *Improved network based algorithms for the assembly line balancing problem.* International Journal of Production Research, vol. 27, no. 11, pages 1901–1915, 1989. (Cité en page 34.)

F. Edgeworth. *The mathematical theory of political economy.* Nature, vol. 40, no. 5, pages 434–436, 1889. (Cité en page 38.)

W.H. Elmaraghy, O.A. Nada et H.A. Elmaraghy. *Quality prediction for reconfigurable manufacturing systems via human error modelling.* International Journal of Computer Integrated Manufacturing, vol. 21, no. 5, pages 584–598, 2008. (Cité en page 8.)

E.W. Endsley, E.E. Almeida et D.M. Tilbury. *Modular finite state machines : Development and application to reconfigurable manufacturing cell controller generation.* Control Engineering Practice, vol. 14, no. 10, pages 1127–1142, 2006. (Cité en page 8.)

E. Erel et H. Gokcen. *Shortest-route formulation of mixed-model assembly line balancing problem.* European Journal of Operational Research, vol. 116, no. 1, pages 194–204, 1999. (Cité en page 34.)

E. Erel, I. Sabuncuoglu et B.A. Aksu. *Balancing of U-type assembly systems using simulated annealing.* International Journal of Production Research, vol. 39, no. 13, pages 3003–3015, 2001. (Cité en page 38.)

E. Falkenauer. *Line balancing in the real world.* In Proceedings of the International Conference on Product Lifecycle Management PLM, volume 5, pages 360–370, 2005. (Cité en page 52.)

A.M. Farid. *Facilitating ease of system reconfiguration through measures of manufacturing modularity.* Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture, vol. 222, no. 10, pages 1275–1288, 2008. (Cité en page 47.)

P. Fattahi et A. Roshani. *A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem.* The International Journal of Advanced Manufacturing Technology, vol. 53, no. 1-4, pages 363–378, 2011. (Cité en page 17.)

B. Finel. *Structuration de lignes d'usinage : méthodes exactes et heuristiques.* PhD thesis, Metz, 2004. (Cité en page 20.)

L.J. Fogel, A.J. Owens et M.J. Walsh. *Intelligent decision making through a simulation of evolution.* Behavioral science, vol. 11, no. 4, pages 253–272, 1966. (Cité en page 37.)

N.H. Ben Fong et R.H. Sturges. *A control design strategy of a reconfigurable manufacturing system.* International Journal of Manufacturing Technology and Management, vol. 17, no. 1, pages 68–81, 2009. (Cité en page 8.)

R. Galan, J. Racero, I. Eguia et J.M. Garcia. *A systematic approach for product families formation in Reconfigurable Manufacturing Systems.* Robotics and Computer-Integrated Manufacturing, vol. 23, no. 5, pages 489–502, 2007. (Cité en page 8.)

R. Gamberini, A. Grassi, M. Gamberi, R. Manzini et A. Regattieri. *U-shaped assembly lines with stochastic tasks execution times : heuristic procedures for balancing and re-balancing problems.* Simulation Series, vol. 36, no. 2, page 137, 2004. (Cité en page 17.)

R. Gamberini, A. Grassi et B. Rimini. *A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem.* International Journal of Production Economics, vol. 102, no. 2, pages 226–243, 2006. (Cité en pages 44 et 52.)

R. Gamberini, E. Gebennini, A. Grassi et A. Regattieri. *A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem.* International Journal of Production Research, vol. 47, no. 8, pages 2141–2164, 2009. (Cité en pages 17, 40 et 53.)

M.R. Gary et D.S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-completeness*, 1979. (Cité en page 26.)

M. Gen, J. Gao et L. Lin. *Multistage-based genetic algorithm for flexible job-shop scheduling problem.* In Intelligent and Evolutionary Systems, pages 183–196. Springer, 2009. (Cité en page 37.)

F. Glover. *Future paths for integer programming and links to artificial intelligence.* Computers & Operations Research, vol. 13, no. 5, pages 533–549, 1986. (Cité en page 38.)

F. Glover. *Tabu search-part I.* ORSA Journal on computing, vol. 1, no. 3, pages 190–206, 1989. (Cité en page 38.)

F. Glover. *Tabu search-part II.* ORSA Journal on computing, vol. 2, no. 1, pages 4–32, 1990. (Cité en page 38.)

H. Gökçen et E. Erel. *A goal programming approach to mixed-model assembly line balancing problem.* International Journal of Production Economics, vol. 24, no. 2, pages 177–185, 1997. (Cité en page 42.)

H. Gökçen, K. Ağpak et R. Benzer. *Balancing of parallel assembly lines.* International Journal of Production Economics, vol. 103, no. 2, pages 600–609, 2006. (Cité en page 11.)

H. Gökçen et K. Agpak. *A goal programming approach to simple U-line balancing problem.* European Journal of Operational Research, vol. 171, no. 2, pages 577–585, 2006. (Cité en page 42.)

D.E. Goldberg et J.H. Holland. *Genetic algorithms and machine learning.* Machine learning, vol. 3, no. 2, pages 95–99, 1988. (Cité en page 37.)

N. Grangeon, P. Leclaire et S. Norre. *Heuristics for the re-balancing of a vehicle assembly line.* International Journal of Production Research, vol. 49, no. 22, pages 6609–6628, 2011. (Cité en page 54.)

S.C. Graves et C.H Redfield. *Equipment selection and task assignment for multiproduct assembly system design.* International Journal of Flexible Manufacturing Systems, vol. 1, no. 1, pages 31–50, 1988. (Cité en page 25.)

K. Gumasta, S. Kumar Gupta, L. Benyoucef et M.K. Tiwari. *Developing a reconfigurability index using multi-attribute utility theory.* International Journal of Production Research, vol. 49, no. 6, pages 1669–1683, 2011. (Cité en page 8.)

E. Gurevsky, O. Battaïa et A. Dolgui. *Stability measure for a generalized assembly line balancing problem.* Discrete Applied Mathematics, vol. 161, no. 3, pages 377–394, 2013. (Cité en page 58.)

O. Guschinskaya et A. Dolgui. *Heuristic methods for a transfer line balancing problem.* In Proceedings of the 19th International Conference on Production Research. Valparaiso, Chile, CD-ROM, 2007. (Cité en page 58.)

O. Guschinskaya et A. Dolgui. *A Transfer Line Balancing Problem by Heuristic Methods : Industrial Case Studies.* Decision Making in Manufacturing and Services, vol. 2, no. (1-2), pages 33–46, 2008. (Cité en page 58.)

O. Guschinskaya et A. Dolgui. *Comparison of exact and heuristic methods for a transfer line balancing problem.* International Journal of Production Economics, vol. 120, no. 2, pages 276–286, 2009. (Cité en page 87.)

O. Guschinskaya, A. Dolgui, N. Guschinsky et G. Levin. *Minimizing makespan for multi-spindle head machines with a mobile table.* Computers & Operations Research, vol. 36, no. 2, pages 344–357, 2009. (Cité en page 22.)

O. Guschinskaya. *Vers une optimisation de la configuration de systèmes d'usinage à botiers multibroches.* Journal Européen des Systèmes Automatisés, vol. 44, no. 7, pages 771–790, 2010. (Cité en page 22.)

A.L. Gutjahr et G.L. Nemhauser. *An algorithm for the line balancing problem.* Management Science, vol. 11, no. 2, pages 308–315, 1964. (Cité en page 34.)

N. Hamta, S.M.T. Fatemi Ghomi, F. Jolai et U. Bahalke. *Bi-criteria assembly line balancing by considering flexible operation times.* Applied Mathematical Modelling, vol. 35, no. 12, pages 5592–5608, 2011. (Cité en page 17.)

A.N. Haq, K. Rengarajan et J. Jayaprakash. *A hybrid genetic algorithm approach to mixed-model assembly line balancing.* The International Journal of Advanced Manufacturing Technology, vol. 28, no. 3-4, pages 337–341, 2006. (Cité en page 37.)

M. Held, R.M. Karp et R. Shareshian. *Assembly-line balancing-dynamic programming with precedence constraints.* Operations Research, vol. 11, no. 3, pages 442–459, 1963. (Cité en page 34.)

W.B. Helgeson et D.P. Birnie. *Assembly line balancing using the ranked positional weight technique.* Journal of Industrial Engineering, vol. 12, no. 6, pages 394–398, 1961. (Cité en page 36.)

H. Hirani, S. Ratchev, N. Lohse et G. Valtchanov. *Methodology for knowledge enriched requirements specification for assembly system reconfiguration.* Assembly Automation, vol. 26, no. 4, pages 307–314, 2006. (Cité en page 8.)

T.R. Hoffmann. *EUREKA : A hybrid system for assembly line balancing.* Management Science, vol. 38, no. 1, pages 39–47, 1992. (Cité en page 33.)

J.H. Holland. *Adaptation in natural and artificial systems.* Complex Adaptive Systems, 1975. (Cité en page 37.)

K.K.B. Hon et S. Xu. *Impact of product life cycle on manufacturing systems reconfiguration.* CIRP Annals-Manufacturing Technology, vol. 56, no. 1, pages 455–458, 2007. (Cité en pages 8 et 50.)

R. Hwang et H. Katayama. *A multi-decision genetic approach for workload balancing of mixed-model U-shaped assembly line systems.* International Journal of Production Research, vol. 47, no. 14, pages 3797–3822, 2009. (Cité en pages 17, 37 et 40.)

R.K. Hwang, H. Katayama et M. Gen. U-*shaped assembly line balancing problem with genetic algorithm.* International Journal of Production Research, vol. 46, no. 16, pages 4637–4649, 2008. (Cité en page 17.)

J.R. Jackson. *A computing procedure for a line balancing problem.* Management Science, vol. 2, no. 3, pages 261–271, 1956. (Cité en page 34.)

R.V. Johnson. *Optimally balancing large assembly lines with ?FABLE ?* Management Science, vol. 34, no. 2, pages 240–253, 1988. (Cité en page 33.)

D.F. Jones, S.K. Mirrazavi et M. Tamiz. *Multi-objective meta-heuristics : An overview of the current state-of-the-art.* European journal of operational research, vol. 137, no. 1, pages 1–9, 2002. (Cité en page 44.)

Y. Kara, U. Ozcan et A. Peker. *An approach for balancing and sequencing mixed-model JIT U-lines.* The International Journal of Advanced Manufacturing Technology, vol. 32, no. 11-12, pages 1218–1231, 2007. (Cité en page 38.)

Y. Kara, T. Paksoy et C.T. Chang. *Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing.* European Journal of Operational Research, vol. 195, no. 2, pages 335–347, 2009. (Cité en pages 17 et 42.)

Y. Kara, H. Gökçen et Y. Atasagun. *Balancing parallel assembly lines with precise and fuzzy goals.* International Journal of Production Research, vol. 48, no. 6, pages 1685–1703, 2010. (Cité en page 42.)

Y. Kara, C. Özgüven, N.Y. Seçme et C.T. Chang. *Multi-objective approaches to balance mixed-model assembly lines for model mixes having precedence conflicts and duplicable common tasks.* International Journal of Advanced Manufacturing Technology, vol. 52, no. 5-8, pages 725–737, 2011. (Cité en page 42.)

T. Kelepouris, C.Y. Wong, A. Farid, A.K.N. Parlikad et D.C.McFarlane. *Towards a reconfigurable supply network model.* 2006. (Cité en page 8.)

O. Kilincci. *Firing sequences backward algorithm for simple assembly line balancing problem of type 1.* Computers & Industrial Engineering, vol. 60, no. 4, pages 830–839, 2011. (Cité en page 107.)

Y.K. Kim et Y.J. Kim. *Genetic algorithms for assembly line balancing with various objectives.* Computers & Industrial Engineering, vol. 30, no. 3, pages 397–409, 1996. (Cité en page 37.)

Y.J. Kim, Y.K. Kim et Y. Cho. *A heuristic-based genetic algorithm for workload smoothing in assembly lines.* Computers & Operations Research, vol. 25, no. 2, pages 99–111, 1998. (Cité en page 37.)

Y.K. Kim, Y. Kim et Y.J. Kim. *Two-sided assembly line balancing : a genetic algorithm approach.* Production Planning & Control, vol. 11, no. 1, pages 44–53, 2000. (Cité en page 37.)

Y.K. Kim, W.S. Song et J.H. Kim. *A mathematical model and a genetic algorithm for two-sided assembly line balancing.* Computers & Operations Research, vol. 36, no. 3, pages 853–865, 2009. (Cité en page 37.)

J. Ko, S.J. Hu et T. Huang. *Reusability assessment for manufacturing systems.* CIRP Annals-Manufacturing Technology, vol. 54, no. 1, pages 113–116, 2005. (Cité en page 53.)

Y. Koren et A.G. Ulsoy. *Reconfigurable manufacturing system having a production capacity method for designing same and method for changing its production capacity*, 2002. US Patent 6,349,237. (Cité en page 6.)

Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy et H. Van Brussel. *Reconfigurable manufacturing systems.* CIRP Annals-Manufacturing Technology, vol. 48, no. 2, pages 527–540, 1999. (Cité en page 6.)

J.F. Kottas et H.S. Lau. *A Cost-Oriented Approach to Stochastic Line Balancing.* IIE Transactions, vol. 5, no. 2, pages 164–171, 1973. (Cité en page 53.)

J.R. Koza. Genetic programming : vol. 1, on the programming of computers by means of natural selection, volume 1. MIT press, 1992. (Cité en page 37.)

J.R. Koza. Genetic programming ii : automatic discovery of reusable programs. MIT press, 1994. (Cité en page 37.)

H. Ku. *A state-time model to measure the reconfigurability of manufacturing areas–key to performance.* Integrated Manufacturing Systems, vol. 12, no. 7, pages 493–499, 2001. (Cité en pages 8 et 48.)

O. Kulak, I.O. Yilmaz et H.O. Günther. *A GA-based solution approach for balancing printed circuit board assembly lines.* OR Spectrum, vol. 30, no. 3, pages 469–491, 2008. (Cité en page 37.)

R. Kumar, S. Kumar et M.K. Tiwari. *An expert enhanced coloured fuzzy Petri net approach to reconfigurable manufacturing systems involving information delays.* The International Journal of Advanced Manufacturing Technology, vol. 26, no. 7-8, pages 922–933, 2005. (Cité en page 8.)

P. Lacomme, C. Prins et M. Sevaux. *A genetic algorithm for a bi-objective capacited arc routing problem.* Computers and Operations Research, vol. 33, no. 12, pages 3473–3493, 2006. (Cité en page 43.)

A.H. Land et A.G. Doig. *An automatic method of solving discrete programming problems.* Econometrica : Journal of the Econometric Society, pages 497–520, 1960. (Cité en page 32.)

S.D. Lapierre, A. Ruiz et P. Soriano. *Balancing assembly lines with tabu search.* European Journal of Operational Research, vol. 168, no. 3, pages 826–837, 2006. (Cité en page 38.)

H. Lee et A. Banerjee. *Executable design and control framework using reconfigurable manufacturing holons and its evolution algorithm.* International Journal of Production Research, vol. 49, no. 5, pages 1405–1423, 2011. (Cité en page 8.)

H.F. Lee et K.E. Stecke. *An integrated design support method for flexible assembly systems.* Journal of Manufacturing Systems, vol. 15, no. 1, pages 13–32, 1996. (Cité en page 12.)

Y.Y. Leu, L.A. Matheson et L.P. Rees. *Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria*.* Decision Sciences, vol. 25, no. 4, pages 581–605, 1994. (Cité en page 37.)

G. Levitin, J. Rubinovitz et B. Shnits. *A genetic algorithm for robotic assembly line balancing.* European Journal of Operational Research, vol. 168, no. 3, pages 811–825, 2006. (Cité en page 37.)

J. Li, X. Dai, Z. Meng, J. Dou et X. Guan. *Rapid design and reconfiguration of Petri net models for reconfigurable manufacturing cells with improved net rewriting systems and activity diagrams.* Computers & Industrial Engineering, vol. 57, no. 4, pages 1431–1451, 2009. (Cité en page 8.)

F.L. Lian, J. Moyne et D. Tilbury. *Implementation of networked machine tools in reconfigurable manufacturing systems.* In 2000 Japan-USA Symposium on Flexible Automation, pages 349–361, 2000. (Cité en page 8.)

B.L. MacCarthy et J. Liu. *A new classification scheme for flexible manufacturing systems.* International Journal of Production Research, vol. 31, no. 2, pages 299–309, 1993. (Cité en page 6.)

L. Makdessian, F. Yalaoui et A. Dolgui. *Partie II. Une approche multicritère.* Journal of Decision Systems, vol. 17, no. 3, pages 337–368, 2008. (Cité en page 43.)

F. Makssoud, O. Battaïa et A. Dolgui. *An Application of Goal Programming Technique for Reconfiguration of Transfer Lines.* In Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics (ICINCO'2012), pages pp 539–542, Rome, Italie, 2012. (Cité en page 105.)

F. Makssoud, O. Battaïa et A. Dolgui. *A mathematical model for a reconfiguration problem of transfer machining lines.* In Information Control Problems in Manufacturing, volume 14, pages 438–443, 2012. (Cité en page 82.)

F. Makssoud, O. Battaïa et A. Dolgui. *A new formulation for the assembly line re-balancing problem.* In Proceedings of the 26th Conference of the European Chapter on Combinatorial Optimization (ECCO XXVI), page p. 128, Paris, France, 2013. (Cité en page 118.)

F. Makssoud, O. Battaïa et A. Dolgui. *An Exact Method for the Assembly Line Re-balancing Problem.* In Vittal Prabhu, Marco Taisch et Dimitris Kiritsis, editeurs, APMS (1), volume 414 of *IFIP Advances in Information and Communication Technology*, pages 159–166. Springer, 2013. (Cité en page 118.)

F. Makssoud, O. Battaïa et A. Dolgui. *An exact approach to improve the reusability of serial machining lines.* In Proceedings of International Conference on Industrial Engineering and Systems Management (IESM'2013), 2013. (Cité en page 82.)

F. Makssoud, O. Battaïa et A. Dolgui. *An Exact Method for the Assembly Line Re-balancing Problem.* In Advances in Production Management Systems. Sustainable Production and Service Supply Chains, pages 159–166. Springer, 2013. (Cité en page 118.)

F. Makssoud, O. Battaïa et A. Dolgui. *Reconfiguration des lignes d'usinage de transfert.* Quatorzième congrès annuel de la Société Française de recherche Opérationnelle et d'Aide à la Décision (ROADEF 2013), 2013. (Cité en page 82.)

F. Makssoud, O. Battaïa et A. Dolgui. *Reconfiguration of Machining Transfer Lines.* In Service Orientation in Holonic and Multi Agent Manufacturing and Robotics, pages 339–353. Springer, 2013. (Cité en page 82.)

F. Makssoud, O. Battaïa et A. Dolgui. *An exact optimization approach for a Transfer Line Reconfiguration Problem.* International Journal of Advanced Manufacturing Technology, page In Press, 2014. (Cité en page 82.)

C. Manuel, C. Oscar et D. Sergio. *An advanced multiobjective genetic algorithm design for the time and space assembly line balancing problem.* Computers & Industrial Engineering, vol. 61, no. 1, pages 103–117, 2011. (Cité en page 17.)

S.M. McGovern et S.M. Gupta. *Ant colony optimization for disassembly sequencing with multiple objectives.* The International Journal of Advanced Manufacturing Technology, vol. 30, no. 5, pages 481–496, 2006. (Cité en page 17.)

S.M. Mcgovern et S.M. Gupta. *Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem.* International Journal of Production Research, vol. 45, no. 18, pages 4485–4511, 2007. (Cité en page 17.)

S.M. McGovern et S.M. Gupta. The disassembly line : Balancing and modeling. McGraw Hill, New York, 2011. (Cité en page 17.)

P.R. McMullen et G.V. Frazier. *Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations.* International Journal of Production Research, vol. 36, no. 10, pages 2717–2741, 1998. (Cité en pages 17 et 38.)

P.R. McMullen et P. Tarasewich. *Using ant techniques to solve the assembly line balancing problem.* IIE transactions, vol. 35, no. 7, pages 605–617, 2003. (Cité en pages 37 et 40.)

P.R. McMullen et P. Tarasewich. *Multi-objective assembly line balancing via a modified ant colony optimization technique.* International Journal of Production Research, vol. 44, no. 1, pages 27–42, 2006. (Cité en pages 11, 17, 37 et 40.)

M.G. Mehrabi, A.G. Ulsoy et Y. Koren. *Reconfigurable manufacturing systems and their enabling technologies.* International Journal of Manufacturing Technology and Management, vol. 1, no. 1, pages 114–131, 2000. (Cité en page 8.)

A.R. Mendes, A.L. Ramos, A.S. Simaria et P.M. Vilarinho. *Combining heuristic procedures and simulation models for balancing a PC camera assembly line.* Computers & Industrial Engineering, vol. 49, no. 3, pages 413–431, 2005. (Cité en page 38.)

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller et E. Teller. *Equation of state calculations by fast computing machines.* The journal of chemical physics, vol. 21, page 1087, 1953. (Cité en page 37.)

K. Miettinen. Nonlinear multiobjective optimization, volume 12. Springer, 1999. (Cité en pages 40 et 41.)

E. Mohamed. *Conception et optimisation d'allocation de ressources dans les lignes d'usinage reconfigurables.* PhD thesis, Saint-Etienne, 2010. (Cité en pages 14 et 19.)

A.C. Nearchou. *Balancing large assembly lines by a new heuristic based on differential evolution method.* The International Journal of Advanced Manufacturing Technology, vol. 34, no. 9-10, pages 1016–1029, 2007. (Cité en page 37.)

A.C. Nearchou. *Multi-objective balancing of assembly lines by population heuristics.* International Journal of Production Research, vol. 46, no. 8, pages 2275–2297, 2008. (Cité en pages 17 et 37.)

A.C. Nearchou. *Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization.* International Journal of Production Economics, vol. 129, no. 2, pages 242–250, 2011. (Cité en page 17.)

M. Nourelfath, D. Ait-kadi et W.I. Soro. *Availability modeling and optimization of reconfigurable manufacturing systems.* Journal of Quality in Maintenance Engineering, vol. 9, no. 3, pages 284–302, 2003. (Cité en page 8.)

A. Nourmohammadi et M. Zandieh. *Assembly line balancing by a new multi-objective differential evolution algorithm based on TOPSIS.* International Journal of Production Research, vol. 49, no. 10, pages 2833–2855, 2011. (Cité en page 17.)

A. Otto et A. Scholl. *Incorporating ergonomic risks into assembly line balancing.* European Journal of Operational Research, vol. 212, pages 277–286, 2011. (Cité en page 17.)

L. Ozbakir, A. Baykasoglu, B. Gorkemli et L. Gorkemli. *Multiple-colony ant algorithm for parallel assembly line balancing problem.* Applied Soft Computing, vol. 11, no. 3, pages 3186–3198, 2011. (Cité en page 17.)

U. Özcan et B. Toklu. *Balancing of mixed-model two-sided assembly lines.* Computers & Industrial Engineering, vol. 57, no. 1, pages 217–227, 2009. (Cité en page 17.)

U. Özcan et B. Toklu. *Multiple-criteria decision-making in two-sided assembly line balancing : a goal programming and a fuzzy goal programming models.* Computers & Operations Research, vol. 36, no. 6, pages 1955–1965, 2009. (Cité en pages 17 et 42.)

U. Özcan et B. Toklu. *A tabu search algorithm for two-sided assembly line balancing.* The International Journal of Advanced Manufacturing Technology, vol. 43, no. 7-8, pages 822–829, 2009. (Cité en pages 17, 38 et 40.)

G. Pahl et W. Beitz. *Engineering design : a systematic approach. 1996*, 1961. (Cité en page 12.)

V. Pareto. Cours d'economie politique. Librairie Droz, 1964. (Cité en page 38.)

R. Pastor et L. Ferrer. *An improved mathematical program to solve the simple assembly line balancing problem.* International Journal of Production Research, vol. 47, no. 11, pages 2943–2959, 2009. (Cité en page 107.)

J.H. Patterson et J.J. Albracht. *Technical Note-Assembly-Line Balancing : Zero-One Programming with Fibonacci Search.* Operations Research, vol. 23, no. 1, pages 166–172, 1975. (Cité en page 28.)

N. Pekin et M. Azizoglu. *Bi criteria flexible assembly line design problem with equipment decisions.* International Journal of Production Research, vol. 46, no. 22, pages 6323–6343, 2008. (Cité en page 17.)

S.G. Ponnambalam, P. Aravindan et G.M. Naidu. *A multi-objective genetic algorithm for solving assembly line balancing problem.* The International Journal of Advanced Manufacturing Technology, vol. 16, no. 5, pages 341–352, 2000. (Cité en page 37.)

H.D. Purnomo, H.M. Wee et H. Rau. *Two-sided assembly lines balancing with assignment restrictions.* Mathematical and Computer Modelling, 2011. (Cité en page 17.)

I. Rechenberg. *Evolutionsstrategie.* Frommann-Holzboog, Stuttgart, Germany, 1973. (Cité en page 37.)

B. Rekiek et A. Delchambre. *Multiple Objective Grouping Genetic Algorithm.* Assembly Line Design : The Balancing of Mixed-Model Hybrid Assembly Lines with Genetic Algorithms, pages 39–56, 2006. (Cité en page 44.)

B. Rekiek, P. De Lit, F. Pellichero, E. Thomas, P. Fouda, E. Falkenauer et A. Delchambre. *A multiple objective grouping genetic algorithm for assembly line design.* Journal of Intelligent Manufacturing, vol. 12, no. 5-6, pages 467–485, 2001. (Cité en page 44.)

S.C. Ren, D. Xu, F. Wang et M. Tan. *Timed event graph-based cyclic reconfigurable flow shop modelling and optimization.* International journal of production research, vol. 45, no. 1, pages 143–156, 2007. (Cité en page 8.)

M. Rigamonti et T. Tolio. *Process analysis and flexible transfer line configuration.* In Digital Enterprise Technology, pages 561–568. Springer, 2007. (Cité en page 9.)

C. Rösiö. *Supporting the design of reconfigurable production systems.* PhD thesis, Jönköping University, JTH. Research area Industrial Production, 2012. (Cité en page 48.)

J. Rubinovitz et G. Levitin. *Genetic algorithm for assembly line balancing.* International Journal of Production Economics, vol. 41, no. 1, pages 343–354, 1995. (Cité en page 37.)

S.M. Saad, A. Baykasoglu et N.N.Z. Gindy. *An integrated framework for reconfiguration of cellular manufacturing systems using virtual cells.* Production Planning & Control, vol. 13, no. 4, pages 381–393, 2002. (Cité en page 8.)

I. Sabuncuoglu, E. Erel et M. Tanyer. *Assembly line balancing using genetic algorithms.* Journal of Intelligent Manufacturing, vol. 11, no. 3, pages 295–310, 2000. (Cité en page 37.)

I. Sabuncuoglu, E. Erel et A. Alp. *Ant colony optimization for the single model U-type assembly line balancing problem.* International Journal of Production Economics, vol. 120, no. 2, pages 287–300, 2009. (Cité en page 37.)

N. Safaei et R. Tavakkoli. *Integrated multi-period cell formation and subcontracting production planning in dynamic cellular manufacturing systems.* International Journal of Production Economics, vol. 120, no. 2, pages 301–314, 2009. (Cité en page 8.)

M. Sakarovitch. Graphes et programmation linéaire. Hermann, 1984. (Cité en page 27.)

M.S. Salerno. *Reconfigurable organisation to cope with unpredictable goals.* International Journal of Production Economics, vol. 122, no. 1, pages 419–428, 2009. (Cité en page 8.)

M.E. Salveson. *The assembly line balancing problem.* Journal of Industrial Engineering, vol. 6, no. 3, pages 18–25, 1955. (Cité en pages 13 et 107.)

A. Scholl, M. Fliedner et N. Boysen. *Absalom : Balancing assembly lines with assignment restrictions.* European Journal of Operational Research, vol. 200, no. 3, pages 688–701, 2010. (Cité en page 34.)

G. Schuh, J. Harre et S. Gottschalk. *Design for Changeability (DFC) in product-oriented production.* CIRP Journal of Manufacturing Systems, vol. 34, no. 5, pages 439–445, 2005. (Cité en page 47.)

K. Sethuraman K.E. Stecke C.W. Chelsea S.E. Chick T.L. Olsen. *A descriptive multi-attribute model for reconfigurable machining system selection examining buyer-supplier relationships.* International Journal of Agile Management Systems, vol. 2, no. 1, pages 33–48, 2000. (Cité en page 8.)

E.C. Sewell et S.H. Jacobson. *A Branch, Bound, and Remember Algorithm for the Simple Assembly Line Balancing Problem.* INFORMS Journal on Computing, vol. 24, no. 3, pages 433–442, 2012. (Cité en pages 34 et 107.)

A.S. Simaria et P.M. Vilarinho. *A genetic algorithm based approach to the mixed-model assembly line balancing problem of type II.* Computers & Industrial Engineering, vol. 47, no. 4, pages 391–407, 2004. (Cité en page 37.)

A.S. Simaria et P.M. Vilarinho. *2-ANTBAL : An ant colony optimisation algorithm for balancing two-sided assembly lines.* Computers & Industrial Engineering, vol. 56, no. 2, pages 489–506, 2009. (Cité en pages 17, 37 et 40.)

S.Y. Son, T.L. Olsen et D. Yip-Hoi. *An approach to scalability and line balancing for reconfigurable manufacturing systems.* Integrated Manufacturing Systems, vol. 12, no. 7, pages 500–511, 2001. (Cité en page 8.)

P. Spicer, Y. Koren, M. Shpitalni et D. Yip-Hoi. *Design principles for machining system configurations.* CIRP Annals-Manufacturing Technology, vol. 51, no. 1, pages 275–280, 2002. (Cité en page 6.)

F.B. Talbot, J.H. Patterson et W.V. Gehrlein. *A comparative evaluation of heuristic line balancing techniques.* Management Science, vol. 32, no. 4, pages 430–454, 1986. (Cité en page 36.)

S.O. Tasan et S. Tunali. *A review of the current applications of genetic algorithms in assembly line balancing.* Journal of Intelligent Manufacturing, vol. 19, no. 1, pages 49–69, 2008. (Cité en pages 17 et 37.)

B. Toklu et U. Özcan. *A fuzzy goal programming model for the simple* U*-line balancing problem with multiple objectives.* Engineering Optimization, vol. 40, no. 3, pages 191–204, 2008. (Cité en pages 17 et 42.)

T.L. Urban. *Note. Optimal balancing of U-shaped assembly lines.* Management Science, vol. 44, no. 5, pages 738–741, 1998. (Cité en page 11.)

P.M. Vilarinho et A.S. Simaria. *A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations.* International Journal of Production Research, vol. 40, no. 6, pages 1405–1420, 2002. (Cité en page 38.)

P.M. Vilarinho et A.S. Simaria. *ANTBAL : an ant colony optimization algorithm for balancing mixed-model assembly lines with parallel workstations.* International Journal of Production Research, vol. 44, no. 2, pages 291–303, 2006. (Cité en page 17.)

H.P. Wiendahl, H.A. ElMaraghy, P. Nyhuis, M.F. Zäh, H.H. Wiendahl, N. Duffie et M.Brieke. *Changeable manufacturing-classification, design and operation.* CIRP Annals-Manufacturing Technology, vol. 56, no. 2, pages 783–809, 2007. (Cité en page 8.)

Z. Xiaobo, W. Jiancai et L. Zhenbi. *A stochastic model of a reconfigurable manufacturing system Part 1 : A framework.* International Journal of Production Research, vol. 38, no. 10, pages 2273–2285, 2000. (Cité en page 8.)

Z. Xiaobo, W. Jiancai et L. Zhenbi. *A stochastic model of a reconfigurable manufacturing system Part 2 : Optimal configurations.* International Journal of Production Research, vol. 38, no. 12, pages 2829–2842, 2000. (Cité en page 8.)

Z. Xiaobo, W. Jiancai et L. Zhenbi. *A stochastic model of a reconfigurable manufacturing system Part 3 : Optimal selection policy.* International Journal of Production Research, vol. 39, no. 4, pages 747–758, 2001. (Cité en page 8.)

B. Xing, J. Eganza, G. Bright et J. Potgieter. *Reconfigurable manufacturing system for agile manufacturing.* In Information Control Problems in Manufacturing, volume 12, pages 509–516, 2006. (Cité en page 8.)

B. Yagmahan. *Mixed-model assembly line balancing using a multi-objective ant colony optimization approach.* Expert Systems with Applications, vol. 38, no. 10, pages 12453–12461, 2011. (Cité en page 17.)

C. Yang, J. Gao et L. Sun. *A Multi-objective Genetic Algorithm for Mixed-model Assembly Rebalancing.* Computers & Industrial Engineering, 2012. (Cité en page 54.)

A.S. Yigit et A. Allahverdi. *Optimal selection of module instances for modular products in reconfigurable manufacturing systems.* International Journal of Production Research, vol. 41, no. 17, pages 4063–4074, 2003. (Cité en page 8.)

A. Yoosefelahi, M. Aminnayeri, H. Mosadegh et H. Davari Ardakahi. *Type II robotic assembly line balancing problem : An evolution strategies algorithm for a multi-objective model.* Journal of Manufacturing Systems, vol. 31, no. 2, pages 139–151, 2012. (Cité en page 17.)

A. Youssef et H.A. ElMaraghy. *Assessment of manufacturing systems reconfiguration smoothness.* The International Journal of Advanced Manufacturing Technology, vol. 30, no. 1-2, pages 174–193, 2006. (Cité en pages 8 et 50.)

A. Youssef et H.A. ElMaraghy. *Performance analysis of manufacturing systems composed of modular machines using the universal generating function.* Journal of Manufacturing Systems, vol. 27, no. 2, pages 55–69, 2008. (Cité en pages 8 et 50.)

J. Yu et Y. Yin. *Assembly line balancing based on an adaptive genetic algorithm.* The International Journal of Advanced Manufacturing Technology, vol. 48, no. 1, pages 347–354, 2010. (Cité en page 17.)

J. Yu, Y. Yin, X. Sheng et Z. Chen. *Modelling strategies for reconfigurable assembly systems.* Assembly Automation, vol. 23, no. 3, pages 266–272, 2003. (Cité en page 8.)

P.T. Zacharia et A.C. Nearchou. *Multi-objective fuzzy assembly line balancing using genetic algorithms.* Journal of Intelligent Manufacturing, pages 1–13, 2012. (Cité en page 17.)

W. Zhang et M. Gen. *An efficient multiobjective genetic algorithm for mixed-model assembly line balancing problem considering demand ratio-based cycle time.* Journal of Intelligent Manufacturing, vol. 22, no. 3, pages 367–378, 2011. (Cité en page 17.)

Y.F. Zhang, G.Q. Huang, T. Qu et O. Ho. *Agent-based workflow management for RFID-enabled real-time reconfigurable manufacturing.* International Journal of Computer Integrated Manufacturing, vol. 23, no. 2, pages 101–112, 2010. (Cité en page 8.)

W. Zhong, V. Maier-Speredelozzi, A. Bratzel, S. Yang, S.E. Chick et S.J. Hu. *Performance analysis of machining systems with different configurations.* In Proceedings of the 2000 Japan-USA Flexible Automation Conference, pages 783–790, 2000. (Cité en page 8.)

Annexe

Paramètres d'entrée des instances :

N	Nombre opérations,
m_0	Nombre de stations maximal autorisé sur la ligne,
n_0	Nombre de blocs maximal sur chaque station de travail,
C_1	Coût d'une station,
C_2	Coût d'un bloc,
C_3	Bénéfice de réutilisation d'une ancienne tête de broches dans la ligne reconfigurée,
τ^b	Temps nécessaire pour activer un bloc,
τ^s	Temps nécessaire pour activer une station,
T_0	Temps de cycle,
T_j	Temps d'exécution d'une opération,
$Pred(i)$	Ensembles de précdécesseurs de l'opération i,
IS	Contraintes d'inclusion au niveau des stations ,
ES	Contraintes d'exclusion au niveau des stations,
EB	Contraintes d'exclusions au niveau des blocs.

Instance 1 :

```
30
15
4
10
2
2
0
0
100

[10,15.6,11.9,18,15.8,14.7,0,0,0,0,11.7,18.5,17.1,15.1,13,10.1,0,13.6,1
1.4,11.6,19.8,14.4,11.1,10,10,11.5,11.2,10.6,14.2,17]

[[],[0],[],[0,2],[],[4],[],[],[],[],[],[10],[],[11,12],[2],[14],[],[3,1
6],[],[13,18],[11,15],[20],[],[12,22],[23],[19],[25],[17],[28],[29]]
[[13,14],[0,2]]

[[0,3],[10,27],[5,13],[4,13],[11,21],[5,12],[3,4],[3,5],[1,20],[1,27],[
3,10],[8,15],[2,19],[1,15],[26,29]]

[[0,22],[5,21],[10,24],[23,24],[4,5],[17,18],[12,21],[19,24],[1,12],[22
,23],[4,11],[2,5],[5,11],[17,27]]

[[2,8,10,11,22],[0,1,4,5,6,7,9,18],[],[],[3,12,13,14,15,16,17,19,23],[2
0,21,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 2 :

```
30
15
4
10
2
1
0
0
100

[10,12.1,12.8,12.7,12.1,17.1,15.4,17.1,18.3,13.7,11.6,0,0,0,0,0,17.2,14
.8,12.5,16,16.8,13.9,10.3,14.3,19,10.2,16.5,14.5,17.5,12.2]

[[],[0],[],[0,2],[],[4],[],[6],[],[8],[],[],[],[],[],[],[3],[16],[9,10]
,[4,18],[],[20],[19],[22],[23],[5,17],[25],[21],[27],[28]]

[[18,19],[10,20]]
```

```
[[0,16],[1,23],[9,19],[1,7],[0,7],[1,5],[4,19],[7,18],[9,18],[4,7],[10,
23],[1,19],[2,3],[24,29]]

[[0,10],[10,18],[20,24],[7,19],[5,9],[16,23],[10,8],[5,21],[6,23],[7,8]
,[5,8],[9,10],[24,26]]

[[2,6,7,10,12,13,20],[],[],[],[0,1,3,4,8,9,11,16,21],[],[],[],[5,14,15,
17,18,19,22,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 3 :

```
30
15
4
10
2
2
0
0
100

[10,15.4,18.2,0,15.2,18.3,16.3,16.2,13.3,0,11.6,18.6,17.3,10.2,13.3,16,
15.4,10.5,0,0,10.3,18.6,0,16.4,13.4,16.1,10.6,9,15.2,17.6]

[[],[0],[],[],[0],[4],[],[6],[],[],[],[10],[],[6,12],[4,6],[14],[6,10],
[2,16],[],[],[],[5,11,20],[],[14],[8,23],[21],[25],[17],[24],[28]]

[[0,2],[7,17]]

[[0,16],[6,10],[9,16],[2,15],[1,10],[12,24],[1,4],[15,23],[0,4],[4,14],
[0,10],[14,23],[5,7],[26,27]]

[[0,13],[1,23],[1,7],[0,7],[1,5],[4,5],[7,11],[13,23],[4,7],[12,23],[1,
12],[9,24],[2,17],[28,29]]

[[0,1,2,12],[],[],[],[4,8,10,11,22],[],[],[],[3,6,7,9,13,14,15,18,20],[
],[],[],[5,16,17,19,21,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 4 :

```
30
15
4
10
2
1
0
```

```
0
100

[10,18.1,0,0,16.2,13.9,18.7,0,18.8,17.4,11,0,15.2,15.9,11.4,14.8,11.8,1
3.6,10.4,16.5,15.6,10,14.5,14.2,0,17.3,15.1,10.5,15.4,12.8]

[[],[0],[],[],[],[0,4],[0],[],[],[4,8],[],[],[],[10,12],[],[14],[8],[16
],[],[18],[13],[20],[1,5,15],[19,22],[],[9],[21],[19],[23],[28]]

[[0,4],[8,17],[21,26]]

[[0,17],[13,23],[10,23],[12,17],[0,6],[0,9],[5,8],[10,18],[5,15],[1,13]
,[0,22]]

[[10,13],[13,21],[10,13],[0,20],[13,14],[19,22],[4,10],[8,12],[4,15],[5
,6],[19,20],[14,18],[9,25]]

[[0,10,11,12,14,15,18,19],[],[],[],[1,2,3,4],[],[],[],[5,6,7,8,9,13,16,
17,20,21,22,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 5 :

```
30
15
4
10
2
1
0
0
100

[10,11.3,14.6,12.9,19.3,15.1,0,0,13.8,15.5,10.9,18.8,11.9,10.9,18.1,10,
10,19.2,10.9,13.7,19.1,14.7,0,0,0,10.2,13.1,15.7,10.7,15.1]

[[],[0],[],[2],[],[4],[],[],[1],[8],[],[5,10],[9],[12],[],[14],[],[11,1
6],[11],[18],[],[3,20],[],[],[],[15],[17,25],[19],[27],[28]]

[[0,3],[5,8]]

[[2,3],[3,8],[0,8],[13,24],[8,23],[5,20],[0,18],[1,17],[2,8],[11,14],[8
,27],[3,4],[1,21]]

[[0,17],[4,27],[3,28],[12,28],[12,17],[2,14],[0,15],[0,2],[2,10],[5,15]
,[0,28],[2,13],[0,27]]

[[2,4,14,15,20,22,23],[],[],[],[0,1,3,10,16,24],[],[],[],[5,6,7,8,9,11,
12,13,18,19,21],[17],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 6 :

```
30
15
4
10
2
2
0
0
100

[0,0,0,0,14.1,0,12,10.9,13.5,10.5,11.3,18.7,17.9,10.5,13.9,17.7,14.5,14
.2,16.5,16.2,11.2,16.9,12.4,19.3,10.1,12.2,10,11.2,9.8,14]

[[],[],[],[],[],[],[],[6],[4],[8],[],[10],[],[12],[],[11,14],[13],[16],
[17],[18],[7],[17,20],[14],[6,9,22],[23],[21],[19,25],[24],[27],[28]]

[[17,20],[4,9]]

[[6,16],[7,18],[8,9],[4,21],[7,13],[10,15],[10,23],[11,16],[9,17],[28,2
9]]

[[12,13],[11,14],[6,18],[4,7],[8,20],[19,26]]

[[0,2,6,7,10,12],[],[],[],[3,13,14,16,18,19,20,22],[1,4,8],[],[],[5,9,1
1,15,21,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[]]
```

Instance 7 :

```
30
15
4
10
2
2
0
0
100

[10,18.4,0,0,17.3,16.7,13,10,18.6,18.7,11.2,18.7,0,15.6,10.7,13,12.7,19
.8,0,0,14.7,11.7,17,11.4,14.6,14.2,12.2,9.7,16.3,16.6]

[[],[0],[],[],[],[4],[],[6],[6],[8],[6],[4,10],[],[12],[],[14],[],[16],
[],[],[],[20],[13],[22],[0,17,23],[1],[25],[21],[24,27],[28]]

[[13,15],[4,8]]
```

```
[[0,5],[5,19],[17,22],[5,11],[4,11],[9,12],[4,10],[11,16],[6,20],[4,14]
,[1,4],[10,23],[1,25]]

[[6,16],[1,7],[4,6],[5,18],[8,9],[4,21],[7,13],[10,15],[6,11],[9,17],[2
0,27]]

[[0,6,12,13,14,15,20,21,22],[],[],[],[1,4,5,8,16,17,23,24],[],[],[],[2,
3,7,9,10,18,19],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[]]
```

Instance 8 :

```
30
15
4
10
2
1.5
0
0
100

[10,11.7,11.9,18.8,10.4,17.9,13.9,19.1,13.6,16.8,0,0,11.4,10.6,17.5,0,0
,0,17.6,10.6,18.2,16.4,11.5,13.5,19.1,9,9.6,11.2,15.6,13.1]

[[],[0],[],[2],[],[4],[],[2,6],[],[8],[],[],[9],[4,12],[],[],[],[],[],[
18],[12],[1,14,18,20],[],[3,7,22],[23],[13],[25],[24],[21],[28]]

[[4,7,8]]

[[0,23],[2,14],[5,12],[0,22],[20,22],[6,8],[4,17],[9,14],[3,24],[2,5],[
9,23],[12,18],[3,5],[4,12],[13,25]]

[[0,5],[5,19],[17,22],[4,8],[9,12],[2,24],[2,8],[4,6],[2,20],[2,4],[1,2
],[7,23],[13,25]]

[[0,2,3,6,14,15],[],[],[],[4,5,7,8,9,16,18,22],[],[],[],[1,10,11,12,13,
17,19,20,21],[23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 9 :

```
30
15
4
10
2
2
0
```

```
0
100

[10,14.9,17.3,16.2,0,0,14.9,18.2,18.6,0,11.1,18.7,0,0,14.3,13.6,19.1,11
.1,18.1,17.8,11.6,11.1,16.1,15.7,13.5,10.6,13.7,11.3,16,12]

[[],[0],[],[2],[],[],[],[6],[],[],[],[1,3,10],[],[],[2],[14],[],[7,16],
[8,10],[18],[],[17,20],[],[22],[15,23],[11],[25],[24,26],[21],[28]]

[[10,14],[19,22]]

[[10,11],[0,24],[1,7],[6,17],[2,16],[16,19],[11,17],[3,16],[0,10],[1,14
],[11,24],[1,7],[7,22],[26,28]]

[[0,23],[2,10],[0,22],[20,22],[6,8],[11,14],[3,24],[2,22],[8,23],[10,18
],[3,14],[11,25]]

[[0,1,2,4,6,7,14,16,20],[8,10],[],[],[3,9,12,13,17,18,19,22],[],[],[],[
5,11,15,21,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 10 :

```
30
15
4
10
2
2
0
0
100

[10,18.2,12.8,13.5,16.7,10.3,15.8,17.3,13.7,13,11.1,18.7,14.9,10.7,11,1
8.9,17.3,16.7,18.7,14.9,0,0,0,0,0,14.1,13.7,11.2,14.8,10.7]

[[],[0],[],[2],[],[4],[5],[6],[],[8],[7],[10],[3],[0,11,12],[],[12,14],
[],[6,16],[8],[18],[],[],[],[],[],[17],[25],[19],[27],[26,28]]

[[2,5],[0,4]]

[[0,18],[17,19],[13,17],[3,4],[10,17],[3,13],[2,16],[12,14],[2,15],[10,
15],[0,14],[1,16],[7,17],[10,14],[26,28]]

[[9,11],[0,29],[4,9],[5,7],[4,17],[2,9],[16,19],[11,17],[3,16],[9,10],[
5,14],[13,29],[1,12],[7,19]]

[[0,1,2,4,5,6],[7,8,10],[],[],[3,9,14,18,20,21,22],[16,17,23,24],[],[],
[11,12,13,15,19],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
```

```
[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[]]
```

Instance 11 :

```
30
15
4
10
2
2
0
0
100

[10,11.5,18.2,0,19.9,11.5,16.8,16.4,18.7,0,0,0,11.6,15.8,17.8,14.2,15.4
,12.3,0,12.1,18.6,10.5,15.3,10,12.4,11.2,17.1,10,15.7,9.8]

[[],[0],[1],[],[],[4],[],[2,6],[],[],[],[],[4],[12],[],[14],[8],[5,13,1
6],[],[15,18],[4],[16,20],[],[22],[4,16,23],[7],[25],[21],[17],[28]]

[[0,2],[13,15]]

[[0,15],[1,5],[4,12],[2,7],[2,20],[2,13],[16,20],[8,22],[14,22],[0,22],
[20,21],[8,16],[2,11],[26,27]]

[[0,14],[17,19],[13,17],[2,4],[15,17],[7,13],[2,16],[12,14],[2,15],[14,
15],[0,14],[1,16],[7,23],[14,28]]

[[0,1,2,4,8,22],[],[],[],[5,6,10,12,13,16,18,23,24],[14,15,19,20],[],[]
,[3,7,9,11,17,21],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 12 :

```
30
15
4
10
2
2
0
0
100

[10,14.8,13.7,18.2,13,0,17.7,15.5,0,0,11,18.7,18.4,10.8,14.6,19.5,13.6,
17.9,0,19.3,0,15.3,19.9,12.1,16.9,10.5,17.2,13.2,14.2,14.4]
```

```
[[],[0],[],[2],[],[],[3],[6],[],[],[1],[10],[],[12],[],[14],[],[4,16],[
],[7,18],[],[17,19,20],[],[22],[23],[11],[21,25],[26],[15,24],[28]]

[[4,6,7],[22,23,24]]

[[0,2],[11,13],[3,21],[10,13],[0,20],[13,14],[19,22],[4,10],[12,15],[4,
15],[2,10],[11,22],[3,16],[3,12],[24,28]]

[[0,15],[1,3],[4,12],[2,3],[10,15],[2,21],[2,13],[13,22],[17,22],[0,10]
,[19,21],[6,11],[26,27]]

[[2,12,16],[3,4,6,7,13,20,22],[],[],[5,8,18,23],[0,1,9,14,17,19,21],[10
,11,15,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[]]
```

Instance 13 :

```
30
15
4
10
2
2
0
0
100

[10,18.1,19.1,15.5,16.2,13.9,18.7,14.6,18.8,17.4,11,18.8,15.2,15.9,0,0,
0,0,0,16.5,15.6,10,14.5,14.2,11.3,9,15.3,10.6,11.1,18.2]

[[],[0],[],[2],[],[0,4],[0],[6],[],[4,8],[],[10],[],[10,12],[],[],[],[]
,[],[],[13],[20],[1,5],[19,22],[23],[21],[25],[19],[24,27],[28]]

[[2,4],[7,8]]

[[0,9],[4,24],[3,23],[12,23],[12,19],[2,6],[0,6],[0,2],[3,8],[2,10],[5,
7],[2,13],[0,22],[10,22]]

[[11,13],[3,21],[10,13],[0,20],[13,19],[19,22],[4,10],[8,12],[4,11],[2,
5],[11,20],[3,6],[23,27]]

[[0,10,11,12,14,15,18,19],[],[],[],[1,2,3,4],[],[],[],[5,6,7,8,9,13,16,
17,20,21,22,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 14 :

```
30
15
```

```
4
10
2
2
0
0
100

[10,0,14.6,12.9,19.3,15.1,19.6,13.7,0,0,10.9,18.8,0,0,18.1,10,10,19.2,1
0.9,13.7,19.1,14.7,19,16.4,15.8,12.2,13.3,11.1,14.4,18]

[[],[],[],[2],[],[4],[],[6],[],[],[],[5,10],[],[],[],[14],[],[11,16],[1
1],[18],[],[3,20],[],[22],[23],[0],[25],[21,26],[27],[28]]

[[0,3],[5,6]]

[[2,3],[3,5],[0,26],[19,24],[6,23],[5,20],[0,18],[3,17],[2,11],[11,14],
[8,22],[3,4],[11,21],[19,29]]

[[0,17],[4,24],[3,23],[11,23],[17,18],[2,6],[0,6],[0,2],[2,10],[5,15],[
0,23],[2,29],[0,22]]

[[2,4,14,15,20,22,23],[],[],[],[0,1,3,10,16,24],[],[],[],[5,6,7,8,9,11,
12,13,18,19,21],[17],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 15 :

```
30
15
4
10
2
1
0
0
100

[10,17.9,15.5,17.6,15.6,17.5,0,0,13.9,11.8,10.8,18.8,15.4,11,11.7,0,16.
4,10.4,0,0,16,14.2,18.2,10.7,14.7,17.9,13.9,10.8,11,11.7]

[[],[0],[],[2],[],[4],[],[],[],[8],[4],[10],[5,8],[12],[],[],[],[3,12,1
4,16],[],[],[17],[13,20],[3,16],[4,22],[23],[9],[11],[21],[24,27],[28]]

[[17,21],[12,14]]

[[16,20],[14,20],[11,12],[1,9],[8,13],[10,23],[2,14],[8,21],[11,25],[5,
22],[26,28]]
```

```
[[2,11],[2,14],[9,10],[21,24],[17,21],[9,14],[17,24],[2,21],[5,17],[3,1
1],[0,8],[4,22],[16,27]]

[[0,1,4,5,6,10,11,14,15,16],[8,12],[],[],[2,3,9,13,18,22,23,24],[],[],[
],[7,17,19,20,21],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]]
```

Instance 16 :

```
30
15
4
10
2
2
0
0
100

[10,11.2,0,0,18.8,18.7,12.4,11,0,0,10.8,18.8,0,16.1,18.4,15.9,14.5,16.1
,12.6,15.2,19.5,18.9,12.8,12.8,19.2,12.7,16.3,15.2,10,11]

[[],[0],[],[],[],[4],[],[6],[],[],[],[10],[],[],[],[14],[4],[13,16],[0]
,[11,18],[],[15,20],[7],[20,22],[23],[19],[25],[17,26],[27],[28]]

[[13,17],[23,24]]

[[0,20],[1,6],[14,18],[1,4],[7,17],[5,14],[7,14],[6,16],[1,14],[13,24],
[11,23],[1,5],[21,23],[24,27]]

[[16,20],[6,20],[11,18],[1,13],[7,13],[10,23],[4,6],[14,21],[6,21],[11,
19],[18,19],[5,18],[15,21],[17,27]]

[[2,3,4,10,12,13,14,15,20,21],[6,7,8,11],[],[],[0,9,16,17,18,22,23,24],
[1,5,19],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 17 :

```
30
15
4
10
2
2
0
0
100
```

```
[0,14.5,16.4,12.3,11.9,19.9,0,0,14,18,10.7,18.8,18.9,11.1,15.2,11.2,0,0
,13.2,12.4,13,13.6,17.4,15,13.6,13.4,14,11.2,15.2,11.1]

[[],[],[],[2],[],[4],[],[],[],[3,8],[],[10],[2],[12],[],[8,14],[],[],[9
],[18],[],[20],[19],[10,13,22],[1,5,23],[15],[25],[11],[26,27],[28]]

[[1,2],[4,8]]

[[8,11],[1,8],[1,10],[13,19],[5,18],[2,4],[1,9],[1,5],[8,14],[2,13],[5,
10],[2,5],[4,15],[1,3],[22,24],[26,27]]

[[1,20],[8,9],[1,4],[14,18],[1,4],[5,14],[9,14],[4,13],[1,14],[13,24],[
3,23],[1,3],[21,23],[18,22],[21,27]]

[[0,4,6,14,16],[],[],[],[1,2,7,8,10,12],[],[],[],[3,5,9,11,13,15,17,20,
21],[],[],[],[18,19,22,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 18 :

```
30
15
4
10
2
1
0
0
100

[10,17.7,11.8,19.6,15.1,11.1,14.3,19.3,19,16.1,10.7,18.8,15.7,16.2,12,0
,0,0,13.7,19.6,0,0,11.9,17.1,18.1,11.4,15.3,13.2,17,16.3]

[[],[0],[],[2],[],[4],[],[6],[],[8],[0,2],[10],[],[12],[3],[],[],[],[],
[7,18],[],[],[],[22],[19,23],[5],[25],[14,26],[24],[28]]

[[6,10]]

[[12,25],[11,18],[18,26],[1,3],[4,27],[3,8],[0,28],[7,15],[1,5],[0,23],
[0,5],[2,7],[8,13],[19,24]]

[[8,11],[0,8],[0,10],[13,19],[5,18],[2,4],[1,9],[6,8],[0,13],[5,7],[2,5
],[0,3]]

[[0,1,2,12,16,22],[4,6,10,13,18],[],[],[7,8,17,19,20,21,23],[],[],[],[3
,5,9,11,14,15,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 19 :

```
30
15
4
10
2
2
0
0
100

[10,11,17.3,16.9,18.2,12.3,15.3,18.4,14.1,14.3,0,18.8,12.4,11.2,18.8,11
.8,0,0,0,0,10,13.1,16.5,19.2,12.5,9,11.1,15.8,12.1,16.3]

[[],[0],[],[2],[],[4],[],[6],[3],[8],[],[],[9],[7,12],[],[1,14],[],[],[
],[],[6],[20],[11],[22],[23],[15],[13],[21],[27],[26,28]]

[[20,21],[2,3]]

[[0,20],[6,24],[22,24],[1,20],[4,11],[1,20],[15,22],[8,21],[5,11],[5,22
],[13,14],[9,22],[26,27]]

[[12,20],[6,11],[20,22],[1,3],[4,20],[3,8],[7,15],[1,5],[0,23],[0,5],[2
,7],[8,13],[9,24],[9,15]]

[[2,3,6,10,11,14,16,20,21,22,23],[],[],[],[0,1,4,8,9,15,24],[],[],[],[5
,7,12,13,17,18,19],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 20 :

```
30
15
4
10
2
0
0
100

[10,14.3,12.7,14.3,11.3,13.5,16.2,17.5,19.1,12.4,10.6,18.9,19.2,16.3,15
.5,17.1,17.3,18.5,14.8,13.9,13.4,17.8,11.1,11.4,17]

[[],[0],[],[2],[],[1,4],[],[6],[],[8],[],[10],[],[12],[2],[14],[2],[16]
,[],[16,18],[],[2,20],[],[21,22],[23]]

[[6,9],[14,15]]
```

```
[[0,24],[12,20],[3,13],[1,24],[1,21],[2,17],[10,16],[0,1],[3,14],[4,13]
,[0,2],[7,9],[4,11],[11,18]]

[[0,20],[18,24],[22,24],[1,20],[4,10],[1,18],[15,19],[8,21],[5,11],[5,2
2],[13,14],[9,22],[18,22]]

[[12,13],[],[],[],[11,22],[],[],[],[3,4,5,7,17,19],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]]
```

Instance 21 :

```
30
15
4
10
2
2
0
0
100

[10,17.6,18.2,11.6,0,0,17.2,16.6,14.2,10.5,10.6,18.9,16,11.3,12.3,12.4,
15.5,14.2,15.4,11.1,16.9,12.5,0,0,0,10.2,15.3,13.2,12.6,14.1]

[[],[0],[],[2],[],[],[],[6],[4],[6,8],[],[10],[10],[12],[],[2,6,14],[],
[16],[],[13,18],[9],[20],[],[],[],[19],[25],[21],[7],[15]]

[[0,2],[13,21]]

[[0,6],[3,20],[2,10],[15,17],[6,12],[2,11],[2,16],[9,21],[5,6],[17,18],
[10,12],[11,17],[6,7],[0,20],[27,28]]

[[0,29],[12,20],[3,13],[1,7],[1,21],[2,17],[10,16],[0,1],[3,14],[8,13],
[0,2],[7,9],[4,11],[11,18],[21,27]]

[[0,4,5,14,18],[2,3],[],[],[1,6,9,10,11,15,20],[],[],[],[7,12,13,16,17,
19,21,22,23,24]

,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[]]
```

Instance 22 :

```
30
15
4
10
2
2
```

0
0
100

[10,10.9,13.6,0,0,0,0,0,19.2,18.6,10.5,18.9,12.7,16.4,19.1,17.7,13.7,19
.8,15.9,18.3,10.4,17.2,10.3,15.7,15.9,16.1,17.1,10.2,19.2,17.7]

[[],[0],[],[],[],[],[],[],[],[8],[],[10],[1],[12],[],[14],[13],[16],[13
],[18],[],[1,20],[],[2,13,22],[23],[9],[11],[27],[25,26,27],[28]]

[[16,19],[19,24]]

[[0,2],[1,15],[0,21],[18,19],[8,21],[8,10],[9,24],[1,9],[9,12],[12,23],
[8,18],[16,22],[10,13],[15,19],[12,21]]

[[0,2],[1,20],[2,10],[15,17],[8,12],[2,11],[2,16],[9,21],[17,18],[10,12
],[11,17],[0,23],[25,26],[26,27]]

[[0,1,4,12,13,14,18,20,22],[],[],[],[5,8,9,16,17,19],[2,6],[],[],[3,7,1
0,11,15,21,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[
],[],[],[],[],[],[],[],[],[]]

Instance 23 :

30
15
4
10
2
2
0
0
100

[10,14.1,19.1,16.3,10.8,17.1,19.1,0,14.2,16.8,0,0,0,0,15.8,13,11.8,15.4
,16.5,15.4,14.9,11,13.9,17.8,11.4,14.9,11,13.9,17.8,11.4]

[[],[0],[],[2],[],[4],[1,2],[],[],[8],[],[],[],[],[],[14],[],[16],[],[0
,14,18],[14],[20],[],[4,18,22],[23],[3],[25],[26],[19],[28]]

[[1,2],[27,14]]

[[0,4],[3,15],[5,8],[0,1],[4,23],[3,20],[9,29],[0,18],[8,14],[26,18],[0
,21],[1,14],[15,23]]

[[0,2],[1,25],[1,15],[18,19],[8,21],[6,26],[9,24],[5,28],[5,23],[8,18],
[16,22],[26,29],[15,19]]

[[0,8,9],[],[],[],[1,2,3,4,6,7,16,17,18],[],[],[],[5,10,11,14,15,20,21]
,[],[],[],[12,13,19,22,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[]

```
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 24 :

```
30
15
4
10
2
2
0
0
100

[10,17.4,14.5,13.7,13.9,18.3,10,0,19.3,14.9,10.4,18.9,16.2,16.5,12.6,18
.3,10,11,0,0,0,0,19.4,19.9,14.8,10,17.1,18.1,10.2,13.3]

[[],[0],[],[2],[],[4],[],[],[5],[8],[],[10],[],[12],[6],[14],[],[15,16]
,[],[],[],[],[11],[22],[15,23],[1],[3],[9],[17],[24]]

[[0,2,3]]

[[0,10],[4,12],[22,24],[13,27],[0,12],[2,23],[5,26],[4,15],[12,14],[2,1
6],[17,22],[12,17],[11,14],[8,9],[28,29]]

[[0,4],[3,15],[5,8],[0,1],[4,23],[3,22],[9,13],[0,18],[8,14],[10,17],[0
,13],[1,14],[15,23],[25,29]]

[[4,5,6,10,11,12,13],[7,8,18,22,23],[],[],[0,2,9,14,15,16,17,19,20],[1,
3,21,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 25 :

```
30
15
4
10
2
2
0
0
100

[10,10.7,10,11,17.1,19.5,0,0,14.3,13,0,0,13,0,19.4,13.5,18.2,16.7,17.6,
19.8,10.9,11.4,14,12.1,19.3,10,10.8,19.1,11.2,13.3]

[[],[0],[],[2],[],[4],[],[],[],[8],[],[],[9],[],[8],[9,14],[5],[16],[],
[18],[],[0,19,20],[8],[22],[23],[21],[17],[15],[27],[24,28]]
```

```
[[1,3],[25,26]]

[[0,5],[14,22],[4,20],[8,20],[8,12],[1,8],[2,9],[4,23],[5,22],[4,19],[9
,21],[14,23],[5,23],[2,18],[9,19]]

[[0,9],[4,20],[22,24],[13,21],[0,12],[2,23],[5,19],[4,15],[12,14],[14,1
6],[17,22],[12,17],[14,23],[8,9]]

[[2,4,5,6,8,10,14,16],[7],[],[],[9,12,13,18,22],[0,1,3,11,15,17,19,20,2
1,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 26 :

```
30
15
4
10
2
2
0
0
100

[10,0,0,0,10.2,10.7,11.9,0,19.3,11.1,10.3,18.9,19.7,16.6,16.2,18.8,16.4
,0,18.2,17,14.3,16.1,18.6,14.2,13.7,10.2,11.2,12.2,13.2,14.2]

[[],[],[],[],[],[4],[],[],[],[8],[6],[8,10],[],[5,12],[],[14],[12],[],[
],[8,18],[0,4],[20],[8],[22],[23],[21],[16],[19],[24],[28]]

[[6,11],[23,24]]

[[0,6],[12,15],[12,21],[5,23],[4,9],[6,9],[6,23],[22,24],[19,21],[0,15]
,[11,19],[6,8],[5,15],[13,26]]

[[0,5],[14,22],[6,20],[8,20],[8,12],[5,10],[0,9],[4,23],[5,22],[4,19],[
14,23],[10,18]]

[[0,1,4,5,6,8,14,18],[7,12,13,19,22],[],[],[2,3,10,11,15,16,17,20,21,23
,24],[9],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 27 :

```
30
15
4
10
```

2
2
0
0
100

[10,17.3,0,15.7,13.4,11.9,0,0,14.4,19.3,10.3,19,16.5,11.6,12.9,14.1,14.6,17.9,0,0,17.8,10.9,13.2,16.4,18.2,10,11.2,12.9,13.4,14.3]

[[],[0],[],[],[],[4],[],[],[],[8],[],[9,10],[],[12],[9],[3,14],[15],[16],[],[],[],[20],[8,21],[13,17,22],[23],[11],[25],[24],[17],[28]]

[[5,8],[13,16]]

[[0,10],[0,9],[0,4],[14,17],[8,22],[8,12],[11,22],[17,22],[10,15],[0,3],[9,21],[13,14],[4,8],[26,29]]

[[12,15],[5,23],[4,9],[1,9],[4,23],[22,24],[13,21],[0,15],[3,20],[0,8],[3,13],[5,15],[9,16],[20,27]]

[[0,1,4,6,12,13,18,20,21],[],[],[],[5,8,10],[],[],[],[2,3,7,9,14,15,22],[],[],[],[11,16,17,19,23,24],[],[],[],[

```
[[0,2,12,16,17],[],[],[],[1,3,6,7,8,9,10,11,14,18,20],[],[],[],[4,5,13,
15,19,21,22,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 29 :

```
30
15
4
10
2
2
0
0
100

[10,13.8,0,0,0,0,0,19.4,14.5,15.5,10.2,19,10,11.7,16.5,14.7,10.9,19.2,1
9.9,18.5,14.8,10.3,12.4,10.6,17.1,15.5,13.2,10,14.4,12]

[[],[0],[],[],[],[],[],[1,6],[],[8],[],[10],[],[12],[],[14],[],[16],[1]
,[18],[],[20],[8,11],[22],[21,23],[7],[19],[13],[27],[28]]

[[7,8],[0,10]]

[[9,13],[1,20],[16,22],[11,14],[8,9],[8,10],[10,13],[0,7],[15,23],[12,2
2],[9,25],[15,26],[9,12]]

[[0,1],[20,21],[12,13],[0,14],[8,15],[18,24],[0,20],[9,23],[10,22],[0,8
],[21,23],[1,24]]

[[2,3,4,6,10,12,14,15,20],[],[],[],[0,8,11,13,21,22,23,24],[],[],[],[1,
5,7,9,16,17,18,19],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 30 :

```
30
15
4
10
2
2
0
0
100

[10,17.1,17.2,17.7,12.8,15.5,15.7,18.5,19.5,13.6,10.2,19,16.8,16.8,13.3
,10,19.1,0,0,0,0,0,12.8,11.6,10.1,11.1,12.1,13.1,14.1,15]
```

```
[[],[0],[],[2],[0],[4],[],[6],[],[8],[],[10],[],[12],[],[14],[11],[],[]
,[],[],[],[5,15],[22],[23],[15,16],[25],[7],[9,27],[28]]

[[13,15],[5,8]]

[[0,10],[0,2],[7,14],[7,9],[1,9],[1,24],[10,23],[4,6],[5,22],[4,9],[1,2
4],[7,8],[3,13],[1,26]]

[[3,13],[1,6],[16,22],[11,14],[8,9],[8,10],[6,13],[0,7],[3,23],[6,22],[
3,9],[9,12],[26,29]]

[[0,1,4,5,8,12,13,14,15,18],[],[],[],[2,3,6,7,10,11,16,19],[],[],[],[9,
17,20,21,22,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 31 :

```
30
15
4
10
2
2
0
0
100

[10,10.4,12.7,15.1,16,16.7,0,17.6,14.5,11.7,0,0,13.5,11.8,0,0,17.3,10.4
,11,12.9,11.8,19.8,11.5,14.9,16,11.2,13.4,15.4,12.2,15.1]

[[],[0],[],[2],[],[4],[],[6],[],[0,8],[],[],[9],[12],[],[],[1],[7,16],[
17],[18],[18],[20],[],[4,19,22],[23],[13],[25],[5],[21],[24]]

[[19,22]]

[[0,1],[9,13],[4,19],[2,13],[1,4],[1,18],[19,24],[4,14],[3,8],[9,24],[1
3,24],[3,9],[1,4],[0,28]]

[[0,22],[0,2],[7,20],[7,9],[1,9],[1,17],[20,23],[4,19],[5,22],[4,9],[7,
8],[9,19],[12,26]]

[[0,6,8,9,12],[],[],[],[1,2,7,14,16],[],[],[],[3,10,13,17,18,19,22],[],
[],[],[4,5,11,15,20,21,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 32 :

```
30
15
```

```
4
10
2
2
0
0
100

[0,0,18.1,12.4,19.1,17.9,17.6,16.7,19.6,19.9,10.1,19,10.3,16.9,16.8,10.
6,0,0,11.5,0,15.2,14.5,16.1,17.1,10.5,12.1,18.3,17.2,11,15.5]

[[],[],[],[2],[],[4],[],[5,6],[],[8],[3],[10],[2,7],[12],[2,7],[14],[],
[],[11],[],[],[7,20],[],[13,22],[23],[18],[22],[15],[26,27],[28]]

[[2,4],[4,7],[6,8]]

[[2,9],[3,10],[7,9],[5,11],[9,14],[3,20],[14,23],[9,18],[6,7],[3,4],[26
,27],[15,21]]

[[9,13],[4,25],[2,13],[4,10],[12,18],[24,25],[4,14],[3,8],[9,24],[6,15]
,[13,24],[9,11],[6,20],[24,29]]

[[0,2,3,6],[1,8],[],[],[4,5,7,10,11,12,13,16,22,23]
,[14,17,18,20,21,24],[],[],[9,15,19],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]]
```

Instance 33 :

```
30
15
4
10
2
2
0
0
100

[10,17,13.6,19.7,12.3,19.1,0,0,14.6,18,10,19,17,11.9,13.6,15.9,13.7,11.
6,12.1,17.2,18.7,19.2,0,0,0,10.7,19.2,14.9,11.2,13.3]

[[],[0],[],[2],[],[3,4],[],[],[],[8],[],[10],[8],[12],[],[5,11,14],[],[
5,16],[],[18],[],[20],[],[],[],[1],[25],[17,26],[15],[27,28]]

[[3,8],[14,19],[13,17]]

[[0,2],[20,21],[16,21],[17,18],[4,13],[2,12],[4,8],[3,19],[10,21],[0,13
],[2,16],[2,4],[12,17],[21,26]]
```

```
[[0,16],[10,13],[2,9],[0,19],[3,10],[5,9],[1,11],[9,14],[3,20],[1,19],[
14,19],[9,18],[3,4],[4,15]]

[[2,3,6,14,18],[],[],[],[8,9,10,11,12,13,16,19,20,22],[],[],[],[0,1,4,5
,7,15,17,21,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 34 :

```
30
15
4
10
2
2
0
0
100

[10,10.2,19,17.1,15.4,10.3,19.5,15,19.7,0,0,0,0,0,10.3,11.2,11.9,17.3,1
2.7,14.4,12.2,13.9,15.3,11.3,19.4,11.11,10.6,12.5,14.4,15]

[[],[0],[],[2],[],[2,4],[0],[6],[],[],[],[],[],[],[],[8,14],[4],[16],[]
,[18],[],[20],[],[22],[23],[3],[5],[17],[27],[26,28]]

[[16,19],[21,23]]

[[0,1],[4,19],[3,18],[5,23],[2,17],[5,19],[19,22],[0,3],[6,16],[12,15],
[8,14],[2,13],[4,14],[8,12],[15,25]]

[[0,2],[20,21],[16,23],[17,18],[4,13],[6,12],[4,8],[3,19],[7,21],[0,24]
,[2,16],[2,4],[12,17],[21,22],[25,26]]

[[2,22],[0,4,5,12,18,20],[],[],[1,3,6,7,8,9,13,21,23],[],[],[],[10,11,1
4,15,16,17,19,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 35 :

```
30
15
4
10
2
2
0
0
100
```

```
[10,13.5,14.5,14.4,18.6,11.5,0,0,0,0,0,19.1,10.5,12,17.1,16.5,10,12.9,1
3.2,11.6,15.7,18.7,19.8,13.5,13.8,9.9,10.6,11.7,12.5,13.8]

[[],[0],[],[2],[],[4],[],[],[],[],[],[4,10],[],[12],[],[14],[14],[16],[
],[2,18],[],[20],[12],[22],[5,19,23],[24],[15],[17],[21],[25]]

[[21,24],[3,5]]

[[4,13],[3,18],[2,17],[1,5],[0,24],[16,22],[13,15],[11,15],[5,16],[19,2
4],[1,17],[1,12],[16,20],[2,22],[12,29]]

[[0,1],[4,19],[5,23],[5,19],[19,22],[0,3],[12,15],[12,14],[2,13],[4,14]
,[2,12],[11,19]]

[[0,2,4,12,18],[1,14,15,16,19],[],[],[6,7,8,9,10,11,13,20,21,22,23],[3,
5,17,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 36 :

```
30
15
4
10
2
2
0
0
100

[0,0,0,11.8,11.7,12.7,11.4,13.2,19.7,12.4,19.9,19.1,17.3,17.1,13.9,11.7
,0,0,13.8,18.8,19.2,13.4,14.4,15.6,18.3,17.2,17.5,16.6,10.2,14.4]

[[],[],[],[2],[3],[4],[],[6],[],[7,8],[],[10],[4,6],[12],[],[3,14],[],[
],[9,12],[18],[11,15],[4,20],[],[22],[10,23],[5],[13],[19],[21,25],[28]
]

[[3,18,20]]

[[6,19],[8,22],[7,13],[19,24],[3,13],[14,21],[6,21],[3,21],[22,23],[8,1
4],[5,10],[5,8],[3,11],[6,26]]

[[4,13],[3,18],[3,9],[7,24],[6,8],[13,15],[11,15],[5,13],[3,10],[12,20]
,[14,22],[8,28]]

[[0,1,2,3,4,14,15,16],[],[],[],[6,7,8,9,10,11,12,17,18,19,20,22],[],[],
[],[5,13,21,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],
[],[],[],[],[],[],[],[],[],[]]
```

Instance 37 :

```
30
15
4
10
2
2
0
0
100

[0,0,15.4,19.1,14.9,13.9,12.4,12.3,14.8,10.5,19.8,19.1,14.1,12.1,10.7,1
7,16.4,14.1,14.3,16,0,0,0,17.8,12.8,14.2,13.5,10,15.4,16.6]

[[],[],[],[2],[],[4],[5],[6],[],[2,6,8],[],[10],[9],[12],[],[14],[],[16
],[3],[18],[],[],[],[13,15,22],[23],[19],[25],[11],[7],[27,28]]

[[11,12,14]]

[[2,5],[2,4],[16,18],[4,17],[9,17],[3,9],[5,7],[3,4],[3,8],[4,10],[19,2
7],[14,16],[7,24],[12,25],[14,23]]

[[2,19],[10,16],[7,13],[16,17],[19,24],[3,13],[4,16],[6,26],[10,23],[8,
14],[6,10],[5,8],[3,11]]

[[0,2,3,20,21,22],[8,10,16],[],[],[4,5,6,9,11,12,13,14,15,18,23,24],[],
[],[],[1,7,17,19],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 38 :

```
30
15
4
10
2
2
0
0
100

[10,13.4,10.9,16.5,18,15.1,13.3,11.4,19.8,18.6,19.8,19.1,10.8,17.2,17.4
,12.3,0,0,14.9,13.1,16.2,12.8,0,0,0,10.8,11.7,12.9,14.6,16.3]

[[],[0],[],[2],[],[4],[],[6],[],[1,2,8],[],[10],[],[12],[11],[7,14],[],
[],[3,14],[18],[],[11,13,20],[],[],[],[9],[19],[15],[21],[28]]

[[1,4,8]]
```

```
[[14,18],[8,12],[3,6],[5,14],[1,21],[1,7],[2,5],[18,27],[0,12],[6,29],[
0,13],[2,6],[9,10],[0,29]]

[[2,5],[4,20],[15,18],[4,27],[9,13],[3,9],[5,7],[3,4],[1,3],[1,5],[19,2
1],[14,20],[7,28],[8,12]]

[[0,1,2,4,8,10,11,12],[3,14,22,23,24],[],[],[6,7,9,13,18,19,20],[5,15,1
6,17,21],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 39 :

```
30
15
4
10
2
2
0
0
100

[10,16.6,0,13.8,0,0,0,0,14.9,16.7,19.8,19.1,17.6,12.2,14.2,17.6,12.8,15
.4,15.4,10.3,19.6,17.6,18.2,12,11.7,10,19.2,16.5,11.8,12.2]

[[],[0],[],[2],[],[],[],[],[],[8],[],[10],[],[8,12],[],[14],[],[10,16],
[0,3],[18],[],[20],[1],[22],[23],[13,17],[25],[19],[24],[27,28]]

[[11,13,15]]

[[0,8],[6,22],[12,23],[0,15],[8,10],[0,3],[8,23],[9,16],[1,9],[9,11],[3
,10],[12,23],[8,15],[20,26]]

[[14,18],[8,16],[3,27],[1,14],[1,21],[2,17],[18,24],[12,17],[18,24],[0,
13],[9,10],[0,29]]

[[0,1,6,10,12,16],[],[],[],[2,8,14,22],[],[],[],[3,4,5,7,9,11,13,15,17,
18,19,20,21,23],[24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 40 :

```
30
15
4
10
2
2
0
```

0
100

[10,19.9,11.8,11.2,0,0,15.2,19.6,0,14.9,19.7,19.1,14.3,17.3,11,12.9,11,
11,16,17.5,0,0,12.8,14.2,16.1,11.8,12.2,17.1,10.9,15.5]

[[],[0],[],[2],[],[],[],[6],[],[3,8],[1],[10],[],[12],[13],[14],[10],[1
6],[3],[18],[],[],[9],[14,22],[6,23],[11],[19],[15],[17,26],[28]]

[[16,17],[12,13]]

[[2,16],[12,15],[0,15],[0,10],[3,24],[11,19],[3,7],[6,24],[19,24],[0,27
],[1,12],[0,22],[0,24]]

[[0,2],[6,22],[2,10],[2,3],[10,23],[9,16],[1,7],[3,10],[12,23],[6,15],[
7,27]]

[[2,4,5,6,8,12,13,14,20],[],[],[],[0,1,3,9,18,19,21],[],[],[],[7,10,11,
15,16,17,22,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],
[],[],[],[],[],[],[],[],[],[]]

Instance 41 :

30
15
4
10
2
1
0
0
100

[10,13.2,17.2,0,17.4,18.7,16.2,18.7,14.9,13,19.7,19.1,0,0,0,0,19.2,16.6
,16.6,14.7,16.6,17,17.3,16.3,10.6,10.1,13.3,17.2,16.1,15]

[[],[0],[],[],[1],[4],[4],[6],[2],[5,8],[0,2],[10],[],[],[],[],[6],[16]
,[],[10,18],[1],[12,20],[0],[6,22],[23],[9],[25],[24],[17],[27,28]]

[[0,2],[8,9]]

[[0,8],[7,11],[0,10],[11,20],[2,6],[19,21],[4,21],[9,16],[6,8],[2,22],[
5,20],[5,9],[2,8],[2,26]]

[[5,16],[0,24],[1,24],[11,19],[0,7],[6,24],[19,24],[0,5],[1,20],[0,22],
[18,28]]

[[0,1,2,4,6,12,14],[],[],[],[3,5,10,15,18,22,23],[11,13,16,24],[],[],[7
,8,9,17,19,20,21],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]

```
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 42 :

```
30
15
4
10
2
2
0
0
100

[10,16.5,12.7,15.8,10.6,19.9,17.1,17.8,10,0,19.6,19.2,0,17.4,14.5,0,17.
3,12.2,17.1,11.9,10.1,0,0,18.5,15,12.2,15.3,17.5,13.4,16.7]

[[],[0],[],[2],[],[4],[],[5,6],[1],[],[],[5,10],[],[12],[10],[],[5],[16
],[],[13,18],[4,18],[],[],[10,13,22],[6,18,23],[3,8],[19],[17],[7],[27,
28]]

[[7,11,25]]

[[0,4],[2,14],[5,8],[0,18],[14,18],[6,14],[0,8],[13,14],[11,18],[11,16]
,[8,24],[24,29]]

[[7,14],[0,10],[11,20],[3,6],[3,6],[13,16],[6,13],[2,11],[5,20],[5,9],[
2,8],[0,2],[5,25]]

[[4,6,10,12,18,20,22],[],[],[],[0,1,2,3,5,13,16,17,19,23,24],[],[],[],[
14],[7,8,9,11,15,21],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 43 :

```
30
15
4
10
2
2
0
0
100

[10,0,18.1,13.2,13.7,11.1,18.1,16.9,0,0,19.6,19.2,0,0,11.3,18.8,15.5,17
.9,17.7,19,13.6,16.5,16.5,10.6,19.5,10.9,11.3,12.7,13.8,14.7]
```

```
[[],[],[],[2],[],[4],[2],[6],[],[],[],[10],[],[],[],[14],[7],[1,8,16],[
5],[14,18],[10],[20],[16],[0,22],[12,23],[3],[21],[25,26],[15],[19,28]]

[[21,22],[0,11]]

[[0,2],[0,6],[3,15],[16,18],[6,14],[7,20],[21,24],[3,4],[4,19],[2,10],[
10,16],[0,22],[24,27]]

[[0,4],[15,22],[2,14],[5,15],[0,18],[14,18],[6,14],[0,3],[4,14],[6,18],
[11,16],[24,29]]

[[0,10,14,15],[],[],[],[1,2,3,4,5,6,7,16,20,21,22,23],[],[],[],[8,17,18
],[9,11,12,13,19,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 44 :

```
30
15
4
10
2
2
0
0
100

[10,13,13.6,10.5,16.9,12.3,19,16,10,17.4,19.5,19.2,11.4,17.5,18.1,0,0,0
,0,0,17.1,11.2,11.1,12.7,14,11.11,12.4,17.5,16.8,15]

[[],[0],[],[2],[],[3,4],[],[6],[],[8],[],[10],[5],[12],[2,4,9],[],[],[]
,[],[],[7,8],[20],[],[9,22],[23],[14],[25],[13],[26,27],[24,28]]

[[3,5]]

[[0,11],[14,21],[1,8],[3,14],[11,24],[0,4],[2,9],[1,2],[11,13],[2,3],[9
,23],[13,21],[3,10],[12,25]]

[[0,8],[0,6],[3,22],[9,14],[7,8],[21,24],[3,12],[4,6],[2,10],[1,9],[6,1
0],[0,1],[21,29]]

[[4,6,8,10],[0,2,22],[],[],[3,5,7,9,11,18,19,20,21],[],[],[],[1,12,13,1
4,15,16,17,23,24],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
,[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[
],[],[],[],[],[],[],[],[],[]]
```

Instance 45 :

```
30
15
```

4
10
2
2
0
0
100

[10,16.3,19,17.9,0,0,10,15.1,15.1,15.5,19.5,19.2,18.1,12.5,14.8,19.4,11
.9,19.1,18.8,13.4,10.5,15.9,0,0,0,15.1,14.2,18.4,10.9,11.5]

[[],[0],[],[2],[],[],[0],[6],[],[8],[],[10],[0],[7,12],[],[14],[],[16],
[],[15,18],[],[20],[],[],[],[13],[9],[19],[17],[27,28]]

[[2,19],[2,6],[12,28]]

[[10,19],[3,16],[11,13],[8,15],[0,8],[17,18],[2,8],[10,11],[7,8],[0,11]
,[15,18],[9,18],[13,27]]

[[0,17],[14,21],[1,8],[14,18],[11,17],[0,18],[2,18],[16,19],[1,2],[2,3]
,[9,14],[13,18],[3,10],[25,26]]

[[0,2,4,5,6,7,10,12,13,14,15,20],[],[],[],[1,3,11,19,21,22,23,24],[],[]
,[],[8,9,16],[
],[],
[],[],[],[],[],[],[],[]]

École Nationale Supérieure des Mines de Saint-Étienne

Fatme MAKSSOUD

Apport de l'optimisation combinatoire pour la reconfiguration des lignes de production

Spécialité : Génie Industriel

Mots clés : Reconfiguration, réutilisabilité, ligne d'usinage, ligne d'assemblage, programmation linéaire, Goal Programming.

Résumé :

Actuellement, les fabricants sont soumis à une pression économique importante et à une concurrence internationale accrue due à la globalisation des marchés. Pour réussir, les fabricants doivent être capables de répondre rapidement aux changements de la demande en adaptant leurs systèmes de production. Cette adaptation aux changements peut être réalisée à travers multiples reconfigurations du système de production.

Les travaux présentés dans ce mémoire portent sur l'élaboration des méthodes de recherche opérationnelle permettant d'accompagner le décideur lors de la reconfiguration d'une ligne de transfert ou d'assemblage. Ce problème apparaît lorsqu'un nouveau produit doit être fabriqué par une ligne existante ou lorsqu'il y a eu des changements dans les caractéristiques du produit. Par conséquent, il devient nécessaire de modifier la configuration du système de production tout en minimisant les coûts induits. Ces coûts sont évalués différemment pour les systèmes automatisés ou manuels. Dans le premier cas, qui correspond au cas des lignes de transfert, pour limiter les investissements, il est souhaitable de réutiliser au maximum les équipements existants à condition que les contraintes techniques et technologiques soient respectées. Dans le cas des lignes manuelles qui sont représentées dans notre étude par les lignes d'assemblage, l'objectif est de minimiser les coûts liés à l'apprentissage des opérateurs causés par la réaffectation de leurs tâches.

Les méthodes de résolution exactes basées sur la modélisation mathématique et la programmation linéaire en nombre mixtes ainsi qu'une méthode de type goal programming sont développées dans ce travail pour argumenter la prise de décisions lors de la reconfiguration des lignes de production. Les méthodes proposées ont été testées avec succès sur des échantillons de problèmes proches des cas industriels et ont montré leur efficacité.

École Nationale Supérieure des Mines de Saint-Étienne

Fatme MAKSSOUD

Contribution of combinatorial optimization for the reconfiguration of manufacturing systems

Speciality : Industrial Engineering

Keywords : Reconfiguration, reusability, machining line, assembly line, mixte integer programm, Goal Programming.

Abstract :

Global competition causes fluctuations in product demand and requires more frequent modifications of product characteristics. As a consequence, the production systems have to be frequently adapted to new production requirements.

This work develops new combinatorial optimization methods for supporting decision makers at the reconfiguration stage considered for transfer and assembly lines. If new products have to be manufactured at the line or existing products are modified, then the line has to be reconfigured in order to meet new production requirements. In highly automated lines, as the transfer lines, the reconfiguration problem is focused on the readjustment of the equipment. To reduce the investment costs, the decision makers aim to reuse the available equipment as much as possible. The existence of compatibility constraints between new operations to be performed and existing facilities makes the reconfiguration problem hard and combinatorial. In manual assembly lines also studied in this thesis, the reconfiguration problem mostly concerns the reassignment of tasks to workers ant the minimization of the cost of retraining operators.

The developed methods are based on the mathematical modelling and mixed integer programming, a goal programming approach is designed as well. These methods were successfully tested on a dataset of problem instances close to real industrial problems. The obtained results show the effectiveness and the efficiency of the solution methods proposed.