

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté des Sciences

Thèse de Doctorat

Présentée par:

MERCHICHI Sabrina

En vue de l'obtention du diplôme de **Doctorat** en:

Filière: INFORMATIQUE

Option: Systèmes Informatiques et Ingénierie des Logiciels

**Approches de résolution exacte du problème de composition de
cellules dans les systèmes cellulaires de production**

Devant le jury :

MAOUCHE	Amin Riad	MC/A	UMBB	Président
AIT ZAI	Abdelhakim	MC/A	USTHB	Examineur
ATIF	Karim	MC/A	USTHB	Examineur
BERRICHI	Ali	MC/A	UMBB	Examineur
BOULIF	Menouar	Prof	UMBB	Directeur de thèse

Année Universitaire : 2015/2016

À ma mère

Remerciements

Je tiens à remercier en premier lieu Allah le Tout Puissant de m'avoir donné courage et santé pour achever ce travail.

Je tiens à exprimer ma gratitude à Mr BOULIF Menouar, mon Directeur de thèse, pour avoir dirigé et encadré mon travail. Je le remercie pour sa disponibilité, son soutien et ses conseils tout au long de ces années de préparation de mon projet de thèse.

Que mes vifs remerciements aillent à M. MAOUCHE Amin Riad, maître de conférences à l'UMBB qui m'a fait l'honneur de présider le jury.

J'exprime ma gratitude à M. AIT ZAI Abdelhakim, maître de conférences à l'USTHB, à M. BERRICHI Ali, maître de conférences à l'UMBB et à M. ATIF Karim, maître de conférences à l'USTHB pour avoir fait partie de ce jury de thèse en tant qu'examinateur.

Je remercie également tous les membres du laboratoire LIMOSE de l'université de Boumerdes, en particulier le professeur MEZGHICHE Mohamed.

Je ne peux aussi oublier Mr HACHMAN, enseignant à INPED, qui m'a beaucoup aidé et encouragé, ni Mr FOUILHOUX Pierre, Maître de Conférences de Université Pierre et Marie Curie (Paris 6) qui a eu l'amabilité de répondre à mes question, ni Mr AÏDER Méziane, Professeur à l'Université de Sciences et de la Technologie Houari Boumediene et Directeur de Recherche qui m'a accueilli au sein de son bureau et qui a eu l'amabilité de me procurer une documentation très riche.

Je remercie par ailleurs tous les membres de ma famille, en particulier mes très chers parents pour le soutien qu'ils m'ont apporté dans les moments difficiles que j'ai connus durant ces dernières années.

Je tiens aussi à remercier l'ensemble du personnel de la Direction de Logement de BOUMERDES, avec lesquels je travaille, pour leur encouragement.

Enfin, mes remerciements vont à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Table des matières

Introduction générale.....	1
Chapitre 1: Les systèmes cellulaires de production	3
1.1. Introduction	3
1.2. Les systèmes de production	3
1.3. Classification des systèmes de production	4
1.3.1. Systèmes à production unitaire	4
1.3.2. Systèmes de production continue.....	4
1.3.3. Systèmes de production manufacturière	4
1.3.3.1. Systèmes de production en ligne « Flow shop »	4
1.3.3.2. Systèmes de production en ateliers spécialisés « Job shop »	5
1.4. Les critères de choix d'un type de systèmes de production	6
1.5. Les systèmes cellulaires de production	7
1.5.1. Conception des SCPs	8
1.5.2. Avantages et inconvénients des SCPs.....	9
1.6. Conclusion	12
Chapitre 2: Composition de cellules: Etat de l'art.....	13
2.1. Introduction	13
2.2. Naissance et évolution du problème de formation de cellules	13
2.3. Description du problème FC	14
2.3.1. Les objectifs	15
2.3.2. Les contraintes	15
2.4. Les méthodes de formation de cellules : état de l'art.....	16
2.4.1. Méthode visuelle	18
2.4.2. Méthodes basées sur la forme des produits.....	18
2.4.3. Méthodes basées sur la gamme de production des produits	18
2.4.3.1. Méthodes de la réorganisation de la matrice (Array-Based Clustering)	18
2.4.3.2. Méthodes basées sur un groupement hiérarchique.....	19
2.4.3.3. Méthodes basées sur un groupement non hiérarchique.....	21
2.4.3.4. Méthodes basées sur la programmation mathématique.....	21
2.4.3.5. Méthodes basées sur la théorie des graphes	23
2.4.3.6. Méthodes approchées	25

2.4.3.7. Les méthodes basées sur l'intelligence artificielle	26
2.4.3.8. Les méthodes exactes	26
2.5. <i>Carences associées aux méthodes de formation de cellules</i>	27
2.6. <i>Conclusion</i>	27
Chapitre 3: Modèles pour le problème de formation de cellules	28
3.1. <i>Introduction</i>	28
3.2. <i>Position du problème</i>	28
3.2.1. La stratégie de formation de cellules	28
3.2.2. Les hypothèses de résolution	29
3.2.3. L'objectif.....	29
3.2.4. Les Contraintes	29
3.3. <i>Modélisation</i>	29
3.3.1. Notations	30
3.3.2. Données et paramètres	30
3.3.3. Les méthodes mathématiques	31
3.3.3.1. Modèle basé sur la relation machine-cellule	31
3.3.3.2. Modèle basé sur la relation machine-médiane	32
3.3.3.3. Une linéarisation efficace	36
3.3.3.4. Modèle basé sur la relation machine-machine	38
3.3.4. Les méthodes graphiques	39
3.3.4.1. Modèle basé sur les arêtes	40
3.3.4.2. Modèle basé sur les co-cycles	41
3.4. <i>Comparaison théorique des modèles</i>	43
3.4.1. Comparaison en termes de nombre de variables et de contraintes	43
3.4.2. Comparaison en termes de problème de redondance.....	45
3.4.3. Comparaison en termes de problème de cécité.....	48
3.5. <i>Conclusion</i>	49
Chapitre 4: Les méthodes de résolution exactes.....	50
4.1. <i>Introduction</i>	50
4.2 <i>Motivation</i>	50
4.3. <i>Les méthodes d'exploration par séparation et évaluation</i>	51
4.4. <i>Généralités sur la structure des grands programmes linéaires</i>	53
4.5. <i>Génération de colonnes pour les problèmes de grande dimension</i>	53
4.5.1. Principe général	53
4.5.2. Décomposition de Dantzig-Wolfe	55
4.5.3. Résolution des PLNE par génération de colonnes	55
4.5.3.1. Décomposition des PLNE	55

4.5.3.2. Méthode de « Branch and Price »	56
4.6. Génération de contraintes pour les problèmes de grande dimension	57
4.6.1. Principe général	57
4.6.2. Décomposition de Benders	58
4.6.3. Résolution de PLNE par la méthode de génération de contraintes.....	59
4.6.3.1. Plans de coupe généraux	59
4.6.3.2. Approche polyédrale	59
4.6.3.3. Méthode de « Branch and Cut ».....	60
4.7 Conclusion.....	61
Chapitre 5: Une méthode par séparation et évaluation cocyclique guidée par les	
contraintes.....	62
5.1. Introduction	62
5.2. Les méthodes SE et le problème FC	62
5.3. Principe général de la méthode SE cocyclique développée	64
5.3.1. L'idée générale.....	64
5.3.2. Règle de séparation	65
5.3.2.1. Procédure de séparation	65
5.3.2.2. Détermination de la liste des co-cycles	65
5.3.3. Déplacement dans l'arborescence	67
5.3.4. L'Evaluation de la borne inférieure	68
5.3.5. Pénalité.....	69
5.4. Amélioration de la méthode SE cocyclique	69
5.4.1. Amélioration basée sur le nombre maximum de cellules autorisé.....	69
5.4.2. Amélioration basée sur le nombre maximum de machines autorisé dans une cellule	
.....	70
5.4.3. Amélioration basée sur les contraintes de cohabitation et de non cohabitation.....	71
5.4.4. Amélioration de la BI.....	72
5.5. Conclusion.....	73
Chapitre 6: Résultats et Analyse.....	74
6.1. Introduction	74
6.2. Environnement.....	74
6.2.1. Ressources informatiques utilisées	74
6.2.2. Données des exemples	75
6.3. Tests effectués.....	76
6.3.1. Tests sur la méthode SE cocyclique.....	76
6.3.2. Tests sur les modèles proposés	76
6.3.3. Comparaison avec les autres méthodes.....	76
6.4. Analyse des résultats	77

6.4.1. Mesures de performance	77
6.4.2. Analyse des résultats de la méthode SE cocyclique	77
6.4.3. Analyse des résultats des modèles	81
6.4.4. Comparaison des méthodes.....	84
6.5. <i>Conclusion</i>	85
Conclusion générale et perspectives	86
Références Bibliographiques	89
Annexe A: Lexique	101
Annexe B: Notion de co-cycle	105
Annexe C: Détails des exemples du chapitre 6	107

Liste des figures

Figure 1.1. Système de production en ligne « flow shop »	5
Figure 1.2. Système de production en ateliers spécialisés « Job shop »	5
Figure 1.3. Système cellulaire de production	6
Figure 1.4. Domaine d'application des différents types de systèmes de production [CAM01].	7
Figure 1.5. L'approche de conception des systèmes cellulaires de production [PRO91].	9
Figure 2.1. Classification des méthodes de formation de cellules	17
Figure 2.2. Exemple d'une matrice d'incidence produits/machines [HAC09]	19
Figure 2.3. Un dendrogramme démontrant le groupement hiérarchique [ALK07]	20
Figure 2.4. L'évolution des coefficients de similarité dans le temps [YIN06]	20
Figure 3.1. Le problème p-médian	33
Figure 3.2. Une instance de graphe de flux	40
Figure 3.3. Exemple d'une composition C	46
Figure 3.4. Les matrices représentantes de la composition C générées par le modèle (L1)	46
Figure 3.5. Les matrices représentantes de la composition C générées par le modèle (L2)	47
Figure 3.6. La matrice représentante de la composition C	47
Figure 3.7. Le problème de redondance avec le modèle (T1)	47
Figure 3.8. Décomposition en co-cycles	48
Figure 3.9. Une solution invisible [BOU06a]	49
Figure 4.1. Algorithme de Branch-and-Bound	52
Figure 4.2 Structure bloc diagonale: a) avec contraintes couplantes b) avec variables couplantes [MIN07]	53
Figure 4.3. Schéma général de la génération de colonne [TOU13]	54
Figure 4.4 Algorithme de Branch and Price	56
Figure 4.5. Le schéma général de la génération de contraintes	58
Figure 4.6 Algorithme de Branch and Cut	60
Figure 5.1. Décomposition en co-cycles	64
Figure 5.2. L'arbre de solutions sans pénalité	65
Figure 5.3. Algorithme de génération de la liste des co-cycles	67
Figure 5.4. Exemple de calcul de la BI	68
Figure 5.5. L'arbre de solutions réduit	70
Figure 5.6. Une solution construite par: (a) des co-cycles arbitraires b. des constructeurs- cellule	71
Figure 5.7. Solution optimale invisible	71
Figure 5.8. Graphe de cohabitation et de non cohabitation	72
Figure 5.9. La coupe minimum de $G2$	73
Figure 5.10. Un exemple de l'amélioration de la BI	73
Figure 6.1. Comparaison des versions de la méthode SE cocyclique	78
Figure 6.2. Impact des améliorations entraînées par les contraintes	79

Figure 6.3. Comparaison des modèles en utilisant le solveur <i>GLPK</i>	82
Figure 6.4. Comparaison des modèles en utilisant le solveur <i>SCIP</i>	82
Figure 6.5. Comparaison entre <i>CBBm2</i> et les autres méthodes.....	85
Figure B.1.Un co-cycle	105

Liste des tableaux

Tableau 1.1. Gains d'implantation des SCPs dans 32 firmes US [WEM89].....	11
Tableau 1.2. Les Performances rapportés par l'étude de 46 firmes US [WEM97]	12
Tableau 3.1. Les données de l'exemple	39
Tableau 3.2. Nombre de variables et de contraintes requises par les différents modèles	44
Tableau 3.3. Exemple de nombre de variables et de contraintes requises par les différents modèles.....	45
Tableau 5.1. La liste complète des co-cycles	66
Tableau 5.2. La liste des co-cycles réduite.....	72
Tableau 6.1. Les données des exemples numériques	75
Tableau 6.2. Résultats de calcul du premier test	80
Tableau 6.3. Résultats de calcul du deuxième test.....	83
Tableau 6.4. Nombre de variables et de contraintes pour chaque exemple	84
Tableau C.1. Matrice d'incidence de l'exemple 1	107
Tableau C.2. Matrice d'incidence de l'exemple 2	107
Tableau C.3. Matrice d'incidence de l'exemple 3	108
Tableau C.4. Matrice d'incidence de l'exemple 4	108
Tableau C.5. Matrice d'incidence de l'exemple 5	108
Tableau C.6. Matrice d'incidence de l'exemple 6	109
Tableau C.7. Matrice d'incidence de l'exemple 7	109
Tableau C.8. Matrice d'incidence de l'exemple 8	109
Tableau C.9. Matrice d'incidence de l'exemple 9	110
Tableau C.10. Matrice d'incidence de l'exemple 10.....	110
Tableau C.11. Matrice d'incidence de l'exemple 11.....	111
Tableau C.12. Matrice d'incidence de l'exemple 12.....	111
Tableau C.13. Matrice d'incidence de l'exemple 13.....	112
Tableau C.14. Matrice d'incidence de l'exemple 14.....	112
Tableau C.15. Matrice d'incidence de l'exemple 15.....	113
Tableau C.16. Matrice d'incidence de l'exemple 16.....	114

Résumé

Cette thèse se situe dans le domaine de l'optimisation, en particulier celui de l'optimisation combinatoire. L'objectif de notre travail est d'étudier le problème de composition de cellules dans les systèmes cellulaires de production.

Dans la première partie de cette thèse, nous nous intéressons à l'importance de la variable de décision choisie dans la modélisation du problème de composition de cellules. Nous commençons par la présentation de trois modèles mathématiques et deux modèles basés sur la théorie des graphes pour ce problème. Ensuite, nous examinons ces différents modèles d'un point de vue théorique. La deuxième partie traite essentiellement la résolution exacte du problème de composition de cellules. La méthode exacte que nous choisissons est de type séparation et évaluation. Nous présentons d'abord, une première version de cette méthode qui utilise une procédure de branchement basée sur les co-cycles. Ensuite, nous proposons une deuxième version de cette méthode qui met en œuvre une amélioration guidée par les contraintes et utilise une nouvelle procédure de calcul de la borne inférieure basée sur la méthode de recherche d'un co-cycle de poids minimum. Différents tests sont effectués sur un jeu d'exemples dans le but de montrer les avantages et les limites des modèles et des méthodes exactes présentées.

Mots-clés: Systèmes cellulaires de production, Composition de cellules, Technologie de groupe, Partitionnement de graphe, Méthode d'exploration par séparation et évaluation, Branch and Bound.

Abstract

This thesis belongs to the field of optimization, especially that of combinatorial optimization. The objective of our work is to study the cell formation problem in cellular manufacturing systems.

In the first part of this thesis, we are interested in the importance of the decision variable chosen in the modeling of cell formation problem. We begin with the presentation of three mathematical models and two other models based on graph theory. Next, we examine these different models from the theoretical point of view. The second part deals mainly with the exact resolution of the cell formation problem. The exact approach we choose is based on the branch-and-bound principle. Initially, we present a first version of this method that uses a branching process based on co-cycles. After, we propose a second version of this method that implements an improvement driven by the constraints and uses a new procedure for calculating the lower bound based on the minimum cut research methodology. Various tests are performed on a set of examples in order to show the advantages and limitations of the models and the exact methods we present.

Key words: Cellular manufacturing systems, Cell formation problem, Group technology, Graph partitioning, Branch and bound.

الأطروحة تدرج في مجال التحسين إلى أقصى حد. الهدف من عملنا هو خلية خلوية.

أهمية اختيار متغير صياغة خلية. صيغ رياضية واثنين قائمة على نظرية الرسم البياني لهذه المشكلة ثم ندرس هذه الصيغ بشكل نظري.

يعالج بالتحديد الحل الدقيق خلية. الطريقة الدقيقة المقترحة هي من نوع التقسيم والتقييم. نقدم في البداية، النسخة الأولى من هذه الطريقة التي تستخدم طرق تقسيم مبنية co-cycle. وبعدها النسخة الثانية من هذه الطريقة التي تطبق تحسين موجه بقيود جديد لحساب الحد. مجموعة من الأمثلة من أجل إظهار مزايا و الصيغ و الطرق الدقيقة المقترحة.

: خلوية إنشاء نظم إنتاج خلوية، تكنولوجيا المجموعة تقسيم الرسم البياني، أسلوب التقسيم والتقييم.

Introduction générale

L'environnement actuel des entreprises est caractérisé par des marchés soumis à une forte concurrence et sur lesquels les exigences et les attentes des clients deviennent de plus en plus fortes en termes de qualité, de coût et de délais de mise à disposition. Cette évolution se renforce par le développement rapide de nouvelles technologies de l'information et de la communication qui permettent une relation directe entre les entreprises (*Business to Business*) et entre les entreprises et leurs clients (*Business to Customer*) [LOP01].

Cette thèse s'intéresse aux Systèmes Cellulaires de Production qui ont pu s'adapter à ces nouvelles exigences en appliquant les principes de la technologie de groupe dans les systèmes de production.

La notion de systèmes cellulaires de production représente relativement une nouvelle pensée qui cherche à décomposer le système de production en cellules de sorte à ce que les produits similaires soient fabriqués dans une même cellule, dans le but d'améliorer la productivité des ateliers.

La conception d'un tel système passe par trois étapes essentielles [PRO91]: (1) La composition de cellules. (2) L'agencement intracellulaire. (3) L'agencement intercellulaire. Notre travail concerne la composition de cellules qui constitue l'étape la plus importante dans la conception.

Pendant les cinquante dernières années différents modèles ont été proposés pour formuler le problème de composition de cellules et plusieurs méthodes ont été mises en place pour le résoudre. Cependant, peu de chercheurs ont tenté d'étudier l'impact de la variable de décision choisie sur les méthodes de résolution et de trouver une solution optimale à ce problème.

L'objectif de notre travail est d'apporter à la communauté des chercheurs une étude théorique et pratique sur l'importance de la variable de décision et d'implémenter des méthodes exactes pour la résolution du problème de composition de cellules.

Nous décomposons cette thèse en six principaux chapitres organisés de la manière suivante.

Le premier chapitre décrit de manière plus détaillée le contexte de la thèse. Dans un premier temps, nous présentons une classification des systèmes de production. Dans un second temps, nous décrivons les systèmes cellulaires de production ainsi que les principales étapes de leur conception.

Dans le second chapitre, nous décrivons le problème de composition de cellules et nous détaillons son état de l'art. L'objectif est d'analyser les travaux de la communauté scientifique pour identifier les méthodes utilisées pour sa résolution et leurs carences qui ont guidé nos recherches.

Dans le troisième chapitre, nous présentons d'abord trois modèles mathématiques et deux modèles basés sur la théorie des graphes pour le problème de composition de cellules. Ensuite, nous comparons ces modèles en étudiant certains de leurs aspects théoriques tels que

le nombre de variables et de contraintes et le problème de redondance et de cécité. Les résultats de cette comparaison sont utiles pour orienter le choix du modèle à utiliser.

Le quatrième chapitre est consacré aux méthodes de résolution exactes et en particulier aux méthodes par séparation et évaluation, Branch and Bound en anglais. Nous commençons par la description de ces méthodes. Nous donnons ensuite une idée sur les méthodes de décomposition et les techniques de résolution des problèmes linéaires de grande taille qui permettent de renforcer leur efficacité.

Dans le chapitre cinq, nous présentons les méthodes exactes développées pour la résolution du problème de composition de cellules. Plus précisément, nous décrivons la méthode d'exploration par séparation et évaluation cocyclique et la méthode d'exploration par séparation et évaluation cocyclique guidée par les contraintes.

Le chapitre six est consacré aux expérimentations effectuées pour tester les méthodes cocycliques présentées dans le chapitre précédent et les modèles mathématiques exposés dans le chapitre trois.

Nous terminons cette thèse par une conclusion générale et certaines perspectives de travaux futurs.

Chapitre 1

Les systèmes cellulaires de production

1.1. Introduction

Devant une compétition internationale de plus en plus forte entre les entreprises, l'amélioration de la productivité est un objectif crucial. Celle-ci peut être obtenue par une meilleure utilisation des ressources. En particulier, la productivité peut être fortement affectée par la manière d'organisation des systèmes de production.

Le problème d'organisation des systèmes de production se pose lorsqu'il s'agit d'implanter ou de réimplanter un système de production [HAC09]. La réimplantation s'impose lorsque les caractéristiques de fabrication des produits évoluent de manière significative par une augmentation importante du volume à fabriquer ou par une modification de certaines caractéristiques des produits traités, ce qui entraîne souvent une évolution dans le nombre et/ou dans l'implantation des ressources utilisées.

Dans ce chapitre, nous présentons quelques généralités sur les systèmes de production, leurs types et les critères de choix d'un type de systèmes de production. Ensuite, nous présentons les systèmes cellulaires de production ainsi que leur intérêt par rapport aux autres types de systèmes.

1.2. Les systèmes de production

Un système de production est généralement vu comme l'association d'un ensemble de ressources en interaction pour réaliser une activité de production [TAM08]. La production s'effectue par une succession d'opérations qui exploitent les ressources disponibles afin de transformer les matières premières entrant dans le système de production en produits finis sortant de ce système.

Ce système doit remplir les fonctions suivantes [SOU94]:

1. Transformation de la matière première par usinage, par déformation ou par assemblage;
2. Manutention des produits en les transportant entre les ressources ou vers les tampons de stockage;
3. Stockage nécessaire dans toutes les étapes de la production. En effet, les produits étant différents, leurs temps de fabrication ne seront pas identiques. Le flux ne sera pas pseudo continu. Ce qui engendre la nécessité de mise de stocks tampons dans les points de discontinuité;
4. Contrôle de qualité par la vérification du respect des normes;

5. Pilotage de la production en dirigeant chaque produit vers le poste de travail ou le tampon de stockage qui convient en fonction de son état d'avancement dans la gamme de production, de la disponibilité des ressources, de son niveau de priorité, etc.

1.3. Classification des systèmes de production

En général, il est très difficile de mettre en place une classification exhaustive de toutes les caractéristiques, publiées dans la littérature, concernant un système de production. Nous allons donc présenter une classification qui met le ton sur les caractéristiques qui nous intéressent dans le cadre de cette étude.

Dans ce contexte, la classification présentée par Hachicha [HAC09] permet de préciser la façon dont sont organisées les ressources pour traiter les flux de matières premières.

En se basant sur cette classification, nous pouvons distinguer trois grands types de systèmes de production: systèmes à production unitaire, systèmes de production continue et systèmes de production manufacturière.

1.3.1. Systèmes à production unitaire

Principalement, la production de type unitaire est un cas où l'entreprise mobilise toutes ces ressources pour la réalisation d'un nombre limité de projets, et ce sur une assez longue période. Dans ce type de production, le produit est généralement immobilisé et les ressources sont arrangées pour assurer le bon déroulement des opérations. Par exemple, on peut citer: la construction d'un grand navire d'une grande taille, les grands travaux publics, etc.

1.3.2. Systèmes de production continue

Les systèmes de production continue se caractérisent par une production qui ne s'arrête jamais et qui se fait sur une succession de ressources en ligne par lesquelles tous les produits vont passer. Ce mode d'organisation se trouve dans les industries lourdes de transformation de matières premières (sidérurgie, pétrochimie, certaines industries agro-alimentaires, etc.).

1.3.3. Systèmes de production manufacturière

Trois classes principales de systèmes de production manufacturière peuvent être distinguées:

1.3.3.1. Systèmes de production en ligne « Flow shop »

Ces systèmes sont conçus pour assurer une grande productivité relative à une petite variété de type de produits. Pour ce genre des systèmes, les ressources sont organisées en lignes dans l'ordre des opérations de sorte que chaque ligne fabrique un ou plusieurs produits utilisant les mêmes ressources (voir figure 1.1 et [BLA91] pour une figure similaire). On trouve couramment de telle structure dans la production de masse (industrie automobile) et plus particulièrement au niveau de l'assemblage final [ARN03].

Les avantages d'un tel système sont: la facilité du contrôle, la réduction du temps de fabrication et la possibilité d'utiliser une main d'œuvre non qualifiée. Cependant, les inconvénients suivants ont été relevés: L'arrêt d'une machine provoque le blocage de la ligne de fabrication et la non flexibilité du système. Nollet et al [NOL86] soulèvent un autre inconvénient des systèmes flow shop en le situant par rapport à la motivation des travailleurs et la qualité de leur travail qui s'y trouve affectée.

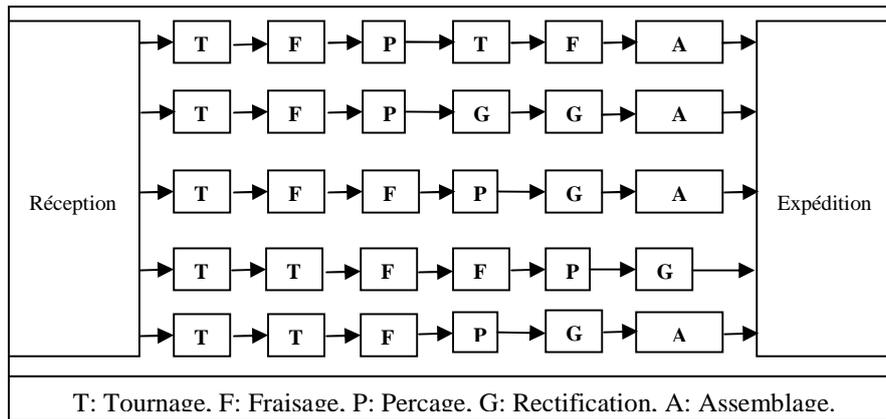


Figure 1.1. Système de production en ligne « flow shop »

1.3.3.2. Systèmes de production en ateliers spécialisés « Job shop »

Ce type de systèmes est conçu pour assurer une petite productivité relative à une grande variété de type de produits. Les machines, qui réalisent les mêmes opérations, sont groupées ensemble pour ce genre des systèmes. Ainsi, les produits passent d'un groupe à un autre pour accomplir les différentes opérations (voir figure 1.2 et [BLA91] pour une figure similaire). Par exemple, pour un atelier de mécanique, les machines de tournage sont réunies ensemble de même que les machines de fraisage, de perçage, etc.

Les avantages de tels systèmes sont: la panne d'une machine ne bloque pas toute la production et la flexibilité. Néanmoins, le transfert des produits entre les groupes des machines conduit à: des en-cours, des délais de production et des coûts de manutention relativement importants et des problèmes complexes d'ordonnancement.

Une étude menée aux Etats-Unis a montré que dans les systèmes job shop, les produits passaient quatre vingt pour cent (80%) de leur temps dans des phases non productives (parcours des trajets et attente dans les stocks tampons) [BOU06a].

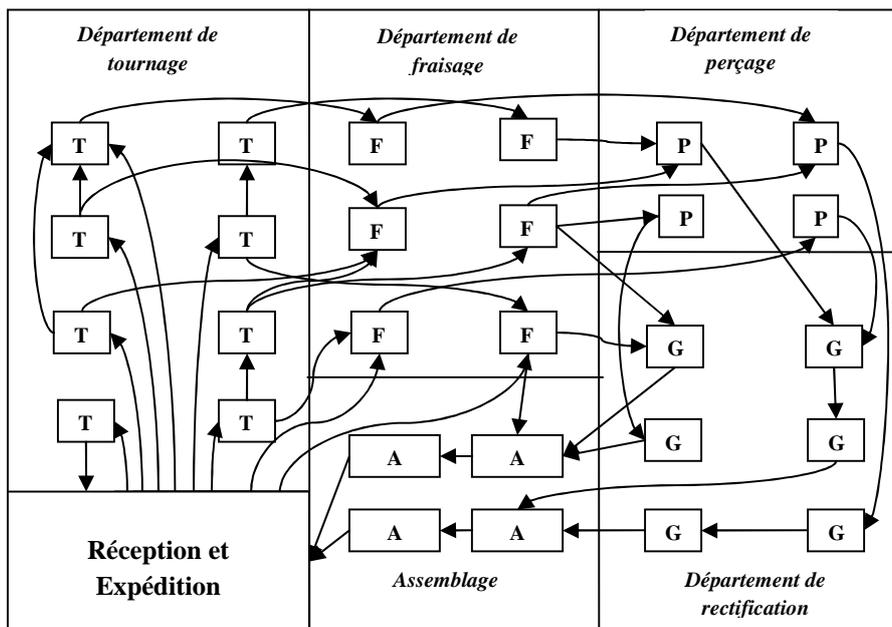


Figure 1.2. Système de production en ateliers spécialisés « Job shop »

Des chercheurs ont eu l'idée d'exploiter les avantages de ces deux types de systèmes de production pour la mise au point d'un troisième: les systèmes cellulaires de production (voir figure 1.3 et [BLA91] pour une figure similaire).

Les systèmes cellulaires de production sont des systèmes hybrides qui associent les avantages d'une production de type "job shop" (flexibilité de produire une grande variété de type de produits) et d'une production de type "flow shop" (mouvement efficace et taux de production élevé) [MUN00]. Dans les systèmes cellulaires de production, les machines sont, en général, proches les unes des autres et participent à la production d'une famille de produits spécifique. Cette organisation particulière permet au système d'avoir un mouvement efficace et un taux élevé de production. Par l'utilisation des machines à usage général, les systèmes cellulaires de production peuvent être facilement reconfigurés pour faciliter la production d'un nouveau type des produits avec un faible effort à la fois en termes de coût et de temps. Cela rend le système de production plus souple en ce qui concerne la production de différentes variétés de type des produits. Ainsi, Un système cellulaire de production peut produire des types des produits dont la variété et le volume sont moyens.

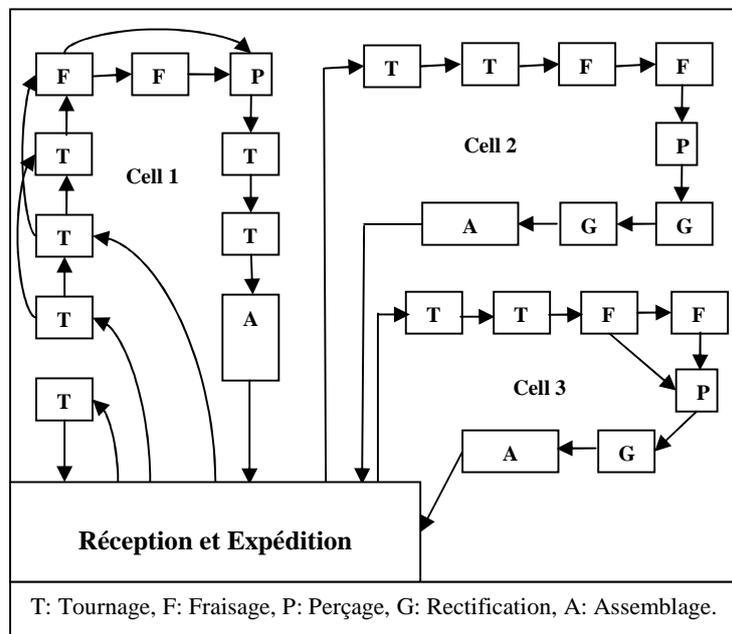


Figure 1.3. Système cellulaire de production

Selon cette classification, nous nous intéressons aux systèmes de production manufacturière, et plus particulièrement aux systèmes cellulaires de production qui seront détaillés dans la section 5.

1.4. Les critères de choix d'un type de systèmes de production

Le problème qui se pose pour l'industriel est le choix d'un des types de systèmes de production que nous avons présentés dans la section précédente.

D'après Campagne [CAM01], le type des systèmes de production manufacturière retenu est essentiellement influé par deux critères: la variété et le volume de la production (voir figure 1.4), où le terme "variété" indique une variété des processus de production et non des produits.

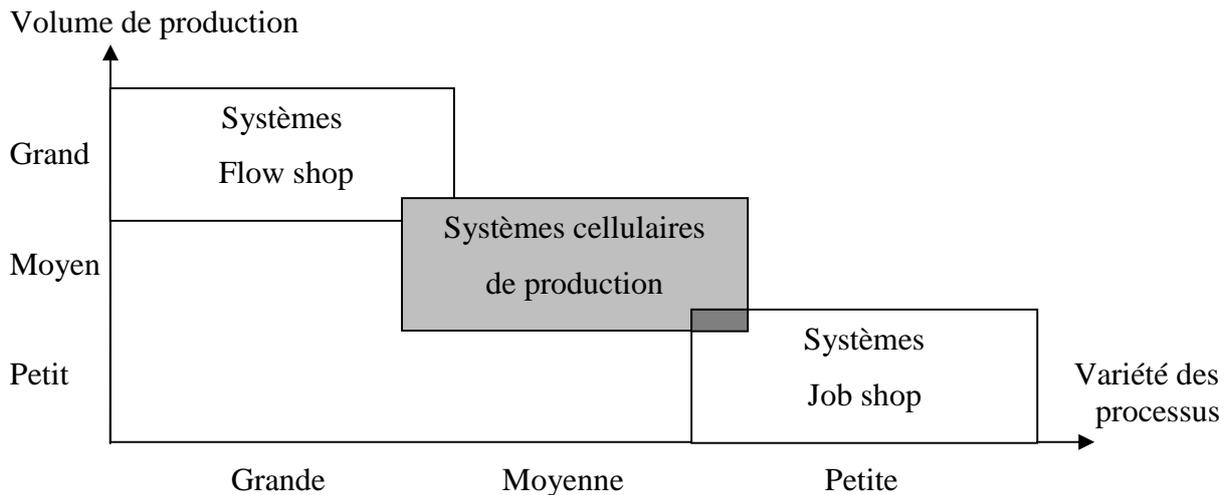


Figure 1.4. Domaine d'application des différents types de systèmes de production [CAM01]

Selon [THO13], Le choix d'un des types de systèmes de production repose sur l'analyse d'un ensemble de critères qui sont: la quantité produite, la nature du processus de production, le mode de gestion de la production et la nature de l'implantation de l'outillage.

a. Les quantités de biens devant être produites

Le type des systèmes variera selon que la production de l'entreprise est unitaire (une seule unité produite à chaque fois) ou concerne des séries plus ou moins importantes (petites, moyennes ou grandes séries). Plus le volume de la production est important, plus la production sera standardisée et reposera sur une structure de production formelle et rigide.

b. La nature du processus de production

Celui-ci peut être de nature continue (la production ne s'arrête jamais et est concentrée en un seul lieu) ou discontinue (la production est fractionnée dans le temps et / ou dans l'espace).

c. Le mode de gestion de la production

La production peut être pilotée soit par l'aval, c'est-à-dire que c'est la commande passée par le client qui déclenche le processus de production, soit par l'amont, c'est-à-dire que le déclenchement de processus de production répond à un cahier des charges prédéfini ce qui peut se traduire par la constitution de stocks de produits finis.

d. La nature de l'implantation de l'outillage

Le processus de production peut reposer soit sur des ateliers spécialisés qui regroupent l'ensemble des postes de travail de même nature, soit par ligne (chaîne de fabrication) où se succèdent les différentes ressources nécessaires à la production d'un bien qui circule d'un bout à l'autre de la chaîne de production, soit sur des ateliers autonomes qui assurent l'ensemble des tâches nécessaires à la réalisation d'une production.

1.5. Les systèmes cellulaires de production

Le professeur Mitrofanov de l'université de Leningrad a introduit en 1946 le concept de "Technologie de groupe (TG)" afin d'établir le lien entre les formes géométriques des produits et les méthodes de fabrication utilisées pour les réaliser [SOU94]. Depuis cette époque, de grands efforts ont été déployés dans ce domaine.

La TG est une philosophie de fabrication, qui consiste à identifier puis regrouper dans des familles les produits similaires afin de tirer avantage de leurs similitudes dans les procédés de fabrication [BUR75].

L'application du principe de la TG dans les systèmes de production a donné naissance aux *Systèmes Cellulaires de Production (SCPs)*.

L'organisation cellulaire permet de diviser le système de production en un certain nombre de cellules correspondant à de véritables "sous-systèmes". Chaque cellule est dédiée à la fabrication d'une famille de produits bien déterminée. La réunion de ces familles donne la totalité de produits que l'entreprise peut fabriquer.

Pour qu'une organisation cellulaire soit performante, toute cellule devrait être indépendante des autres cellules, c'est-à-dire chaque produit de la famille est complètement fabriqué dans sa cellule. Néanmoins, une telle organisation est impossible car les processus de fabrication des produits sont différents. Zolfaghari [ZOL97] a confirmé cela en disant que les mouvements intercellulaires sont souvent inévitables, mais ils peuvent quand même être réduits.

1.5.1. Conception des SCPs

Les différents paramètres à considérer dans la conception d'un SCP provoquent une grande difficulté qui dirige les concepteurs à décomposer le problème en étapes.

Nous proposons dans ce qui suit l'approche adoptée au sein du projet SAGEP [PRO91]. Cette approche vise principalement les ateliers de fabrication cellulaires et se déroule en trois étapes, avec éventuellement une possibilité de retour arrière entre les étapes (voir figure 1.5).

- La première étape consiste à regrouper les machines en cellules et les produits en familles, en optimisant un ou plusieurs critères et en respectant certaines contraintes. Le résultat obtenu à l'issue de cette étape constituera une partie des données de l'étape suivante.
- La deuxième étape réalise l'agencement intracellulaire, c'est-à-dire le choix du moyen de manutention et la disposition des machines à l'intérieur des cellules (construction physique des cellules).
- La troisième et la dernière étape est consacrée à l'agencement des cellules sur le sol de l'atelier.

L'étude de la première étape constitue l'objet de notre travail et sera détaillée dans le reste de cette thèse.

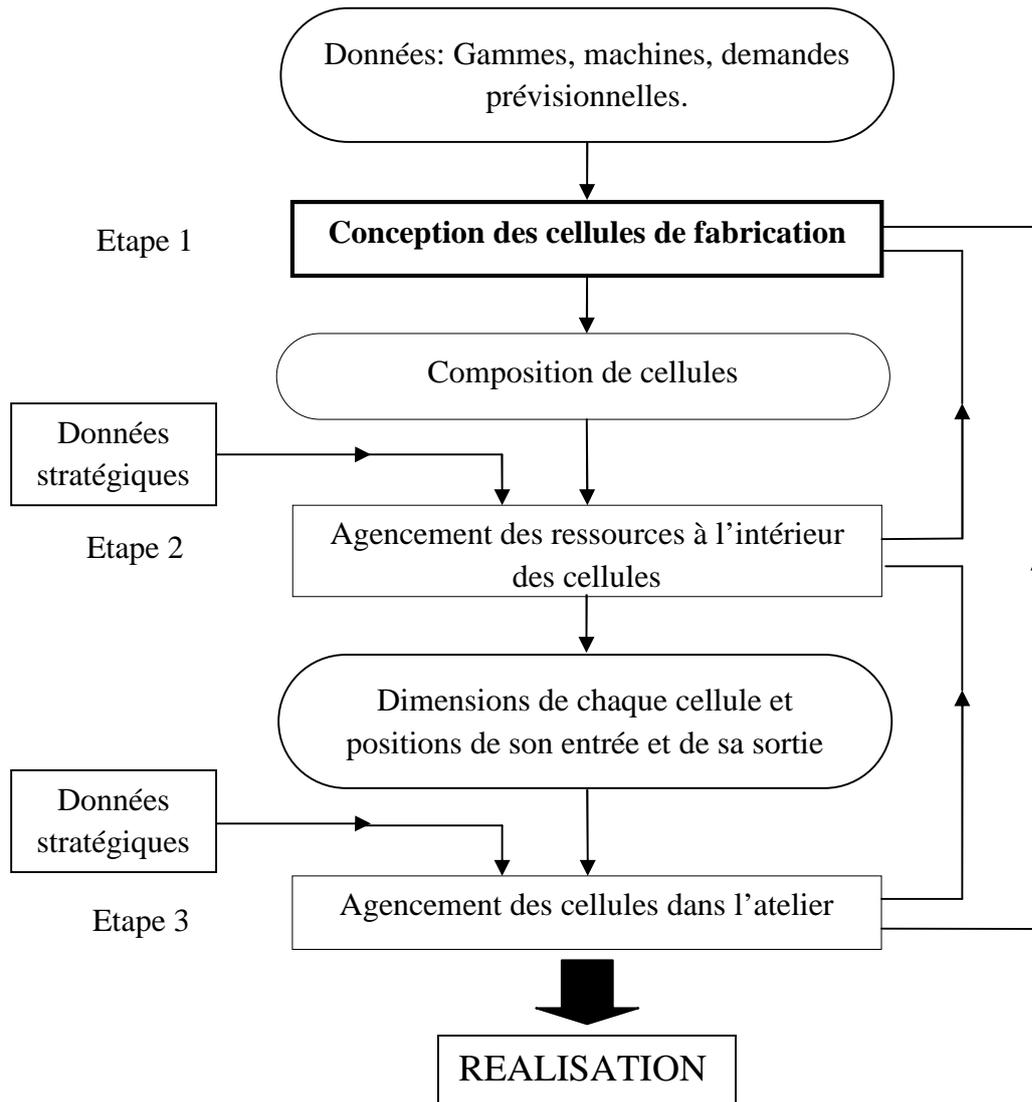


Figure 1.5. L'approche de conception des systèmes cellulaires de production [PRO91]

1.5.2. Avantages et inconvénients des SCPs

Comparés aux systèmes de production traditionnels, les avantages engendrés par des systèmes cellulaires de production sont nombreux. Ces derniers ont été établis grâce à des études analytiques ou des implémentations réelles en industrie. Les avantages les plus cités sont résumés ci-dessous [BER87, SIN96, WEM97, MUN00, BOU02, BOU06a et ATE07]:

- Réduction des temps de réglage des machines: Un système cellulaire est conçu pour traiter des familles de produits ayant des caractéristiques similaires. Cette similitude se caractérise par l'utilisation d'un outillage similaire. Ce qui réduit le recours fréquent au changement d'outils, d'où le gain de temps.
- Réduction des stocks: A cause de la prise en charge plus rapide des produits, on obtient une réduction des tailles des stocks tampons.
- Réduction des en-cours: Les deux facteurs précédents entraînent automatiquement la réduction des produits en cours de fabrication (en-cours).

- Les coûts et les temps de manutention sont réduits: En production cellulaire, les produits sont essentiellement traités dans leurs cellules. Ce qui engendre une réduction des trajets.
- Amélioration de la qualité: Les produits étant subdivisés en familles et affectés aux cellules, les ouvriers (ou opérateurs) d'une même cellule acquièrent une certaine habileté et un professionnalisme par rapport à la famille de produits qu'ils traitent. Si un défaut de fabrication survient, il est aisé d'arrêter le processus de fabrication et de réparer le dysfonctionnement (le contrôle de la qualité est intégré dans la cellule).
- Meilleur contrôle de la production: Dans les systèmes de production Job shops, les produits peuvent circuler dans l'atelier tout entier produisant des cheminements complexes. L'ordonnancement et le contrôle de la production deviennent des tâches compliquées. Dans les systèmes cellulaires, la majorité des trajets est réalisée à l'intérieur des cellules. Ce qui simplifie les deux tâches évoquées.
- La réduction des espaces: La réduction des en-cours et des stocks de produits finis conduit à la réduction de l'espace utile et nécessaire au bon fonctionnement de l'unité industrielle.
- La modulation des processus: Si un segment du processus de fabrication devrait être changé, seule la cellule particulière serait affectée et non pas la ligne de production.
- Les systèmes cellulaires favorisent le travail en équipe, encouragent l'esprit de groupe et permettent une meilleure communication entre les membres d'équipe.
- La réduction de la taille des lots: À la suite de la diminution des temps de réglage des machines, la mise en place d'une séquence de production en petits lots est possible et offre des avantages économiques considérables. De plus, la fabrication par petits lots permet un flux de production uniforme.

D'autre part, les inconvénients retenus sont [BER87]:

- La nécessité d'un personnel qualifié (chaque personne est responsable d'un grand nombre d'opérations).
- Le coût élevé de la mise en œuvre de la production cellulaire, mais les gains obtenus sont si élevés que l'opération peut se rentabiliser en quelques semaines.

Les bénéfices recensés dont nous parlerons ci-dessous sont issus aussi bien à partir d'exemples théoriques et de cas pratiques du secteur manufacturier.

Hyer [HYE84] a rassemblé et recueilli des informations sur une vingtaine de sociétés américaines ayant implanté des cellules de production comme approche manufacturière. Le constat initial a démontré que les SCPs ont apporté des avantages réels et ont dépassé les attentes attendues. En effet, l'auteur a observé une diminution des délais de fabrication et une baisse des en-cours et des coûts de main-d'œuvre. De plus, il a indiqué la facilité de mise en place d'un programme de production et les économies liées aux coûts de manutention.

En France, la société Brown Boveri a créé plusieurs cellules dans ses ateliers. Le travail y est réalisé par des opérateurs polyvalents; chacun d'entre eux effectue consécutivement sur les différentes machines de la cellule plusieurs opérations relatives à une même pièce. Alors que l'on avait auparavant un opérateur par machine, la polyvalence ainsi créée a nettement accru la productivité. Les délais de fabrication des pièces traitées dans les cellules ont été ramenés d'une ou deux semaines à deux ou trois minutes [BER87].

L'organisation des ateliers de fabrication en cellules est très répandue au Japon. La polyvalence des opérateurs, leur passage d'une machine à l'autre dans la même cellule, semble être la règle. C'est le cas dans plusieurs usines de Toyota, Sankyo, Yamaha, Toshiba

et Nukabe. Les cellules de machines sont l'une des principales explications des délais de fabrication très courts et des niveaux de stocks extrêmement faibles des usines japonaises [BER87].

Fry et al [FRY87] ont souligné le fait que plusieurs firmes implantées aux Etats-Unis se sont converties des systèmes classiques de production aux SCPs. Durant la période de cinq années se terminant en 1989, le nombre estimé de SCPs, fonctionnant aux U.S.A, a évolué de cinq cents vingt cinq (525) à plus de huit mille (8.000) firmes. Ce nombre accuse une augmentation continue [CHO92].

Wemmerlöv et Hyer [WEM89] ont rapporté des gains d'application des SCPs dans une étude réalisée sur 32 firmes U.S. Ces 32 firmes ont été choisies de façon à représenter une large variété de lignes de production telles que la machinerie, l'outillage des machines, les équipements d'agriculture et de construction, les équipements hospitaliers et médicaux, les équipements militaires, les pièces détachés et les engins. Le Tableau 1.1 montre les gains rapportés.

Types de Bénéfice	Nombre de Réponses	Pourcentage d'amélioration %		
		Moyen	Minimum	Maximum
1. Réduction des temps de fabrication.	25	45.6	5.0	90.0
2. Réduction des en-cours.	23	41.4	8.0	90.0
3. Réduction du coût de manutention.	26	39.3	10.0	83.0
4. Satisfaction des opérateurs améliorée.	16	34.4	15.0	50.0
5. Réduction des temps de réglage.	23	32.0	2.0	95.0
6. Réduction dans l'espace requis.	9	31.0	1.0	85.0
7. Amélioration de la qualité des produits.	26	29.6	5.0	90.0
8. Réduction du nombre de produits finis.	14	29.2	10.0	75.0
9. Réduction du coût de la main d'œuvre.	15	26.2	5.0	75.0
10. Augmentation de l'utilisation des machines dans les cellules.	6	23.3	10.0	40.0

Tableau 1.1. Gains d'implantation des SCPs dans 32 firmes US [WEM89]

La transformation d'un atelier multi-gammes en SCPs réalisée pour le compte de la compagnie *PMI food Equipement Group* et publiée par Howard et Newman [HOW93] a montré là aussi des avantages indéniables. Entre autres, la diminution des stocks, une amélioration du service à la clientèle et une meilleure communication entre les employés.

Collet et Spicer [COL95] ont démontré de nombreuses améliorations obtenues à la suite de l'implantation d'un SCP dans une petite et moyenne entreprise (PME). Ils ont observé une réduction du temps de fabrication ainsi qu'une diminution de l'aire de travail en raison de l'amélioration du processus de production.

L'analyse de Lévassuer et al [LEV95] a confirmé les gains excessifs obtenus à la suite de la mise en place et de l'exploitation d'un SCP. Ils ont eu, comme valeurs numériques, près de 86% de réduction des délais de fabrication et plus de 80% d'économie réalisée sur le niveau des en-cours de fabrication et des stocks. Ils ont observé également une baisse des taux de rejets entraînant un accroissement de la qualité des produits manufacturés. Comme dans le cas précédent, ils ont observé une diminution de la surface de travail et une réduction des coûts reliés à la main d'œuvre directe.

Wemmerlov et John [WEM97] ont conduit une étude sur les avantages des SCPs recensés dans 46 usines. Dans cette étude, les industries visées étaient celles des produits et des composants électriques et électroniques, les équipements de contrôle, la machinerie et l'outillage, les produits et composants chaud-et-froid et les engins. Le Tableau 1.2 montre les performances recensées.

Mesures de Performance	Nombre de Réponses	% moyen d'amélioration	% minimum d'amélioration	% maximum d'amélioration
1. Réduction des trajets des produits	37	61.3	15.0	99.0
2. Réduction des temps de fabrication	40	61.2	12.5	99.5
3. Réduction des temps de réponse aux commandes	37	50.1	0.0	93.2
4. Réduction en en-cours	40	48.2	10.0	99.7
5. Réduction des temps de réglage	33	44.2	0.0	96.6
6. Réduction dans l'inventaire des produits finis	38	39.3	0.0	100.0
7. Amélioration de la qualité des produits	39	28.4	0.0	62.5
8. Réduction des coûts des produits	38	16.0	0.0	60.0

Tableau 1.2. Les Performances rapportés par l'étude de 46 firmes US [WEM97]

1.6. Conclusion

Les systèmes de production manufacturière ont connu le développement du concept des systèmes cellulaires de production (SCPs), une application de la technologie de groupe (TG). Cette dernière est l'une des approches qui ont connu une large utilisation dans les industries manufacturières. Les SCPs ont prouvé être un moyen efficace pour améliorer la productivité dans les systèmes de production. Leur conception passe par plusieurs étapes parmi lesquelles l'étape de composition de cellules qui constitue le point sur lequel cette thèse est consacrée. Nous commencerons le chapitre suivant par la description de problème de composition de cellules ensuite nous passerons en revue les différentes méthodes appliquées pour sa résolution.

Chapitre 2

Composition de cellules: Etat de l'art

2.1. Introduction

Les systèmes cellulaires de production consistent à décomposer le système de production en cellules de machines, de sorte à ce que les produits similaires, qui formeront une famille, soient fabriqués dans une même cellule. La conception de tels systèmes passe par trois étapes essentielles: (1) La composition (ou la formation) de cellules (FC), (2) L'agencement intracellulaire et (3) l'agencement intercellulaire.

Ce chapitre se compose de deux parties principales. La première décrit le problème FC, qui constitue le point le plus important dans la conception, en mettant l'accent sur les différents objectifs et contraintes utilisés dans la littérature. La deuxième présente un état de l'art des méthodes proposées pour le résoudre.

2.2. Naissance et évolution du problème de formation de cellules

Le principe de la production cellulaire a été développé indépendamment par les entreprises allemandes et américaines [HYE02]. Le livre "*Group production*", qui décrit l'expérience de Daimler avec la production cellulaire dans l'une de ses usines, a été publié en Allemagne dans les années 1922, et Flanders a délivré en 1925 un papier, qui présente comment la production a été standardisée dans une industrie machine-outil au début des années 1920, à la Société Américaine des Ingénieurs Mécaniques.

En 1937, Sokolovski a proposé de regrouper des produits ayant des similitudes quelconques afin qu'ils soient manufacturés dans des processus opérationnels standards [ATE07].

Mitrofanov [MIT66] était le premier à publier le principe de la TG comme suit: "*les choses semblables devraient se faire similairement*". Ce principe se traduit par le regroupement de produits ayant des similitudes de conception en familles de produits afin de les fabriquer ensemble [ASK93].

Malgré le fait que le principe de la production cellulaire ait été pratiqué en Allemagne, France, Suisse et la Russie de 1930 à 1960, la TG fut largement appliquée dans les industries grâce à Burbidge et sa méthode "*Production Flow Analysis (PFA)*" publiée dans [BUR63]. Cette méthode n'était pas basée sur des fondements théoriques rigoureux (du moins dans sa première version) [BOU06a]. En effet, PFA repose sur l'expérience du décideur (celui chargé de la planification de la production).

Jusqu'aux travaux de Burbidge, les méthodes de mise en œuvre de la TG étaient basées sur des techniques d'analyse de données. La résolution du problème FC en utilisant des techniques d'optimisation peut être ramenée aux travaux de McAuley et Purcheck. Le premier a proposé, en 1972, d'utiliser ce que l'auteur appelle des coefficients de similarité [MCA72]. Tandis que le deuxième a publié, en 1975, la première approche basée sur la programmation mathématique [PUR75].

Deux différentes écoles se sont formées depuis cette époque [BOU06a]: l'*école pragmatique*, fidèle aux prescriptions de Burbidge, qui s'articule sur des procédés de conception basés sur l'intuition et le savoir-faire acquis du concepteur, et l'*école optimale* qui se base sur la modélisation du problème afin de le résoudre par les techniques d'optimisation.

2.3. Description du problème FC

Pour donner une description au problème FC, nous ferons référence à la définition donnée par Wu et Salvendy [WU93]:

“ if the number, types, and capacities of production machines, the number and types of parts to be manufactured, and the routing plans and machine standards for each part are known, which machines and their associated parts should be grouped together to form cells?”

Cette définition nous oriente vers les trois principales étapes du processus de FC:

- La formation des familles de produits;
- Le regroupement des machines en cellules de production;
- L'assignation des familles aux cellules.

Cependant, ces trois étapes ne sont pas nécessairement exécutées de manière séquentielle ni dans l'ordre présenté ci-dessus.

Papaioannou et Wilson [PAP10] ont identifié trois stratégies de formation de cellules fondées sur la procédure employée pour former les familles de produits et les cellules de machines:

- D'abord les familles de produits sont formées, ensuite les machines sont regroupées dans des cellules selon les besoins opérationnels de ces familles. Cette stratégie est appelée "*Identification de familles de produits*".
- Les cellules de production sont créées en premier selon la similitude associée au flux opérationnel des produits. Par la suite, les produits sont attribués aux cellules. Cette stratégie est appelée "*Identification de groupes de machines*".
- Les familles de produits et les cellules de production sont formées simultanément. Cette dernière stratégie est appelée "*stratégie de regroupement produits-machines*".

La dernière stratégie de formation englobe, en général, les méthodes utilisant des informations de la matrice d'incidence produits /machines (voir la section 2.4.3.1). Ces méthodes ne prennent pas en compte plusieurs paramètres permettant de définir le problème réel tels que les séquences opératoires, les volumes de production, etc. Le choix entre les deux autres stratégies de formation dépend de l'aspect le plus intéressant au regard du concepteur du système et la taille du problème (i.e. nombre de machines et de produits).

En abordant le problème FC, l'objectif (ou les objectifs) à optimiser et les contraintes à respecter doivent être précisés. Pour mieux choisir ces éléments, nous présentons dans ce qui suit un aperçu sur les objectifs et les contraintes cités dans la littérature.

2.3.1. Les objectifs

Tout au long de l'histoire du problème FC, plusieurs objectifs ont été proposés par différents auteurs. Les premières méthodes de FC, méthodes utilisant la matrice d'incidence produits/machines, ont cherché seulement à minimiser le nombre des éléments exceptionnels (voir la section 2.4.3.1) dans la matrice d'incidence [voir par exemple, ADI96]. Assez rapidement, d'autres objectifs ont été proposés pour répondre aux besoins du monde réel de la production, parmi lesquels nous pouvons citer:

- La minimisation des mouvements intercellulaires [DAH93]: Cet objectif est une généralisation de l'objectif de minimisation des éléments exceptionnels lorsqu'on tient en compte le routage de chaque produit;
- La minimisation des mouvements intracellulaires [GUP95] ;
- La minimisation du coût de duplication de (ou d'investissement en) machines [ASK90] ;
- La maximisation des similarités (ou la minimisation des dissimilarités) entre produits de même famille [GUP90] ;
- La minimisation du trafic intercellulaire [NAG90]: Cet objectif est une généralisation de l'objectif de minimisation des mouvements intercellulaires lorsqu'on tient en compte le volume de production de chaque produit;
- La minimisation des écarts des charges intercellulaires [MAN02];
- L'efficacité de groupement qui permet de minimiser les opérations intercellulaires et maximiser les opérations intracellulaires simultanément [ELB12 et BYC14];
- ...etc.

Pour une liste détaillée des objectifs cités dans la littérature, nous invitons le lecteur à voir l'étude présentée par Mansouri et al [MAN00], la revue de Papaioannou et Wilson [PAP10] et l'analyse critique donnée par Boulif et Atif [BOU10]. Les conclusions dégagées par les auteurs de ces travaux peuvent être résumées comme suit:

- Le survol réalisé par Mansouri et al [MAN00] sur dix-neuf articles lui a permis de conclure que les objectifs les plus utilisés dans la littérature sont: La minimisation du coût de duplication de machines, la minimisation du trafic intercellulaire, la minimisation des mouvements intercellulaires et la minimisation des écarts des charges intracellulaires.
- L'étude faite par Boulif et Atif [BOU10], ajoutant plus d'une trentaine d'articles aux travaux analysés par Mansouri et al [MAN00], a montré que l'objectif de minimisation du trafic intercellulaire, qui est une généralisation de l'objectif de minimisation des mouvements intercellulaires, est l'objectif le plus populaire. D'après Boulif [BOU06a], cette popularité vient du fait que le plus grand gaspillage dans une entreprise est lié à la manutention. En effet, la majorité des bénéfices de la TG, tels que la réduction des trajets, des en-cours, des délais de livraison, etc., sont liés à l'indépendance des cellules. Par conséquent, ils sont perdus si les produits passent la plupart de leur temps à se déplacer entre les cellules.
- D'après la revue effectuée par Papaioannou et Wilson [PAP10] sur cinquante-deux articles, la plupart des formulations proposées pour le problème FC sont des modèles mathématiques avec l'objectif principal de minimisation des mouvements intercellulaires, qui est un cas particulier de l'objectif de minimisation du trafic intercellulaire.

2.3.2. Les contraintes

Il n'y a pas que les objectifs de FC qui devraient être pris en compte lors de la conception d'un système de production. En effet, d'autres données stratégiques dites contraintes devraient

être satisfaites pour qu'une solution au problème FC soit retenue. Selon Boulif et Atif [BOU10], il est possible de diviser les contraintes en quatre types :

1. *Contraintes pratiques*: ce sont des contraintes dont on est obligé d'en tenir compte pour avoir des solutions réalisables. Parmi elles, nous avons :
 - a) La taille des cellules: C'est la contrainte la plus considérée dans la littérature. On entend par « taille des cellules » le nombre de machines que celles-ci peuvent couvrir. Ce nombre doit être modéré (et donc borné) pour ne pas perdre les gains apportés par les SCPs.
 - b) La taille des familles: La taille des familles représente le nombre de produits qui les composent. Il est préférable de modérer cette taille pour maîtriser l'objectif de la qualité.
 - c) Le nombre de cellules: Pour faciliter la gestion de l'atelier, ce nombre peut être limité par deux bornes: une borne supérieure et une borne inférieure. Toutefois, si la contrainte de la taille des cellules est considérée, on peut négliger ces deux bornes car elles existent toujours. En effet, l'utilisation d'une borne supérieure au nombre de machines, pouvant être abritées dans une cellule, induit une contrainte sur le nombre minimum de cellules, égal à l'entier immédiatement supérieur au nombre de machines, divisé par le nombre maximum de machines par cellule et une contrainte sur le nombre maximum de cellules, égal au nombre de machines.
 - d) Cohabitation entre machines: exprime la nécessité de placer certaines machines dans la même cellule. La raison est que, par exemple, ces machines peuvent être dangereuses ou d'une source d'énergie particulière exigeant qu'elles soient placées à une proximité fermée afin de faciliter leur surveillance et de satisfaire leurs besoins énergétiques.
 - e) Non cohabitation entre machines: Certaines machines doivent être éloignées les unes des autres. Par exemple, écarter les machines de haute précision ou bien celles qui génèrent de grandes vibrations.
2. *Contraintes de simplification*: Certaines contraintes ajoutées aux modèles proposés pour le problème FC représentent une supposition simplificatrice qui permet de réduire l'espace de recherche de façon significative. Or, une telle simplification peut conduire à des modèles très éloignés de la réalité. Nous présentons dans ce qui suit quelques contraintes de simplification:
 - a) Fixation de nombre de cellules : une bonne partie des travaux de recherche publiés dans la littérature suppose que le nombre de cellules est fixé au préalable.
 - b) Fixation de la taille des familles : Quelques auteurs considèrent des tailles fixes pour chaque famille au lieu de borner la taille des familles.
3. *Contraintes critères*: Pour les problèmes à objectifs multiples, certains auteurs choisissent un objectif principal et considèrent les autres objectifs comme contraintes.
4. *Contraintes de cohérence*: Elles sont utilisées dans les modèles mathématiques et servent à préserver la cohérence des variables du modèle.

2.4. Les méthodes de formation de cellules : état de l'art

Dans la littérature, il existe un grand nombre de méthodes pour résoudre le problème FC. Ces méthodes ont été classifiées de différentes manières, voir par exemple [SEL98, ALB03, ALK07, HAC09 et PAP10]. Du point de vue général, la classification montrée dans la figure 2.1 peut être adoptée. Cette classification consiste à décrire des caractéristiques remarquables communes à un ensemble de méthodes.

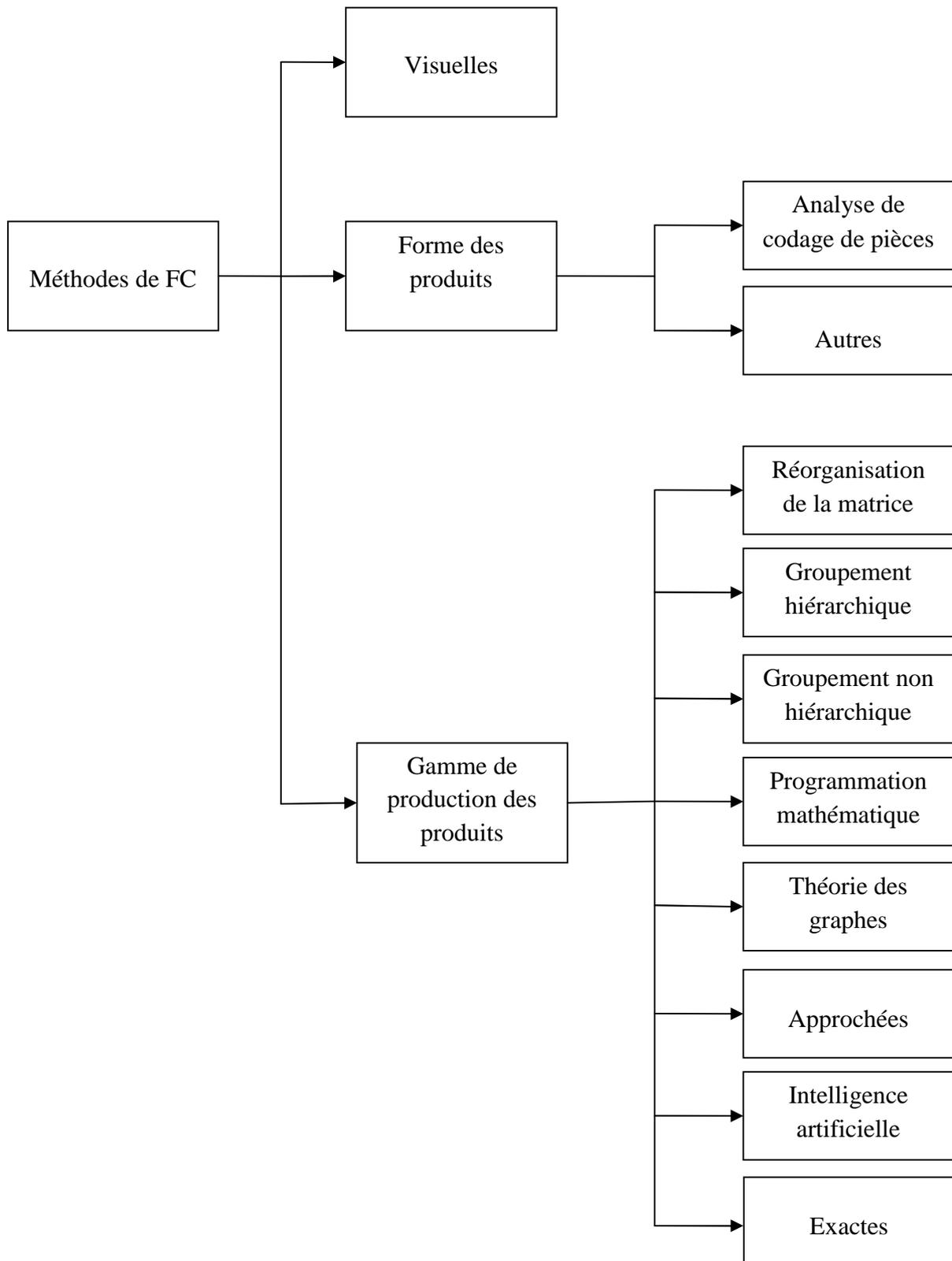


Figure 2.1. Classification des méthodes de formation de cellules

Au plus haut niveau, les méthodes de FC peuvent être classées en : méthode visuelle, méthodes basées sur la forme des produits et méthodes basées sur la gamme de production des produits.

2.4.1. Méthode visuelle

Elle consiste à classer les produits dans des familles et les machines dans des cellules par une simple détection visuelle des caractéristiques physiques des produits. Cette méthode est triviale seulement lorsque le nombre des produits et des machines est petit.

2.4.2. Méthodes basées sur la forme des produits

Les principales méthodes qui constituent cette classe sont les méthodes d'analyse du codage des pièces.

Les méthodes d'analyse du codage des pièces étaient le principal outil de la TG dans les années 1960 et 1970. Ces méthodes utilisent un système de codage qui attribue une valeur numérique dénommée « poids » à l'ensemble des produits pour en faire, par la suite, une classification en familles de produits. La codification est basée sur les attributs de conception et/ou de fabrication du produit. Les produits sont groupés en familles suivant certains schémas de classification.

Le professeur Opitz est considéré comme l'un des innovateurs de ces méthodes. Dans les années soixante, il a proposé un système de codage [OPI70] pour permettre à la TG d'atteindre un stade de perfectionnement avancé. Un survol des méthodes d'analyse de codage des pièces est présenté par Askin et Chiu [ASK90].

L'inconvénient principal de ces méthodes est qu'elles demandent un effort considérable pour pouvoir codifier tous les produits lorsque leur nombre est important [BOU06a].

2.4.3. Méthodes basées sur la gamme de production des produits

Cette classe de méthodes se base sur les gammes de production et exploite la similarité des opérations pour grouper les produits en familles. Ainsi, en utilisant les gammes de production, les produits exigeant le même ensemble d'opérations forment une famille de produits et les machines associées à ces opérations forment un groupe de machines. La plupart des méthodes de FC proposées au cours de dernières décennies appartiennent à cette classe. Les méthodes basées sur la gamme de production des produits peuvent être encore classées en: méthodes de la réorganisation de la matrice, méthodes basées sur un groupement hiérarchique, méthodes basées sur un groupement non hiérarchique, méthodes basées sur la programmation mathématique, méthodes basées sur la théorie de graphe, méthodes approchées et méthodes basées sur l'intelligence artificielle.

2.4.3.1. Méthodes de la réorganisation de la matrice (Array-Based Clustering)

Ces méthodes utilisent une matrice appelée matrice d'incidence produits/machines (MIPM) qui est une matrice booléenne, ayant autant de lignes que de produits, et autant de colonnes que de machines, (voir figure 2.2.a). Un élément quelconque m_{ij} de cette matrice indique si la machine j fait partie du routage du produit i ou non.

Les méthodes de la réorganisation de la matrice d'incidence consistent alors à arranger l'ordre des lignes et des colonnes afin d'obtenir une matrice formée de blocs tout le long de la

diagonale (voir figure 2.2.b). Ces blocs permettent d'identifier les groupes de machines formant les cellules ainsi que les familles de produits qui peuvent être fabriquées par ces cellules. Tous les éléments non nuls à l'extérieur de ces blocs représentent les éléments exceptionnels « *exceptional elements* », tandis que les éléments nuls qui sont situés dans les blocs sont les éléments vides « *void elements* ».

	M_1	M_2	M_3	M_4	M_5	M_6
P_1	1	0	0	0	1	1
P_2	1	1	0	0	1	1
P_3	0	1	1	1	0	0
P_4	1	1	0	1	0	0
P_5	0	1	1	0	0	1
P_6	0	1	0	1	0	0
P_7	0	1	1	0	0	0

a) données initiales.

	M_2	M_4	M_3	M_1	M_5	M_6
P_3	1	1	1	0	0	0
P_6	1	1	0	0	0	0
P_4	1	1	0	1	0	0
P_5	1	0	1	0	0	1
P_7	1	0	1	0	0	0
P_1	0	0	0	1	1	1
P_2	1	0	0	1	1	1

b) Résultat de groupement en cellules.

Figure 2.2. Exemple d'une matrice d'incidence produits/machines [HAC09]

Plusieurs méthodes ont été développées pour la réorganisation de la matrice d'incidence dont notamment les plus connues sont: la méthode BEA (Bond Energy Analysis) proposée par McCormick et al [MCC72], la méthode ROC (Rank Order Clustering) présentée par King [KIN80] et la méthode DCA (Direct Clustering Analysis) développée par Chan et Milner [CHA82]. Une comparaison détaillée de ces trois méthodes est donnée par Chu et Tsai [CHU90].

Les avantages de ces méthodes résident dans la formation simultanée des groupes de machines et des familles de produits et la simplicité de leur mise en application mais, elles présentent plusieurs inconvénients. D'une part, elles ne tiennent pas compte de séquences opératoires de chaque produit (i.e., l'ordre dans lequel les opérations sont effectuées sur chaque produit) et d'autres paramètres de conception cellulaire comme la capacité maximale de chaque cellule et la quantité de produits à usiner. D'autre part, elles nécessitent une intervention et une inspection manuelle pour valider la composition finale des cellules de production [ALB03].

2.4.3.2. Méthodes basées sur un groupement hiérarchique

Ces méthodes opèrent sur un ensemble de données d'entrée décrites en termes d'un coefficient de similarité ou de dissimilitude entre les machines (ou les produits) et produisent une hiérarchie. A chaque niveau de la hiérarchie, il peut y avoir un nombre différent de groupes avec un nombre différent de membres. Contrairement aux méthodes de la réorganisation de la matrice, les méthodes basées sur un groupement hiérarchique ne forment pas les groupes de machines et les familles de produits simultanément. Essentiellement, ces méthodes peuvent être classées en deux [PAP10]: (a) méthodes de division qui commencent avec toutes les données (machines ou produits) dans un seul groupe et créent une série de partitions jusqu'à ce que chaque machine (produit) soit dans un singleton groupe. (b)

Méthodes d'agglomération qui commencent avec des groupes singleton et procèdent à les fusionner en partitions plus grandes jusqu'à ce qu'une partition contenant l'ensemble entier est obtenue. Le regroupement hiérarchique peut être représenté par un dendrogramme [ALK07] qui illustrera la fusion ou la division faite à chaque étape de l'analyse. Un exemple d'un tel dendrogramme est donné par la figure 2.3. Le concepteur doit choisir un seuil de similitude pour définir le nombre de groupes. Tant que le seuil augmente, le nombre de cellules augmente tandis que la taille de chaque cellule diminue.

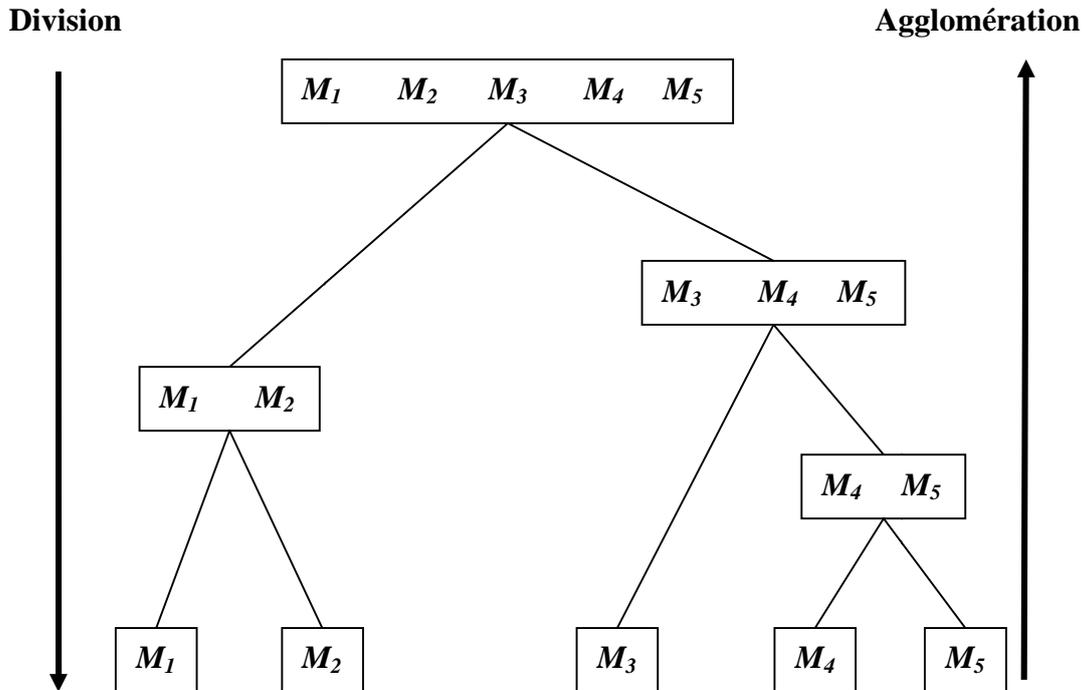


Figure 2.3. Un dendrogramme démontrant le groupement hiérarchique [ALK07]

McAuley [MCA72] est le premier à avoir défini un coefficient de similarité entre deux machines basé sur l'indice de Jaccard [JAC08]. Depuis cette méthode, les coefficients de similarité ont connu beaucoup d'amélioration afin d'introduire dans cette mesure des informations permettant de définir le problème réel (voir la Figure 2.4).

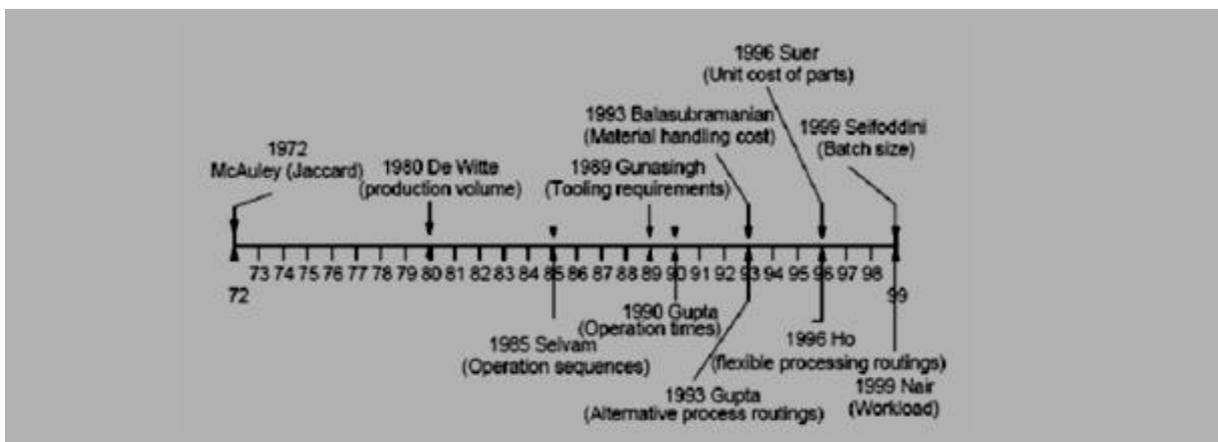


Figure 2.4. L'évolution des coefficients de similarité dans le temps [YIN06]

Quelques travaux de recherche ont exposé une revue de ces coefficients comme par exemple [SIN96, YIN06 et OWS09]. La proposition de plusieurs définitions de coefficients de similarité a rendu les méthodes hiérarchiques très variées. Parmi ces méthodes, en plus de ceux déjà cités, nous trouvons [JAY98, WAN03, ALB05 et JEO06].

Bien que les méthodes basées sur un groupement hiérarchique offrent la possibilité d'intégrer plusieurs paramètres de production, elles présentent un certain nombre d'inconvénients. L'un des inconvénients principaux de ces méthodes est l'utilisation de procédures intuitives. Il est soutenu qu'une méthode basée sur un coefficient de similarité fait des hypothèses fortes qui empêchent parfois des résultats corrects d'être générés [DEW80].

2.4.3.3. Méthodes basées sur un groupement non hiérarchique

Les méthodes basées sur un groupement non hiérarchique sont des méthodes itératives. Elles commencent soit avec une solution initiale générée aléatoirement ou obtenue par un procédé heuristique afin de démarrer avec une bonne solution, soit par le choix de quelques points de semences. En comparaison avec les méthodes basées sur un groupement hiérarchique, ces méthodes permettent à une machine (ou un produit) de modifier l'appartenance au groupe pendant le processus de formation de cellules.

Chandrasekharan et Rajagopalan [CHA86] sont parmi les premiers chercheurs qui ont proposé une méthode non-hiérarchique pour le groupement des produits et des machines. Dans la littérature, cette méthode est notée par « *Ideal Seed Non-hierarchical Clustering (ISNC)* » qu'on peut traduire, en français, par « *la méthode de la graine idéale pour les groupements non hiérarchiques* ». Les auteurs ont utilisé un indicateur de performance appelé efficacité de groupement qui mesure les mouvements intercellulaires et le taux d'utilisation des machines. Pour surmonter l'inconvénient de la fixation du nombre de groupes à obtenir avant de commencer la résolution du problème de formation, la méthode consiste à appliquer en première étape la théorie des graphes. En seconde étape, ils ont proposé de fixer une valeur initiale de k des groupes à former qu'ils ont nommée graine « seed ». En troisième étape, ils ont appliqué une analyse multi variée de la matrice d'incidence.

Un an plus tard, les mêmes auteurs ont proposé le modèle *ZODIAC* « *Zero-One Data Ideal seed Algorithm for Clustering* » [CHA87]. Ce modèle constitue une amélioration du modèle initial *ISNC*. Ils ont ajouté une limite critique pour l'indicateur d'efficacité minimum qu'il faut respecter durant toutes les étapes de classification.

D'autres méthodes non-hiérarchiques trouvées dans la littérature, telles que la méthode *GRAFICS* qui a été proposée par les chercheurs Srinivasan et Narendran [SRI91] et les algorithmes de groupement non-hiérarchique: *CASE* « *Clustering Algorithm for Sequence Data* » et *ACCORD* (*A bicriterion Clustering algorithm for Cell formation using Ordinal and Ratio-level Data*) proposés par Jayakrishnan Nair et Narendran [JAY98 et JAY99].

2.4.3.4. Méthodes basées sur la programmation mathématique

Les méthodes basées sur la programmation mathématique sont largement utilisées dans la conception des systèmes cellulaires de production. Selim et al [SEL98] ont classé ces méthodes en quatre grands groupes en fonction du type de formulation: (a) la Programmation Linéaire (PL), (b) La Programmation Linéaire et Quadratique en nombres entiers (PLQ), (c) La Programmation Dynamique (PD) et (d) La Programmation avec Buts « Goal Programming » (PB). Ces méthodes offrent l'avantage d'intégrer facilement des

caractéristiques spécifiques au système industriel considéré durant le processus de conception telles-que les temps opérationnels, les coûts, les séquences opératoires, l'ensemble des ressources disponibles, etc.

Kusiak [KUS87] a proposé le modèle p-médian comme l'un des premiers modèles mathématiques pour la formation des familles de produits. Le modèle p-médian est utilisé pour regrouper n produits en p familles de produits. La matrice d'incidence et la valeur de p constituent l'entrée du modèle. Le modèle p-médian cherche à maximiser la somme des coefficients de similitude pour un nombre fixe de familles de produits avec la contrainte qu'un produit est affecté à une seule famille seulement et la contrainte qu'un produit peut être attribué seulement à une famille de produits qui a été formée. Plusieurs travaux de recherches, comme [WON00, WAN03, WON04a, WON04b, WON06] et d'autres, ont amélioré le modèle p-médian: soit en modifiant l'expression du coefficient de similarité, soit en améliorant la fonction objectif et/ou les contraintes.

Shtub [SHT89] a montré que le problème FC simple et son extension, le cas où plusieurs routages sont pris en compte pour chaque type de produits, sont équivalents au problème d'affectation général. En se basant sur cette démonstration, Srinivasan et al [SRI90] ont proposé un modèle d'affectation pour le problème de formation des cellules de machines et familles de produits. Ils ont fourni une procédure séquentielle pour identifier les cellules de machines suivie par l'identification des familles de produits. L'objectif du modèle était de maximiser la similitude. La caractéristique principale de cette approche c'est le fait que le nombre de cellules n'est pas fixé a priori. Les auteurs ont affirmé que ce modèle est plus performant que le modèle p-médian en termes de la qualité de la solution et du temps de calcul.

Heragu et Chen [HER98] ont présenté un modèle de programmation non linéaire mixte pour la conception des systèmes cellulaires de production. Le modèle intègre trois aspects essentiels: l'utilisation des ressources, le routage alternatif et les contraintes pratiques. Parce que les modèles de programmation non linéaire sont généralement plus difficiles à résoudre de manière optimale que les modèles de programmation linéaire, les auteurs ont proposé de reformuler le modèle initial à un modèle de programmation linéaire mixte qui a non seulement moins de variables entières par rapport à la plupart des autres modèles de la littérature, mais qui permet l'application de la décomposition de Benders.

Une nouvelle approche mathématique pour former les cellules de production a été proposée par Albadawi et al [ALB05]. Cette approche comporte deux phases. Dans la première phase, les cellules sont identifiées par l'application d'une analyse factorielle à la matrice des coefficients de similarité. Dans la deuxième phase, un modèle de programmation en nombres entiers est utilisé pour attribuer les produits aux cellules identifiées.

Tavakkoli-Moghaddam et al [TAV08] ont présenté une approche de programmation linéaire en nombres entiers pour le problème de formation de cellules dans un environnement dynamique. L'objectif est de minimiser le mouvement entre les cellules et le coût des machines simultanément. Le modèle proposé est résolu par une méthode de recuit simulé.

Arkat et al [ARK12] ont proposé un modèle de programmation non linéaire en nombres entiers pour le problème FC. Puisque la fonction objectif du modèle a une forme non linéaire, une technique efficace de linéarisation a été appliquée.

Krushinsky et Goldengorin [KRU12] ont montré que le problème FC est équivalent au problème de multi-coupes minimales. Ensuite, ils ont présenté deux programmes linéaires mixtes pour le problème de p-coupes minimales (MINpCUT) et montré leur flexibilité en matière d'introduction des nouvelles contraintes.

Elbenani et Ferland [ELB12] sont les premiers chercheurs qui ont proposé la résolution exacte du problème FC avec l'objectif d'efficacité de regroupement en utilisant un algorithme nommé 'Dinkelbach algorithm'. L'efficacité de regroupement permet de maximiser le nombre d'opérations intracellulaires et de minimiser le nombre d'opérations intercellulaires simultanément. La restriction principale de ce modèle était dans la fixation de nombre de cellules a priori. Cette restriction a été éliminée par le modèle proposé par Bychkov et al [BYC14].

La programmation avec but a été appliquée principalement pour la conception de systèmes cellulaires de production multicritères. L'idée générale de cette technique est d'établir un but à atteindre pour chaque critère (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 solution choisie en programmation avec but est celle qui minimise la somme pondérée des écarts en valeur absolue par rapport aux buts. Sankaran et Rodin [SAN90] présentent un modèle de la programmation avec but qui aborde le problème de développement des solutions alternatives par rapport à des objectifs contradictoires et un ordre préférentiel des différents buts. Shafer et Rogers [SHA91] ont présenté trois modèles de la programmation avec but correspondant à trois types de situations. Les critères pris en compte visent à minimiser les temps de configuration, le mouvement intercellulaire et l'investissement dans les nouveaux équipements et à maintenir l'utilisation des machines à un niveau acceptable. Les modèles ont été résolus en utilisant une heuristique.

La programmation dynamique consiste à décomposer un problème en plusieurs sous-problèmes similaires au problème initial mais plus simple à résoudre. La combinaison des solutions des sous-problèmes permet d'avoir la solution optimale du problème global. Steudel et Ballakur [STE87] ont développé une heuristique en deux étapes pour résoudre le problème de la formation de cellules, où la première étape utilise une approche de programmation dynamique pour déterminer la séquence de machines qui maximise la similitude entre les machines. La deuxième étape divise la séquence maximale pour former des cellules respectant la contrainte de la taille des cellules.

2.4.3.5. Méthodes basées sur la théorie des graphes

Ces méthodes peuvent être utilisées pour définir le problème FC dans un nombre de structures à travers un certain nombre d'approches, y compris les approches de partitionnement de graphe, les approches basées sur des graphes bipartis (Bipartite graph), les approches basées sur l'arbre couvrant de poids minimal (Minimum Spanning Tree), et les approches basées sur le réseau de flot (Network Flow).

a) Les approches de partitionnement de graphe

Ces méthodes traitent les machines et/ou les produits comme des nœuds et le processus de fabrication des produits comme des arcs reliés aux nœuds. Ces méthodes visent l'obtention de sous graphes composés des combinaisons machines-machines ou produits-machines permettant d'identifier les cellules de production [SEL98].

Les premiers à mettre en œuvre une approche de partitionnement de graphe au problème FC étaient Rajagopalan et Batra [RAJ75]. Ils ont suggéré l'utilisation des coefficients de similarité de Jaccard [JAC08] et la théorie des graphes pour former des groupes de machines. Chaque sommet du graphe représente une machine, alors que, chaque arête indique la similitude entre les machines. La formation de cellules a été effectuée par l'assemblage de cliques déterminées à partir du graphe de telle sorte que les mouvements intercellulaires soient minimisés. Une contrainte sur la taille des cellules a été imposée par les auteurs afin de spécifier le nombre de machines que celles-ci peuvent abriter.

Faber et Carter [FAB86] ont développé un algorithme basé sur la théorie des graphes pour regrouper les machines et les produits dans des cellules de production. La matrice de similarité entre les machines est représentée par un réseau cluster (Cluster Network) qui est partitionné en cellules par la résolution du problème de flot à coût minimum (minimum cost flow problem).

L'approche développée par Kumar et al [KUM86] est également fondée sur le partitionnement d'un graphe. Elle résout le problème de regroupement des machines en cellules et des produits en familles pour un nombre connu de cellules tout en respectant la contrainte de la taille des cellules.

Askin et Chiu [ASK90] ont proposé une formulation mathématique basée sur les coûts et une heuristique pour le problème FC. La méthode de partitionnement de graphe de Lin-Kernighan a été adaptée et appliquée dans un algorithme de partitionnement en deux phases. La première phase assigne des produits à des machines spécifiques. La deuxième phase groupe des machines dans des cellules.

Spiliopoulos et Sofianopoulou [SPI98] ont proposé une approche basée sur le concept de partitionnement de graphe. La caractéristique principale de cette approche c'est le fait que le nombre de cellules n'est pas fixé a priori, mais il est obtenu dans la solution.

Boulif et Atif [BOU06b] ont proposé une formulation basée sur le partitionnement de graphe pour le problème FC en considérant des contraintes du monde réel. Le modèle proposé est résolu par une combinaison d'un algorithme génétique et une méthode B&B.

b) Les approches basées sur des graphes bipartis

Dans les graphes bipartis, un type de nœud représente les produits et l'autre type présente les machines [SIN96]. Un arc reliant deux nœuds de types différents indique l'utilisation de la machine j pour la fabrication du produit i . Suite à des itérations progressives, on supprime les arcs (les plus faibles) et on obtient k réseaux indépendants. Il est à noter que le paramètre k est équivalent à p dans le modèle p -médian (voir la section 2.4.3.4).

Une nouvelle version plus efficace de l'algorithme *ROC* proposé par King [KIN80] est décrite par King et Nakornchai [KIN82]. Les auteurs ont utilisé un graphe biparti qui exprime les besoins de produits en machines.

Al-qattan [ALQ90] a utilisé l'algorithme de séparation et évaluation «*branch and bound*» pour la composition de cellules. La démarche utilisée se base sur l'analyse d'un graphe biparti.

c) Les approches basées sur l'arbre couvrant de poids minimal

Gower et Ross [GOW69], un algorithme basé sur l'Arbre Couvrant de Poids Minimal (ACPM) est considéré. Comme dans les cas précédents, la formation de cellules peut être structurée dans un graphe pondéré où chaque sommet correspond à une machine et le poids des arêtes représente la dissimilitude entre les machines correspondantes. Supposons, qu'un ACPM se trouve dans un tel graphe. La suppression de $K-1$ arêtes les plus lourdes du ACPM produit K sous-arbres qui peuvent être interprétés comme des cellules de production. Une telle procédure assure que la dissimilitude entre les machines dans une cellule est minimale.

d) Les approches basées sur le réseau de flot

Vohra et al [VOH90] ont présenté un algorithme, qui se base sur une matrice d'incidence avec des durées opératoires, pour la composition de cellules. L'utilisation de réseau de flot est la phase la plus importante de leur algorithme. Ce réseau est utilisé pour la classification simultanée des machines et des produits tout en respectant le critère de minimisation des mouvements intercellulaires.

En tenant compte des séquences opératoires, Wu et Salvendy [WU93] ont modélisé le problème FC par un réseau. Deux algorithmes efficaces sont présentés suivant ce modèle. Le premier algorithme partitionne le réseau en trouvant l'ensemble minimal de coupes. Le deuxième algorithme est une version simplifiée du premier algorithme pour réduire le temps de calcul.

Nsakanda et al [NSA06] ont proposé un modèle mathématique qui intègre plusieurs autres problèmes de fabrication au problème FC. En relaxant ce modèle, les auteurs ont observé que le nouveau modèle a une structure de flot à coût minimum qui s'adapte facilement à l'approche de décomposition de Dantzig-Wolfe. Ainsi, ils ont proposé une approche de résolution qui combine un algorithme génétique et des techniques de résolution des problèmes de grande taille.

2.4.3.6. Méthodes approchées

Les méthodes approchées jouent, aujourd'hui, un rôle primordial dans la résolution des problèmes d'optimisation difficiles. Elles sont regroupées d'une manière générale en deux grandes familles: Les heuristiques réservées à un type de problème particulier et dont l'usage ne peut être étendu à une problématique différente, et les méta-heuristiques fondées sur des principes généraux qui s'adaptent, généralement, à différents types de problèmes. Bien que ces méthodes ne garantissent pas l'optimalité de la solution trouvée, elles sont très utiles dans la production d'une solution acceptable dans un délai raisonnable.

Il existe plusieurs chercheurs qui ont utilisé des heuristiques pour résoudre le problème FC en littérature. Citons par exemple [CHE01 et MUK02]

Deux familles de méta-heuristiques peuvent être distinguées :

- Les méta-heuristiques à solution unique qui sont basées sur le principe de recherche de voisinage. En effet, en partant d'une solution initiale, L'algorithme tente d'améliorer cette solution pas à pas en choisissant une nouvelle solution dans son voisinage. Le recuit simulé [SOU95] et la méthode tabou [LEI06] sont les méta-heuristiques à solution unique utilisées dans la littérature pour résoudre le problème FC.

- Les méta-heuristiques à population qui améliorent au fur et à mesure, un ensemble de solutions, appelé population, et non une seule solution. Les colonies de fourmis, les algorithmes d'optimisation par essaim de particules et les algorithmes génétiques sont des méta-heuristiques à population très répandues dans la littérature. Des exemples d'application de ces méthodes sur le problème FC peuvent être trouvés dans [WU07, MEG08 et SPI08].

La tendance dans cette classe de méthodes concerne le développement des méthodes hybrides, eu égard aux bonnes performances que cette hybridation peut procurer. Parmi les travaux intéressés par cette nouvelle direction, nous citons par exemples [BOU06b, NSA06 et JAM07].

2.4.3.7. Les méthodes basées sur l'intelligence artificielle

Les méthodes basées sur l'intelligence artificielle sont des outils pour résoudre les problèmes compliqués de la vie réelle produisant, en général, des solutions de haute qualité dans un délai raisonnable. Ainsi l'application de ces méthodes sur le problème FC a été prouvée d'être un domaine très prometteur. La logique floue et les réseaux de neurones sont les principales approches de cette classe.

Carpenter et al [CAR91] ont développé la théorie ART1 (Adaptive Resonance Theory) qui se base sur un apprentissage non supervisé. Cet apprentissage permet de décomposer les vecteurs d'entrée en groupes en se basant sur des coefficients de similarité. Plusieurs chercheurs ont proposé des modèles qui se basent sur le principe des réseaux de neurones et en particulier Venkumar et Noorul Haq [VEN05] qui ont amélioré le modèle ART1. Ils ont affirmé que les résultats de formation des cellules sont identiques à ceux obtenus par les algorithmes génétiques.

Xu et Wang [XU89] sont parmi les premiers chercheurs qui ont utilisé la logique floue pour la formation des cellules de production. Ils ont appliqué plusieurs vecteurs qui décrivent les différentes entités des produits. Le nombre de familles à former étant fixe d'avance. Un nouveau type de produit est affecté à l'une des familles les plus proches en utilisant une technique de reconnaissance. Cette méthode n'est applicable que pour des matrices d'incidence binaire. Lozano et al [LOZ02] ont amélioré les résultats de formation et ils ont rendu la méthode capable d'estimer le nombre de cellules à former.

Boulif et Atif [BOU08] ont proposé une approche pour résoudre le problème FC sous son aspect dynamique. Deux stratégies ont été considérées: la stratégie passive et la stratégie active. Pour la première stratégie, les auteurs ont montré qu'elle est équivalente au problème FC dans un système statique. Pour la deuxième stratégie, ils ont proposé une heuristique de plus court chemin, un algorithme génétique et une nouvelle méthode qui combine l'algorithme génétique et la logique floue pour résoudre le problème.

2.4.3.8. Les méthodes exactes

Les méthodes exactes telles que la programmation linéaire en nombres entiers, les procédures de recherche arborescente ou la programmation dynamique, offrent en théorie la garantie de trouver une solution optimale d'un problème et d'en prouver l'optimalité [MAN04].

Dans la littérature, il existe très peu de méthodes exactes traitant le problème FC. La plupart de ces méthodes sont basées sur les techniques énumératives s'appuyant, généralement, sur

des méthodes par séparation et évaluation, Branch & Bound en anglais. Il s'agit d'énumérer d'une manière implicite les solutions et d'en choisir la meilleure parmi toutes. Ces méthodes diffèrent par leurs procédures d'évaluation et de séparation.

Pour plus de détails sur cette classe de méthodes et leur application sur le problème FC, nous invitons le lecteur à voir les chapitres 4 et 5.

2.5. Carences associées aux méthodes de formation de cellules

Au lieu de citer tous les désavantages des méthodes citées plus haut, nous mettons l'accent sur les carences qui ont guidé nos recherches.

D'abord, au mieux de nos connaissances, peu d'études ont tenté de trouver une solution optimale au problème FC. En particulier, presque tous les modèles suggérés ont, au moins, l'un des inconvénients suivants:

- Le modèle est d'une nature intuitive.

En fait, les méthodes utilisant un coefficient de similitude (ou de dissimilitude) vues dans la section 2.4.3.2 sont toutes fondées sur des considérations intuitives et il n'y a pas un raisonnement rigoureux qui explique pourquoi un coefficient est meilleur qu'un autre. Donc, si une telle mesure inexacte est incluse dans un modèle, le modèle entier n'est rien de plus qu'une approximation au problème initial.

- Le modèle est résolu par des procédures heuristiques.

Puisque le problème FC est NP-difficile [BOU06b], plusieurs méthodes approchées ont été développées pour le résoudre. La qualité des solutions obtenues par ces méthodes est difficile à évaluer et il a été seulement démontré qu'elles donnent des résultats satisfaisants pour certains cas particuliers.

De plus, une grande partie des méthodes de résolution n'arrivent pas à évaluer le flux réel entre les cellules car elles s'appuient sur des modèles qui négligent des informations essentielles telles que le volume de production et la séquence d'opérations de chaque produit. C'est le cas de toutes les méthodes basées uniquement sur la matrice d'incidence produits/machines.

Enfin, plusieurs modèles de la littérature considèrent le nombre de cellules comme une entrée. Cette restriction empêche les méthodes de résolution de chercher le nombre idéal de cellules qui correspond le mieux aux hypothèses du problème.

2.6. Conclusion

Nous avons identifié dans ce chapitre le problème de formation de cellules et les différentes décisions devant être prises pour sa résolution. Nous avons ensuite mis en avant plusieurs méthodes pour le résoudre. Toutefois, nous avons constaté certaines limites, qui ont dirigé nos recherches, à l'application de ces méthodes.

Dans le chapitre suivant, nous allons aborder la modélisation du problème FC par la présentation d'un ensemble de modèles qui considèrent des contraintes pratiques, ne nécessitent pas d'information sur le nombre de cellules à former et prennent en considération le volume et la séquence d'opérations pour chaque produit.

Chapitre 3

Modèles pour le problème de formation de cellules

3.1. Introduction

Les nombreuses limites liées à la modélisation évoquées dans le chapitre précédent nous ont conduits à mener des recherches plus approfondies selon cette voie afin d'aboutir à des modèles efficaces pour le problème FC.

Dans ce chapitre, nous insistons, donc, sur un aspect dont l'impact sur la complexité d'algorithmes de résolution a été souvent négligé par les chercheurs. Il s'agit des variables de décision choisies. Dans une première partie, nous positionnons notre problème. Ensuite, nous présentons cinq différents modèles qui ne nécessitent pas d'informations sur le nombre de cellules à former, prennent en considération le volume et la séquence d'opérations pour chaque produit et considèrent des contraintes pratiques. Enfin, nous proposons une étude théorique comparative entre ces différents modèles.

3.2. Position du problème

Pour décrire clairement le problème FC que nous allons résoudre, nous présentons, dans cette section, la stratégie de formation choisie, les hypothèses de résolution, l'objectif à atteindre et les contraintes à respecter.

3.2.1. La stratégie de formation de cellules

Comme nous l'avons vu dans le chapitre 2, le problème FC consiste à regrouper les machines en cellules et les produits en familles. Chaque famille de produits est dédiée à une cellule de machines, de telle sorte que les produits de la même famille soient traités essentiellement dans leur cellule associée.

Lorsque les deux tâches de regroupement (produits et machines) ne sont pas faites simultanément (voir les stratégies de formation de cellules, chapitre 2 section 2.3), il est possible de déduire le regroupement de produits du regroupement de machines, et vice versa. En se basant sur l'observation suivante: « Dans la plupart des problèmes FC, le nombre de produits surpasse le nombre de machines », notre approche s'appuie sur la stratégie " *Identification de groupes de machines* " et se concentre uniquement sur le regroupement des machines en cellules. Le regroupement de produits en familles peut être déduit dans une deuxième étape.

3.2.2. Les hypothèses de résolution

Le système de production est supposé respecter les hypothèses suivantes:

1. Le nombre de produits à fabriquer durant l'horizon de planification est supposé connu et constant;
2. Les ressources nécessaires pour fabriquer ces produits sont connues;
3. Les machines sont disponibles pendant tout l'horizon de planification (zéro pannes);
4. Les machines disponibles permettent de réaliser la totalité de la commande en produits;
5. Chaque produit est fabriqué suivant un unique routage;
6. Le volume de production de chaque produit est connu et constant.

3.2.3. L'objectif

L'objectif de minimisation de trafic intercellulaire étant l'objectif le plus populaire (voir la section 2.3.1). Ainsi, nous présentons dans ce qui suit des modèles tenant compte de cet objectif.

3.2.4. Les Contraintes

Les contraintes à respecter par toute composition sont:

- La taille des cellules est bornée supérieurement;
- Certains couples de machines doivent être placés dans la même cellule;
- Certains couples de machines doivent être séparés dans des cellules différentes.

3.3. Modélisation

La modélisation est, en général, une étape très importante dans le processus de résolution d'un problème. En fait, nous pouvons la voir comme l'écriture formelle d'un problème. Les méthodes de FC utilisent les trois méthodes de modélisation suivantes:

- Les méthodes matricielles;
- Les méthodes mathématiques;
- Les méthodes graphiques.

Les méthodes matricielles consistent à construire la matrice d'incidence produits / machines dont nous avons vu les inconvénients dans le chapitre précédent.

Afin de représenter l'espace de recherche, les deux dernières méthodes s'appuient sur un ensemble de variables de décision. Tout au long de ce chapitre, nous verrons à quel point le choix de ces variables est important. En effet, un mauvais choix donne des modèles qui aggravent la complexité du problème et affaiblissent la capacité des méthodes de résolution, qu'elles soient de nature exacte ou approchée.

Dans ce qui suit, nous introduisons les notations, les données et les paramètres utilisés dans la construction des différents modèles. Ensuite, nous présentons deux classes de modèles pour le problème FC. La première s'appuie sur les méthodes mathématiques tandis que la deuxième sur les méthodes graphiques.

3.3.1. Notations

i, q, u	Indices de machines
k, h	Indices de produits
j, g	Indices de cellules
s	Indice de co-cycles
l	Indice de couples de machines à cohabiter
n	Indice de couples de machines à ne pas cohabiter
m	Nombre de machines
p	Nombre de produits
J	Nombre de cellules formées
N	Nombre maximum de machines par cellule
s_1	Nombre de couples de machines à cohabiter
s_2	Nombre de couples de machines à ne pas cohabiter
a	Nombre désiré de familles de produits
M_i	Machine i
P_k	Produit k
R_k	Routage de produit k
$\langle M_{k,t} \rangle$	La t -ième machine visitée par le produit k
r_k	Le volume de production de produit P_k par unité de temps
SC	Ensemble de couples de machines à cohabiter
SN	Ensemble de couples de machines à ne pas cohabiter
$(i, q)_l$	l -ième couple de machines à cohabiter
$(i, q)_n$	n -ième couple de machines à ne pas cohabiter
G	Le graphe de flux
C_j	Cellule j
v_{kiq}	Le nombre de fois que M_i suit M_q et inversement dans le routage de P_k
e_{iq}	L'arête qui relie la machine M_i à la machine M_q
t_{iq}	Trafic entre la machine M_i et la machine M_q
$W(e_{iq})$	La fonction de pondération des arêtes
CS	Un sous-ensemble de co-cycles de G
ES	Un sous-ensemble d'arêtes intercellulaires de G
w_s	Co-cycle s
\mathcal{O}	Ensemble de co-cycles

3.3.2. Données et paramètres

- (1) Considérons $M \equiv \{M_1, M_2, \dots, M_m\}$ l'ensemble de m machines disponibles et $P = \{P_1, P_2, \dots, P_p\}$ l'ensemble de p types de produits à fabriquer durant l'horizon de planification. Les produits de P sont fabriqués sur les machines de M .
- (2) Pour chaque type de produits $P_k (k = 1, \dots, p)$, on suppose donné:
 - (i) Une séquence unique de machines, dite routage et notée R_k , à être visitées par le produit P_k :

$R_k = (< M_{k, 1} >, \dots, < M_{k, s_k} >)$, où $< M_{k, t} > \in M$ ($t = 1, \dots, s_k$) et s_k est le nombre de machines dans le routage R_k .

(ii) r_k : Volume de production du produit P_k par unité de temps.

(3) Les données d'entrées des contraintes:

(i) Un ensemble de couples de machines SC .

(ii) Un autre ensemble de couples de machines SN , tel que :

$$SN \cap SC = \emptyset .$$

(iii) Un nombre entier N .

(4) Pour chaque triplet $(P_k, M_i, M_q) \in P \times M \times M$, on note par v_{kij} le nombre de fois que M_i suit M_q et inversement ($i, q = 1, \dots, m$) dans le routage de P_k .

(5) Pour chaque couple $(M_i, M_q) \in M \times M$, on note le trafic entre les machines M_i, M_q ($i, q = 1, \dots, m$) par: $t_{iq} = \sum_{k=1}^p r_k \cdot v_{kij}$

3.3.3. Les méthodes mathématiques

Les méthodes mathématiques décrivent un problème en utilisant un langage mathématique. Ces méthodes sont courantes et très utilisées; elles permettent, en général, d'obtenir une écriture simple et puissante. Dans cette section, nous présentons trois différents modèles mathématiques. Chaque modèle formule le problème FC comme un programme linéaire en nombres entiers.

3.3.3.1. Modèle basé sur la relation machine-cellule

Ce modèle emploie la variable de décision x_{ij} définie comme suit:

$$x_{ij} = \begin{cases} 1 & \text{si la machine } i \text{ est affectée à la cellule } j, \\ 0 & \text{sinon.} \end{cases}$$

Cette variable de décision est largement utilisée dans les articles publiés [BOU10]. Parmi ceux-ci, nous avons [ASK90, VAK90, CHE99, ARK12 et KRU12] et la liste est loin d'être exhaustive.

Dans ce qui suit, nous présentons donc le premier modèle mathématique, que nous nommons (Q1) [MER14]:

(Q1)

Fonction objectif

$$\text{Minimiser } \sum_{j=1}^m \sum_{i=1}^{m-1} \sum_{q=i+1}^m x_{ij} \cdot (1 - x_{qj}) \cdot t_{iq} \quad (1)$$

Sous les contraintes

$$\sum_{j=1}^m x_{ij} = 1 \quad i = 1, \dots, m \quad (2)$$

$$\sum_{i=1}^m x_{ij} \leq N \quad j = 1, \dots, m \quad (3)$$

$$x_{ij} - x_{qj} = 0 \quad (i, q)_l \in SC, l = 1, \dots, s_1, j = 1, \dots, m \quad (4)$$

$$x_{ij} + x_{qj} \leq 1 \quad (i, q)_n \in SN, n = 1, \dots, s_2, j = 1, \dots, m \quad (5)$$

$$x_{ij} \in \{0,1\} \quad i = 1, \dots, m, j = 1, \dots, m \quad (6)$$

La fonction objectif (1) minimise le trafic intercellulaire. La première contrainte (2) assure que chaque machine est affectée à une et une seule cellule. La deuxième contrainte (3) garantit que chaque cellule contient au plus N machines. La troisième contrainte (4) assure que chaque couple de machines à cohabiter est affectée à la même cellule. La quatrième contrainte (5) assure que chaque couple de machines à ne pas cohabiter n'est pas attribuée à la même cellule. Enfin, la dernière expression détermine le type de variables utilisées dans le modèle.

Remarque 3.1: Dans ce modèle j prend ses valeurs entre 1 et m (et non pas un nombre donné de cellules) pour permettre la détermination du nombre optimal de cellules qui sera déduit de la solution optimale après avoir éliminé les cellules vides.

3.3.3.2. Modèle basé sur la relation machine-médiane

a. Le problème p -médian(PMP)

PMP est l'un des problèmes de location-allocation les plus connus qui permettent la prise des décisions concernant la localisation des facilités et l'allocation des points de demande à une ou plusieurs facilités [REE06].

Pour: $I = \{1, \dots, b\}$ un ensemble de b points de localisation;

$H = \{1, \dots, d\}$ un ensemble de d clients;

$C = [c_{iq}]$ une matrice $b \times d$ avec des distances (coûts de déplacement);

$R = (M, E)$ un graphe pondéré, où: l'ensemble de sommets $M = I \cup H$ et l'ensemble d'arcs E est défini comme suit:

$$E = \{e_{iq}/(M_i, M_q) \quad I \times H; i = 1, \dots, b; q = 1, \dots, d \text{ et } c_{iq} \neq 0\}$$

Le problème PMP consiste à trouver I_p sommets (les médians, $1 \leq |I_p| \leq |I|$) tel que la distance totale entre les sommets de I_p et les sommets de H qui leur sont associés est

minimale (un sommet de H est associé à un sommet de I_p s'il se trouve proche de ce dernier que tous les autres).

Si $I = H$ alors tous les sommets du graphe R ont la qualité de localisation (voir figure 3.1.).

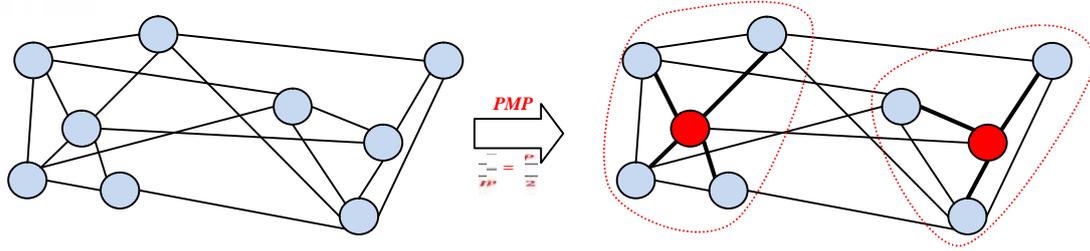


Figure 3.1. Le problème p-médian

Pour plus de détails sur PMP, voir les introductions détaillées proposées par Reese [REE06] et Mladenovic et al [MLA07].

b. PMP appliqué au problème FC

En termes de FC, les sommets représentent les machines et la distance c_{iq} peut représenter: la similitude, la dissimilitude, etc. entre les machines i et q .

Ainsi, L'application de PMP au problème FC consiste à trouver I_p machines qui sont les meilleures représentantes (*médianes*) de I_p cellules de production. Une fois les *médianes* trouvées, les cellules peuvent être produites en attribuant chaque machine restante à l'une des *médianes*, de telle sorte que la distance totale est maximale (ou minimale) selon la distance choisie. À noter que le nombre désiré de cellules (I_p) est, en général, l'un des inputs du modèle et doit être connu à l'avance.

c. Le modèle p-médian originel

Kusiak [KUS87] a proposé le modèle p-médian comme l'un des premiers modèles mathématiques pour la formation de familles de produits. Ce modèle est utilisé pour regrouper p produits en a familles de produits. La matrice d'incidence et la valeur de a constituent les entrées du modèle. La formulation proposée par Kusiak [KUS87] est la suivante:

(P1)

Fonction objectif

$$\text{Maximiser } \sum_{k=1}^p \sum_{h=1}^p s_{kh} x_{kh} \tag{7}$$

Sous les contraintes

$$\sum_{h=1}^p x_{kh} = 1 \quad k = 1, \dots, p \tag{8}$$

$$\sum_{h=1}^p x_{hh} = a \quad (9)$$

$$x_{kh} \leq x_{hh} \quad k, h = 1, \dots, p \quad (10)$$

$$x_{kh} \in \{0, 1\} \quad k, h = 1, \dots, p \quad (11)$$

Où $x_{kh} = 1$ si le produit k est affecté à la famille où le produit h constitue son médian et $x_{kh} = 0$, sinon, a : Nombre de familles de produits désiré et p : nombre total de produits. s_{kh} : Similarité entre le produit k et h définie comme étant le nombre de machines que les deux produits ont en commun.

La fonction objectif du modèle cherche à maximiser la somme des coefficients de similitude. Il est à noter que la contrainte (8) permet de s'assurer que chaque produit est affecté à une et une seule famille. La contrainte (9) sert à limiter le nombre de familles à a . La contrainte (10) permet de s'assurer que chaque produit k est affecté à la famille h seulement si cette famille a été formée. La dernière contrainte (11) assure que la solution sera à composantes binaires.

A l'origine, Kusiak [KUS87] a proposé ce modèle pour la formation de familles de produits, mais avec le même principe, le modèle peut aussi servir à former des cellules.

Depuis que Kusiak [KUS87] a suggéré l'application de PMP au problème FC, plusieurs travaux de recherche ont rapporté des améliorations réussies à ce modèle, comme [DEU98, WON00, WON04a, WON04b et WON06]. L'amélioration principale proposée par Wang et Roze [WAN97] consiste à remplacer la contrainte (10) par la contrainte (12) suivante:

$$\sum_{k=1}^p x_{kh} \leq u_a x_{hh} \quad h = 1, \dots, p \quad (12)$$

Avec u_a : est le nombre maximal de produits par famille.

Cependant, tous ces travaux ont, au moins, l'un des inconvénients suivants:

- La nature intuitive du modèle. En effet, la fonction à optimiser est basée sur la définition d'un coefficient de similitude / dissimilitude.
- Le nombre de familles de produits (ou de cellules) est considéré comme une entrée du modèle et non un paramètre à optimiser.
- Le modèle se base sur la matrice d'incidence produits/machines qui ne prend pas en compte plusieurs paramètres permettant de définir le problème réel.

Le but de la section suivante est d'introduire un modèle p-médian modifié pour le problème FC.

d. Un modèle p-médian modifié

Compte tenu des inconvénients cités ci-dessus, le modèle p-médian proposé par Kusiak [KUS87] et amélioré par Wang et Roze [WAN97] peut être modifié par la réécriture de

l'équation (7), la suppression de la contrainte (9) et l'ajout de deux nouvelles contraintes. En résumé, le modèle p-médian modifié, que nous nommons (Q2), est le suivant:

(Q2)

Fonction objectif

$$\text{Minimiser } \sum_{j=1}^m \sum_{i=1}^{m-1} \sum_{q=i+1}^m x_{ij} \cdot (1 - x_{qj}) \cdot t_{iq} \quad (13)$$

Sous les contraintes

$$\sum_{j=1}^m x_{ij} = 1 \quad i = 1, \dots, m \quad (14)$$

$$\sum_{i=1}^m x_{ij} \leq N \cdot x_{jj} \quad j = 1, \dots, m \quad (15)$$

$$x_{ij} - x_{qj} = 0 \quad (i, q)_l \in SC, l = 1, \dots, s_1, j = 1, \dots, m \quad (16)$$

$$x_{ij} + x_{qj} \leq 1 \quad (i, q)_n \in SN, n = 1, \dots, s_2, j = 1, \dots, m \quad (17)$$

$$x_{ij} \in \{0,1\} \quad i = 1, \dots, m, j = 1, \dots, m \quad (18)$$

Où :

$$x_{ij} = \begin{cases} 1 & \text{si la machine } i \text{ est affectée à la cellule où la machine } j \text{ constitue sa médiane,} \\ 0 & \text{sinon.} \end{cases} \quad i \neq j$$

Et

$$x_{ii} = \begin{cases} 1 & \text{si la machine } i \text{ est une médiane,} \\ 0 & \text{sinon.} \end{cases} \quad i = j$$

La fonction objectif (13) minimise le trafic intercellulaire. La première contrainte (14) assure que chaque machine est affectée à une et une seule cellule. La deuxième contrainte (15) assure l'attribution de N machines au plus à une cellule formée. La troisième contrainte (16) assure que chaque couple de machines à cohabiter est affectée à la même cellule. La quatrième contrainte (17) garantit que chaque couple de machines à ne pas cohabiter n'est pas attribuée à la même cellule. La cinquième contrainte (18) garantit que la solution est binaire.

Remarque 3.2: Comme pour le modèle précédent, l'indice j prend aussi ses valeurs de 1 à m (voir la remarque 3.1). Ceci afin de permettre la détermination du nombre idéal de cellules qui sera donné par le nombre de médianes de la solution optimale.

Afin de diminuer le problème de redondance causé par le choix de la variable de décision (voir la section 3.4.2), nous proposons d'apporter des modifications au modèle ci-dessus. En fait, la contrainte suivante doit être respectée dans le processus d'affectation des machines aux cellules:

Définition 1 (Contrainte-affectation) : Chaque machine ne peut être affectée qu'aux cellules dont l'indice de la médiane est inférieur ou égal à son indice.

Ainsi, la machine 1 peut être affectée exclusivement à la cellule où elle est la médiane; la machine 2 peut être affectée soit à la cellule où la machine 1 est la médiane, soit à la cellule où la machine 2 est la médiane, et ainsi de suite.

Par conséquent, le modèle basé sur la relation machine-médiane, que nous nommons (Q2'), est le suivant:

(Q2')

Fonction objectif

$$\text{Minimiser } \sum_{i=1}^{m-1} \sum_{q=i+1}^m \sum_{j=1}^i x_{ij} \cdot (1 - x_{qj}) \cdot t_{iq} \quad (19)$$

Sous les contraintes

$$\sum_{j=1}^i x_{ij} = 1 \quad i = 1, \dots, m \quad (20)$$

$$\sum_{i=j}^m x_{ij} \leq N \cdot x_{jj} \quad j = 1, \dots, m \quad (21)$$

$$x_{ij} - x_{qj} = 0 \quad (i, q)_l \in SC, l = 1, \dots, s_1, j = 1, \dots, \min(i, q) \quad (22)$$

$$x_{ij} + x_{qj} \leq 1 \quad (i, q)_n \in SN, n = 1, \dots, s_2, j = 1, \dots, \min(i, q) \quad (23)$$

$$x_{ij} \in \{0,1\} \quad i = 1, \dots, m, j = 1, \dots, i \quad (24)$$

3.3.3.3. Une linéarisation efficace

La fonction objectif des modèles (Q1), (Q2) et (Q2') (c'est-à-dire., Equations (1), (13) et (19)) a une forme non linéaire. Ainsi, une technique efficace de linéarisation utilisée par Arkat et al [ARK12] est utilisée pour la transformer en une forme linéaire.

Soit y_{iqj} une variable binaire. L'équation de transformation des modèles (Q1) et (Q2) est la suivante:

$$x_{ij} \cdot (1 - x_{qj}) = y_{iqj} \quad 1 \quad i < q \leq m, j = 1, \dots, m \quad (25)$$

Tandis que celle du modèle (Q2') est comme suit:

$$x_{ij} \cdot (1 - x_{qj}) = y_{iqj} \quad 1 \quad i < q \leq m, j = 1, \dots, i \quad (26)$$

Les équations ci-dessus signifient que y_{iqj} est égale à un si la machine i est affectée à la cellule j alors que la machine q ne l'est pas; sinon elle est égale à zéro. Les contraintes suivantes sont ajoutées aux contraintes originelles des modèles ((Q1) et (Q2)) et (Q2') respectivement pour appliquer les équations de transformation (c'est-à-dire., Equations (25) et (26)):

$$x_{ij} + (1 - x_{qj}) - y_{iqj} \leq 1 \quad 1 \quad i < q \leq m, j = 1, \dots, m \quad (27)$$

$$x_{ij} + (1 - x_{qj}) - y_{iqj} \leq 1 \quad 1 \quad i < q \leq m, j = 1, \dots, i \quad (28)$$

Les contraintes (27) et (28) garantissent que si la machine i est affectée à la cellule j mais la machine q ne l'est pas (c'est-à-dire., $x_{ij} = 1$ et $x_{qj} = 0$) alors la nouvelle variable correspondante y_{iqj} prend la valeur un. Dans les autres cas (c'est-à-dire., $x_{ij} = 0$ et / ou $x_{qj} = 1$), la variable y_{iqj} prend la valeur zéro. Compte tenu de la technique de linéarisation susmentionnée, les suivants modèles linéaires équivalents aux modèles (Q1), (Q2) et (Q2') respectivement sont obtenus:

<p>(L1)</p> <p><i>Fonction objectif</i></p> $\text{Minimiser } \sum_{j=1}^m \sum_{i=1}^{m-1} \sum_{q=i+1}^m y_{iqj} \cdot t_{iq}$ <p><i>Sous les contraintes</i></p> <p>(2)-(6), la contrainte (27) et la contrainte suivante:</p> $y_{iqj} \in \{0,1\} \quad 1 \quad i < q \leq m, j = 1, \dots, m$	<p>(L2)</p> <p><i>Fonction objectif</i></p> $\text{Minimiser } \sum_{j=1}^m \sum_{i=1}^{m-1} \sum_{q=i+1}^m y_{iqj} \cdot t_{iq}$ <p><i>Sous les contraintes</i></p> <p>(14)-(18), la contrainte (27) et la contrainte suivante:</p> $y_{iqj} \in \{0,1\} \quad 1 \quad i < q \leq m, j = 1, \dots, m$
<p>(L2')</p> <p><i>Fonction objectif</i></p> $\text{Minimiser } \sum_{i=1}^{m-1} \sum_{q=i+1}^m \sum_{j=1}^i y_{iqj} \cdot t_{iq}$ <p><i>Sous les contraintes</i></p> <p>(2⁰)-(2⁴), la contrainte (28) et la contrainte suivante:</p> $y_{iqj} \in \{0,1\} \quad 1 \quad i < q \leq m, j = 1, \dots, i$	

3.3.3.4. Modèle basé sur la relation machine-machine

Sofianopoulou [SOF97] a proposé d'utiliser une nouvelle variable de décision qui indique si pour tout couple de machines, celles-ci sont ou ne sont pas dans la même cellule peu importe son indice. Ainsi,

$$x_{iq} = \begin{cases} 1 & \text{si la machine } i \text{ et la machine } q \text{ sont affectées à la même cellule,} \\ 0 & \text{sinon.} \end{cases}$$

Contrairement à la variable décrite dans la section 3.3.3.1, cette variable a été utilisée, directement ou indirectement, par très peu d'auteurs. Citons par exemple: Deutsch et al [DEU98] et Krushinsky et Goldengorin [KRU12].

Le modèle proposé par Sofianopoulou [SOF97], que nous nommons (L3), est le suivant:

(L3)

Fonction objectif

$$\text{Minimiser } \sum_{i=1}^{m-1} \sum_{q=i+1}^m t_{iq} \cdot (1 - x_{iq}) \quad (29)$$

Sous les contraintes suivantes

$$\sum_{i=1}^{t-1} x_{it} + \sum_{q=t+1}^N x_{tq} \leq N - 1 \quad t = 1, \dots, m \quad (30)$$

$$x_{iq} + x_{it} - x_{qt} \leq 1$$

$$x_{iq} - x_{it} + x_{qt} \leq 1 \quad i = 1, \dots, m - 2, q = i + 1, \dots, m - 1, t = q + 1, \dots, m \quad (31)$$

$$-x_{iq} + x_{it} + x_{qt} \leq 1$$

$$x_{iq} = 1 \quad (i, q)_l \in SC, l = 1, \dots, s_1 \quad (32)$$

$$x_{iq} = 0 \quad (i, q)_n \in \bar{S}N, n = 1, \dots, s_2 \quad (33)$$

$$x_{iq} \in \{0,1\} \quad i = 1, \dots, m - 1, q = i + 1, \dots, m \quad (34)$$

La fonction objectif (29) minimise le trafic intercellulaire. La première contrainte (30) nécessite que la taille maximale de chaque cellule soit N . Les contraintes (31) sont imposées pour préserver la structure triangulaire de la formulation ci-dessus. Elles assurent que les machines q et t sont regroupées ensemble ($x_{qt} = 1$) si et seulement si les machines i et q ainsi que les machines i et t sont les deux affectées à la même cellule ($x_{iq} = 1$ et $x_{it} =$

1 respectivement). La troisième contrainte (32) garantit que chaque couple de machines à cohabiter est affectée à la même cellule. Le quatrième (33) assure que chaque couple de machines à ne pas cohabiter n'est pas attribuée à la même cellule. La cinquième contrainte (34) garantit que la solution est binaire.

Noter que les contraintes de cohabitation et de non cohabitation sont ajoutées par Sofianopoulou [SOF99].

Remarque 3.3: La première contrainte (30) est reprise fidèlement de l'article [SOF97] dont l'auteur ne précise pas son explication de façon détaillée. Nous estimons que cette formule aurait été mieux formulée ainsi: $\sum_{i=1}^{t-1} x_{it} + \sum_{q=t+1}^m x_{tq} \leq N - 1 \quad t = 1, \dots, m - 1$

3.3.4. Les méthodes graphiques

Les premiers à avoir modélisé le problème FC en utilisant un graphe constitué par:

- Des nœuds représentant les machines;
- Des arêtes indiquant la similitude entre les machines.

sont Rajagopalan et Batra [RAJ75].

Boulif et Atif [BOU06b] ont été parmi les auteurs qui ont utilisé la même représentation, mais avec une définition différente de l'ensemble des arêtes. En effet, ils ont proposé le modèle suivant:

- (1) On définit le *graphe de flux* non orienté $G = (M, E)$, pour lequel l'ensemble des sommets M est l'ensemble des machines et l'ensemble des arêtes E est l'ensemble des couples non ordonnés des machines reliées par un flux positif:

$$E = \{e_{iq}/(M_i, M_q) \mid M \times M; i, q = 1, \dots, m; i \neq q \text{ et } t_{iq} \neq 0\} \quad SN \cup SC$$

- (2) Une fonction de pondération des arêtes, W , est d'autre part définie comme suit:

$$W(e_{iq}) = t_{iq} \quad i \text{ et } q \in \{1, \dots, m\}$$

L'exemple suivant est utilisé à titre illustratif. Soit un atelier de quatre produits et cinq machines. Les séquences d'opérations et le volume de production de chaque produit sont définis dans le tableau 3.1 suivant:

Numéro de produit	Routing	Volume
1	M ₁ M ₂ M ₃ M ₄ M ₅	1
2	M ₁ M ₂ M ₃ M ₄ M ₅	3
3	M ₁ M ₂ M ₃ M ₄ M ₅	2
4	M ₁ M ₂ M ₃ M ₄ M ₅	1

Tableau 3.1. Les données de l'exemple

La figure 3.2 illustre le graphe de flux, $G = (M, E)$, associé à cet exemple.

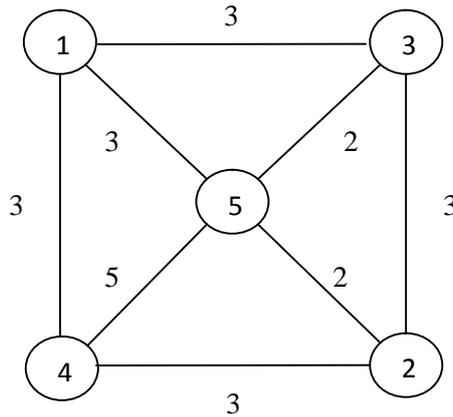


Figure 3.2. Une instance de graphe de flux

Dans le reste de cette section, nous présentons deux différents modèles basés sur ce même concept graphique.

Remarque 3.4: Nous supposons, dans ce qui suit, que le graphe de flux est connecté. Pour cela, nous ajoutons des arêtes fictives avec des poids nuls à tout graphe non connecté.

Remarque 3.5: Si les machines, qui doivent ou ne doivent pas cohabiter, ne sont pas reliées par une arête, elles doivent être reliées en ajoutant des arêtes fictives avec des poids nuls. Ainsi, E est étendu à $(E \cup SC \cup SN)$.

3.3.4.1. Modèle basé sur les arêtes

En se basant sur le graphe $G = (M, E)$ décrit ci-dessus et le modèle présenté par Boulif et Atif [BOU06b], nous proposons le modèle (T1) suivant:

(T1)

Variable de décision

Nous associons à chaque arête $e_{iq} \in E, i$ et $q \in \{1, 2, \dots, m\}$, une variable de décision x_{iq} définie comme suit:

$$x_{iq} = \begin{cases} 1 & \text{si } e_{iq} \text{ est une arête intercellulaire,} \\ 0 & \text{sinon.} \end{cases}$$

Traitements intermédiaires

(1) Soit $ES = \{e_{iq} \in E / (M_i, M_q) \quad C_j \times C_g; j \text{ et } g \in \{1, \dots, J\}; g \neq j; i \text{ et } q \in \{1, 2, \dots, m\}\}$ un sous-ensemble des arêtes de G . En fait, ES contient les arêtes associées à des variables mises à un.

(2) Soit $C = \{C_1, C_2, \dots, C_J\}$ un ensemble de composantes connexes du graphe G :

$$C = (M, E - ES)$$

C est une partition de $M \in J$ cellules, c'est-à-dire :

$$C_j \neq \emptyset, \forall j \in \{1, \dots, J\}; \bigcup_{j=1}^J C_j = M \text{ et } C_j \cap C_g = \emptyset, \forall j, g \in \{1, \dots, J\}, j \neq g$$

(3) Le trafic intercellulaire associé au sous-ensemble ES est donné par:

$$T(ES) = \sum_{e_{iq} \in ES} W(e_{iq})$$

Contraintes

Pour avoir une solution réalisable, ES doit respecter les contraintes suivantes:

1- Le nombre maximum de machines permis dans une cellule, N : nous considérons seulement les sous-ensembles des arêtes, ES , dont la partition C respecte:

$$C_j \in C (j = 1, \dots, J); \text{Card}(C_j) \leq N$$

Où $\text{Card}(C_j)$ est le cardinal de C_j .

2- Etant donné un ensemble de couples de machines à cohabiter (placées dans la même cellule), SC , il faut que:

$$(M_i, M_q) \in SC, \exists C_j \in C: M_i, M_q \in C_j \text{ où } i, q \in \{1, \dots, m\} \text{ et } j \in \{1, \dots, J\}$$

3- Etant donné un ensemble de couples de machines à ne pas cohabiter (séparées dans des cellules différentes), SN , il faut que:

$$(M_i, M_q) \in SN, \exists C_j, C_g \in C: M_i \in C_j \text{ et } M_q \in C_g \text{ où } i, q \in \{1, \dots, m\}, j, g \in \{1, \dots, J\} \text{ et } j \neq g$$

Fonction objectif

Soit S l'ensemble des sous-ensembles d'arêtes qui respectent les contraintes précédentes. Le problème est de trouver la solution $ES^* \in S$, telle que:

$$Z(ES^*) = \min_{ES \in S} T(ES)$$

C'est-à-dire qu'on cherche un sous-ensemble d'arêtes qui respecte toutes les contraintes et qui a la quantité minimale de trafic intercellulaire.

3.3.4.2. Modèle basé sur les co-cycles

Dans cette section, nous présentons un nouveau modèle, que nous nommons ($T2$), pour le problème FC. Ce modèle s'appuie sur la notion de co-cycle, discutée en détail dans L'annexe B.

En se basant sur le graphe $G = (M, E)$ décrit ci-dessus, le modèle ($T2$) proposé est le suivant [MER14]:

(T2)

Variables de décision

Soit $\Omega = \{w_1, w_2, \dots, w_{|\Omega|}\}$ l'ensemble de $|\Omega|$ co-cycles liés au graphe de flux G (dans un graphe connexe de m sommets, $|\Omega|$ est égal à $2^{n-1} - 1$, voir l'annexe B). Nous associons à chaque co-cycle $w_s, s \in \{1, 2, \dots, |\Omega|\}$, une variable de décision x_s définie comme suit:

$$x_s = \begin{cases} 1 & \text{si } w_s \text{ est utilisé pour construire la solution courante,} \\ 0 & \text{sinon.} \end{cases}$$

Traitements intermédiaires

(1) Soit $CS = \{w_1, w_2, \dots, w_{|CS|}\}$ un sous-ensemble de co-cycles de G tel que:

$$\left\lceil \frac{m}{N} \right\rceil \leq |CS| < m$$

Où $\lceil x \rceil$ est la partie entière du nombre x et $|X|$ est le cardinal de l'ensemble X .

En fait, CS contient les co-cycles associés à des variables mises à un.

(2) Soit $C = \{C_1, C_2, \dots, C_J\}$ un ensemble de composantes connexes du graphe G :

$$C = (M, E - \bigcup_{s=1}^{|CS|} w_s)$$

C est une partition de M en J cellules, c'est-à-dire:

$$C_j \neq \emptyset, \forall j \in \{1, \dots, J\}; \bigcup_{j=1}^J C_j = M \text{ et } C_j \cap C_g = \emptyset, \quad j, g \in \{1, \dots, J\}, j \neq g$$

(3) On définit $E(CS)$ comme étant l'ensemble des arêtes intercellulaires associées au sous-ensemble CS :

$$E(CS) = \bigcup_{s=1}^{|CS|} w_s$$

(4) Le trafic intercellulaire total associé au sous-ensemble CS est donné par:

$$T(CS) = \sum_{e_{iq} \in E(CS)} W(e_{iq})$$

Contraintes

Pour avoir une solution réalisable, CS doit respecter les contraintes suivantes :

- 1- Le nombre maximum de machines permis dans une cellule, N_C : nous considérons seulement les sous-ensembles de co-cycles, CS , dont la partition C respecte :

2- Contrainte de cohabitation:

$$(M_i, M_q) \in SC, \forall w_s \in CS: e_{iq} \notin w_s, \text{ où } i, q \in \{1, \dots, m\} \text{ et } s \in \{1, \dots, |CS|\}$$

3- Contrainte de non-cohabitation:

$$(M_i, M_q) \in SN, \exists w_s \in CS: e_{iq} \in w_s, \text{ où } i, q \in \{1, \dots, m\} \text{ et } s \in \{1, \dots, |CS|\}$$

Fonction objectif

Soit S l'ensemble de sous-ensembles de co-cycles qui respectent les contraintes précédentes. Le problème est de trouver la solution $CS^* \in S$, telle que:

$$Z(CS^*) = \min_{CS \in S} T(CS)$$

C'est-à-dire qu'on cherche un sous-ensemble de co-cycles qui respecte toutes les contraintes et qui a la quantité minimale de trafic intercellulaire.

3.4. Comparaison théorique des modèles

Dans cette section, nous proposons une comparaison théorique des différents modèles présentés dans ce chapitre pour le problème FC. Les modèles seront comparés et analysés en termes de nombre de variables et de contraintes, problème de redondance et problème de cécité.

3.4.1. Comparaison en termes de nombre de variables et de contraintes

- Les modèles (L1) et (L2)

Les modèles (Q1) et (Q2) contiennent m^2 variables binaires et $(s_1 + s_2 + 2) \cdot m$ contraintes. Une technique de linéarisation a été utilisée pour transformer leur fonction objectif en forme linéaire (voir la section 3.3.3.3). Cette technique utilise $C_m^2 \cdot m^{(1)}$ variables et $C_m^2 \cdot m$ contraintes. Par conséquent, les modèles (L1) et (L2) obtenus contiennent un total de $(m + C_m^2) \cdot m$ variables binaires et $(C_m^2 + s_1 + s_2 + 2) \cdot m$ contraintes.

- Le modèle (L2')

Pour le modèle (Q2'), le nombre de variables est égal au nombre de combinaisons de m machines 2 à 2 (C_m^2) plus le nombre de médianes possibles (m), tandis que le nombre de contraintes ne dépasse pas $(2 + s_1 + s_2) \cdot m - (s_1 + s_2)$ contraintes.

La transformation de la fonction objectif du modèle (Q2') en une forme linéaire nécessite Nb variables et Nb contraintes. Ce nombre peut être calculé selon la formule suivante:

$$Nb = \sum_{i=1}^{m-1} i \cdot m - i^2$$

(1) C_m^n : Combinaison de m éléments pris n à n

Ainsi, le modèle (L2') obtenu contient un total de $C_m^2 + m + Nb$ variables et un nombre qui ne dépasse pas $(2 + s_1 + s_2).m - (s_1 + s_2) + Nb$ contraintes.

- Le modèle (L3)

Le modèle (L3) contient C_m^2 variables binaires et $3 C_m^3 + m + s_1 + s_2$ contraintes.

- Le modèle (T1)

Le nombre de variables du modèle (T1) est inférieur ou égal à $\frac{n(m-1)}{2}$ variables. En effet, un graphe complet à m sommets possède $\frac{n(m-1)}{2}$ arêtes. En plus, le modèle (T1) se base sur un graphe G connecté mais il n'est pas obligatoirement complet. Ainsi, Le nombre d'arêtes du graphe G est égal au plus à celui d'un graphe complet.

- Le modèle (T2)

Le dernier modèle fait référence à la notion de co-cycle (voir l'annexe B). Par conséquent, le modèle (T2) contient $2^{n-1} - 1$ variables.

Le tableau 3.2 résume le nombre total de variables et de contraintes requises par chacun des modèles présentés dans ce chapitre.

		Nombre de variables	Nombre de contraintes
M O D È L E S	(L1)	$(m + C_m^2) \cdot m$	$(C_m^2 + s_1 + s_2 + 2) \cdot m$
	(L2)	$(m + C_m^2) \cdot m$	$(C_m^2 + s_1 + s_2 + 2) \cdot m$
	(L2')	$C_m^2 + m + Nb$	$(2 + s_1 + s_2) \cdot m - (s_1 + s_2) + Nb$
	(L3)	C_m^2	$3 C_m^3 + m + s_1 + s_2$
	(T1)	$\frac{m(m-1)}{2}$	Non applicable (voir la remarque 3.7)
	(T2)	$2^{n-1} - 1$	Non applicable (voir la remarque 3.7)

Tableau 3.2. Nombre de variables et de contraintes requises par les différents modèles

Pour une illustration de cette comparaison, considérons une instance du problème FC avec 10 machines et prenons $s_1 = 0$ et $s_2 = 0$ (voir le tableau 3.3).

		<i>Nombre de variables</i>	<i>Nombre de contraintes</i>
M O D È L E S	<i>(L1)</i>	550	470
	<i>(L2)</i>	550	470
	<i>(L2')</i>	220	185
	<i>(L3)</i>	45	370
	<i>(T1)</i>	45	/
	<i>(T2)</i>	511	/

Tableau 3.3. Exemple de nombre de variables et de contraintes requises par les différents modèles

Par conséquent, les résultats suivants sont obtenus:

1. Le modèle *(T1)*, suivi immédiatement par le modèle *(L3)*, requiert moins de variables binaires car :
Pour $m \leq 10$, nous avons :

$$C_m^2 = \frac{m(m-1)}{2} < C_m^2 + m + Nb < 2^{m-1} - 1 < (m + C_m^2).m$$

Et pour $m \geq 11$, nous avons:

$$C_m^2 = \frac{m(m-1)}{2} < C_m^2 + m + Nb < (m + C_m^2).m < 2^{m-1} - 1$$

2. Le modèle *(L2')* requiert moins de contraintes que les autres modèles.

Remarque 3.6: Il convient de souligner que, contrairement aux autres modèles susmentionnés, le modèle *(T2)* permet de trouver une solution réalisable même s'il existe des variables de décision qui n'ont pas encore de valeur. En effet, l'ajout des co-cycles à une solution réalisable trouvée, c-à-d la mise à un de ces variables, conduit à la dégradation de cette solution par l'augmentation de son nombre de cellules et son trafic intercellulaire. Par conséquent, ces variables prennent automatiquement la valeur nulle dès qu'une solution réalisable est définie. C'est cette caractéristique qui nous motive à utiliser ce modèle dans le chapitre 5 malgré le nombre exponentiel de ses variables.

Remarque 3.7: L'évaluation du nombre de contraintes pour les modèles *(T1)* et *(T2)* n'est pas applicable dans notre cas car elle est réalisée dans notre approche de façon algorithmique.

3.4.2. Comparaison en termes de problème de redondance

Définition 2 (redondance): La redondance signifie qu'une composition a plus d'une matrice représentante (solution réalisable) en se basant sur une variable de décision particulière.

Le modèle *(L1)* souffre du problème de redondance. Pour illustrer un tel inconvénient, considérons la composition suivante: $\{M_1, M_2\}$ et $\{M_3\}$ (voir figure 3.3). Cette composition a

six matrices représentantes possibles obtenues par des permutations des indices de cellules (voir figure 3.4, où les valeurs de la variable x_{ij} sont représentées par une matrice). Cette redondance agrandit l'espace de recherche à explorer, rendant plus difficile le travail des méthodes de résolution, qu'elles soient de nature exacte ou approchée.

Ce problème devient de plus en plus critique pour un nombre important de cellules. En effet, plus le nombre de cellules d'une composition est important, plus elle a des matrices représentantes.

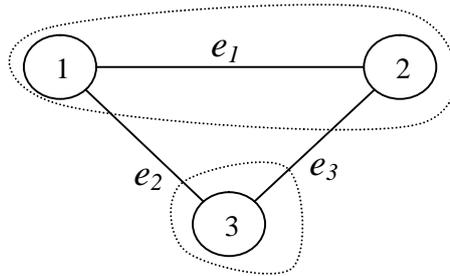


Figure 3.3. Exemple d'une composition C

		Cellule		
		1	2	3
Machine	1	1	0	0
	2	1	0	0
	3	0	1	0

(1)

		Cellule		
		1	2	3
Machine	1	0	1	0
	2	0	1	0
	3	1	0	0

(2)

		Cellule		
		1	2	3
Machine	1	1	0	0
	2	1	0	0
	3	0	0	1

(3)

		Cellule		
		1	2	3
Machine	1	0	0	1
	2	0	0	1
	3	1	0	0

(4)

		Cellule		
		1	2	3
Machine	1	0	1	0
	2	0	1	0
	3	0	0	1

(5)

		Cellule		
		1	2	3
Machine	1	0	0	1
	2	0	0	1
	3	0	1	0

(6)

Figure 3.4. Les matrices représentantes de la composition C générées par le modèle (L1)

Le modèle (L2) génère un taux de redondance plus faible par rapport au modèle (L1). Pour le démontrer, nous prenons la même composition précédente (voir figure 3.3). Cette composition a deux matrices représentantes possibles (voir figure 3.5), ce qui est faible par rapport à six matrices représentantes générées par le modèle (L1).

		Médiane		
		1	2	3
Machine	1	1	0	0
	2	1	0	0
	3	0	0	1

(1)

		Médiane		
		1	2	3
Machine	1	0	1	0
	2	0	1	0
	3	0	0	1

(2)

Figure 3.5. Les matrices représentantes de la composition C générées par le modèle (L2)

L'impact de la *Contrainte-affectation* sur le modèle (L2') et la variable de décision choisie sur le modèle (L3) est clair. En effet, deux solutions réalisables ne peuvent figurer pour la même composition. Par conséquent, l'inconvénient des solutions réalisables redondantes est éliminé avec ces deux modèles mathématiques. Pour le même exemple de la figure 3.3, cette composition peut être représentée par une seule solution réalisable (voir la matrice de la figure 3.6.a pour le modèle (L2') et la matrice de la figure 3.6.b pour le modèle (L3)).

		Médiane		
		1	2	3
Machine	1	1	×	×
	2	1	0	×
	3	0	0	1

a)

		Machine		
		1	2	3
Machine	1	×	1	0
	2	×	×	0
	3	×	×	×

b)

Figure 3.6. La matrice représentante de la composition C

Le problème de redondance se présente aussi avec le modèle (T1). Pour l'illustrer, considérons la composition présentée dans la figure 3.7.a. Cette composition peut être déterminée par quatre sous-graphes du graphe G dont les composantes connexes de chaque sous-graphe déterminent les cellules de la solution associée (voir figure 3.7.b).

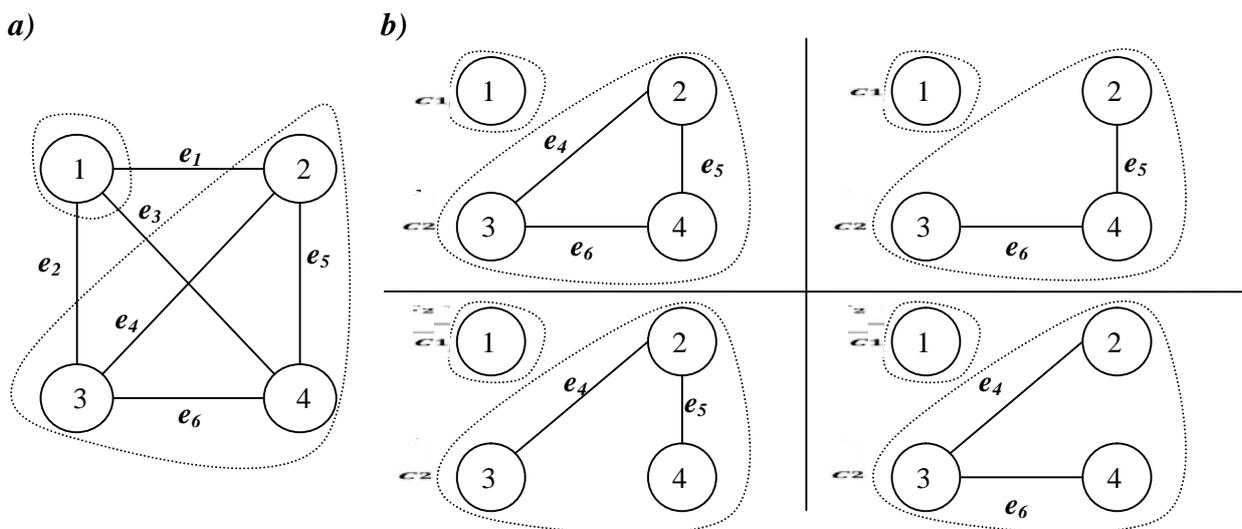


Figure 3.7. Le problème de redondance avec le modèle (T1)

Le modèle (T2) n'a pas été épargné par le problème de redondance. En effet, La composition de la figure 3.8 peut être définie par la somme de deux des co-cycles w_1, w_2 et w_3 (pour plus d'informations sur les co-cycles et la somme des co-cycles, voir le chapitre 5 et l'annexe B), c'est-à-dire cette composition est obtenue par la somme de w_1 et w_2 ou w_1 et w_3 ou w_2 et w_3 .

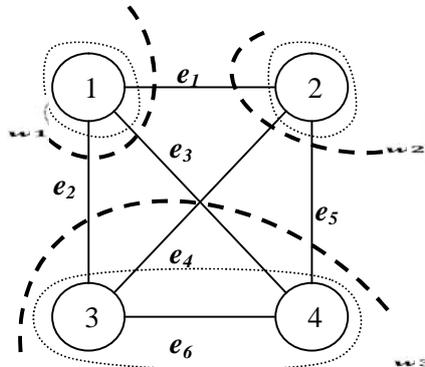


Figure 3.8. Décomposition en co-cycles

3.4.3. Comparaison en termes de problème de cécité

Définition 3 (cécité): La cécité signifie que la variable de décision utilisée est dans l'incapacité de représenter certaines solutions réalisables de l'espace de recherche.

Le deuxième problème qui se pose avec les modèles (T1) et (T2) est la cécité. Pour illustrer ce problème, considérons le graphe de flux de la figure 3.9 comportant quatre machines et trois arêtes. La solution suivante: $C_1 = \{M_1, M_2\}$ et $C_2 = \{M_3, M_4\}$, que nous nommons *solution 1*, ne peut être représentée en utilisant les variables de (T1) et (T2) parce qu'il n'y a aucun chemin intracellulaire reliant M_1 et M_2 . Heureusement, dans ce cas-ci, les variables des deux modèles arrivent à représenter la solution suivante: $C_1 = \{M_1\}$, $C_2 = \{M_2\}$ et $C_3 = \{M_3, M_4\}$, que nous nommons *solution 2*, qui a un trafic intercellulaire égal à celui de la *solution 1*. Cependant, si le nombre maximum de cellules est égal à deux, nous aurons un problème car la seule solution que les variables de (T1) et (T2) peuvent représenter devient non réalisable. Deux solutions à ce problème ont été proposées par Boulif [BOU06a]. La première consiste à ajouter des arêtes fictives de poids nuls au graphe de flux jusqu'à ce qu'il soit complet. La deuxième considère que le nombre maximum de cellules n'est borne supérieurement.

Cette cécité peut être considérée comme étant l'une des vertus de ces modèles: Les solutions non vues par la méthode de résolution présentent des cellules avec des groupes indépendants de machines. Ainsi, si le nombre de cellules n'est pas borné supérieurement, la méthode de résolution essaiera de définir un nombre de cellules qui est en totale harmonie avec l'esprit de la philosophie TG et engendré par la nature de l'instance du problème [BOU06a].

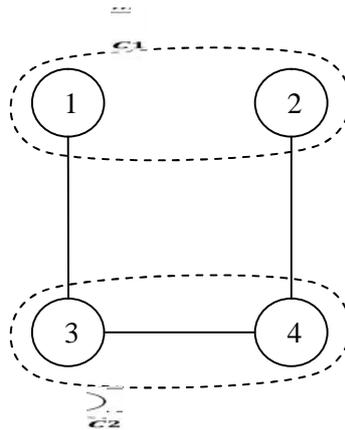


Figure 3.9. Une solution invisible [BOU06a]

3.5. Conclusion

Cinq modèles ont été présentés dans ce chapitre pour le problème FC. Le nombre de variables et de contraintes, le problème de redondance et le problème de cécité, trois éléments importants qui ont un impact direct sur les performances des méthodes de résolution, sont théoriquement comparés. Cette comparaison révèle l'inefficacité du modèle le plus utilisé dans la littérature et donc l'inutilité de l'utiliser pour construire des modèles plus complexes et la supériorité des modèles qui s'appuient sur une relation machines-machines par rapport aux modèles qui s'appuient sur une relation machines-cellules. Les résultats de cette étude comparative seront avantageux pour la communauté des chercheurs développant des modèles plus complexes pour le problème FC.

Dans la perspective de vérifier ces résultats théoriques, le chapitre suivant sera consacré à la présentation des méthodes de résolution exacte.

Chapitre 4

Les méthodes de résolution exactes

4.1. Introduction

Nous avons vu au chapitre deux que de nombreuses méthodes de résolution peuvent être utilisées pour aborder le problème de formation de cellules. Dans cette thèse, nous nous intéressons principalement à l'apport des méthodes exactes pour ce problème et en particulier aux méthodes d'exploration par séparation et évaluation qui fournissent des techniques permettant de déterminer une solution optimale entière sans avoir à énumérer explicitement toutes les solutions possibles.

Face à des problèmes difficiles, les méthodes d'exploration par séparation et évaluation perdent rapidement leur efficacité. Des méthodes de décomposition et des techniques spécifiques pour la résolution des problèmes linéaires de grande taille peuvent alors être associées à ces méthodes afin de les renforcer.

Ainsi dans ce chapitre, nous présentons en premier lieu les méthodes d'exploration par séparation et évaluation « Branch-and-Bound ». Puis nous décrivons la méthode de génération de colonnes, la décomposition de Dantzig-Wolfe et la méthode du « Branch-and-Price ». Nous faisons ensuite un tour d'horizon de la méthode de génération de contraintes, la décomposition de Benders et les concepts de base de la théorie polyédrale et nous terminons par la présentation de la méthode « Branch-and-Cut ».

4.2. Motivation

Les méthodes exactes offrent en théorie la garantie de trouver une solution optimale d'un problème et d'en prouver l'optimalité. Sur des problèmes linéaires en nombres entiers, elles sont efficaces en pratique sur des instances de petite taille, mais sont rapidement confrontées à l'aspect combinatoire des problèmes considérés. Cependant, l'application des méthodes exactes sur ces problèmes présente plusieurs intérêts [MAN04]. D'une part, la résolution exacte d'une version relâchée d'un problème (en simplifiant ou éliminant certaines contraintes) permet d'obtenir des bornes (supérieures ou inférieures, suivant que l'on considère une fonction objectif à minimiser, et c'est ce que nous supposons dans ce qui suit, ou à maximiser) qui servent à évaluer la qualité des différentes méthodes heuristiques proposées pour la recherche de solutions admissibles. D'autre part, l'étude de la modélisation d'un problème, de sa structure et des possibilités de décomposition peuvent permettre de ramener la résolution d'un problème globalement très complexe à celle d'un ensemble de sous-problèmes plus faciles à résoudre, et/ou mieux connus. On peut alors faire appel à des méthodes exactes pour résoudre tous ou une partie de ces sous-problèmes. Dans le second cas, la résolution exacte de certains aspects du problème global, peut être intégrée efficacement dans une méthode de résolution heuristique.

4.3. Les méthodes d'exploration par séparation et évaluation (SE)

Les méthodes SE, plus connues sous leur appellation anglaise Branch and Bound (B&B), sont des méthodes exactes permettant de résoudre des problèmes d'optimisation combinatoire. Elles réalisent une énumération intelligente de l'espace de solutions à la recherche des meilleures solutions.

Ces méthodes ont été utilisées pour la première fois par Land et Doig [LAN60] pour la programmation discrète. Elles sont ensuite devenues l'un des outils les plus couramment utilisés pour résoudre les problèmes NP-difficile. Le nom de « Branch and Bound » est apparu pour la première fois dans [LIT63] lors de la résolution du problème de voyageur de commerce (TSP).

Les méthodes SE sont basées sur deux principes élémentaires [GON09]: la séparation et l'évaluation. La séparation consiste à explorer l'ensemble des solutions réalisables en des sous-ensembles de plus en plus petits de façon à isoler une solution optimale dans l'un de ces sous-ensembles. Pour présenter cette exploration, on construit une arborescence dont le sommet de base correspond à l'ensemble des solutions réalisables et dont les autres sommets correspondent à des sous-ensembles de . Pour définir plus précisément la procédure d'exploration, nous devons définir une règle de séparation et une règle pour se déplacer dans l'arborescence. L'évaluation consiste à déterminer pour chaque sous-ensemble une borne inférieure. L'intérêt principal de cette borne est de permettre, lorsque l'on possède déjà une bonne solution réalisable, le parcours implicite des branches entières de l'arborescence sans examiner chacun des nœuds.

Selon Minoux [MIN07], il existe autant d'algorithmes de B&B que de façon de choisir:

- La procédure de calcul de la borne inférieure. Il est nécessaire que cette dernière soit la plus précise possible afin d'arrêter plus rapidement l'exploration.
- La règle de déplacement dans l'arborescence. Plusieurs stratégies ont été proposées, citons par exemple:
 - Profondeur d'abord qui consiste à choisir le sommet de niveau le plus élevé parmi les sommets non encore séparés. Cette stratégie a pour but de mettre en évidence le plus tôt possible une solution du problème;
 - Largeur d'abord qui consiste à choisir systématiquement le sommet ayant l'évaluation la plus faible. Le risque, avec cette méthode, est d'avoir à explorer une fraction importante de l'arborescence avant de découvrir une solution. En revanche, la première solution trouvée est généralement de meilleure qualité que celle déterminée par une stratégie profondeur d'abord.
- La variable suivant laquelle s'effectuera la séparation du sommet choisi. Très souvent, l'ordre des variables est fixe une fois pour toutes, soit de façon totalement arbitraire, soit suivant des critères justifiés de façon intuitive.

Un exemple d'un algorithme de B&B est présenté dans la figure 4.1 où les notations suivantes sont utilisées:

UB: La meilleure solution réalisable calculée;

x^* : Une solution optimale du problème;

Liste: Liste des problèmes à traiter.

Entrée: Une instance de *PLNE*
Sortie: Une solution optimale de *PLNE*

1. Initiation

- $UB = +\infty$;{ ou $UB =$ à la solution d'une heuristique };
- $Liste = \{PLNE\}$;

2. Tant que ($Liste \neq \emptyset$) faire

- a) *Sélection du nœud*: Choisir un problème dans *liste*;
- b) *Evaluation du nœud*: Résoudre la relaxation continue du problème choisi (*PR*);
 Si *PR* est non réalisable, supprimer le de *liste* et Aller à l'étape 2;
 Sinon, soit \bar{x} sa solution optimale et $sol_{courante}$ la valeur de la fonction objectif correspondante.
- c) *Elagage*:
 Si $sol_{courante} \geq UB$, supprimer le *PR* de *liste* et Aller à l'étape 2;
 Sinon, Si \bar{x} n'est pas entière, aller à l'étape *d*;
 Sinon Poser $UB = sol_{courante}$ et $x^* = \bar{x}$, supprimer le *PR* de *liste* et
 Aller à l'étape 2;
- d) *Branchement*: Effectuer un branchement et créer deux ou plusieurs *sous-problèmes* de *PR*. Ajouter les *sous-problèmes* à *liste* et Aller à l'étape 2;

Fin Tant que

3. Fin : Sortir avec une solution optimale x^* de *PLNE*;

Figure 4.1. Algorithme de Branch-and-Bound

Remarque 4.1: L'algorithme de B&B présenté ci-dessus résout des problèmes d'optimisation modélisés comme des programmes linéaires en nombres entiers (PLNE). Cependant la méthode de B&B peut être appliquée à des problèmes qui ne sont pas formalisés sous forme de PLNE (voir le chapitre 5).

Pour plus de détails sur les méthodes SE et leurs principes, voir [PAP81, MIN07 et GON09].

Les méthodes SE ont été utilisées pour résoudre plusieurs problèmes tels que : le problème de formation de cellules [KUS93] et le problème du sac à dos (Knapsack Problem) [KOZ04].

Devant des PLNE difficile, les méthodes SE peuvent s'avérer inutiles. En effet, la résolution de la relaxation continue de ces problèmes donne souvent une borne de mauvaise qualité qui rend la tâche d'une méthode SE très difficile. Une solution à ce problème consiste à utiliser des méthodes de décomposition qui donnent lieu à des nouveaux modèles, permettant de ramener la résolution de tels problèmes à la résolution d'une suite de problème plus faciles, et fournissent de meilleures bornes. Cependant, ces modèles sont de trop grande taille et ne peuvent donc être résolus par un algorithme tel que le simplexe. Par conséquent, nous faisons appel à des techniques de résolution des programmes linéaires de grande taille à savoir, la génération de colonnes et la génération de contraintes qui seront détaillées dans les sections suivantes.

4.4. Généralités sur la structure des grands programmes linéaires

Il est rare de rencontrer, dans les applications, des programmes linéaires de grande taille sans structure particulière [MIN07]. Tout d'abord, les matrices de contraintes de ces programmes linéaires sont généralement « très creuses », c'est-à-dire comportent une forte proportion de termes nuls. Par ailleurs, non seulement la proportion des termes nuls est élevée, mais ces termes nuls ne sont généralement pas distribués de façons quelconque. La figure 4.2 montre les structures des matrices de contraintes les plus typiques rencontrées dans les applications.

- La figure 4.2.a correspond au cas de K blocs associés à K sous-problèmes qui seraient indépendants, sans la présence d'un certain nombre de contraintes, dites *contraintes couplantes* (Programmes de type 1).
- La figure 4.2.b correspond au cas de K blocs associés à K sous-problèmes qui seraient indépendants, sans la présence d'un certain nombre de variables indépendantes, dites *variables couplantes* (Programmes de type 2).

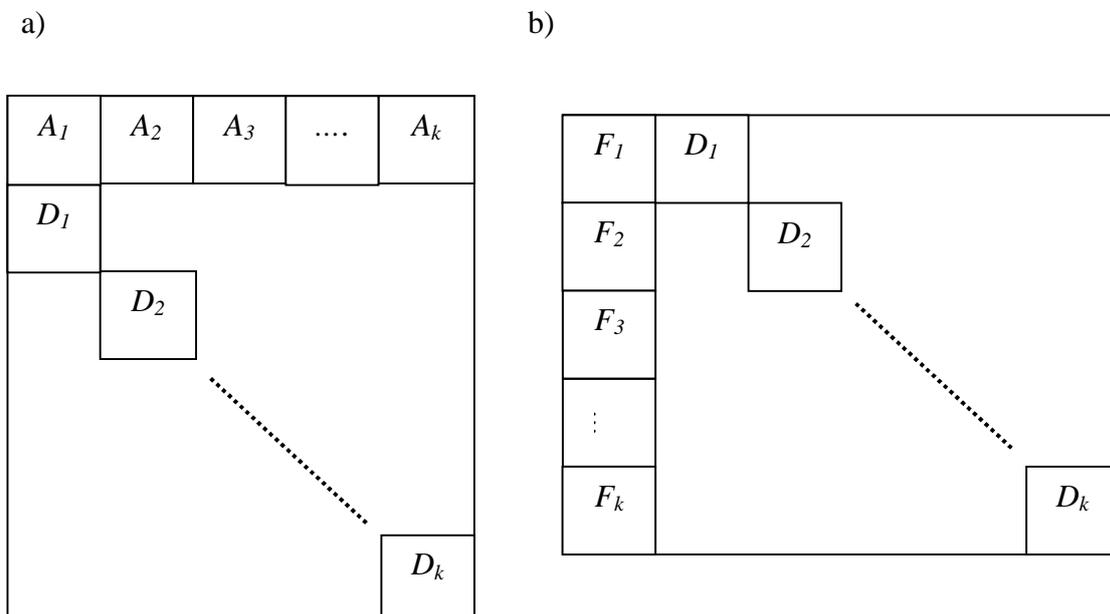


Figure 4.2 Structure bloc diagonale: a) avec contraintes couplantes
b) avec variables couplantes [MIN07]

4.5. Génération de colonnes pour les problèmes de grande dimension

4.5.1. Principe général

La méthode de génération de colonnes, ou programmation linéaire généralisée, permet de résoudre un programme linéaire en nombres réels lorsque les variables sont trop nombreuses pour une résolution directe par un solveur [TOU13]. Elle a été introduite par Gilmore et Gomory entre 1961 et 1965 [GIL61, GIL63 et GIL65] pour résoudre le problème de découpe industrielle. L'idée principale de cette méthode repose sur le fait que toutes les variables ne sont pas nécessaires pour trouver la solution optimale [TOU13]. En effet, lors de la résolution d'un programme linéaire par la méthode de simplexe, plusieurs variables sont hors base et, très souvent, la plupart d'entre elles sont nulles à l'optimum, et peuvent être donc négligées.

En s'appuyant sur cette idée, la méthode de génération de colonnes consiste à ne manipuler, à la fois, qu'un sous-ensemble de variables (colonnes) de petite taille constituant le problème maître restreint (*PMR*) et à identifier les variables entrantes en base (variables améliorantes) par la résolution d'un sous-problème. Si une variable améliorante est trouvée alors elle est ajoutée au *PMR* et le processus se poursuit. Sinon cela signifie qu'il n'existe plus de variables améliorantes et la solution du *PMR* représente donc la solution optimale du problème initial. L'algorithme qui résout les sous-problèmes s'appelle « l'algorithme générateur » ou aussi « l'algorithme de pricing ». Le schéma général de cette méthode est décrit dans la figure 4.3.

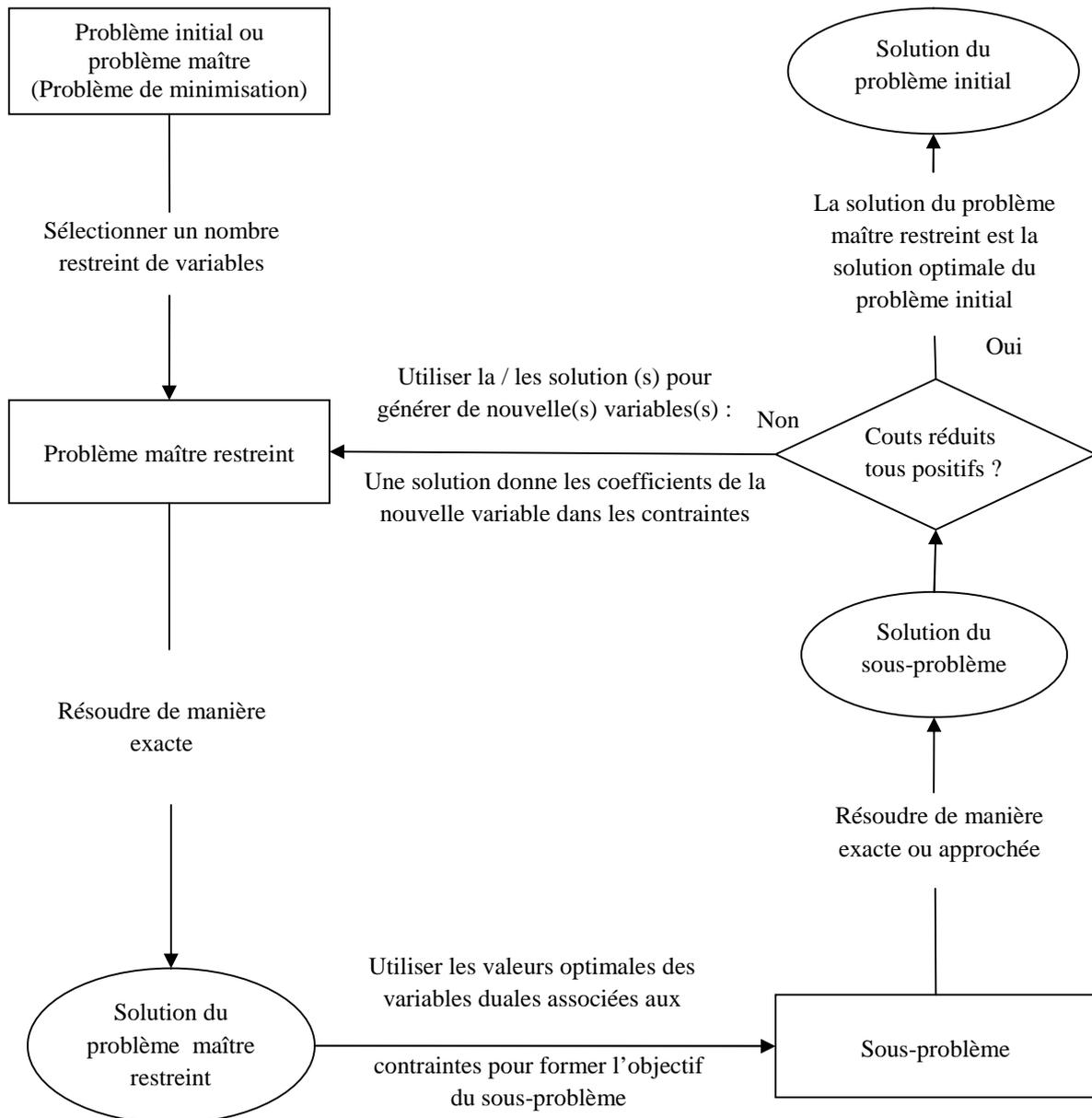


Figure 4.3. Schéma général de la génération de colonne [TOU13]

Le rôle de l'algorithme générateur est, à chaque itération, de fournir une colonne améliorante ou de prouver qu'il n'en existe aucune. La rapidité de la méthode de génération de colonnes est donc très dépendante de l'efficacité de l'algorithme générateur. N'importe quelle colonne avec un coût réduit négatif pour un problème de minimisation peut contribuer à améliorer la

solution courante, il n'est donc pas toujours utile de résoudre exactement chaque sous-problème. Ainsi, le sous-problème peut être résolu de manière heuristique. Cependant si l'heuristique ne donne pas de variables améliorantes, nous sommes obligés de le résoudre de manière exacte soit pour trouver une variable améliorante, soit pour prouver qu'il n'existe plus de variable améliorante. Nous pouvons trouver une description détaillée de la méthode de génération de colonnes dans [MIN07 et TOU13].

De nombreux programmes linéaires peuvent être reformulés de façon à présenter une structure adaptée à la résolution par génération de colonnes. Nous utilisons classiquement la méthode de décomposition de Dantzig-Wolfe que nous présentons dans la suite.

4.5.2. Décomposition de Dantzig-Wolfe

Dans ce paragraphe, nous nous intéressons aux programmes linéaires décomposables de type 1 (voir la section 4.4) c.-à-d. avec contraintes couplantes.

La décomposition de Dantzig-Wolfe [DAN60] associe un polyèdre à chaque bloc et exprime les variables du problème comme combinaison linéaire des sommets de ces polyèdres, permettant ainsi une formulation, présentée le programme maître (PM), équivalente au programme linéaire de départ (P), dans laquelle seules les contraintes couplantes sont explicitées [MAN04]. Si le PM présente moins de contraintes que P , il introduit un nombre "exponentiel" de variables, celui-ci étant égal au nombre de points extrêmes du polyèdre. Pour le résoudre, il est donc nécessaire d'utiliser la méthode de génération de colonnes. L'algorithme générateur doit chercher, dans ce cas, les sommets des K polyèdres qui minimisent la valeur de K fonctions objectifs, c'est-à-dire résoudre les K sous-problèmes.

Le processus itératif, engendré par la génération de colonnes, s'arrête lorsqu'il n'existe plus de solution strictement négative à ces problèmes. La solution optimale du PMR est alors la solution optimale du PM et elle fournit également la solution de P .

La décomposition de Dantzig-Wolfe a été appliquée avec succès dans de nombreuses applications, nous citons par exemple: [GAB06] pour le problème de planification de missions spatiales, [NSA06] pour la résolution du problème de formation de cellules de production et [SAL11] pour la résolution du problème de tournées de véhicules.

Voir [MIN07] pour une présentation détaillée de la décomposition de Dantzig-Wolfe.

4.5.3. Résolution des PLNE par génération de colonnes

4.5.3.1. Décomposition des PLNE

La méthode de Dantzig-Wolfe permet, comme il a été vu précédemment, de décomposer des programmes linéaires de grande taille et de les résoudre par génération de colonnes. En pratique, cette méthode est utilisée aussi pour la résolution des programmes linéaires en nombres entiers ($PLNE$) avec une grande taille. En effet, la relaxation continue d'un $PLNE$ a la forme requise pour être résolue par la méthode de génération de colonnes.

L'utilisation de la décomposition de Dantzig-Wolfe permet de réduire le saut d'intégrité entre le $PLNE$ et sa relaxation continue [VIR05]. Autrement dit, cette reformulation de $PLNE$ réduit l'écart existant entre la valeur de la solution de la relaxation continue de $PLNE$ et la valeur de sa solution entière, renforçant ainsi nos capacités à résoudre le $PLNE$ à l'optimalité ou, lorsque ce n'est pas possible, à mieux évaluer la qualité d'une solution approchée.

4.5.3.2. Méthode de « Branch and Price »

La génération de colonnes appliquée à un problème maître en nombres entiers (*PM*) fournit une solution optimale de sa relaxation continue qui, si elle n'est pas entière, peut être utilisée comme une bonne borne inférieure de la solution entière de *PM* [MAN04].

Pour trouver la solution entière, nous utilisons une méthode, appelée Branch and Price (B&P), qui combine les performances de B&B et génération de colonnes (voir figure 4.4). Plus précisément, la génération de colonnes est utilisée dans la procédure d'évaluation des nœuds.

Pour plus d'indication sur la méthode B&P, nous renvoyons le lecteur à voir [MIN07 et TOU13].

Entrée: Une instance de *PLNE*

Sortie: Une solution optimale de *PLNE*

1. Initiation

- $UB = +\infty$; { ou $UB =$ à la solution d'une heuristique };
- $Liste = \{PLNE\}$;

2. Tant que ($Liste \neq \emptyset$) faire

- a) *Sélection du nœud:* Choisir un problème dans *liste*;
- b) *Evaluation du nœud:* Résoudre la relaxation continue du problème choisi (*PR*) par la méthode de *génération de colonnes*;
Si *PR* est non réalisable, supprimer le de *liste* et Aller à l'étape 2;
Sinon, soit \bar{x} sa solution optimale et $sol_{courante}$ la valeur optimale correspondante.
- c) *Elagage:*
Si $sol_{courante} \geq UB$, supprimer le *PR* de *liste* et Aller à l'étape 2;
Sinon, Si \bar{x} n'est pas entière, aller à l'étape d);
Sinon Poser $UB = sol_{courante}$ et $x^* = \bar{x}$, supprimer le *PR* de *liste* et Aller à l'étape 2;
- d) *Branchement:* Effectuer un branchement et créer deux ou plusieurs *sous-problèmes* de *PR*. Ajouter les *sous-problèmes* à *liste* et Aller à l'étape 2;

Fin Tant que

- e) **Fin** : Sortir avec une solution optimale x^* de *PLNE*;

Figure 4.4 Algorithme de Branch and Price

Les premières mises en œuvre de la méthode B&P ont concerné le problème de tournées de véhicules avec fenêtre de temps [DES84]. D'autres problèmes ont, par la suite, été résolus par cette méthode, tels que: le problème de coloration de graphe [MEH96] et les problèmes d'affectation et d'ordonnancement de tâches [VAN00].

4.6. Génération de contraintes pour les problèmes de grande dimension

4.6.1. Principe général

La description sous la forme d'un programme linéaire d'un problème d'optimisation combinatoire peut être trop grande pour être représentée explicitement en mémoire ou pour tenir dans un solveur de programmes linéaires. Dans ce cas, la méthode de génération de contraintes, ou méthode des plans de coupe, résout une relaxation du problème d'optimisation en enlevant la plupart de ces contraintes [BOU98]. En effet, cette méthode repose sur la manipulation d'un sous-ensemble de contraintes constituant le problème maître restreint (*PMR*) et l'identification des contraintes du problème originel (*PM*), appelées les plans de coupe, violées par la solution courante. Si une contrainte est trouvée alors elle est ajoutée au *PMR* et le processus se poursuit. Sinon, cela signifie qu'il n'existe plus de contrainte violée et donc la solution courante constitue une solution optimale du *PM*.

Notons que l'évaluation de la solution optimale de *PMR* est inférieure ou égale à celle d'une solution optimale de *PM* (cas d'un problème de minimisation). En effet, l'ensemble des solutions réalisables du *PM* est inclus dans celui du *PMR*.

La méthode de génération de contraintes n'impose pas d'avoir une liste exhaustive explicite des plans de coupe, mais seulement une méthode pour générer efficacement des contraintes valides violées. L'identification de ces contraintes est appelé le problème de séparation [BOU98].

Le problème de séparation

Soit \mathbf{C} un ensemble de contraintes d'un programme linéaire. Étant donné un point $x \in \mathbb{R}^n$, le *problème de séparation* associé à \mathbf{C} et x consiste à déterminer si x satisfait toutes les contraintes de \mathbf{C} et sinon de trouver une contrainte de \mathbf{C} violée par x .

Le schéma général de la méthode de génération de contraintes est décrit dans la figure 4.5. Nous pouvons trouver une description détaillée de la méthode de génération de contraintes dans [MIN07].

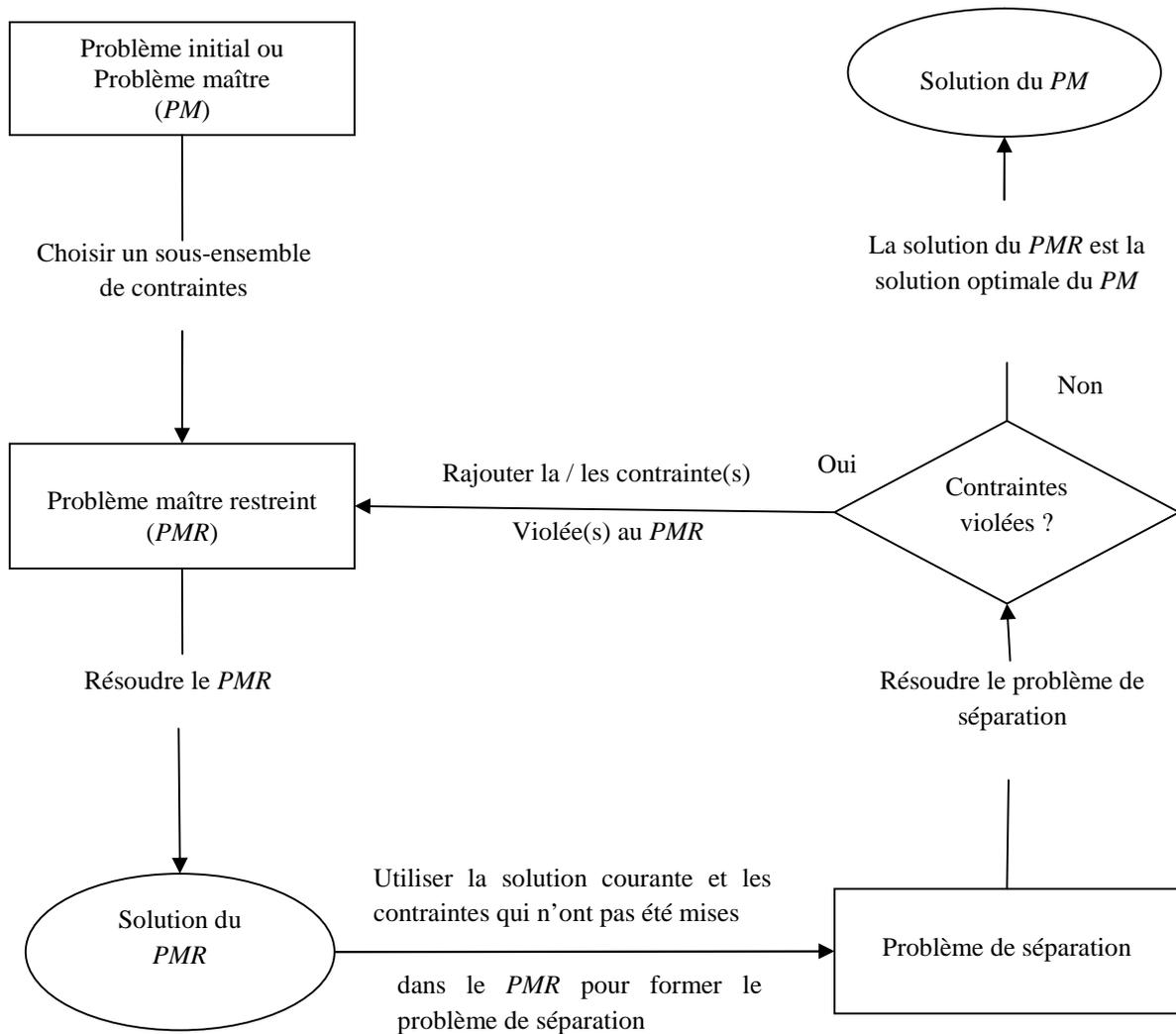


Figure 4.5. Le schéma général de la génération de contraintes

4.6.2. Décomposition de Benders

Dans ce paragraphe, nous nous intéressons aux programmes linéaires décomposables de type 2 (voir la section 4.4) c.-à-d. avec variables couplantes.

La méthode de décomposition de Benders, appelée aussi méthode de décomposition par partitionnement de variables, permet de décomposer un programme mathématique mixte, i.e. présentant des variables entières et continues, afin de traiter de façon séparée la partie entière de la partie continue [VIR05]. En effet, dès l'instant où la valeur des variables couplantes est fixée, nous allons résoudre un problème restreint aux autres variables (*PE*, programme esclave). Cependant, nous ne pouvons pas choisir n'importe comment les valeurs des variables couplantes, il faut au moins que *PE* ait une solution pour pouvoir trouver une solution au problème originel (*P*). Afin de traduire cette condition, nous associons au problème avec variables couplantes (*PM*, problème maître) des contraintes d'optimalité et de réalisabilité correspondant respectivement aux points extrêmes et aux rayons extrémaux du polyèdre des contraintes du *PE*. De ce fait, le nombre de contraintes peut être énorme et il ne sera pas possible de toutes les énumérer, mais plutôt de procéder par génération d'un sous-ensemble d'entre elles en suivant le principe de génération de contraintes. La résolution du

problème de séparation est guidée, dans ce cas, par la résolution optimale de PE pour la solution courante du PMR . Le processus itératif s'arrête lorsque le test d'optimalité est satisfait. La solution optimale du PMR est alors la solution optimale du PM et elle fournit également la solution de P .

L'intérêt de la méthode de décomposition de Benders réside dans le fait qu'elle s'applique, non seulement aux grands programmes linéaires décomposables, mais également à beaucoup d'autres problèmes, par exemple les programmes linéaires mixtes et certains programmes partiellement non linéaires [MIN07].

La décomposition de Benders a été introduite par Benders en 1962 pour résoudre des programmes en nombres entiers mixtes [BEN62]. Ensuite, cette méthode a été appliquée avec succès dans de nombreuses applications de la programmation linéaire mixte, nous citons par exemple: [MIN01] pour la résolution de problèmes d'optimisation de réseaux à coût discret et [HER98 et GHO11] pour la résolution du problème de formation de cellules de production.

Pour une description détaillée de la méthode de décomposition de Benders, nous invitons le lecteur à voir [BAP05, VIR05 et MIN07].

4.6.3. Résolution de PLNE par la méthode de génération de contraintes

En pratique, la méthode de génération de contraintes est utilisée aussi pour résoudre la relaxation continue d'un PLNE dans deux cas [TOU13]:

- Nous avons un PLNE avec un très grand nombre de contraintes et nous ne souhaitons pas (ou nous ne pouvons pas) toutes les énumérer.
- Nous avons un PLNE et nous souhaitons utiliser des plans de coupe afin d'améliorer la valeur de la relaxation.

4.6.3.1. Plans de coupe généraux

Il existe des plans de coupe généraux qui ne sont pas spécifiques au problème résolu et peuvent être utilisés pour tous les programmes linéaires en nombres entiers. Les plans de coupe de Gomory [GOM58] ou les coupes générées par la méthode dite du « *lift and project* » introduite par Balas et al dans [BAL93a, BAL93b et BAL96] sont des exemples. Cependant, L'utilisation de ces plans s'avère souvent limitée pendant la résolution de problèmes difficiles. D'où l'intérêt de l'étude des plans de coupe spécifiques au problème résolu que nous décrirons dans la section suivante.

4.6.3.2. Approche polyédrale

L'approche polyédrale consiste à ramener la résolution d'un problème d'optimisation combinatoire à la résolution d'un programme linéaire en décrivant l'enveloppe convexe de ses solutions ($conv(S)$) par un système d'inégalités linéaires [MAH04]. En effet, tout polyèdre peut être écrit sous la forme d'un système formé d'inéquations linéaires. Par conséquent, si nous caractérisons le polyèdre $conv(S)$ par un système d'inégalités linéaires, alors nous ramenons la résolution d'un problème d'optimisation combinatoire à la résolution d'un programme linéaire. La caractérisation complète du polyèdre $conv(S)$ est généralement difficile à obtenir [FOU12]. Par ailleurs, une description complète du polyèdre peut comporter un nombre exponentiel d'inégalités. Cependant, un nombre réduit de ces inégalités peut être suffisant pour résoudre le problème à l'aide de la méthode de Branch and Cut, décrite dans la section suivante.

Plusieurs travaux ont essayé de caractériser le polyèdre associé au problème étudié, citons par exemple [BOU98] pour le problème de voyageur de commerce.

Pour plus de détails sur l'approche polyédrale voir [BOU98, SCH03, MAH04 et FOU12].

4.6.3.3. Méthode de « Branch and Cut »

Nous venons de voir que la méthode de génération de contraintes appliquée au PLNE fournit une solution qui est, généralement, fractionnaire (non optimale). Pour trouver une solution entière optimale, nous faisons appel à une méthode, appelée Branch and Cut (B&C), qui conjugue les efforts de la méthode B&B avec la méthode de génération de contraintes. En effet, pour résoudre un PLNE, la méthode B&C commence par résoudre une relaxation continue du problème puis elle applique l'algorithme de génération de contraintes sur la solution trouvée. Si celle-ci n'arrive pas à obtenir une solution entière alors le problème est divisé en plusieurs sous-problèmes qui seront résolus de la même façon. Le fait d'ajouter des contraintes (appliquer la méthode de génération de contraintes) pour améliorer la valeur de relaxation réduit le nombre de branchements à effectuer, on construit donc un arbre plus petit (moins de mémoire nécessaire, parcours de l'arbre plus rapide) [TOU13].

Un algorithme de B&C est présenté dans la figure 4.7. Pour plus d'indications sur la méthode de B&C, nous renvoyons le lecteur à voir [MIN07, BOU98 et TOU13].

<p>Entrée: Une instance de <i>PLNE</i></p> <p>Sortie: Une solution optimale de <i>PLNE</i></p>
<p>1. Initiation</p> <ul style="list-style-type: none"> • $UB = +\infty$; { ou $UB =$ à la solution d'une heuristique }; • $Liste = \{PLNE\}$. <p>2. Tant que ($Liste \neq \emptyset$) faire</p> <p>a) <i>Sélection du nœud</i>: Choisir un problème dans <i>liste</i>;</p> <p>b) <i>Evaluation du nœud</i>: Résoudre la relaxation continue du problème choisi (<i>PR</i>) par la méthode de <i>génération de contraintes</i>;</p> <p>Si <i>PR</i> est non réalisable, supprimer le de <i>liste</i> et Aller à l'étape 2;</p> <p>Sinon, soit \bar{x} sa solution optimale et $sol_{courante}$ la valeur optimale correspondante.</p> <p>c) <i>Elagage</i>:</p> <p>Si $sol_{courante} \geq UB$, supprimer le <i>PR</i> de <i>liste</i> et Aller à l'étape 2;</p> <p>Sinon, Si \bar{x} n'est pas entière, aller à l'étape d);</p> <p style="padding-left: 40px;">Sinon Poser $UB = sol_{courante}$ et $x^* = \bar{x}$, supprimer le <i>PR</i> de <i>liste</i> et Aller à l'étape 2;</p> <p>d) <i>Branchement</i>: Effectuer un branchement et créer deux ou plusieurs <i>sous-problèmes</i> de <i>PR</i>. Ajouter les <i>sou-problèmes</i> à <i>liste</i> et Aller à l'étape 2;</p> <p>Fin Tant que</p> <p>3. Fin : Sortir avec une solution optimale x^* de <i>PLNE</i>;</p>

Figure 4.6 Algorithme de Branch and Cut

Une telle approche a été utilisée pour la première fois pour le problème de « *Linear Ordering* » dans [GRO84]. Le terme de « *Branch and Cut* » était introduit par Padberg et Rinaldi [PAD87] pour un algorithme de résolution du problème du voyageur de commerce.

Une version généralisée de ce dernier est présentée et parallélisée par Bouzgarrou [BOU98]. La méthode de B&C a été utilisée aussi avec succès pour la résolution de divers autres problèmes d'optimisation combinatoire tel que les problèmes de tournées de véhicules (voir, par exemple, [FIS94]) et les problèmes de satisfiabilité (voir, par exemple, [JOY97]).

4.7 Conclusion

Dans ce quatrième chapitre, nous avons présenté un recueil des méthodes exactes qui nous serviront dans les chapitres à venir pour résoudre le problème FC.

Les méthodes SE sont basées sur deux techniques élémentaires: la séparation et l'évaluation. La séparation permet d'obtenir une méthode efficace pour énumérer toutes les solutions tandis que l'évaluation permet d'éviter l'énumération explicite de toutes les solutions. Selon les façons de choisir: la procédure de calcul de la borne inférieure, la règle de déplacement dans l'arborescence et la variable suivant laquelle s'effectuera la séparation, différents algorithmes peuvent être développés.

Pour des problèmes d'optimisation combinatoire de très grande taille, ou pour lesquels la relaxation continue ne fournit pas de bornes de qualité suffisante, deux variantes de la Programmation Linéaire, appelée respectivement Génération de Colonnes et Génération de contraintes permettent de développer des méthodes de résolution efficaces.

Dans le chapitre suivant, nous décrirons les méthodes d'exploration par séparation et évaluation proposées pour la résolution du problème FC.

Chapitre 5

Une méthode par séparation et évaluation cocyclique guidée par les contraintes

5.1. Introduction

Les méthodes de résolution exactes peuvent être utilisées pour borner les problèmes d'optimisation combinatoire qui se posent et ainsi évaluer la qualité des algorithmes développés pour la recherche de solutions. Couplées à une approche par décomposition des problèmes, elles peuvent également être intégrées dans les processus de résolution, en traitant de façon exacte certains sous problèmes. Malgré tous ces intérêts, l'application des méthodes exactes sur le problème de formation de cellules (FC) a reçu peu d'attention à cause de sa complexité.

Dans ce chapitre, nous proposons deux méthodes d'exploration par séparation et évaluation (SE), tout en tenant compte des contraintes réelles de la vie des systèmes de production. La première méthode est basée sur la notion des co-cycles dans la génération des solutions. La deuxième a une structure similaire à la première, sauf qu'elle est améliorée par les contraintes.

5.2. Les méthodes SE et le problème FC

Dans ce qui suit, nous présentons des études qui ont examiné les méthodes SE, ou B&B en anglais, pour la résolution exacte ou approximative du problème FC. Notez que ces approches peuvent différer de manière significative dans leurs entrées, variables de décision et / ou fonctions objectives.

En s'appuyant sur l'analyse d'un graphe, Al-Qattan [ALQ90] a développé une heuristique basée sur une méthode B&B pour spécifier les cellules de machines et les familles de produits. La méthode proposée transfère la matrice d'incidence produits / machines initiale à une matrice diagonale (c.-à-d. matrice dans laquelle il n'y a pas d'éléments exceptionnels) en dupliquant les machines goulot d'étranglement. L'objectif de l'algorithme proposé est de minimiser le nombre de machines doubles.

Kusiak et al [KUS93] ont développé une méthode A*, qui s'appuie sur un modèle quadratique, pour résoudre le problème FC. Parce que l'algorithme proposé est incapable de traiter les grands problèmes, les auteurs ont proposé une heuristique basée sur la méthode B&B. Le procédé de branchement proposé est effectué sur la matrice d'incidence produits / machines.

Ce chapitre est basé sur l'article [MER14]

Une formulation quadratique a été proposée par Cheng et al [CHE96] pour résoudre un problème similaire à celui décrit par Kusiak et al [KUS93], mais avec des données supplémentaires telles que l'ordonnement des machines, les coûts des mouvements intercellulaires et le nombre maximum de machines admises dans une cellule. La résolution du modèle est effectuée en utilisant une approche A* et, encore une fois, en raison de la faiblesse de cet algorithme dans le traitement des problèmes de grande taille, les auteurs ont développé une heuristique pour accélérer la procédure de branchement.

Ramabhatta et Nagi [RAM98] ont proposé une formulation mathématique pour le problème FC en considérant le routage alternatif. Un algorithme B&B est développé qui commence à partir d'une solution heuristique et tente de l'améliorer via une procédure de recherche systématique.

L'approche basée sur le concept de partitionnement de graphe proposée par Spiliopoulos et Sofianopoulou [SPI98] est une méthode de recherche arborescente optimale qui emploie deux bornes inférieures différentes: une borne classique pour le problème d'affectation quadratique et une borne proposée pour le problème de transport quadratique. Le nombre de cellules n'est pas fixé a priori, mais il est obtenu dans la solution.

Caux et al [CAU00] ont proposé un algorithme qui a pour fonction objectif la minimisation du trafic intercellulaire en tenant compte de la contrainte des capacités disponibles des différentes machines du problème. Leur amélioration consiste à intégrer l'approche B&B à la méthode de recuit simulé et ceci pour choisir la meilleure séquence de chaque produit qui donne une meilleure solution. L'algorithme utilise aussi une matrice d'incidence produits / machines.

Bouliif et Atif [BOU06b] ont proposé une formulation basée sur le partitionnement de graphe pour le problème FC en considérant des données de production et des contraintes du monde réel. Le modèle proposé est résolu par une combinaison d'un algorithme génétique et d'une méthode B&B. La méthode B&B proposée est utilisée pour déterminer les zones qui peuvent potentiellement contenir de bonnes solutions. Ensuite, ces régions sont explorées au moyen d'un algorithme génétique.

Les mêmes auteurs [BOU06c] ont adopté une approche epsilon-contrainte afin d'améliorer la borne inférieure de l'étape de branchement de la méthode B&B. Toutefois, le travail présenté n'a pas fourni des résultats expérimentaux.

Arkat et al [ARK12] ont formulé le problème FC comme un modèle de programmation linéaire pur. Les auteurs ont mis au point trois versions de l'algorithme B&B. Le premier algorithme utilise un schéma de branchement binaire basé sur les variables de décision. Contrairement au premier algorithme qui s'appuie sur le modèle mathématique proposé, le second est conçu sur la base de la structure du problème. Le dernier algorithme a une structure similaire à celle de second algorithme, sauf qu'il a la capacité d'éliminer les nœuds en double dans les branches de l'arbre.

Goldengorin et al [GOL13] ont présenté un algorithme B&B pour la résolution du problème FC bi-critère originalement suggéré par Zilinskas et al [ZIL13]. Deux modèles bi-critère bien connus ont été utilisés: le premier modèle considère le nombre d'éléments exceptionnels et des vides tandis que le deuxième modèle considère les mouvements intercellulaires et les écarts de charge intercellulaires.

Zilinskas et al [ZIL15] ont proposé un algorithme B&B exact pour créer un front de Pareto-optimal pour le problème FC bi-critère.

5.3. Principe général de la méthode SE cocyclique développée

5.3.1. L'idée générale

Le codage utilisé pour la codification des solutions est un codage binaire basé sur le modèle de la théorie des graphes discuté en détails dans le paragraphe 3.3.4.3 et dans [MER14]. Pour le problème FC, ce codage définit une chaîne binaire de $Card(E)$ allèles. Chaque allèle représente une arête du graphe de flux G et indique si oui ou non cette arête est intra ou intercellulaire (i.e. les machines correspondant aux deux extrémités de cette arête sont, ou ne sont pas, dans la même cellule). Ainsi, si une arête est intercellulaire, elle est notée un (1), sinon elle est notée zéro (0). Pour plus de détails sur le codage binaire, voir [BOU06b].

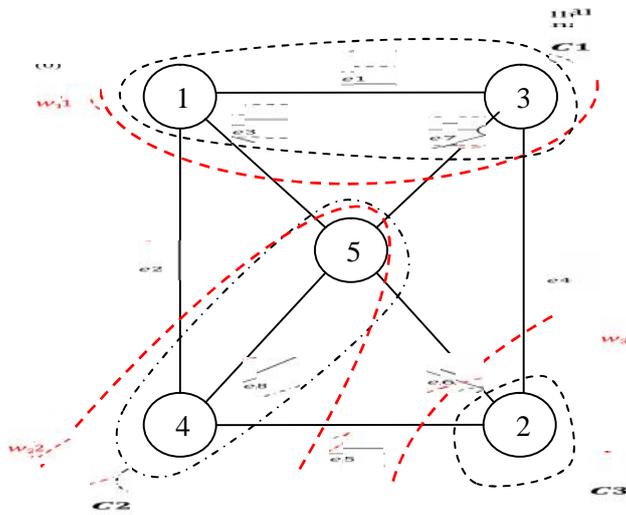


Figure 5.1. Décomposition en co-cycles

A titre d'exemple, considérons la solution définie par la composition de cellules suivante: $C_1 = \{M_1, M_3\}$, $C_2 = \{M_4, M_5\}$ et $C_3 = \{M_2\}$ (voir figure 5.1). Cette solution est codée par la chaîne binaire suivante:

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
0	1	1	1	1	1	1	0

Puisqu'une solution au problème FC est une partition des sommets du graphe de flux, elle peut, dans le cas d'un graphe connexe, être représentée par une somme de co-cycles [BOU06b], en utilisant l'opérateur booléen OR, notée \oplus .

La solution représentée par la figure 3.2, par exemple, peut être définie par la somme de deux des co-cycles w_1, w_2, w_3 (voir figure 5.1). Par exemple, la somme de $w_1 = (0, 1, 1, 1, 0, 0, 1, 0)$ et $w_2 = (0, 1, 1, 0, 1, 1, 1, 0)$ donne $w_1 \oplus w_2 = (0, 1, 1, 1, 1, 1, 1, 0)$. Ce qui n'est autre que le codage de la solution.

5.3.2. Règle de séparation

Soit $\mathcal{W} = \{w_1, w_2, \dots, w_{|\Omega|}\}$ l'ensemble de $|\Omega|$ co-cycles liés au graphe de flux G (voir l'annexe B), et soit S l'ensemble de toutes les solutions possibles. La séparation est effectuée selon Boulif et Atif [BOU06b] comme suit: L'ensemble S est d'abord divisé en deux sous-ensembles enfants, S_0 et S_1 , selon que le co-cycle w_1 est ou n'est pas utilisé pour construire les solutions de S_1 et S_0 (S_0 est le sous-ensemble de solutions qui ne sont pas construites par w_1 , et S_1 est le sous-ensemble de solutions qui sont construites par w_1). Ensuite, chacun des sous-ensembles S_0 et S_1 est divisé en deux sous-ensembles enfants (S_0 donne S_{01} , S_{00} et S_1 donne S_{11} et S_{10}) selon que le co-cycle w_2 est ou n'est pas utilisé pour construire les solutions de chaque sous-ensemble, et ainsi de suite.

5.3.2.1. Procédure de séparation

Le sous-ensemble S_{00101} , par exemple, désigne le sous-ensemble atteint en suivant la séquence de décisions, $\bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 x_5$, qui s'interprète par la mise de x_1, x_2, x_4 à zéro et de x_3, x_5 à un où x_s désigne la variable de décision associée à w_s (voir la section 3.3.4.3). Ce sous-ensemble contient la solution construite par la somme $w_3 \oplus w_5$ ainsi que toutes les solutions pouvant être construites par l'ajout de tout co-cycle restant à cette somme [BOU06b].

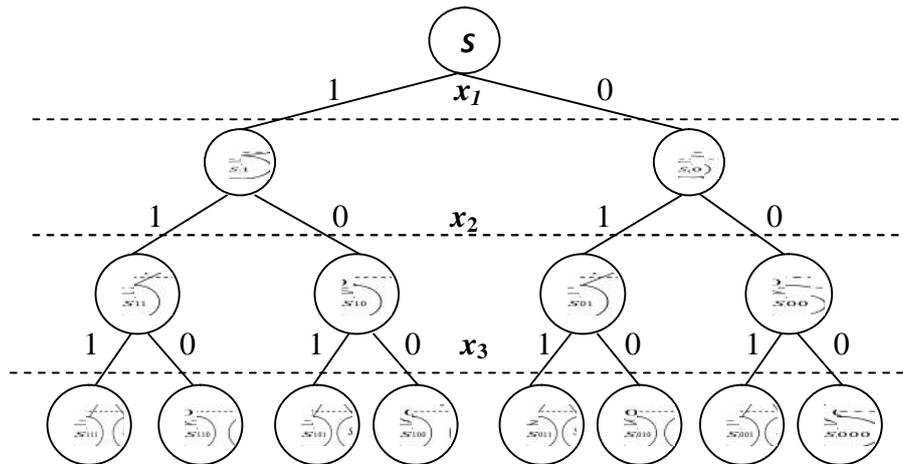


Figure 5.2. L'arbre de solutions sans pénalité

La figure 5.2 illustre un exemple d'un arbre des solutions résultant de la procédure de séparation. Dans cet exemple, nous supposons qu'il ya trois machines.

5.3.2.2. Détermination de la liste des co-cycles

Pour résoudre le problème FC de façon optimale en utilisant la méthode B&B, nous devons assurer la complétude du procédé de séparation. L'énumération de toutes les solutions possibles nécessite, a priori, la génération de la totalité des co-cycles. La technique proposée par Boulif et Atif [BOU06b] crée une liste de co-cycles d'une taille limitée qui ne garantit pas l'optimalité de la méthode proposée. Nous proposons dans ce qui suit une technique qui permet de générer la totalité de la liste des co-cycles.

En utilisant la représentation par sommets d'un co-cycle (voir l'annexe B), nous pouvons énumérer tous les co-cycles du graphe G de façon directe. Il suffit de déterminer toutes les combinaisons différentes pour m machines.

Donc, nous pouvons déterminer tous les co-cycles d'un graphe en énumérant les nombres entiers de 1 à $2^{m-1}-1$ codés en binaire. Puisque le branchement est basé sur la représentation par arêtes d'un co-cycle (voir l'annexe B), une conversion de chaque co-cycle de la représentation par sommets à une représentation par arêtes est effectuée. Différentes combinaisons de co-cycles peuvent donner le même sous ensemble, ce qui pose un problème de redondance. Afin d'atténuer ce problème, nous éliminons certains co-cycles de la liste des co-cycles. Un co-cycle étant une combinaison de co-cycles élémentaires sans arcs communs, nous pouvons donc ignorer tout co-cycle non élémentaire.

L'organigramme de la génération de la liste des co-cycles est représenté par la figure 5.3. L'exemple de la figure 3.2 donne la liste des co-cycles représentée par le tableau 5.1. Les co-cycles trois et douze seront supprimés car ils sont non élémentaires.

Numéro de co-cycle	Représentation par sommets					Représentation par arêtes							
	M_5	M_4	M_3	M_2	M_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
1	0	0	0	0	1	1	1	1	0	0	0	0	0
2	0	0	0	1	0	0	0	0	1	1	1	0	0
3	0	0	0	1	1	1	1	1	1	1	1	0	0
4	0	0	1	0	0	1	0	0	1	0	0	1	0
5	0	0	1	0	1	0	1	1	1	0	0	1	0
6	0	0	1	1	0	1	0	0	0	1	1	1	0
7	0	0	1	1	1	0	1	1	0	1	1	1	0
8	0	1	0	0	0	0	1	0	0	1	0	0	1
9	0	1	0	0	1	1	0	1	0	1	0	0	1
10	0	1	0	1	0	0	1	0	1	0	1	0	1
11	0	1	0	1	1	1	0	1	1	0	1	0	1
12	0	1	1	0	0	1	1	0	1	1	0	1	1
13	0	1	1	0	1	0	0	1	1	1	0	1	1
14	0	1	1	1	0	1	1	0	0	0	1	1	1
15	0	1	1	1	1	0	0	1	0	0	1	1	1

Tableau 5.1. La liste complète des co-cycles

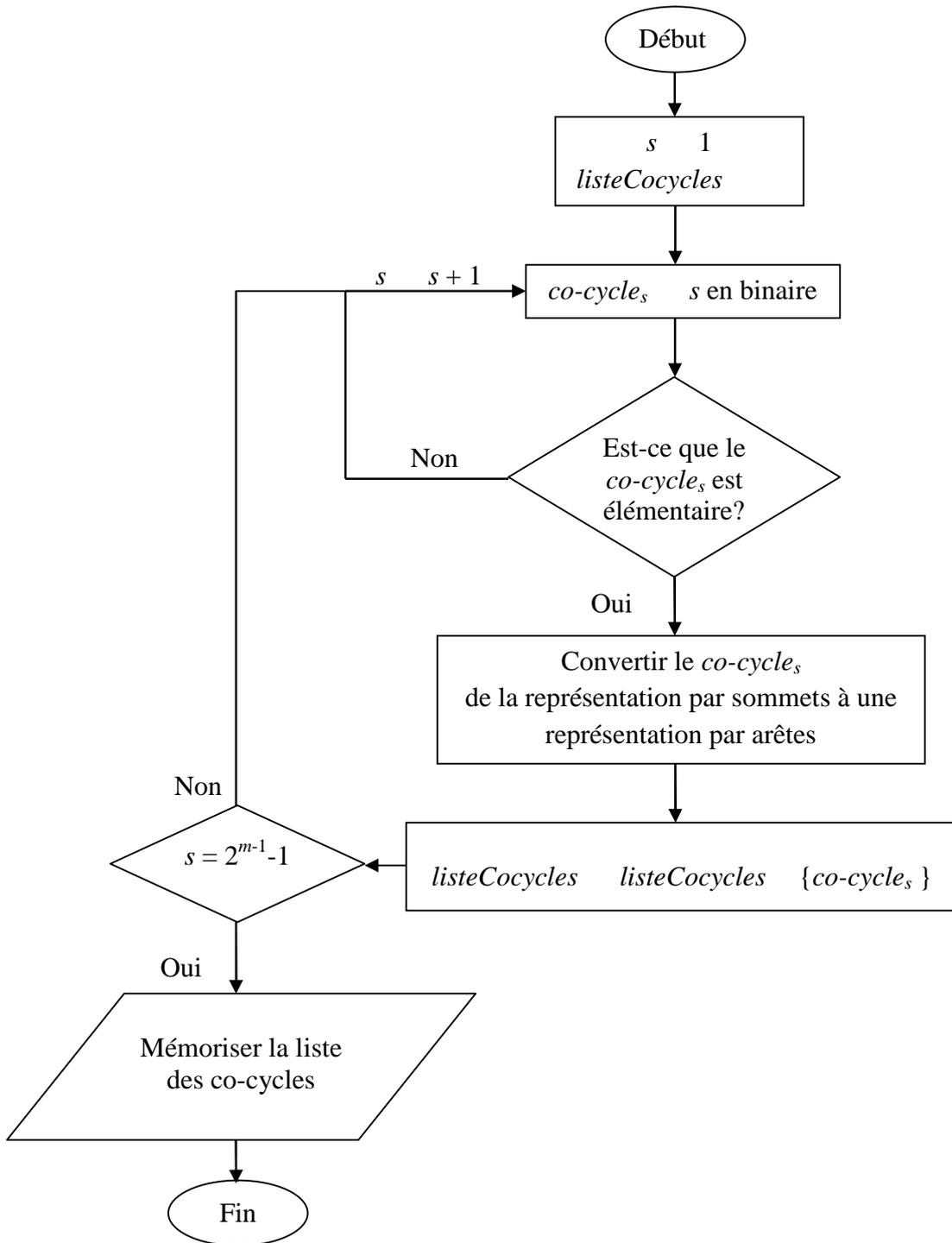


Figure 5.3. Algorithme de génération de la liste des co-cycles

5.3.3. Déplacement dans l'arborescence

Nous utilisons une Recherche en Profondeur D'abord (RPD) [PAP81] pour explorer la structure de l'arbre décrite ci-dessus. Formellement, RPD est une recherche qui progresse en élargissant le premier sous-nœud, qui apparaît, de l'arbre d'énumération et allant ainsi progressivement plus profond jusqu'à ce que l'une des pénalités est atteinte. Ensuite, la recherche de retour arrière vers le nœud le plus récent qui n'a pas encore été examiné.

5.3.4. L'Évaluation de la borne inférieure

La fonction objectif de la solution définie par la somme des co-cycles ayant des variables de décision égales à 1 est utilisée pour établir une évaluation de la borne inférieure (*BI*) de sous-ensemble [BOU06b].

Lorsqu'un co-cycle w_s est utilisé dans la séparation d'un sous-ensemble en deux sous-ensembles enfants. La *BI* de l'enfant associé à une variable de décision nulle est égale à la *BI* du parent, tandis que la *BI* de l'autre enfant prend la somme de la *BI* du parent lorsqu'elle n'est pas par défaut et le poids des arêtes de w_s qui n'appartiennent pas aux co-cycles qui composent le sous-ensemble de solutions du parent.

L'ajout des co-cycles à une solution réalisable définie conduit à la dégradation de cette solution par l'augmentation de son nombre de cellules et son trafic intercellulaire. Par conséquent, nous arrêtons la procédure de séparation dès qu'une solution réalisable est définie.

Pour l'évaluation par défaut des sous-ensembles associés qu'aux variables de décision nulles, nous utilisons la somme de poids des K premières arêtes de poids minimum. Ici, K étant le nombre minimum d'arêtes pouvant définir un co-cycle.

Afin de clarifier la procédure de calcul de la *BI*, nous considérons l'exemple décrit dans le tableau 3.1. Nous calculons la *BI* de S , S_I et S_{11} (voir figure 5.4), en utilisant les co-cycles $w_1 = (0,1,1,1,0,0,1,0)$ et $w_2 = (0,1,1,0,1,1,1,0)$ (voir tableau 5.1).

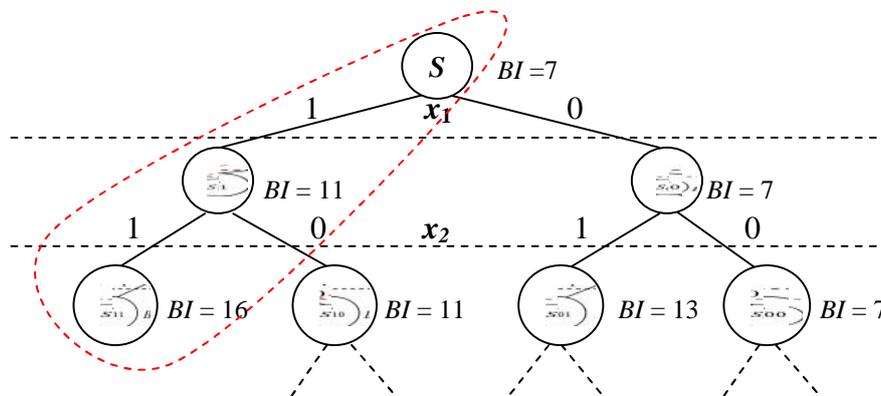


Figure 5.4. Exemple de calcul de la *BI*

S représente le nœud racine de l'arbre de solutions ainsi une évaluation par défaut est appliquée. Pour K égal à trois, la *BI* est égale à sept. S_I correspond au sous-ensemble de solutions construit en utilisant le co-cycles w_1 . Le poids des arêtes de w_1 (c.-à-d., arêtes e_2, e_3, e_4 et e_7) définit l'évaluation de l'enfant puisque le parent a une évaluation par défaut. La valeur de sa *BI* est donc égale à onze.

Enfin, S_{11} est le sous-ensemble de solutions construit par la somme de w_1 et w_2 . La valeur de sa *BI* est égale à l'évaluation du père, à laquelle nous ajoutons le poids des arêtes additionnelles de w_2 (c.-à-d., arêtes e_5 et e_6), ce qui donne un *BI* de seize.

5.3.5. Pénalité

Un sous-ensemble n'a pas besoin d'être traité plus loin une fois que l'une des conditions suivantes se produit:

- La BI de sous-ensemble est plus que la borne supérieure courante (la fonction objective de la meilleure solution réalisable trouvée pour l'instant).
- La solution associée au sous-ensemble n'est pas réalisable en raison d'une contrainte autre que la taille de la cellule ou la non cohabitation.
- Une solution réalisable est obtenue. Si la valeur de la fonction objectif de cette solution est inférieure à la borne supérieure courante, elle sera mise à jour.

Remarque 5.1: La borne supérieure peut être antérieurement établie en utilisant la fonction objectif d'une solution réalisable obtenue par une heuristique.

5.4. Amélioration de la méthode SE cocyclique

Pour être performante dans la résolution d'un problème donné, une méthode doit tirer profit des informations spécifiques à ce problème. En effet, ces informations peuvent être exploitées afin de guider la recherche de solutions. C'est dans ce sens que nous présentons la méthode SE cocyclique améliorée.

L'amélioration entraînée par les contraintes proposée ici conduit à une réduction de la taille de la liste des co-cycles et, par conséquent, à une diminution du nombre total de nœuds générés. De plus, nous présentons une nouvelle BI basée sur l'algorithme de coupe minimum. Cette BI plus resserrée stimule le processus de pénalité et produit des gains d'efficacité.

5.4.1. Amélioration basée sur le nombre maximum de cellules autorisé

Dans notre modèle, le nombre de cellules n'est pas prédéterminé. Cette liberté permet à la méthode de résolution de chercher la meilleure répartition des cellules quel que soit leur nombre. Cependant, puisque nous n'autorisons pas les cellules vides, il ne peut pas y avoir plus de m cellules (m est le nombre de machines). Ainsi, Nous pouvons voir que le fait de prendre seulement $m - 1$ co-cycles suffira pour construire toute solution. Cela nous amène à conclure que la procédure de branchement peut être arrêtée dès que le nombre de variables de décision mises à un est égal à $m - 1$. Si nous prenons un exemple de trois machines, l'arbre de solutions précédent sera tel que représenté sur la figure 5.5.

Comme le montre la figure 5.5, le nombre total de nœuds est réduit à treize, ce qui est moins que quinze nœuds de la figure 5.2.

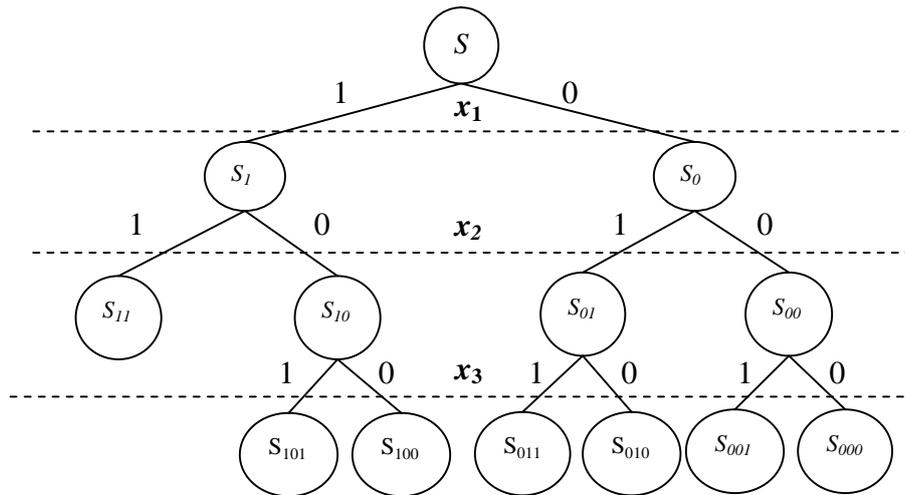


Figure 5.5. L'arbre de solutions réduit

5.4.2. Amélioration basée sur le nombre maximum de machines autorisé dans une cellule

Définition 1 (constructeur-cellule): nous définissons un co-cycle comme étant un *constructeur-cellule* si et seulement si la cardinalité de l'un des deux sous-ensembles de sommets associés est inférieure ou égale au nombre maximum de machines admis dans une cellule.

Définition 2 (constructeur-cellule exact): nous définissons un *constructeur-cellule* comme étant *exact* si et seulement si la cardinalité de l'un des deux sous-ensembles de sommets associés est égale au nombre maximum de machines admis dans une cellule.

Il est assez simple de montrer que nous pouvons énumérer toutes les solutions possibles en ne considérant que des *constructeurs-cellule*. Cette réduction ne modifie pas la recherche d'exhaustivité, puisque chaque solution réalisable peut être représentée par une somme de tels co-cycles (qu'ils soient élémentaires ou non).

Un exemple est présenté ici pour clarifier l'idée de cette amélioration. Pour N égal à deux (N est le nombre maximum de machines autorisé dans une cellule), la solution à quatre cellules (voir figure 5.6(a)) construite par la somme de $w_1 = (0,1,0,1,1,0,1,0,1)$ et $w_2 = (0,1,1,0,1,1,1,0,0)$ peut être construite également par la somme de $w_3 = (0,0,1,0,0,0,1,0,1)$, $w_4 = (0,1,1,1,1,0,0,0,0)$ et $w_5 = (0,1,0,0,1,1,0,0,1)$ (voir figure 5.6(b)).

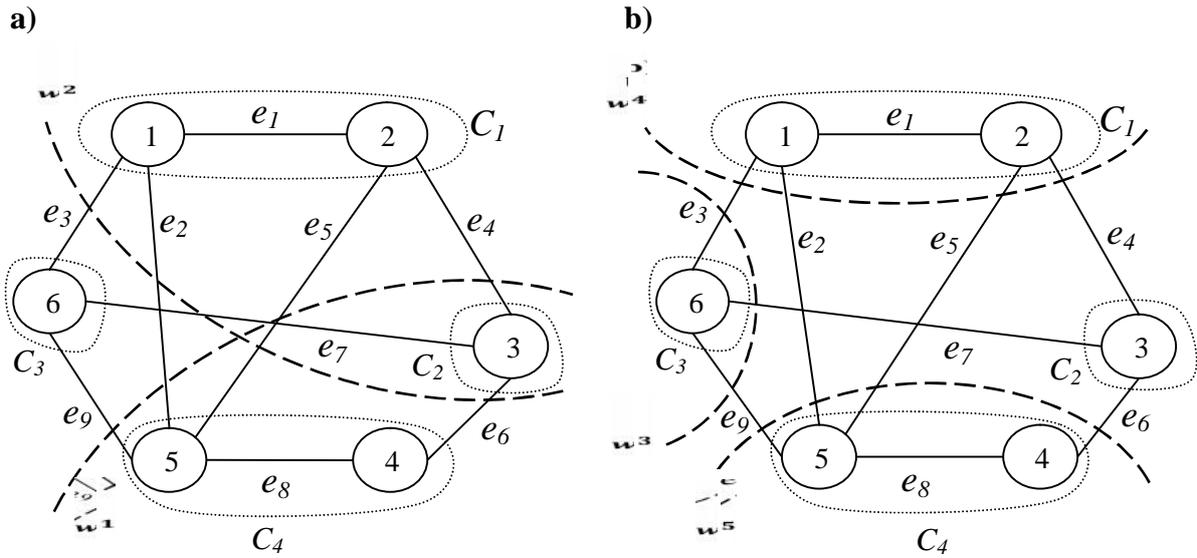


Figure 5.6. Une solution construite par: (a) des co-cycles arbitraires b. des constructeurs-cellule

Pour éviter l'impression que seuls les constructeurs-cellule exacts assurent l'exhaustivité de la recherche, considérons le graphe de flux de la figure 5.7 comportant quatre machines, quatre arêtes et N est supposé égal à trois. La solution optimale minimisant le trafic intercellulaire est représentée par la composition suivante: $C_1 = \{M_1, M_2\}, C_2 = \{M_3, M_4\}$. Les constructeurs-cellule exacts sont incapables de représenter cette solution.

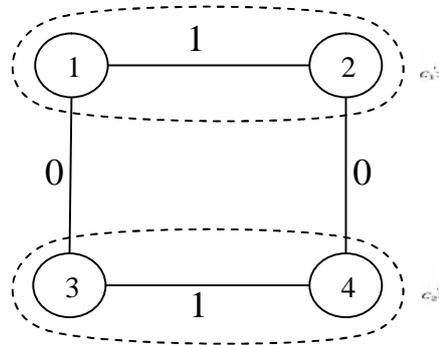


Figure 5.7. Solution optimale invisible

5.4.3. Amélioration basée sur les contraintes de cohabitation et de non cohabitation

Pour chaque couple de machines à cohabiter, il y'a deux valeurs *identiques* dans la représentation par sommets. Pour l'exemple de la figure 3.2, si M_1 et M_5 sont liées par une contrainte de cohabitation alors nous avons $w_s = (0, ?, ?, ?, 0)$ ou $w_s = (1, ?, ?, ?, 1)$.

De plus, pour chaque couple de machines à ne pas cohabiter, il y'a deux valeurs *différentes* dans la représentation par sommets. Pour le même exemple, si M_4 et M_5 sont liées par une contrainte de non cohabitation alors nous avons $w_s = (?, ?, ?, 1, 0)$ ou $w_s = (?, ?, ?, 0, 1)$.

Ainsi, nous pouvons comprendre que pour la représentation par sommets certaines valeurs peuvent être déterminées en fonction d'autres valeurs. Par conséquent, le fait de prendre toutes les machines dans la procédure de génération de la liste des co-cycles était inutile.

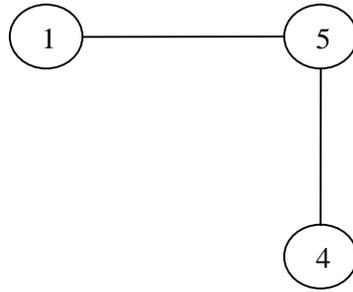


Figure 5.8. Graphe de cohabitation et de non cohabitation

Pour illustrer les avantages de cette amélioration, nous considérons l'exemple de la figure 3.2. Les machines M_1 et M_5 doivent être réunies alors que M_4 et M_5 doivent être séparées. Ce peut être représenté par le graphe illustré dans la figure 5.8. Dans ce graphe de cohabitation et non cohabitation, toutes les machines qui apparaissent dans des contraintes de cohabitation et/ou de non cohabitation sont représentées et il ya un lien entre deux machines si elles sont soumises à des contraintes de cohabitation ou de non cohabitation. Pour chaque composante connexe de ce graphe, nous prenons au hasard une machine représentante. Ensuite, nous n'utilisons que ces représentantes avec le sous-ensemble de machines qui ne sont pas concernées par les contraintes de cohabitation et de non cohabitation pour définir la représentation par sommets. Cette représentation réduite sera utilisée pour générer la liste de co-cycles.

Numéro de co-cycle	Représentation par sommets requise			Représentation par sommets complète					Représentation par arêtes							
	M_3	M_2	M_1	M_5	M_4	M_3	M_2	M_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1
2	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1
3	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1
4	0	1	1	1	0	0	1	1	1	1	0	1	1	0	1	1

Tableau 5.2. La liste des co-cycles réduite

Revenons à notre exemple, si nous prenons M_1 en tant que représentante de la composante connexe $\{M_1, M_4, M_5\}$, seules trois machines seront utilisées dans la génération de co-cycles. La liste complète de co-cycles générés est donnée dans le tableau 5.2. Avec trois machines, nous n'avons que quatre co-cycles, ce qui est beaucoup moins que les quinze du tableau 5.1.

5.4.4. Amélioration de la BI

Dans cette section, nous tenons à améliorer la procédure de calcul de la borne inférieure car si elle est efficacement serrée, elle améliorera considérablement la performance de la méthode SE.

Si la solution liée au sous-ensemble actuel ne respecte pas la contrainte de la taille des cellules, il faudra diviser les cellules qui dépassent le nombre autorisé de machines. Boulif et Atif [BOU06c] ont proposé d'ajouter le poids le plus faible des arêtes intracellulaires de chacune de ces cellules à l'évaluation de base. Nous proposons plutôt d'ajouter le coût de la coupe minimum de ces sous-graphes de cellules. Pour identifier le coût d'une coupe minimum, nous pouvons utiliser l'un des algorithmes efficaces disponibles dans la littérature.

Dans notre implémentation, nous avons utilisé l'algorithme polynomial décrit par Stoer et Wagner [STO97]. Pour la *BI* par défaut, la procédure fondée sur la coupe minimum peut également être utilisée car elle est plus serrée que celle proposée par Boulif et Atif [BOU06c] et plus efficace.

Afin de clarifier l'amélioration de la procédure de calcul de la *BI* de la méthode SE, nous considérons la même instance de la section 5.3.4 avec au plus deux machines par cellule. Le co-cycle w_1 divise le graphe initial en deux sous-graphes G_1 et G_2 (voir figure 5.1). La contrainte de la taille des cellules n'est pas respectée par G_2 dans ce cas (voir figure 5.9).

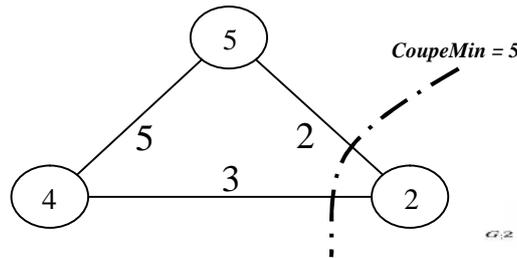


Figure 5.9. La coupe minimum de G_2

Ainsi, nous ajoutons le coût de la coupe minimum de G_2 à l'évaluation de base de S_1 (voir figure 5.10).

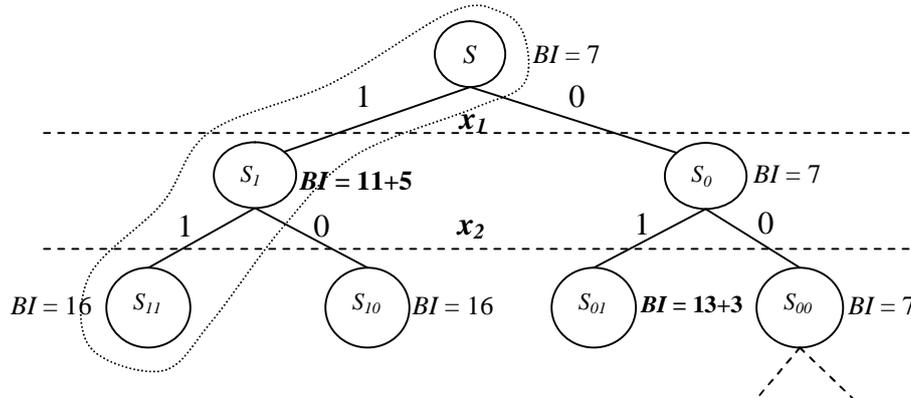


Figure 5.10. Un exemple de l'amélioration de la *BI*

5.5. Conclusion

Dans ce chapitre, nous avons présenté une méthode SE pour le problème FC. Cette méthode utilise une procédure de branchement basée sur les co-cycles. Les performances de cette méthode ont été améliorées par la suite grâce à une exploitation des informations spécifiques à ce problème. En effet, nous avons développé une deuxième méthode SE basant sur une procédure de branchement qui met en œuvre pour la première fois, au mieux de nos connaissances, une amélioration guidée par les contraintes et sur une procédure d'évaluation qui utilise un algorithme de coupe minimum.

Le chapitre suivant sera consacré à la présentation des résultats obtenus par les méthodes présentées dans ce chapitre et la comparaison de leurs performances par rapport à celles déjà existantes.

Chapitre 6

Résultats et Analyse

6.1. Introduction

Dans ce chapitre, nous décrivons d'abord les ressources informatiques utilisées, ensuite nous présentons les expérimentations effectuées sur les méthodes SE proposées dans le chapitre précédent et les modèles mathématiques présentés dans le chapitre 3, et enfin nous analysons les différents résultats obtenus, en se basant sur quatre mesures de performance.

6.2. Environnement

6.2.1. Ressources informatiques utilisées

Les méthodes SE ont été codées en C++ et à l'aide de compilateur intégré MinGW. Les tests ont été effectués sur un ordinateur personnel avec un processeur Intel Core2 Duo à 2,5 GHz, 2 Go de RAM.

Deux solveurs libres ont été utilisés pour résoudre les modèles mathématiques et comparer leur efficacité. Ces solveurs sont GLPK version 4.34 et SCIP version 3.02.

GLPK (GNU Linear Programming Kit) est un outil performant pour résoudre des problèmes d'optimisation linéaires en variables continues ou mixtes. Il est composé de:

- Un langage de modélisation GNU Mathlog;
- Une librairie de fonctions en C;
- Un solveur GLPSol.

Il est disponible sur le site internet suivant:

<http://www.gnu.org/software/glpk/>

Il intègre les composants suivants:

- La méthode du Simplexe;
- La méthode de points intérieurs primal-dual;
- La méthode de Branch and Bound.

SCIP (Solving Constraint Integer Programs) est un Framework pour la manipulation des algorithmes de coupes, génération de colonnes et branchement (Branch-Cut-and-Price). Il est parmi les solveurs non commerciaux les plus performants [MIT15]. Il s'agit d'un produit universitaire de l'université ZIB de Berlin:

<http://SCIP.zib.de/>

Il est composé de:

- Un langage de modélisation ZIMPL;

Ce chapitre est basé sur l'article [MER14]

- Un solveur de programme linéaire SOPLex basé sur Branch-and-Cut.
- Un solveur Branch-Cut-and-Price GCG;
- Un Framework parallèle pour résoudre les programmes linéaires ou non linéaires en variables mixtes.

6.2.2. Données des exemples

Nous évaluons les performances des méthodes SE et l'efficacité des modèles mathématiques sur un ensemble de seize exemples (voir l'annexe C). Le tableau 6.1 présente les données des exemples numériques, y compris le nombre de produits, nombre de machines, nombre maximal de machines dans une cellule et les contraintes de cohabitation et de non cohabitation. Les exemples sont classés selon le nombre de machines.

N°Exemple	Nombre de machines	Nombre de produits	Taille de cellule	Contraintes	
				Contrainte de Cohabitation	Contrainte de non-cohabitation
1	3	5	2
2	5	5	2	...	M_1, M_5
3	6	17	3	M_2, M_6	...
4	7	7	3	...	M_2, M_5 & M_2, M_6
5	8	20	4
6	9	6	3	M_2, M_7	M_7, M_8
7	10	12	4	M_5, M_8	M_2, M_7
8	12	16	4	M_8, M_4 & M_8, M_9	M_2, M_{11}
9	12	19	4	...	M_7, M_{12}
10	15	20	5	M_6, M_{15}	M_8, M_{11} & M_{15}, M_5
11	17	10	5	M_1, M_6 & M_6, M_{10}	M_{11}, M_{17}
12	18	35	6	M_9, M_{16}	M_2, M_{12}
13	20	23	7	M_2, M_3 & M_3, M_6	M_2, M_{10} & M_4, M_{16}
14	20	20	5
15	20	51	5
16	25	40	4

Tableau 6.1. Les données des exemples numériques

Les cinquième, quatorzième et seizième exemples sont tirés de [JAY98]. Les neuvième, douzième et quinzième sont cités dans [VAK90, SU98 et NAG90], respectivement. Le reste des exemples sont générés aléatoirement. Il est important de noter que les contraintes de cohabitation et de non cohabitation ainsi que la taille maximale des cellules des exemples tirés de [JAY98] ont été ajoutées par Boulif et Atif [BOU06b].

6.3. Tests effectués

6.3.1. Tests sur la méthode SE cocyclique

Avant d'effectuer nos tests sur la méthode *SE cocyclique* proposée, nous avons associé une routine à chaque amélioration entraînée par une contrainte. La première, que nous appelons *maxnc*, met en œuvre l'amélioration basée sur le nombre maximum de cellules autorisé (voir section 5.4.1). La seconde, *maxnm*, implémente l'amélioration basée sur le nombre maximum de machines autorisé dans une cellule (voir section 5.4.2). La troisième, *cnc*, met en pratique l'amélioration basée sur les contraintes de cohabitation et de non-cohabitation (voir section 5.4.3). La dernière, *mc*, emploie la procédure de calcul de la borne inférieure fondée sur la coupe minimum (voir section 5.4.4).

Pour tester la performance de notre méthode, nous effectuons deux types d'expérimentations. D'abord, nous proposons d'évaluer la performance de la méthode SE cocyclique lorsque les différentes routines sont introduites. Plus précisément, nous comparons les performances des méthodes suivantes : *BB* pour la méthode *SE co-cyclique* telle que décrite par Boulif et Atif [BOU06b], *CBB* pour *BB* améliorée par les trois premières routines telle que décrite par Boulif et Atif [BOU06c], mais en prenant seulement un seul critère, *CBBmc1* pour *CBB* améliorée par la dernière routine pour le cas *mono-cellule* (La routine est appliquée seulement à l'une des cellules qui dépassent le nombre autorisé de machines par cellule), *CBBmc2* pour *CBB* améliorée par la dernière routine pour le cas *multi-cellules* (la routine est appliquée pour toutes les cellules dont la taille dépasse le nombre de machines admis dans une cellule). Ensuite, nous testons l'impact de chacune des trois premières routines sur les performances de la méthode *CBBmc2*. Pour ce faire, nous comparons les trois méthodes suivantes : *CBBmc2-nomaxnc* pour *CBBmc2* avec la première routine désactivée, *CBBmc2-nocnc* pour *CBBmc2* avec la troisième routine désactivée et *CBBmc2-nomaxnm* pour *CBBmc2* avec la deuxième routine désactivée, par rapport à la méthode *CBBm2*.

6.3.2. Tests sur les modèles proposés

Afin de confirmer les résultats théoriques obtenus de la comparaison des modèles mathématiques *L1*, *L2*, *L2'* et *L3* présentés dans le chapitre 3, nous résolvons ces modèles en utilisant deux solveurs non commerciaux : GLPK et SCIP. Notons que le solveur GLPK utilise la méthode B&B pour résoudre des programmes linéaires en nombres entiers tandis que le solveur SCIP utilise la méthode Branch-and-Cut qui conjugue les efforts de la méthode B&B et B&C (voir le chapitre 4 pour plus de détails sur ces deux méthodes).

Pour les résultats obtenus de ces tests, en plus des informations conclues sur l'efficacité de chaque modèle, nous avons aussi une idée sur les performances des deux solveurs dans la résolution du problème FC.

6.3.3. Comparaison avec les autres méthodes

Plusieurs raisons empêchent la comparaison de notre méthode avec celles proposées dans la littérature. Citons par exemple:

- La différence dans les fonctions objectif;
- L'utilisation de certaines informations permettant de simplifier le problème telles que le nombre de cellules à former;

- La non utilisation de certaines informations utiles telles que la séquence d'opérations, le volume de production et les machines à, ou à ne pas, cohabiter.

Pour cela, nous proposons de comparer les résultats obtenus par la méthode *CBBm2* avec les meilleurs résultats obtenus de la comparaison faite entre les modèles mathématiques c.à.d. nous comparons la méthode *CBBm2* avec la méthode B&B de GLPK appliquée au modèle *L3*, notée *GLPK3*, la méthode B&C de SCIP appliquée au modèle *L1*, notée *SCIP1*, ainsi que la méthode B&C de SCIP appliquée au modèle *L2'*, notée *SCIP2'*.

Comme il est difficile de trouver des solutions exactes dans un horizon proche pour les instances de grande taille, nous considérons un maximum de deux heures de temps lors de l'exécution de chacun des tests cités ci-dessus sur les exemples. En d'autres termes, si un test échoue à obtenir l'optimum global dans les deux heures, alors la meilleure solution trouvée sera rapportée.

6.4. Analyse des résultats

Avant de procéder à l'analyse des résultats des tests effectués, nous présentons d'abord les mesures de performance que nous avons adoptées pour évaluer notre méthode.

6.4.1. Mesures de performance

Pour effectuer les différentes comparaisons citées ci-dessus, nous avons défini quatre principales mesures de performance comme base de comparaison:

- Temps de résolution (CPU en secondes);
- La valeur de la solution optimale / meilleure solution trouvée par la méthode (FO).
- Le nombre des exemples résolus à l'optimum.
- L'écart (GAP en %) relatif entre la meilleure solution trouvée (*MS*) et la solution optimale (*SO*) calculé selon la formule suivante:

$$GAP = \frac{MS - SO}{MS} \times 100$$

6.4.2. Analyse des résultats de la méthode *SE cocyclique*

Dans cette section, nous comparons les méthodes *BB*, *CBB*, *CBBm1*, *CBBm2*, *CBBm2-nomaxnc*, *CBBm2-nocnc* et *CBBm2-nomaxnm*, en tenant compte du temps d'exécution moyen et du nombre des exemples résolus. Le tableau 6.2 résume les résultats obtenus pour chacune des versions de la méthode *SE cocyclique*. La colonne 1 du tableau reporte le numéro de l'exemple traité. Le contenu des colonnes 2 à 15 reportent: le temps d'exécution et la valeur de la solution optimale / la meilleure solution trouvée par chacune des versions.

A la suite de ces résultats, nous pouvons remarquer que:

1. L'amélioration de la méthode *SE cocyclique* par les trois premières routines a un bon comportement, puisqu'elle permet de réduire le temps d'exécution moyen (1465,03 secondes) et de résoudre les dixième, onzième, douzième et treizième exemples, ce qui était impossible avec la méthode *BB*.
2. La combinaison des trois premières routines et la routine *mc* appliquée au cas *mono-cellule* fait augmenter la performance de la méthode *SE cocyclique*. En effet, cette fois-ci

le temps d'exécution décroît (qui vaut 1088,62 secondes en moyenne) et le nombre des exemples non résolus diminue (il passe à 2).

3. La méthode *CBBmc2*, qui combine les trois premières routines et la routine *mc* appliquée en cas *multi-cellules*, peut être considérée comme la version la plus performante (voir figure 6.1), parce qu'elle permet un temps d'exécution moyen légèrement meilleur (1086,16 secondes versus 1088,62 secondes pour la méthode *CBBmc1*).

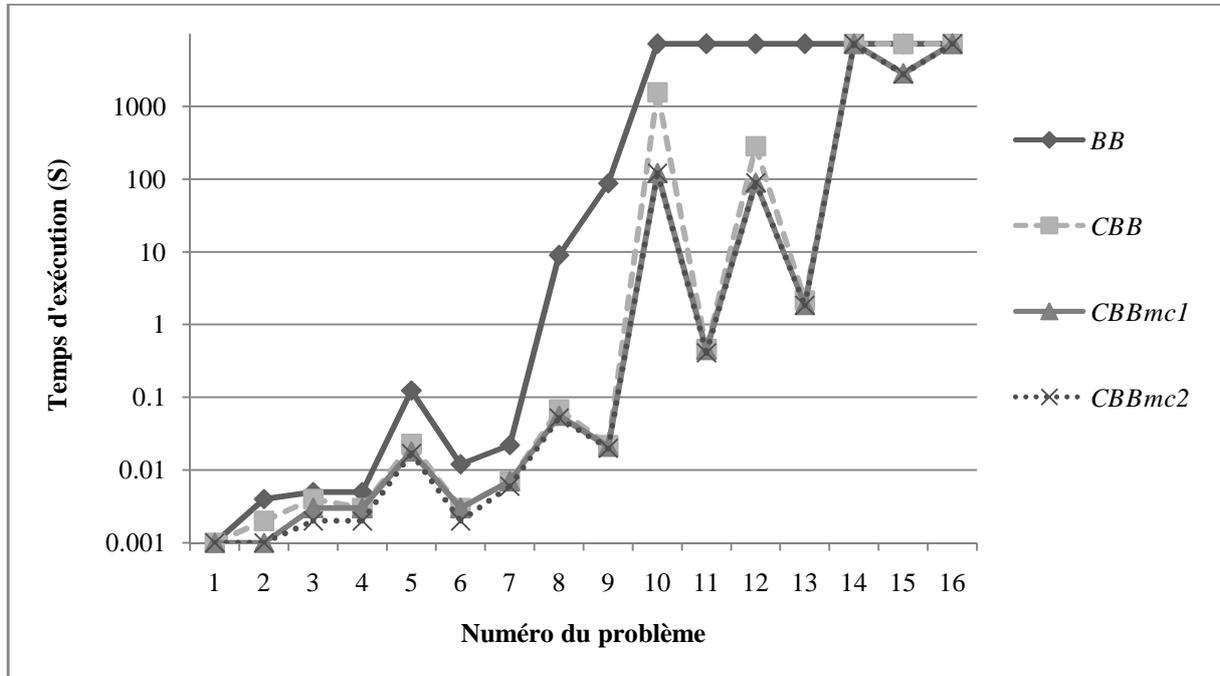


Figure 6.1. Comparaison des versions de la méthode SE cocyclique

De plus, nous pouvons remarquer des résultats de la deuxième expérimentation que:

- 1- La méthode *CBBm2* est bien plus performante que les trois autres méthodes (voir figure 6.2). En effet, elle permet d'obtenir les solutions optimales / meilleures solutions après près de 18 minutes et 10 secondes en moyenne, tandis que les méthodes *CBBm2-nomaxnc*, *CBBm2-nocnc* et *CBBm2-nomaxnm* prennent en moyenne près de 18 minutes et 15 secondes, 33 minutes et 27 minutes respectivement.
- 2- Le temps d'exécution moyen de la méthode *CBBm2* a légèrement augmenté après la désactivation de la première routine, essentiellement en raison de la liberté donnée à la méthode *SE cocyclique* afin de rechercher la meilleure répartition des cellules quel que soit leur nombre, mais nous estimons que cette routine aura plus d'influence sur les performances de notre méthode si nous bornons supérieurement le nombre de cellules avec une valeur bien définie.
- 3- La suppression de la routine 2 permet de ralentir la résolution des exemples, en particulier, l'exemple 15 où la méthode atteint le temps limite que nous avons fixé avant d'arriver à la solution optimale.
- 4- La méthode *CBBm2* sans la routine 3 donne un temps d'exécution assez grand relativement à la méthode *CBBm2*, notamment pour les exemples 10 et 12 où la méthode arrive à la limite de temps avant de trouver la solution optimale.
- 5- Pour le nombre des exemples non résolus à l'optimum, Les méthodes *CBBm2* et *CBBm2-nomaxnc* ne permettent pas de résoudre deux exemples parmi les seize exemples traités,

tandis que nous avons quatre et trois exemples non résolus pour les méthodes *CBBm2-noncnc* et *CBBm2-nomaxnm* respectivement.

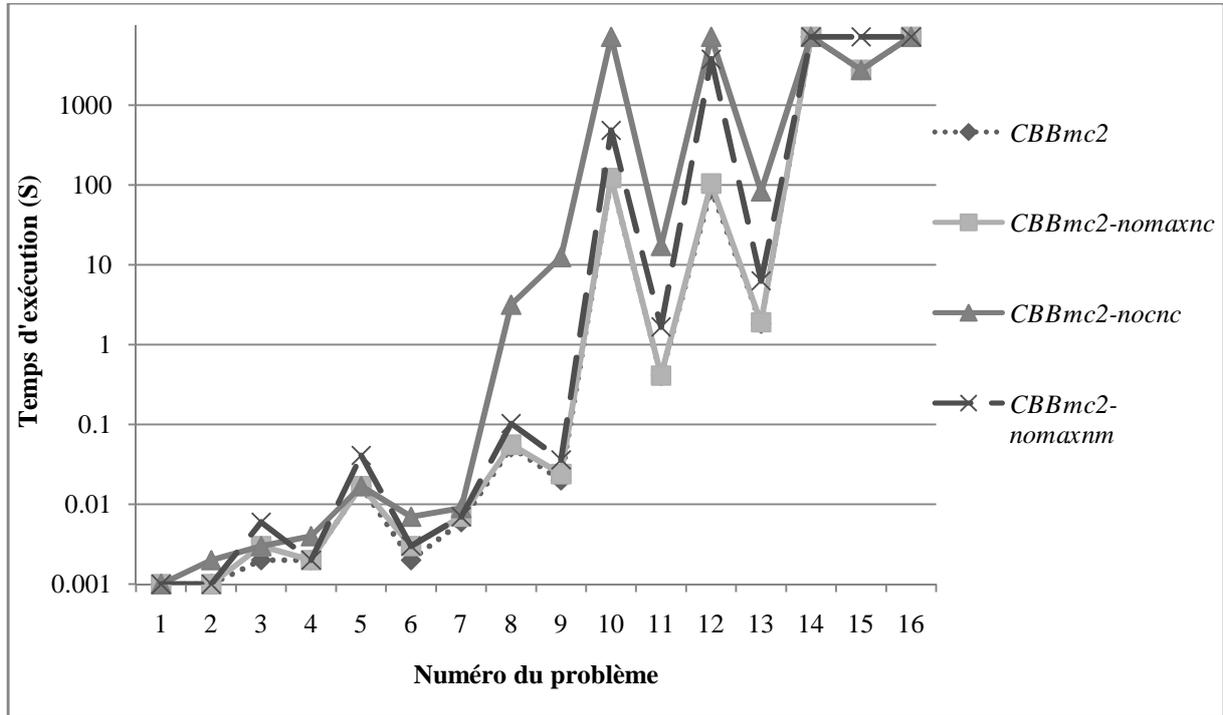


Figure 6.2. Impact des améliorations entraînées par les contraintes

Numéro de l'exemple	BB		CBB		CBBmc1		CBBmc2		CBBmc2-nomaxnc		CBBmc2-nocnc		CBBmc2-nomaxnm	
	CPU	FO	CPU	FO	CPU	FO	CPU	FO	CPU	FO	CPU	FO	CPU	FO
1	0	8	0	8	0	8	0	8	0	8	0	8	0	8
2	0,004	27	0,002	27	0,001	27	0	27	0,001	27	0,002	27	0,001	27
3	0,005	57	0,004	57	0,003	57	0,002	57	0,003	57	0,003	57	0,006	57
4	0,005	5	0,003	5	0,003	5	0,002	5	0,002	5	0,004	5	0,002	5
5	0,124	13	0,023	13	0,018	13	0,017	13	0,017	13	0,017	13	0,041	13
6	0,012	5	0,003	5	0,003	5	0,002	5	0,003	5	0,007	5	0,003	5
7	0,022	6	0,007	6	0,007	6	0,006	6	0,007	6	0,009	6	0,007	6
8	8,9625	41	0,068	41	0,056	41	0,052	41	0,056	41	3,168	41	0,103	41
9	87,341	42	0,022	42	0,021	42	0,020	42	0,024	42	12,479	42	0,036	42
10	7200	107*	1555,4	93	121,128	93	120,887	93	122,95	93	7200	93*	486,035	93
11	7200	20*	0,46	20	0,448	20	0,408	20	0,412	20	17,253	20	1,676	20
12	7200	74*	282,495	56	89,161	56	89,16	56	104,77	56	7200	74*	3761	56
13	7200	36*	2,141	36	1,854	36	1,835	36	1,909	36	83,493	36	6,33	36
14	7200	26*	7200	26*	7200	26*	7200	26*	7200	26*	7200	26*	7200	26*
15	7200	120*	7200	120*	2805,33	82	2766,17	82	2786,11	82	2766,17	82	7200	120*
16	7200	44*	7200	44*	7200	44*	7200	44*	7200	44*	7200	44*	7200	44*
Moyenne	3156,029		1465,039		1088,627		1086,160		1088,516		1980,162		1615,952	

Note: * La meilleure solution après deux heures de temps d'exécution.

Tableau 6.2. Résultats de calcul du premier test

6.4.3. Analyse des résultats des modèles

Pour comparer les modèles mathématiques $L1$, $L2$, $L2'$ et $L3$, nous nous basons à la fois sur le GAP, la durée d'exécution et le nombre d'exemples résolus. Le tableau 6.3 regroupe les performances des solveurs GLPK et SCIP appliqués à ces modèles. La colonne 1 du tableau reporte le numéro de l'exemple et le contenu des colonnes 2 à 19 reportent la solution optimale / la meilleure solution trouvée, le temps d'exécution et le GAP pour chacun des modèles. En plus de ces résultats, nous avons surveillé le nombre de variables et de contraintes générées pour chacun des exemples (voir le tableau 6.4).

À partir de ces deux tableaux, nous pouvons remarquer que:

- 1- Quel que soit le solveur appliqué au modèle $L3$ (GLPK ou SCIP), une solution optimale a été trouvée pour tous les exemples (voir figure 6.3 et figure 6.4). Par conséquent, Le modèle $L3$ vient en première position suivi par le modèle $L2'$ où tous les exemples ont été résolus par le solveur SCIP versus 13 sur 16 exemples résolus par GLPK et le modèle $L2$ où tous les exemples ont été résolus par le solveur SCIP versus 6 sur 16 exemples résolus par GLPK. Le pire nombre est celui du modèle $L1$ (13 sur 16 exemples résolus par SCIP versus 5 sur 16 exemples résolus par GLPK). En effet, cela peut se justifier par le nombre de variables générées (voir le tableau 6.4) et par l'application de principe de p-médian et la *contrainte-affectation* (voir la section 3.3.3.2).
- 2- La *contrainte-affectation* a un grand impact sur le modèle $L2'$ puisqu'elle permet d'accélérer la résolution de la pluparts des instances. Ainsi, le temps d'exécution moyen passe de 4846,6 secondes pour $GLPK2$ (resp. 143,11 secondes pour $SCIP2$) à 1659,5 secondes pour $GLPK2'$ (resp. 3,81 secondes pour $SCIP2'$).
- 3- Les mauvais résultats sont donnés par le modèle le plus utilisé dans la littérature c.à.d. le modèle $L1$. En effet, quel que soit le solveur appliqué sur ce modèle (GLPK ou SCIP), le temps d'exécution moyen est le pire.
- 4- Le solveur SCIP est plus performant que le solveur GLPK. En effet, SCIP nécessite un temps d'exécution moyen total de 529,76 secondes comparé aux 2864,59 secondes que nécessite le solveur GLPK pour effectuer tous les tests (la résolution des 16 exemples pour les 4 modèles) et il permet de trouver la solution optimale de 95% des exemples versus 62% pour le solveur GLPK. De plus, il permet d'avoir des GAPs moyens total de 0,48% versus 1,72% obtenus par le solveur GLPK. Ceci peut être justifié par le fait que le solveur SCIP utilise la méthode B&C qui est plus performante que la méthode B&B.

Nous pouvons finalement conclure que les deux modèles $L2'$ et $L3$ sont prometteurs. En effet:

- Le modèle $L3$ a permis de résoudre tous les exemples de l'étude proposée dans cette thèse quelque soit le solveur utilisé. Notons que ce modèle a été négligé par la plupart des chercheurs étudiant le problème FC.
- SCIP appliqué au modèle $L2'$ résout toutes les instances avec le meilleur temps moyen (voir le tableau 6.3).

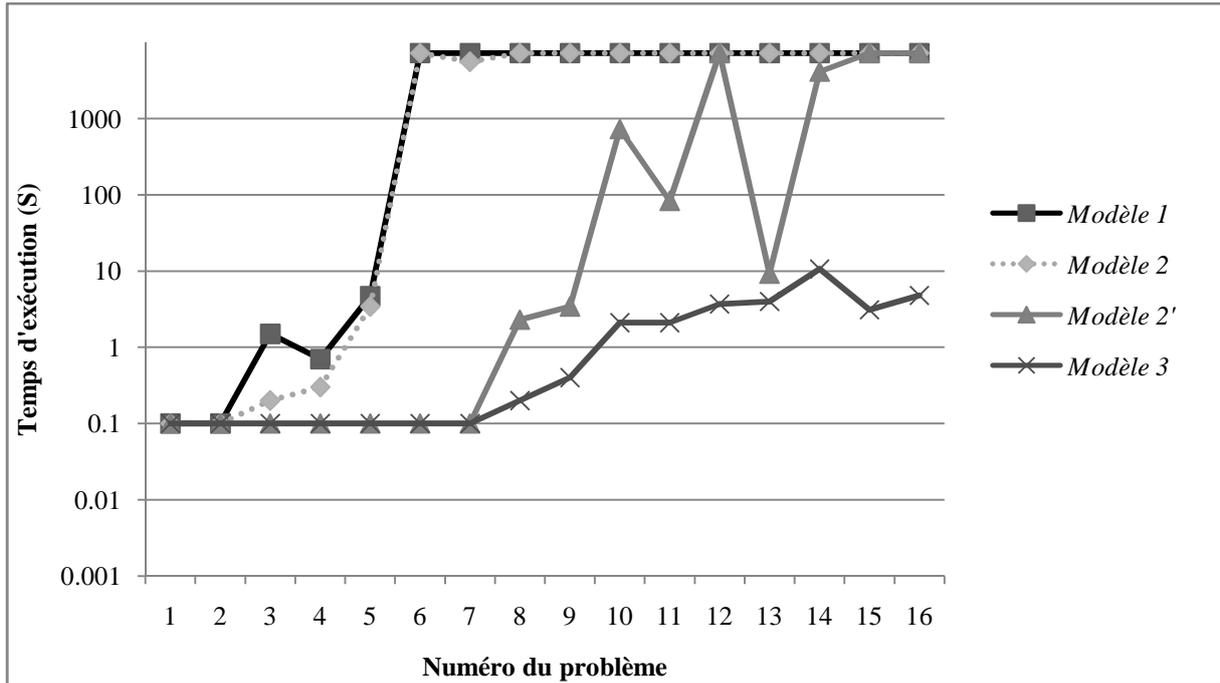


Figure 6.3. Comparaison des modèles en utilisant le solveur GLPK

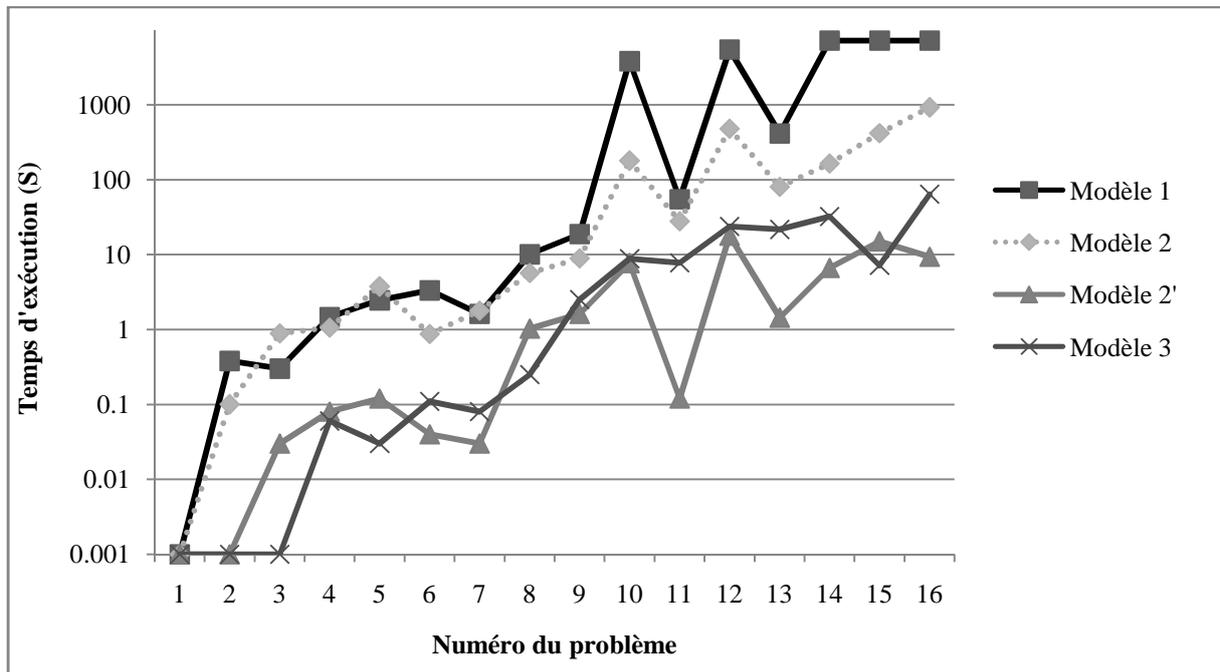


Figure 6.4. Comparaison des modèles en utilisant le solveur SCIP

Numéro de l'exemple	Modèle 1						Modèle 2						Modèle 2'						Modèle 3						CBBm2		
	GLPK			SCIP			GLPK			SCIP			GLPK			SCIP			GLPK			SCIP			FO	CPU	GAP
	FO	CPU	GAP	FO	CPU	GAP	FO	CPU	GAP	FO	CPU	GAP	FO	CPU	GAP	FO	CPU	GAP	FO	CPU	GAP	FO	CPU	GAP			
1	8	0,1	0,0	8	0	0,0	8	0,1	0,0	8	0	0,0	8	0,1	0,0	8	0	0,0	8	0,1	0,0	8	0	0,0	8	0	0,0
2	27	0,1	0,0	27	0,38	0,0	27	0,1	0,0	27	0,01	0,0	27	0,1	0,0	27	0	0,0	27	0,1	0,0	27	0	0,0	27	0	0,0
3	57	1,5	0,0	57	0,30	0,0	57	0,2	0,0	57	0,89	0,0	57	0,1	0,0	57	0,03	0,0	57	0,1	0,0	57	0	0,0	57	0,002	0,0
4	5	0,7	0,0	5	1,46	0,0	5	0,3	0,0	5	1,06	0,0	5	0,1	0,0	5	0,08	0,0	5	0,1	0,0	5	0,06	0,0	5	0,002	0,0
5	13	4,7	0,0	13	2,45	0,0	13	3,4	0,0	13	3,74	0,0	13	0,1	0,0	13	0,12	0,0	13	0,1	0,0	13	0,03	0,0	13	0,017	0,0
6	5*	7200	0,0	5	3,32	0,0	5*	7200	0,0	5	0,87	0,0	5	0,1	0,0	5	0,04	0,0	5	0,1	0,0	5	0,11	0,0	5	0,002	0,0
7	6*	7200	0,0	6	1,62	0,0	6	5542,3	0,0	6	1,78	0,0	6	0,1	0,0	6	0,03	0,0	6	0,1	0,0	6	0,08	0,0	6	0,006	0,0
8	43*	7200	4,65	41	10,06	0,0	41*	7200	0,0	41	5,67	0,0	41	2,3	0,0	41	1,03	0,0	41	0,2	0,0	41	0,25	0,0	41	0,052	0,0
9	43*	7200	2,32	42	18,80	0,0	42*	7200	0,0	42	8,89	0,0	42	3,4	0,0	42	1,62	0,0	42	0,4	0,0	42	2,51	0,0	42	0,020	0,0
10	93*	7200	0,0	93	3810,1	0,0	98*	7200	5,10	93	178,36	0,0	93	726,2	0,0	93	7,66	0,0	93	2,1	0,0	93	8,81	0,0	93	120,88	0,0
11	20*	7200	0,0	20	54,71	0,0	20*	7200	0,0	20	27,77	0,0	20	82,8	0,0	20	0,12	0,0	20	2,1	0,0	20	7,77	0,0	20	0,408	0,0
12	56*	7200	0,0	56	5470,1	0,0	57*	7200	1,75	56	479,28	0,0	56*	7200	0,0	56	17,87	0,0	56	3,7	0,0	56	23,81	0,0	56	89,16	0,0
13	40*	7200	10	36	413,5	0,0	36*	7200	0,0	36	80,40	0,0	36	9,2	0,0	36	1,43	0,0	36	4	0,0	36	21,66	0,0	36	1,83	0,0
14	18*	7200	0,0	18*	7200	0,0	18*	7200	0,0	18	164,05	0,0	18	4127,4	0,0	18	6,64	0,0	18	10,6	0,0	18	32,24	0,0	26*	7200	30,76
15	94*	7200	12,76	83*	7200	1,20	83*	7200	1,20	82	417,30	0,0	82*	7200	0,0	82	14,93	0,0	82	3,1	0,0	82	7,16	0,0	82	2766,1	0,0
16	55*	7200	40	47*	7200	29,78	47*	7200	29,78	33	919,72	0,0	34*	7200	2,94	33	9,36	0,0	33	4,8	0,0	33	63,91	0,0	44*	7200	25
Moyenne		4950,3	4,35		1961,6	1,93		4846,6	2,36		143,11	0,0		1659,5	0,18		3,81	0,0		1,98	0,0		10,52	0,0		1086,1	3,48

Note: * La meilleure solution après deux heures d'exécution.

Tableau 6.3. Résultats de calcul du deuxième test

Numéro de l'exemple	Modèle 1		Modèle 2		Modèle 2'		Modèle 3	
	variable	contrainte	variable	contrainte	variable	contrainte	variable	contrainte
1	18	15	18	15	10	10	3	6
2	75	65	75	65	35	31	10	36
3	126	108	126	108	56	49	15	67
4	196	175	196	175	84	74	21	114
5	288	240	288	240	120	100	28	176
6	405	360	405	360	165	147	36	263
7	550	490	550	490	220	192	45	372
8	936	852	936	852	364	324	66	675
9	936	828	936	828	364	317	66	673
10	1800	1650	1800	1650	680	609	105	1383
11	2601	2397	2601	2399	969	860	136	2060
12	3078	2826	3078	2828	1140	1016	153	2468
13	4200	3920	4200	3922	1540	1381	190	3444
14	4200	3840	4200	3840	1540	1370	190	3440
15	4200	3840	4200	3840	1540	1370	190	3440
16	8125	7550	8125	7550	2925	2650	300	6925

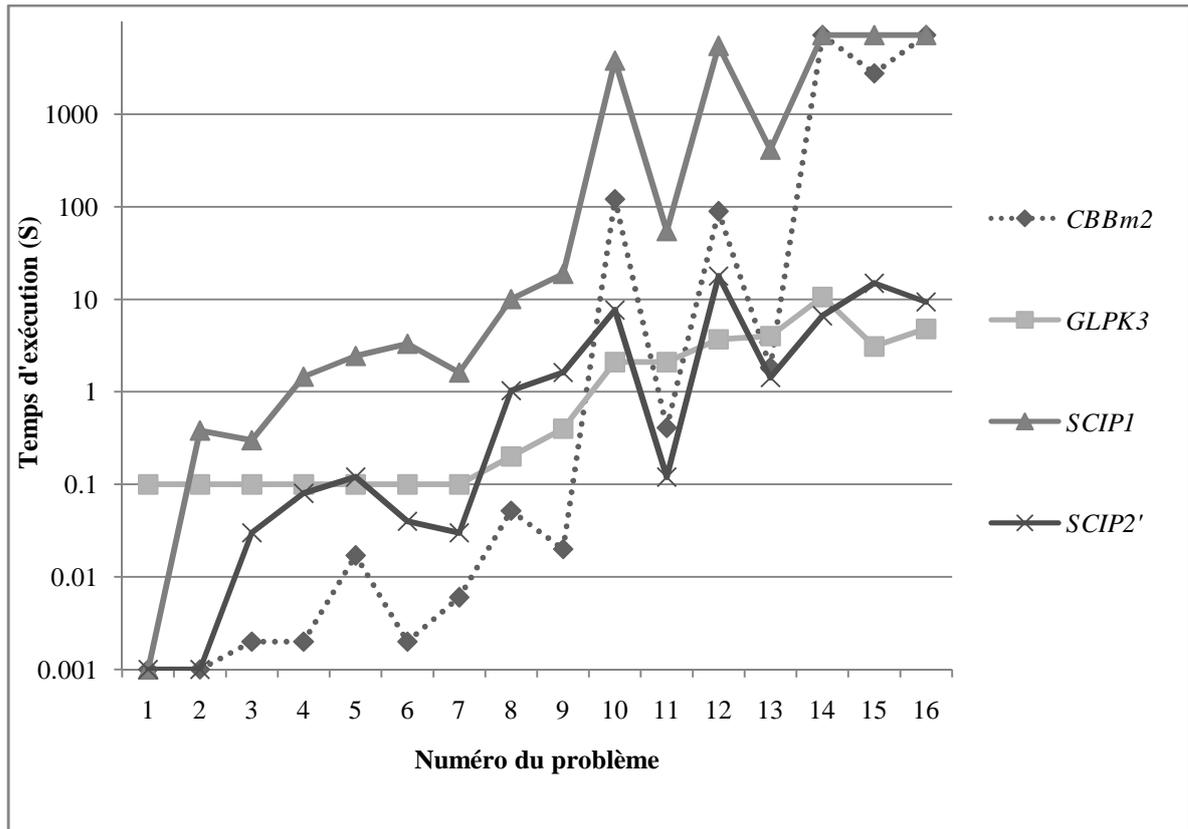
Tableau 6.4. Nombre de variables et de contraintes pour chaque exemple

6.4.4. Comparaison des méthodes

Nous comparons dans cette section la méthode *CBBm2* avec les méthodes *GLPK3*, *SCIP1* et *SCIP2'*, en tenant compte du GAP, la durée d'exécution et le nombre des exemples résolus.

Une lecture du tableau 6.3 permet de constater que:

- 1- La méthode *CBBm2* est bien plus performante que la méthode *SCIP1*. En effet, elle permet d'obtenir les solutions après près de 18 minutes en moyenne, de résoudre 14 sur 16 exemples à l'optimum et d'avoir un GAP moyen total de 0,18%, tandis que la méthode *SCIP1* prend en moyenne près de 32 minutes, résout 13 sur 16 exemples à l'optimum et a un GAP moyen total de 3,48%.
- 2- La méthode la plus performante sur ce jeu d'exemples est *GLPK3* (voir figure 6.5). En effet, dans ce cas le temps d'exécution moyen vaut 1,98 secondes, le nombre des exemples résolus passe à 16 (tous les exemples ont été résolus) et le GAP moyen diminue à 0%. Néanmoins, notre méthode assure le meilleur temps d'exécution pour la plupart des exemples (11 sur 16 exemples) comparée à *GLPK3*.
- 3- *SCIP2'* se classe en deuxième position après *GLPK3*, en termes de performance, avec un temps d'exécution moyen de 3,81 secondes, un nombre d'exemples résolus de 16 et un GAP moyen de 0%.

Figure 6.5. Comparaison entre *CBBm2* et les autres méthodes

6.5. Conclusion

Dans ce chapitre, nous avons mené quatre études expérimentales sur un jeu d'exemples composé d'exemples de la littérature et d'autres aléatoires. À partir des deux premières études, nous constatons l'impact de chacune des quatre routines sur l'amélioration des performances de la méthode *SE cocyclique*. La troisième étude expérimentale nous confirme les résultats de la comparaison théorique présentée dans le chapitre 3 sur l'inefficacité du modèle le plus utilisé dans la littérature (le modèle *L1*) et nous révèle deux autres modèles plus prometteurs (Modèles *L2'* et *L3*) pour la construction de modèles mathématiques plus complexes. De plus, ces résultats montrent la supériorité du solveur non-commercial SCIP comparé à GLPK et nous encourageons donc à l'utiliser pour résoudre des exemples de plus grande taille. La dernière étude dénote l'efficacité de notre méthode pour résoudre le problème FC. D'une part, elle performe mieux que *SCIP1* pour tous les exemples, et d'autre part, elle reste compétitive par rapport aux autres méthodes.

Conclusion générale et perspectives

Notre travail s'est focalisé sur la modélisation et la résolution exacte du problème de composition de cellules. Plus précisément, l'accent a été mis sur la variable de décision choisie et son impact sur la résolution exacte du problème. A cet effet, nous avons présenté trois modèles mathématiques et deux modèles basés sur la théorie des graphes, en tenant compte du critère le plus utilisé dans la littérature et des contraintes réelles des systèmes de production. Une comparaison théorique de ces différents modèles dévoile l'inefficacité du modèle le plus utilisé dans la littérature et la supériorité des modèles basés sur une relation machine-machine par rapport aux modèles basés sur une relation machine-cellule.

Étant donné ce constat, nous avons présenté, pour un modèle basé sur la relation machine-machine, une méthode d'exploration par séparation et évaluation, qui met en œuvre une amélioration guidée par les contraintes.

Les contributions proposées dans cette thèse peuvent être énumérées comme suit:

1. Modèle basé sur la relation machine-médiane:

Dans la littérature, la modélisation du problème FC basée sur le problème p-médian utilise dans la plupart des cas:

- Une matrice d'incidence produits/machines;
- Un coefficient de similitude/dissimilitude.

Le modèle qui utilise:

- La séquence d'opérations et le volume de chaque produit pour évaluer le flux réel entre les cellules;
- La *contrainte-affectation* afin de diminuer la redondance.

est un travail original.

2. Modèle basé sur les co-cycles:

Le modèle utilisant les co-cycles pour définir une composition est une amélioration du modèle proposé par Boulif et Atif [BOU06b].

3. Étude comparative théorique et appliquée:

Pour la première fois, au mieux de nos connaissances, une étude comparative théorique et appliquée a été réalisée pour la résolution exacte du problème FC. L'étude inclue un ensemble de modèles tirés de la littérature et d'autres proposés pour la première fois. Les résultats de

cette étude ont permis d'identifier le modèle le plus avantageux pour le développement de modèles plus généraux.

4. Méthode par séparation et évaluation cocyclique guidée par les contraintes:

La méthode par séparation et évaluation cocyclique a été proposée par Boulif et Atif [BOU06b et BOU06c]. Cependant, l'implémentation d'une méthode par séparation et évaluation cocyclique guidée par les contraintes pour la résolution exacte du problème FC est un nouveau travail.

5. Amélioration de la borne inférieure (*BI*):

L'amélioration de la procédure de calcul de la *BI* par l'application de l'algorithme de Stoer et Wagner de la coupe minimum [STO97] est un apport original.

Les connaissances acquises tout au long de cette thèse nous ont permis de dégager plusieurs idées. Nous avons pu réaliser certaines de ces idées et envisager plusieurs pistes qui peuvent être investies. Nous pouvons citer entre autres:

1. La modélisation

Comme nous l'avons évoqué dans le chapitre deux, le critère à optimiser varie d'une publication à une autre ceci rend l'orientation vers l'optimisation multicritères plus qu'une nécessité. Pour cela, l'une des extensions possibles à notre travail de modélisation est de proposer de nouveaux modèles, qui se basent sur les modèles $L2'$ et $L3$, prenant en considération plusieurs critères à optimiser ainsi que d'autres contraintes liées à la vie réelle des systèmes de production.

2. La résolution

Vu l'influence des calculs de bornes sur les performances de la méthode d'exploration par séparation et évaluation, il serait très intéressant de considérer des procédures plus efficaces pour les calculer. En effet, la valeur de la borne supérieure peut être améliorée en utilisant des heuristiques plus puissantes, tandis que la valeur de la borne inférieure peut être améliorée de différentes manières, par exemple par les méthodes de décomposition.

Des techniques utilisées par les méthodes de résolution présentées dans ce manuscrit peuvent être utilisées afin d'implémenter une méthode d'exploration par séparation et évaluation basée sur le modèle graphique (*TI*).

Une autre direction de recherche qui nous semble intéressante est la résolution du problème de composition de cellules en utilisant d'autres méthodes exactes. En particulier, les méthodes Branch and price et Branch and Cut qui ont montré leur efficacité dans la résolution de plusieurs problèmes d'optimisation combinatoire.

Dans cette thèse, nous avons supposé que les commandes en produits sont constantes en genre et en quantité. Cette supposition, très restrictive, entrainera une éventuelle dégradation des performances du système de production une fois qu'une nouvelle commande aura été effectuée. Pour cela, nous pensons à la résolution exacte du problème de composition de cellules sous son aspect dynamique.

3. L'implémentation

La parallélisation des méthodes de résolution est une voie très importante vu la réduction du temps d'exécution qu'elle peut procurer. Par conséquent, nous pensons à la parallélisation des méthodes proposées. Celle-ci devient plus qu'une nécessité lors de l'application de ces méthodes pour des instances réelles du problème.

Références Bibliographiques

A

- [ADI96] **ADIL**, G. K., **RAJAMANI**, D. and **STRONG**, D., “*Cell formation considering alternate routings*”. International Journal of Production Research, 34(5):1361-80, **1996**.
- [ALB03] **ALBADAWI**, Z., “*A mathematical approach for designing cell formation in cellular manufacturing systems*”, Thèse de doctorat, Université de Québec, **2003**.
- [ALB05] **ALBADAWI**, Z., **BASHIR**, H. A. and **CHEN**, M., “*A mathematical approach for the formation of manufacturing cells*”, Computers & Industrial Engineering, 48(1):3–21. **2005**.
- [ALK07] **AL-Kioon**, s., “*Comprehensive models and solution procedures for integrated cellular manufacturing systems design*”, Thèse de doctorat, Université de Québec, **2007**.
- [ALQ90] **AL-QATTAN**, I., “*Designing flexible manufacturing cells using branch and bound method*”, International Journal of Production Research, 28(2):325-336, **1990**.
- [ARK12] **ARKAT**, J., **ABDOLLAHZADEH**, H. and **GHAHVE**, H., “*A new branch and bound algorithm for cell formation problem*”, Applied Mathematical Modelling, 36(10):5091-5100, **2012**.
- [ARN03] **ARNOULD**, P. and **RENAUD**, J., “*Flux de production: les outils d’amélioration*”, Afnor, Paris, **2003**.
- [ASK90] **ASKIN**, R. G. and **CHIU**, K. S., “*A graph partitioning procedure for machine assignment and cell formation in group technology*”, International Journal of production research, 28(8):1555-1572. **1990**.
- [ASK93] **ASKIN**, R. G. and **STANDRIDGE**, C. R., “*Modeling and analysis of manufacturing system*”. New York: John Wiley & Sons, **1993**.
- [ATE07] **ATEME-NGUEMA**, B. H., “*Conception optimale des cellules de fabrication flexibles basée sur l’approche par réseaux de neurones*”, Thèse de doctorat, Université de Québec, **2007**.

B

- [BAL93a] **BALAS**, E., **CERIA**, S. and **CORNUÉJOLS**, G., “*A lift-and-project cutting plane algorithm for mixed 0-1 programs*”, Mathematical Programming, 58(1-3): 295–324, **1993**.
- [BAL93b] **BALAS**, E., **CERIA**, S. and **CORNUÉJOLS**, G., “*Solving mixed 0-1 programs by a lift-and-project method*”, In: Proceedings of the Fourth Annual ACM / SIGACT-SIAM Symposium on Discrete Algorithms, 25-27, Austin, Texas, **1993**.
- [BAL96] **BALAS**, E., **CERIA**, S. and **CORNUÉJOLS**, G., “*Mixed 0-1*

- programming by lift-and-project in a branch-and-cut framework*”, Management Science, 42(9): 1229-1246, **1996**.
- [BAP05] **BAPTISTE, P., GIARD, V., HAIT, A. and SOUMIS, F.**, “*Gestion de production et ressources humaines: méthodes de planification dans les systèmes productifs*”, Presses Internationales Polytechniques, Montréal, Canada, **2005**.
- [BEN62] **BENDERS, J. F.**, “*Partitioning procedures for solving mixed-variable programming problems*”, Numerische Mathematik, 4(1):238–252. **1962**.
- [BER87] **BERANGER, P.**, “*Les nouvelles règles de la production-vers l'excellence industrielle*”, Dunod, Paris, **1987**.
- [BLA91] **BLACK, J. T.**, “*The design of the factory with a future*”, McGraw-Hill, New York, **1991**.
- [BOU98] **BOUZGARROU, M. E.**, “*Parallélisation de la méthode du « branch-and-cut » pour résoudre le problème du voyageur de commerce*”, Thèse de doctorat, Institut National Polytechnique de Grenoble, **1998**.
- [BOU02] **BOUZEKOUK, S.**, “*Produire vraiment sans gaspiller: L'après juste à temps*”, Édition d'Organisation, Paris, **2002**.
- [BOU06a] **BOULIF, M.**, “*Composition de cellules flexibles dans les systèmes cellulaires de production statiques et dynamiques : Approche multicritère*”, Thèse de doctorat, Université des Sciences et de la Technologie Houari Boumediene, **2006**.
- [BOU06b] **BOULIF, M. and ATIF, K.**, “*A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem*”, Computers & Operations Research, 33(8):2219-2245, **2006**.
- [BOU06c] **BOULIF, M. and ATIF, K.**, “*An exact multiobjective epsilon-constraint approach for the manufacturing cell formation problem*”, in International Conference on Service Systems and Service Management, Troyes, France, pp.883–888, **2006**.
- [BOU08] **BOULIF, M. and ATIF, K.**, “*A new fuzzy genetic algorithm for the dynamic bi-objective cell formation problem considering passive and active strategies*”, International Journal of Approximate Reasoning, 47: 141–165, **2008**.
- [BOU10] **BOULIF, M. and ATIF, K.**, “*Pour une meilleure modélisation du problème de composition de cellules: expose des décisions et analyse critique des contraintes et des critères*”, International conference on Industrial Engineering and Manufacturing ICIEM'10, Batna, Algeria, May 9-10, **2010**.
- [BUR63] **BURBIDGE, J. L.**, “*Production flow analysis*”, Production Engineer, 42(12):742-752, **1963**.
- [BUR75] **BURBIDGE, J. L.**, “*The introduction of group technology*”, Heinemann, London, **1975**.
- [BYC14] **BYCHKOV, I., BATSYN, M. and PARDALOS, P. M.**, “*Exact model for the cell formation problem*”. Optimization Letters, 8 (8): 2203-2210, **2014**.

C

- [CAM01] **CAMPAGNE, J. P.**, “*Chapitre 1: La problématique de la maîtrise*”

- des flux*", de l'ouvrage: Maîtrise et organisation des flux industriels, Hermes Sciences publications, Paris, **2001**.
- [CAR91] **CARPENTER**, G. A., **GROSSBERG**, S. and **ROSEN**, D. B., "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", *Neural Networks*, 4(6):759-771, **1991**.
- [CAU00] **CAUX**, C., **BRUNIAUX**, R. and **PIERREVAL**, H., "Cell formation with alternative process plans and machine capacity constraints: A new combined approach", *International Journal of Production Economics*, 64(1-3):279-284, **2000**.
- [CHA82] **CHAN**, H. M. and **MILNER**, D. A., "Direct clustering algorithm for group formation in cellular manufacturing", *Journal of Manufacturing Systems*, 1(1):65-75, **1982**.
- [CHA86] **CHANDRASEKHARAN**, M. P. and **RAJAGOPALAN**, R., "An ideal seed non-hierarchical clustering algorithm for cellular manufacturing", *International Journal of Production Research*, 24(2):451-463, **1986**.
- [CHA87] **CHANDRASEKHARAN**, M. P. and **RAJAGOPALAN**, R., "ZODIAC: an algorithm for concurrent formation of part families and machines cells", *International Journal of Production Research*, 25(6): 835-850, **1987**.
- [CHE96] **CHENG**, C. H., **GOH**, C. H. and **LEE**, A., "Solving the generalized machine assignment problem in group technology", *Journal of the Operational Research Society*, 47(6):794-802, **1996**.
- [CHE99] **CHEN**, S. J. and **HEARAGU**, S., "Stepwise decomposition approaches for large scale cell formation Problems", *European Journal of Operational Research*, 113(1):64-79, **1999**.
- [CHE01] **CHENG**, C. H., **GOH**, C. H. and **LEE**, A., "Designing group technology manufacturing systems using heuristics branching rules", *Computers and Industrial engineering*, 40(1-2):117-131, **2001**.
- [CHO92] **CHOI**, M., "Manufacturing cell design", *Production and Inventory management Journal*, 33(2):66-69, **1992**.
- [CHU90] **CHU**, C. H. and **TSAI**, M., "A comparison of three array-based clustering techniques for manufacturing cell formation", *International Journal of Production Research*, 28(8):1417-1433, **1990**.
- [COL95] **COLLET**, S. and **SPICER**, R., "Improving productivity through cellular manufacturing", *Production and Inventory Management Journal*, 36(1): 71-75, **1995**.
- D**
- [DAH93] **DAHEL**, N. E. and **SMITH**, S. B., "Designing flexibility into cellular manufacturing systems", *International Journal of Production Research*, 31(4):933-945, **1993**.
- [DAN60] **DANTZIG**, G. B. and **WOLFE**, P., "Decomposition principle for linear programs", *Operations Research*, 8(1):101-111, **1960**.
- [DES84] **DESROSIERS**, J., **SOUMIS**, F. and **DESROCHERS**, M., "Routing with time windows by column generation", *Networks*, 14(4):545-

- 565, 1984.
- [DEU98] **DEUTSCH**, S. J., **FREEMAN**, S. F. and **HELANDER**, M., “*Manufacturing cell formation using an improved p-median model*”, *Computers and Industrial Engineering*, 34(1):135-146, 1998.
- [DEW80] **DE WITTE**, J., “*The use of similarity coefficients in production flow analysis*”, *International Journal of Production Research*, 18(4), 503-514, 1980.
- E**
- [ELB12] **ELBENANI**, B. and **FERLAND**, J. A., “*Cell Formation Problem Solved Exactly with the Dinkelbach Algorithm*”, Publication CIRRELT-2012-07, University of Montreal, Canada, 2012.
- F**
- [FAB86] **FABER**, Z. and **CARTER**, N. W., “*A new graph theory approach for forming machine cells in cellular production systems*”, in *Flexible Manufacturing Systems: Method and Studies*, Elsevier Science Publishers B.V, 1986.
- [FIS94] **FISCHETTI**, M., **TOTH**, P. and **VIGO**, D., “*A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs*”, *Operations Research*, 42(5):846-859, 1994.
- [FOU12] **FOUILHOUX**, P., “*Programmation mathématique Discrète et Modèles Linéaires*”, notes de cours, Université Pierre et Marie Curie, 2012.
- [FRY87] **FRY**, T., **BREEN**, M. and **WILSON**, M., “*A successful implementation of group technology and cell manufacturing*”, *Production and Inventory Management Journal*, 28(3):4-6, 1987.
- G**
- [GAB06] **GABREL**, V. “*An extensive comparison of 0-1 linear programs for the daily satellite mission planning*”, *ANNALES du LAMSADE*, N4-5, 2006.
- [GHO11] **GHOTBODDINI**, M. M., **RABBANI**, M. and **RAHIMIAN**, H., “*A comprehensive dynamic cell formation design: Benders’ decomposition approach*”, *Expert Systems with Applications*, 38(3):2478-2488, 2011.
- [GIL61] **GILMORE**, P. C. and **GOMORY**, R. E., “*A linear programming approach to the cutting-stock problem*”, *Operations Research*, 9(6):849– 859, 1961.
- [GIL63] **GILMORE**, P. C. and **GOMORY**, R. E., “*A linear programming approach to the cutting stock problem–Part II*”, *Operations Research*, 11(6):863–888, 1963.
- [GIL65] **GILMORE**, P. C. and **GOMORY**, R. E., “*Multistage cutting stock problems in two or more dimensions*”, *Operations Research*, 13(1):94–120, 1965.
- [GOL13] **GOLDENGORIN**, B., **KRUSHINSKY**, D. and **PARDALOS**, P.

- M., “Chapter 7: Two Models and Algorithms for Bi-Criterion Cell Formation”, In Cell Formation in Industrial Engineering: Theory, Algorithms and Experiments, Springer, USA, **2013**.
- [GOM58] **GOMORY**, R. E., “Outline of an algorithm for integer solutions to linear programs”, Bulletin of the American Mathematical Society, 64(5):275–278, **1958**
- [GON09] **GONDRAN**, M. and **MINOUX**, M., “Graphes et algorithmes”, 4thed, Tec & Doc, Lavoisier, Paris, **2009**.
- [GOW69] **GOWER**, J. C. and **ROSS**, G. J. S., “Minimum spanning trees and single linkage cluster analysis”, Applied Statistics, 18(1), 54-64, **1969**.
- [GRO84] **GRÖTSCHEL**, M., **JUNGER**, M. and **REINELT**, G., “A cutting plane algorithm for the linear ordering problem”, Operations Research, 32(6):1195–1220, **1984**.
- [GUP90] **GUPTA**, T. and **SEIFODDINI**, H., “Production data based similarity coefficient for machine-component grouping decision in the design of a cellular manufacturing system”, International Journal of Production Research, 28(7):1247-1269, **1990**.
- [GUP95] **GUPTA**, Y., **GUPTA**, M., **KUMAR**, A. and **SUNDRAM**, C., “Minimizing total intercell and intracell moves in cellular manufacturing: A genetic algorithm approach”, International Journal of Computer Integrated Manufacturing, 8(2):92-101, **1995**.
- H**
- [HAC09] **HACHICHA**, W., “Nouvelles approches pour la formation des cellules de production dans le cadre d’une démarche de conception”, Thèse de doctorat, Université de Sfax, **2009**.
- [HER98] **HERAGU**, S. S. and **CHEN**, J. S., “Optimal solution of cellular manufacturing system design: Bender’s decomposition approach”, European Journal of Operational Research, 107(1):175-192, **1998**.
- [HOW93] **HOWARD**, M. and **Newman**, R., “From job shop to just-in-time - successful conversion”, Production and Inventory Management Journal, 34(3):70-74, **1993**.
- [HYE84] **HYER**, N. L., “The potential of group technology for U.S. manufacturing”, Journal of Operations Management, 4(3):183-202, **1984**.
- [HYE02] **Hyer**, N. and **WEMMERLOV**, U., “Reorganizing the factory: Competing through cellular manufacturing”, Productivity Press, **2002**.
- J**
- [JAC08] **JACCARD**, P., “Nouvelles recherches sur la distribution florale”, Bulletin de la Société Vaudoise des Sciences Naturelles, 44(163):223-270, **1908**.
- [JAM07] **JAMES**, T. L., **BROWN**, E. C. and **KEELING**, K. B., “A hybrid grouping genetic algorithm for the cell formation problem”, Computers & Operations Research, 34(7):2059–2079, **2007**.
- [JAY98] **JAYAKRISHNAN NAIR**, G. and **NARENDRAN**, T. T., “CASE:

- A clustering algorithm for cell formation with sequence data”, International Journal of Production Research, 36(1):157-180, **1998**.
- [JAY99] **JAYAKRISHNAN NAIR, G.** and **NARENDRAN, T. T.**, “*ACCORD: A bicriterion algorithm for cell formation using ordinal and ratio level data*”, International Journal of Production Research, 37(3):539-556, **1999**.
- [JEO06] **JEON, G.** and **LEEP, H. R.**, “*Forming part families by using genetic algorithm and designing machine cells under demand changes*”, Computers & Operations Research, 33(1):263–283, **2006**.
- [JOY97] **JOY, S.**, **MITCHELL, J.** and **BORCHERS, B.**, “*A branch-and-cut algorithm for MAX-SAT and weighted MAX-SAT*”, In **DU, D.**, **GU, J.** and **PARDALOS, P. M.**, “*Satisfiability problem: theory and applications*”, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 35:519-536, American Mathematical Society, **1997**.
- K**
- [KIN80] **KING, J. R.**, “*Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm*”, International Journal of Production Research, 18(2):213-232, **1980**.
- [KIN82] **KING, J. R.** and **NAKORNCHAI, V.**, “*Machine-component group formation in group technology: review and extension*”, International Journal of Production Research, 20(2):117–133, **1982**.
- [KOZ04] **KOZANIDIS, G.**, **MELACHRINOUDIS, E.**, “*A Branch & Bound Algorithm for the 0-1 Mixed Integer Knapsack Problem with Linear Multiple Choice Constraints*”, Computers & Operations Research, 31(5): 695-711, **2004**.
- [KRU12] **KRUSHINSKY, D.**, **GOLDENGORIN, B.**, “*An exacte model for cell formation in group technology*”, Computational Management Science, 9(3): 323-338, **2012**.
- [KUM86] **KUMAR, K. R.**, **KUSIA, A.** and **VANNELLI, A.**, “*Grouping of parts and components in flexible manufacturing systems*”, European Journal of Operational Research, 24(3):387–397, **1986**.
- [KUS87] **KUSIAK, A.**, “*The generalized group technology concept*”, International Journal of Production Research, 25(4):561-569, **1987**.
- [KUS93] **KUSIAK, A.**, **BOE, W. J.** and **CHENG, C.**, “*Designing cellular manufacturing systems: branch- and-bound and A* approaches*”, IIE Transactions, 25(4):46-56, **1993**.
- L**
- [LAN60] **LAND, A. H.**, and **DOIG, A. G.**, “*An automatic method of solving discrete programming problems*”, Econometrica, 28(3):497–520, **1960**.
- [LEI06] **LEI, D.** and **WU, Z.**, “*Tabu search for multiple-criteria manufacturing cell design*”, International Journal of Advanced Manufacturing Technology, 28(9):950–956, **2006**.
- [LEV95] **LEVASSEUR, G. A.**, **HELMS, M. M.** and **ZINK, A. A.**, “*A*

- conversion from a functional to a cellular manufacturing layout at steward, Inc.*" Production and Inventory Management Journal, 36(3):37-42, **1995**.
- [LIE07] **LIEDLOFF**, M., "*Algorithmes exacts et exponentiels pour les problèmes NP-Difficiles : domination, variantes et généralisation*", thèse de doctorat, Université de Paul Verlaine-METZ, **2007**.
- [LIT63] **LITTLE**, J. D. C., **MURTY**, K. G., **SWEENEY**, D. W. and **KAREL**, C., "*An algorithm for the traveling salesman problem*", Operations Research, 11(6):972–989, **1963**.
- [LOP01] **LOPEZ**, P. and **ROUBELLAT**, F., "*Ordonnancement de la production*". 2^{ème} Edition, Hermès Science, **2001**.
- [LOZ02] **LOZANO**, S., **DOBADO**, D., **LARRANETA**, J. and **ONIEVA**, L., "*Modified fuzzy C-means algorithm for cellular manufacturing*", Fuzzy Sets and Systems, 126(1):23–32, **2002**.
- M**
- [MAH04] **MAHJOUB**, A. R., "*Chapitre 1. Approches Polyédrales, de Optimisation combinatoire: Concepts fondamentaux*", **2004**.
- [MAN00] **MANSOURI**, S. A., **MOATTAR-HUSSEINI**, S. M. and **NEWMAN**, S. T., "*A review of the modern approaches to multi-criteria cell design*", International Journal of Production Research, 38(5):1201-1218, **2000**.
- [MAN02] **MANSOURI**, S. A., **MOATTAR-HUSSEINI**, S. M. and **ZEGORDI**, S. H., "*Multi-criterion tackling bottleneck machines and exceptional parts in cell formation using genetic algorithms*", in Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture, pp. 181—192, **2002**.
- [MAN04] **MANCEL**, C., "*Modélisation et résolution de problèmes d'optimisation combinatoire issus d'applications spatiales*", Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, **2004**.
- [MCA72] **McAULEY**, J., "*Machine grouping for efficient production*", Production Engineer, 51(2):53-57, **1972**.
- [MCC72] **McCORMICK**, W. T., **SCHWEITZER**, P. J. and **WHITE**, T. W., "*Problem decomposition and data reorganization by a clustering technique*", Operations Research, 20(5):993-1009, **1972**.
- [MEG08] **MEGALA**, N., **RAJENDRAN**, C. and **GOPALAN**, R., "*An ant colony algorithm for cell-formation in cellular manufacturing systems*", European Journal of Industrial Engineering, 2(3):298–336, **2008**.
- [MEH96] **MEHROTRA**, A. and **TRICK**, M. A., "*A column generation approach for graph coloring*", Informs Journal on Computing, 8(4):344-354, **1996**.
- [MER14] **MERCHICHI**, S. and **BOULIF**, M. (in press), "*Constraint-driven exact algorithm for the manufacturing cell formation problem*", European Journal of Industrial Engineering, **2014**.
- [MIN01] **MINOUX**, M., "*Discrete cost multicommodity network optimization problems and exact solution methods*", Annals of Operations Research, 106(1-4):19-46, **2001**.

- [MIN07] **MINOUX**, M., “Programmation Mathématique: Théorie et algorithmes”, 2nd ed, Tec & Doc, Lavoisier, Paris, **2007**.
- [MIT66] **MITROFANOV**, S. P., "*Scientific principles of group technology*", National Lending Library for Science and technology, Boston, **1966**.
- [MIT15] <http://plato.asu.edu/ftp/milpc.html> site supervisé par le chercheur Mittelman, H. D.
- [MLA07] **MLADENOVIC**, N., **BRIMBERG**, J., **HANSEN**, P. and **MORENO-PEREZ**, J. A., “*The p-median problem: A survey of meta-heuristic approaches*”, European Journal of Operational Research, 179(3):927-939, **2007**.
- [MUK02] **MUKATTASH**, A. M., **ADIL**, M. B. and **TAHBOUB**, K. K., “*Heuristic approaches for part assignment in cell formation*”, Computers & Industrial Engineering, 42(2): 329-341, **2002**.
- [MUN00] **MUNGWATTANA**, A., "*Design of cellular manufacturing systems for dynamic and uncertain production requirements with presence of routing flexibility*", PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, **2000**.
- N**
- [NAG90] **NAGI**, R., **HARHALAKIS**, G. and **PROTH**, J. M., “*Multiple routings and capacity considerations in Group Technology applications*”, International Journal of Production Research, 28(12):2243-2257, **1990**.
- [NOL86] **NOLLET**, J., **KELADA**, J. and **DIORIO**, M., "*La gestion des opérations et de la production: une approche systémique*", 1^{ère} ed, Gaétan Morin, Paris, **1986**.
- [NSA06] **NSAKANDA**, A. L., **DIABY**, M. and **PRICE**, W. L., “*Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings*”, European Journal of Operational Research, 171(3):1051–1070, **2006**.
- O**
- [OPI70] **OPITZ**, H., “*A classification system to describe workpieces*”, Pergamon Press, Oxford, **1970**.
- [OWS09] **OWSINSKI**, J. “*Machine-part grouping and cluster analysis: similarities, distances and grouping criteria*”, Bulletin of the Polish Academy of sciences: Technical sciences, 57(3):217-228, **2009**.
- P**
- [PAD87] **PADBERG**, M. and **RINALDI**, G., “*Optimization of a 532-city symmetric traveling salesman problem by branch-and-cut*”, Operations Research Letters, 6(1):1–7, **1987**.
- [PAP81] **PAPADIMITRIOU**, C. H. and **STEIGLITZ**, K., “*Combinatorial Optimization: Algorithm and Complexity*”, PrenticeHall, Englewood Cliffs, NJ, **1981**.
- [PAP10] **PAPAIOANNOU**, G. and **WILSON**, J. M., “*The evolution of cell*

- formation problem methodologies based on recent studies (1997-2008): Review and directions for future research*", European Journal of Operational Research, 206(3):509-521, **2010**.
- [PRO91] **PROTH**, J. M. and **SOUILAH**, A., "Agencement des systèmes de fabrication cellulaires", Rapport de recherche n° 1442, unité de recherche INRIA, **1991**.
- [PUR75] **PURCHECK**, G. F. K., "A linear-programming method for the combinatorial grouping of an incomplete power set", Journal of Cybernetics, 5(4):51-76, **1975**.
- R**
- [RAJ75] **RAJAGOPALAN**, R. and **BATRA**, J. L., "Design of cellular production systems: a graph-theoretic approach", International Journal of Production Research, 13(6):567-579, **1975**.
- [RAM98] **RAMABHATTA**, V. and **NAGI**, R., "An Integrated formulation of manufacturing cell formation with capacity planning and multiple routings", Annals of Operations Research, 77(0):79-95, **1998**.
- [REE06] **REESE**, J., "Solution methods for the p -median problem: An annotated bibliography", Networks, 48(3):125-142, **2006**.
- S**
- [SAL11] **SALANI**, M. and **VACCA**, I. "Branch and price for the vehicle routing problem with discrete split deliveries and time windows", European Journal of Operational Research, 213(3):470-477, **2011**.
- [SAN90] **SANKARAN**, S. and **RODIN**, E. Y., "Multiple objective decision making approach to cell formation: A goal programming model", Mathematical computer modelling, 13(9):71-81, **1990**.
- [SCH03] **SCHRIJVER**, A., "Combinatorial optimization: polyhedra and efficiency", Springer, **2003**.
- [SEL98] **SELIM**, H. M., **ASKIN**, R. G. and **VAKHARIA**, A. J., "Cell formation in Group technology: review, evaluation and directions for future research", Computers and Industrial engineering, 34(1): 3-20, **1998**.
- [SHA91] **SHAFER**, S. M. and **ROGERS**, D. F., "A goal programming approach to the cell formation problem", Journal of Operations management, 10(1):28-43, **1991**.
- [SHT89] **SHTUB**, A. "Modeling group technology cell formation as a generalized assignment problem", International Journal of Production Research, 27(5):775-782, **1989**.
- [SIN96] **SINGH**, N. and **RAJAMANI**, D., "Cellular manufacturing systems: Design, Planning and control", Chapman & Hall, New York. USA, **1996**.
- [SOF97] **SOFIANOPOULOU**, S., "Application of simulated annealing to a linear model for the formulation of machine cells in group technology", International Journal of Production Research, 35(2):501-511, **1997**.
- [SOF99] **SOFIANOPOULOU**, S., "Manufacturing cells design with alternative process plans and/or replicate machines", International

- Journal of Production Research, 37(3), 707-720, **1999**.
- [SOU94] **SOUILAH, A.**, "*Les systèmes cellulaires de production: L'agencement intercellulaires*", Thèse de Doctorat, Université de Metz, **1994**.
- [SOU95] **SOUILAH, A.**, "*Simulated annealing for manufacturing systems layout design*". European Journal of Operational Research, 82(3):592-614, **1995**.
- [SPI98] **SPILIOPOULOS, K.** and **SOFIANOPOULOU, S.** "*An optimal tree search method for the manufacturing systems cell formation problem*", European Journal of Operational Research, 105(3):537–551, **1998**.
- [SPI08] **SPILIOPOULOS, K.**, **SOFIANOPOULOU, S.**, "*An efficient ant colony optimization system for the manufacturing cells formation problem*", International Journal of Advanced Manufacturing Technology, 36(5):589–597, **2008**.
- [SRI90] **SRINIVASAN, G.**, **NARENDRAN, T. T.** and **MAHADEVAN, B.**, "*An assignment model for the part-families problem in Group technology*", International Journal of Production Research, 28(1):145-152, **1990**.
- [SRI91] **SRINIVASAN, G.** and **NARENDRAN, T. T.**, "*GRAFICS: A nonhierarchical clustering-algorithm for group technology*", International Journal of Production Research, 29(3):463-478, **1991**.
- [STE87] **STEUDEL, H. J.** and **BALLAKUR, A.**, "*A dynamic programming based heuristic for machine grouping in manufacturing cell formation*", Computers & Industrial Engineering, 12(3):215-222, **1987**.
- [STO97] **STOER, M.** and **WAGNER, F.**, "*A simple min-cut algorithm*", Journal of the ACM, 44(4):585–591, **1997**
- [SU98] **SU, C. T.** and **HSU, C. M.**, "*Multi-objective machine-part cell formation through parallel simulated annealing*", International Journal of Production Research, 36(8):2185–2207, **1998**.
- T**
- [TAM08] **TAMANI, K.**, "*Développement d'une méthodologie de pilotage intelligent par régulation de flux adaptée aux systèmes de production*", Thèse de Doctorat, Université de Savoie, **2008**.
- [TAV08] **TAVAKKOLI-MOGHADDAM, R.**, **SAFAEI, N.** and **SASSANI, F.**, "*A new solution for a dynamic cell formation problem with alternative routing and machine costs using simulated annealing*", Journal of the Operational Research Society, 59:443-454, **2008**.
- [THO13] www.thotme.com: Site exposant un ensemble de concepts et définitions liés au domaine de la production. Consulté le 25 janvier **2013**.
- [TOU13] **TOUSSAINT, H.**, "*Introduction au Branch Cut and Price et au solveur SCIP (Solving Constraint Integer Programs)* ", Rapport de recherche LIMOS/RR-13-07, **2013**.
- V**
- [VAK90] **VAKHARIA, A. J.** and **WEMMERLOV, U.**, "*Designing a cellular*

- manufacturing system: a material flow approach based on operation sequences*", IIE Transactions, 22(1):84-97, **1990**.
- [VAN00] **VAN DEN AKKER**, J. M., **HURKENS**, C. A. J. and **SAVELSBERGH**, M. W. P., "Time-indexed formulations for machine scheduling problems: column generation", *Inform Journal on Computing*, 12(2):111-124, **2000**.
- [VEN05] **VENKUMAR**, P. and **NOORUL HAQ**, A., "Manufacturing cell formation using modified ART1 networks", *International Journal of Advanced Manufacturing Technology*, 26(7):909-916, **2005**.
- [VIR05] **VIRGINIE**, G., "Résolution de programmes linéaires en nombres entiers de grande taille: décomposition et renforcement pour une résolution exacte et approchée", Mémoire d' Habilitation, Université Paris Dauphine, **2005**.
- [VOH90] **VOHRA**, T., **CHEN**, D. S., **CHANG**, J. C. and **CHEN**, H. C., "A network approach to cell formation in cellular manufacturing", *International Journal of Production Research*, 28(11):2075-2084, **1990**.
- W**
- [WAN97] **WANG**, J. and **ROZE**, C., "Formation of machine cells and part families in cellular manufacturing: an experimental study", *International Journal of Production Research*, 35(5):1259-1286, **1997**.
- [WAN03] **WANG**, J., "Formation of machine cells and part families in cellular manufacturing systems using a linear assignment algorithm", *Automatica*, 39(9):1607-1615, **2003**.
- [WEM89] **WEMMERLOV**, U. and **HYER**, N., "Cellular Manufacturing in the US Industry: A Survey of Users", *International Journal of Production Research*, 27 (9):1511-1530, **1989**.
- [WIK15] https://fr.wikipedia.org/wiki/Optimisation_combinatoire: Site exposant un ensemble de concepts et définitions liés au domaine de l'optimisation combinatoire. Consulté le 05 février **2015**.
- [WEM97] **WEMMERLOV**, U. and **JOHN**, D., "Cellular Manufacturing at 46 User Plants: Implementation experiences and Performance Improvements", *International Journal of Production Research*, 35(1):29-49, **1997**.
- [WON00] **WON**, Y., "New p -median approach to cell formation with alternative process plans", *International Journal of Production Research*, 38(1):229-240, **2000**.
- [WON04a] **WON**, Y. and **LEE**, K. C. "Modified p -median approach for efficient GT cell formation", *Computers & Industrial Engineering*, 46(3):495-510, **2004**.
- [WON04b] **WON**, Y. and **CURRIE**, K. R. "Efficient p -median mathematical programming approaches to machine-part grouping in group technology manufacturing", *Engineering Optimization*, 36(5):555-573, **2004**.
- [WON06] **WON**, Y. and **CURRIE**, K. R. "An effective p -median model considering production factors in machine cell / part family formation", *Journal of Manufacturing Systems*, 25(1):58-64, **2006**.

- [WU93] **WU, N.** and **SALVENDY, G.**, “*A modified network approach for the design of cellular manufacturing systems*”, *International Journal of Production Research*, 31(6):1409-1421, **1993**.
- [WU07] **WU, X.**, **CHU, C. H.**, **WANG, Y.** and **YAN, W.**, “*A genetic algorithm for cellular manufacturing design and layout*”, *European Journal of Operational Research*, 181(1):156–167, **2007**.
- X**
- [XU89] **XU, H.** and **WANG, H. P.**, “*Part family formation for GT applications based on fuzzy mathematics*”, *International Journal of Production Research*, 27(9): 1637–1651, **1989**.
- Y**
- [YIN06] **YIN, Y.** and **YASUDA, K.**, “*Similarity coefficient methods applied to the cell formation problem: A taxonomy and review*”, *International Journal of Production Economics*, 101(2):329–352, **2006**.
- Z**
- [ZIL13] **ZILINSKAS, J.**, **GOLDENGORIN, B.** and **PARDALOS, P. M.**, “*Branch and bound algorithm for bi-criterion cell formation problems*”, In *Cell Formation in Industrial Engineering: Theory, Algorithms and Experiments*, Springer, Unpublished manuscript, **2013**.
- [ZIL15] **ZILINSKAS, J.**, **GOLDENGORIN, B.** and **PARDALOS, P. M.** “*Pareto optimal front of cell formation problem in group technology*”, *Journal of Global Optimization*, 61(1):91-108, **2015**.
- [ZOL97] **ZOLFAGHARI, S.**, “*Design and Planning for Cellular Manufacturing: Application of Neural Networks and Advanced Search Techniques*”, School of Graduate Studies and Research, University of Ottawa, **1997**.

Annexe A

Lexique

Nous présentons dans cette annexe les plus importants mots techniques utilisés dans cette thèse [BOU06a, MIN07, LIE07 et WIK15].

A

Allèle: Dans le langage génétique se dit d'un caractère situé sur un chromosome. Dans le codage binaire, il s'agit d'un élément du vecteur représentatif d'une solution donnée (case dans la chaîne représentative).

B

Bloc diagonal: (de MIPM) L'ensemble des cases de MIPM appartenant à l'intersection des lignes des produits d'une famille avec les colonnes des machines de la cellule associée.

C

Cellule: (de machines) Ensemble de ressources groupées et destinées à réaliser, aussi complètement que possible, une famille de produits.

Combinaison convexe: Etant donné p points de \mathbb{R}^n (x^1, x^2, \dots, x^p), on dit que $x \in \mathbb{R}^n$ est une combinaison convexe de ces p points s'il existe des coefficients u_1, u_2, \dots, u_p ($u_i \geq 0$, $i = 1, 2, \dots, p$) tels que: $\sum_{i=1}^p u_i = 1$ et $x = \sum_{i=1}^p u_i \cdot x^i$.

Convexe: Voir ensemble convexe.

Critère: Objectif défini au préalable et dont l'optimalité est attendue ou souhaitée.

E

Ecoles optimale et pragmatique: Deux doctrines antagonistes adoptées par les chercheurs dans le domaine de la conception des SCP. La première opte pour les techniques d'optimisation pour modéliser et résoudre le problème. La deuxième s'articule, dans cette résolution, sur l'intuition et la compétence du concepteur.

Ensemble convexe: Un ensemble $S \subset \mathbb{R}^n$ est dit convexe si et seulement si, pour deux points quelconques x et y pris dans S , le segment $[x, y]$ est tout entier contenu dans S .

Enveloppe convexe: Soit $A \subset \mathbb{R}^n (A \neq \emptyset)$, on appelle enveloppe convexe de A , le plus petit ensemble convexe contenant A , noté $conv(A)$, ($A \subset conv(A)$).

Espace de solutions: Ensemble de toutes les solutions.

F

Famille: (de produits) Ensemble de produits présentant des caractéristiques similaires qui permettent de réaliser la totalité sinon la majorité des opérations de leurs gammes dans une même cellule de machines.

G

Gamme de production: Suite ordonnée d'opérations subites par un produit donné durant sa fabrication avec les temps opératoires correspondants.

H

Horizon de planification: Etendue de temps sur laquelle le système cellulaire de production sera implanté.

M

Manutention: Ensemble des moyens engagés pour le transport des produits à l'intérieur de l'atelier de fabrication.

Matrice d'Incidence Produits/Machines (MIPM): Une matrice booléenne indiquant pour chaque produit les machines utilisées pour le fabriquer. En général, les lignes sont associées aux produits et les colonnes aux machines. Dans les travaux récents, on y incorpore l'information des routages. Dans ce cas, la matrice n'est pas booléenne et chaque élément non nul de celle-ci indique le rang de la machine dans le routage du produit.

O

Opération: Action réalisée sur un produit à l'aide d'une machine.

P

Point extrême: On appelle point extrême d'un polyèdre X , tout point $x \in X$ qui ne peut pas être exprimé comme combinaison convexe d'autres points $y \in X (y \neq x)$.

Polyèdre: Un ensemble convexe de la forme $X = \{x/Ax = b, x \geq 0\}$ est appelé un polyèdre.

Problème d'optimisation: Etant donné une fonction $f: A \rightarrow \mathbb{R}$ définie sur un ensemble A à valeurs dans l'ensemble \mathbb{R} des nombres réels, trouver un élément \bar{x} de A tel que $f(\bar{x}) \leq f(x)$ pour tous les x dans A . Où: - La fonction f porte divers noms: fonction-coût, fonction objectif ou simplement objectif, critère, etc; - L'ensemble A est appelé l'ensemble admissible; Les points de A sont appelés les points admissibles; - Le point \bar{x} est appelé solution optimale du problème d'optimisation.

Problème d'optimisation combinatoire (ou discrète): Il consiste à trouver dans un ensemble discret un parmi les meilleurs sous-ensembles (ou solutions) réalisables, la notion de meilleure solution étant définie par une fonction objectif. Formellement :

$$\max_{S \subseteq N} \{ f(S) : S \in \mathcal{R} \}$$

Où : N un ensemble discret; $f: 2^N \rightarrow \mathbb{R}$ une fonction objectif et \mathcal{R} un ensemble de sous-ensembles de N dont les éléments sont appelés les solutions réalisables.

Problème NP (pour Non Déterministe Polynomial): Un problème NP est un problème de décision que l'on peut résoudre à l'aide d'un algorithme non déterministe polynomial.

Problème NP-difficile: Un problème P est dit NP-difficile si l'existence d'un algorithme polynomial (déterministe) pour P implique l'existence d'un algorithme polynomial (déterministe) pour n'importe quel problème de NP.

Productivité: La productivité du travail est le rapport entre la quantité et/ou la valeur de la production et le nombre d'heures de travail employées.

Programme linéaire (PL): Un programme linéaire est un problème d'optimisation qui consiste à minimiser (ou maximiser) une fonction linéaire sous contraintes linéaires.

Programme linéaire en nombres entiers (PLNE): Un programme linéaire en nombres entiers est un problème d'optimisation linéaire dans lequel les variables sont astreintes à ne prendre que des valeurs entières (ou certaines valeurs entières).

Programme linéaire mixte: Un programme linéaire mixte est un problème d'optimisation linéaire dans lequel certaines variables sont astreintes à ne prendre que des valeurs entières.

Programme non linéaire: Un programme non linéaire est un problème d'optimisation dans lequel la fonction à optimiser ou les contraintes à respecter peuvent être non linéaires.

R

Relaxation continue d'un PLNE : Elle représente le programme obtenu à partir de PLNE en relaxant (en oubliant) les contraintes d'intégrité.

Routage : Suite ordonnée des machines que visite le produit durant sa fabrication. Elle est extraite de la *gamme de fabrication* du produit et de la composition du parc machines de l'atelier. Par abus de langage, on utilise aussi ce terme pour décrire une ligne de MIPM et parfois même une séquence opératoire.

S

Solution réalisable: Solution où toutes les contraintes sont satisfaites.

Système: Ensemble d'objets interactifs. Un système est défini par trois entités: 1. Les entrées 3. Les sorties 3. L'état des objets qui le constituent.

T

Technologie de groupe (TG): Logique basée sur l'analyse de données, cherchant à regrouper les produits en familles présentant des caractéristiques similaires du point de vue de la conception et/ou de la fabrication, afin de tirer une meilleure gestion du temps et des moyens engagés.

Annexe B

Notion de co-cycle

Dans cette annexe, nous présentons les concepts théoriques utiles à la construction des éléments de notre procédure de branchement, à savoir la notion de co-cycle. Pour plus de détails, nous invitons le lecteur à voir [GON09].

Soit $G = (M, E)$ un graphe non orienté où $M = \{M_1, M_2, \dots, M_m\}$ et $E = \{e_1, e_2, \dots, e_{|E|}\}$. Soit A un sous-ensemble de M .

B.1. Notion de co-cycle

Nous appelons co-cycle associé à A , que nous notons $w(A)$, l'ensemble des arcs (arêtes) ayant l'une de ses extrémités dans le sous-ensemble A et la deuxième dans le sous-ensemble $M - A$.

Un co-cycle peut être représenté par les sommets de l'un des deux sous-ensembles associés, A et $M - A$, ou par les arêtes qui le composent.

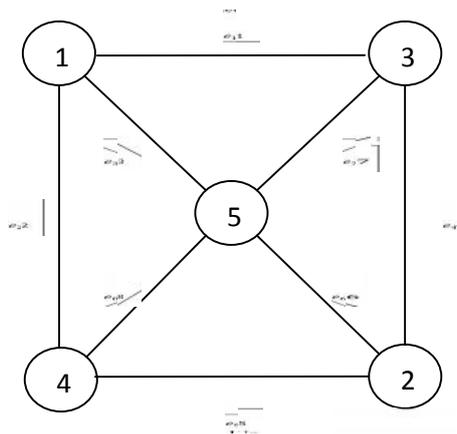


Figure B.1. Un co-cycle

Exemple de la figure B.1, le co-cycle associé à $A = \{M_1, M_4\}$ est représenté soit par $w(A) = \{M_1, M_4\}$, soit par $w(A) = \{M_2, M_3, M_5\}$ ou bien par $w(A) = \{e_1, e_3, e_5, e_8\}$.

Proposition: Le nombre de co-cycles d'un graphe connexe avec m sommets est égal à $2^{m-1} - 1$.

Preuve: le même co-cycle est associé à une combinaison de deux sous-ensembles complémentaires. De toutes ces combinaisons, nous devons éliminer la combinaison de sous-ensemble complet et vide.

B.2. Co-cycle élémentaire

Un co-cycle est élémentaire s'il ne contient pas un autre co-cycle.

B.3. Représentation par sommets d'un co-cycle

A tout co-cycle $w(A)$, nous pouvons associer un vecteur v de taille égale au nombre de machines m . Les cases de vecteur représentent les machines M_i et ils sont définis comme suit:

$$v(i) = \begin{cases} 1 & \text{Si } M_i \in A \\ 0 & \text{Sinon} \end{cases} \quad i \in \{1, 2, \dots, m\}$$

Donc, le vecteur associé au co-cycle de la figure B.1 est le suivant: $v = (1, 0, 0, 1, 0)$.

B.4. Représentation par arêtes d'un co-cycle

A tout co-cycle $w(A)$, nous pouvons associer un vecteur w de taille égale au nombre d'arêtes, $|E|$, de graphe G . Les cases de vecteur représentent les arêtes e_j et ils sont définis comme suit:

$$w(i) = \begin{cases} 1 & \text{Si } e_j \in w(A) \\ 0 & \text{Sinon} \end{cases} \quad j \in \{1, 2, \dots, |E|\}.$$

Donc, le vecteur associé au co-cycle de la figure B.1 est le suivant: $w = (1, 0, 1, 0, 1, 0, 0, 1)$.

Annexe C

Détails des exemples du chapitre 6

Le premier exemple présente un atelier de 5 produits et 3 machines. Le tableau C.1 illustre la matrice d'incidence produits / machines avec les informations des séquences d'opérations et le volume de production de chaque produit. Le nombre maximum de machines par cellule est égal à 2.

		<i>Machines</i>			<i>Volume</i>
		1	2	3	
<i>Produits</i>	1	2	1		5
	2		1	2	5
	3		1		3
	4	2	3	1	2
	5	2	3	1	1

Tableau C.1. Matrice d'incidence de l'exemple 1

Dans le deuxième exemple, il s'agit d'un atelier de 5 produits et 5 machines. Les deux machines M_1 et M_5 sont contraintes à ne pas cohabiter. D'autre part, le nombre maximum de machines par cellule est fixé à 2. Les routages des produits sont mentionnés dans le tableau C.2.

		<i>Machines</i>					<i>Volume</i>
		1	2	3	4	5	
<i>Produits</i>	1		2		1,3	4	3
	2	2		1			2
	3	1	3			2	7
	4		1,4	3	2		1
	5	1		2		3	8

Tableau C.2. Matrice d'incidence de l'exemple 2

Le troisième exemple concerne un atelier de 17 produits et 6 machines. Les deux machines M_2 et M_6 sont contraintes à cohabiter. Le nombre maximum de machines par cellule est égal à 3. Les routages associés à cet exemple sont présentés dans le tableau C.3.

		<i>Machines</i>						<i>Volume</i>
		1	2	3	4	5	6	
<i>Produits</i>	1	1	2			4	3	1
	2	3	1	2	4			3
	3				2	1,3		5
	4	1	2,4	3				2
	5	1		3	2,4			2
	6	1	3,6	5	2,4			3
	7	1	2,4	5	3,6			1
	8	1	3		2,4			3
	9	1	4		3	2		2
	10	1		3	4	2,5		1
	11	1		3	2,4			2
	12	4	3	1,6	5,7	2		1
	13			4	1,3	2		2
	14		2			1	3	1
	15	3		2		1		3
	16		2		3		1	2
	17	1	2		4	3,5	6	1

Tableau C.3. Matrice d'incidence de l'exemple 3

L'exemple quatre présente un atelier de 7 produits et 7 machines. Le nombre maximum de machines par cellule est égal à 3. Les deux machines M_2 et M_5 , ainsi que les deux machines M_2 et M_6 sont contraintes à cohabiter.

		<i>Machines</i>							<i>Volume</i>
		1	2	3	4	5	6	7	
<i>Produits</i>	1	3	1	2					1
	2	2	1						1
	3			1	2				1
	4			1	2				1
	5					3	2	1	1
	6			2		4	3	1	1
	7				3	1	2	4	1

Tableau C.4. Matrice d'incidence de l'exemple 4

Le cinquième exemple est tiré de [JAY98]. Il s'agit d'un atelier de 20 produits et 8 machines. Le tableau C.5 illustre les routages des produits. Le nombre maximum de machines par cellule est égal à 4.

		<i>Machines</i>								<i>Volume</i>
		1	2	3	4	5	6	7	8	
<i>Produits</i>	1					2	1			1
	2	1		2						1
	3	2	1		5			3	4	1
	4		1		2			3	4	1
	5					2	1			1
	6		1		2	5		3	4	1
	7		4		2			3	1	1
	8	1		2						1
	9	1		3			2			1
	10				2	3	1			1
	11	3		2				1		1
	12					1	3	2		1
	13	1		2						1
	14	1	2	3						1
	15				1	2				1
	16	1		2						1
	17	3		1		2				1
	18		2		1			4	3	1
	19	1		2						1
	20		2		1		3	4	5	1

Tableau C.5. Matrice d'incidence de l'exemple 5

Dans le sixième exemple, nous avons 6 produits et 9 machines. Les deux machines M_2 et M_7 sont contraintes à cohabiter. De plus, les deux machines M_7 et M_8 sont contraintes à ne pas cohabiter. La taille de la cellule est égale à 3.

		Machines									Volume
		1	2	3	4	5	6	7	8	9	
Produits	1			3	4	1				2	1
	2			2				1			1
	3		1					2			1
	4	2			1		3				1
	5	2				4	3		1		1
	6		1	2			3				1

Tableau C.6. Matrice d'incidence de l'exemple 6

L'exemple sept présente un atelier de 12 produits et 10 machines. Les deux machines M_5 et M_8 sont contraintes à cohabiter. De plus, les deux machines M_2 et M_7 sont contraintes à ne pas cohabiter. Le nombre maximum de machines par cellule est égal à 4. Les routages des produits sont mentionnés dans le tableau C.7.

		Machines										Volume
		1	2	3	4	5	6	7	8	9	10	
Produits	1			1,3			2					1
	2		1			3			4		2	1
	3		2			4			1		3	1
	4				3			1		2		1
	5	3		5	4,6		1		2			1
	6							1,3		2		1
	7		2			3					1	1
	8		2								1	1
	9	4		1			2		3,5			1
	10	3		1			2					1
	11							1				1
	12		1				2			3		1

Tableau C.7. Matrice d'incidence de l'exemple 7

Le huitième exemple concerne un atelier de 16 produits et 12 machines. Le nombre maximum de machines par cellule est égal à 4. Les deux machines M_4 et M_8 , ainsi que les deux machines M_8 et M_9 sont contraintes à cohabiter, tandis que les machines M_2 et M_{11} sont contraintes à ne pas cohabiter.

		Machines												Volume	
		1	2	3	4	5	6	7	8	9	10	11	12		
Produits	1	1			2				3	4					2
	2	1			2,4			3,6	5						3
	3	1	2		3			4	5	6					1
	4	1			2			3		4					3
	5	1					2	4		5	3				2
	6						1	3	4	5	2				1
	7				2		1		3	4					2
	8		3	1	5	2	4		6	7					1
	9			1	4	2	3		5	6					1
	10				1,3			2	4						2
	11							2				1	3		1
	12											1	2		1
	13							2			3	1			3
	14	1						2			4	3,5	6		3
	15							2				1	3		1
	16						1	2			3				3

Tableau C.8. Matrice d'incidence de l'exemple 8

Le neuvième exemple est dû à [VAK90]. Il concerne un atelier de 19 produits et 12 machines. Les deux machines M_7 et M_{12} sont contraintes à ne pas cohabiter. Le nombre maximum de machines par cellule est égal à 4. Les routages des produits sont mentionnés dans le tableau C.9.

		Machines												Volume		
		1	2	3	4	5	6	7	8	9	10	11	12			
Produits	1	1			2				3	4						2
	2	1			2,4			3,6	5							3
	3	1	2		3			4	5	6						1
	4	1			2			3		4						3
	5	1					2	4		5	3					2
	6						1	3	4	5	2					1
	7				2		1		3	4						2
	8		3	1	5	2	4		6	7						1
	9			1	4	2	3		5	6						1
	10				1,3			2	4							2
	11						1									3
	12							2					1	3		1
	13												1	2		1
	14							2				3	1			3
	15	1						2				4	3,5	6		1
	16	1						2				4	3,5	6		2
	17							2					1	3		1
	18							1	2			3				3
	19													1		2

Tableau C.9. Matrice d'incidence de l'exemple 9

Dans l'exemple dix, nous avons 20 produits et 15 machines. Le nombre maximum de machines par cellule est égal à 5. Les deux machines M_6 et M_{15} sont contraintes à cohabiter, tandis que les deux machines M_8 et M_{11} , ainsi que les deux machines M_5 et M_{15} sont contraintes à ne pas cohabiter.

		Machines															Volume
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Produits	1			6	5	4	2	3,7		1							1
	2	6	1		3	4					7			2,5	8		3
	3		1	2													3
	4	1	2,4		3	7		5						6			1
	5	2		1						3							3
	6					2		6	3	5	7			8		1,4	2
	7					1		3								2	2
	8		1								2				3		2
	9		2								1				3		1
	10	7		8	1,3		2	5			4			6			3
	11	3				5					2	4,6		1			3
	12	2	6							3,5		4				1	1
	13	3		8			1,5	2	4		7					6	3
	14			2										1		3	2
	15					1	4				3,6				2,5		1
	16		2	3	4				1								2
	17	1,4				2		3									1
	18		1				6	3	4		2		7			5	3
	19		4	6		1,3							5		2		3
	20								3			2	4	1			2

Tableau C.10. Matrice d'incidence de l'exemple 10

L'exemple onze concerne un atelier de 10 produits et 17 machines. Les deux machines M_1 et M_6 , ainsi que les deux machines M_6 et M_{10} sont contraintes à cohabiter, tandis que les deux machines M_{11} et M_{17} sont contraintes à ne pas cohabiter. La taille de la cellule est égale à 5.

		Machines																Volume	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		17
Produits	1	1		2	4				3										2
	2	2		1															3
	3		2					1		3									1
	4		1,4		2,5	3													4
	5	2					1,3				4								1
	6			2					4	5	1	3							1
	7						4	1			2		3						2
	8					2,5				1				3		4			1
	9		4							2				1	3				3
	10		3						2								1,4,6	5	4

Tableau C.11. Matrice d'incidence de l'exemple 11

Dans le douzième exemple, tiré de [SU98], il s'agit d'un atelier de 35 produits et 18 machines. Les deux machines M_2 et M_{12} sont contraintes à ne pas cohabiter, tandis que les machines M_9 et M_{16} sont contraintes à cohabiter. D'autre part, le nombre maximum de machines par cellule est fixé à six. Les routages des produits sont mentionnés dans le tableau C.12.

		Machines																		Volume	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		
Produits	1		1,4		2						6			3	5					1	
	2							2						5	1	4		3		1	
	3			5					3		2						4		1	1	
	4	2				3						4				1				1	
	5						3			2						1				1	
	6									1	3					2,4				1	
	7			2							1							3		1	
	8			3							2,5						1		4	1	
	9				4		5				2,6		1			3				1	
	10		2,5			4							3	1						1	
	11		2		3		1,5							4						1	
	12		2		1		3	5					4	6						1	
	13							2,4							5	1		3		1	
	14			5					3		1						6		2,4	1	
	15								1		4						3		2	1	
	16	3				1,4						2				5				1	
	17	2				4						1				3,5				1	
	18						3			2,4						1				1	
	19	4				2						3				1				1	
	20								1,3									2		4	1
	21						1			4	2,5					3				1	
	22			6				1						3	2,5	4				1	
	23		3,5		2		1						4	6						1	
	24				2,6		4				3		1	5						1	
	25			1,3				2							5			4		1	
	26			3				1,6							5	2			4	1	
	27			1					3,6							2	5		4	1	
	28	2,4										1				3				1	
	29	1,3										4				2				1	
	30						2			4	1					3		5		1	
	31						5			3	2					1			4	1	
	32			3					5		2,4						6		1	1	
	33							4,6						2	5	3		1		1	
	34		1		3						5		4	2						1	
	35			4					3						1	6		2,5		1	

Tableau C.12. Matrice d'incidence de l'exemple 12

Le treizième exemple présente un atelier de 23 produits et 20 machines. Le nombre maximum de machines par cellule est égal à 7. Les deux machines M_2 et M_3 , ainsi que les deux machines M_3 et M_6 sont contraintes à cohabiter, tandis que les deux machines M_2 et M_{10} , ainsi que les deux machines M_4 et M_{16} sont contraintes à ne pas cohabiter.

		Machines																			Volume	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		20
Produits	1	2													4		3		1		3	
	2												3		1		4		2		3	
	3				3	4		1											2		2	
	4	3				1					4			2							5	
	5		2	1					3													3
	6								1												2	3
	7	1	5					2,4		3												1
	8		1													3			2			1
	9				3												2		1			1
	10				4									2							1,3	1
	11	5			1	3		2						4								1
	12										2					1						1
	13		2							3	1											1
	14				2											3					1	1
	15		4		5										3	1		2				1
	16															2						2
	17			5,7			6	2			4					1			3			2
	18	1									2					3						2
	19										2	3			4	1						2
	20			2								1,3		4								1
	21												2	1								1
	22	1												2								2
	23			7		6	4							1	3		2				5	2

Tableau C.13. Matrice d'incidence de l'exemple 13

L'exemple quatorze est tiré de [JAY98]. Il s'agit d'un atelier de 20 produits et 20 machines. Le tableau C.14 illustre les routages des produits. Le nombre maximum de machines par cellule est égal à 5.

		Machines																			Volume		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		20	
Produits	1	2							3										4		5	1	
	2		2	3							1											1	
	3								1												3	2	1
	4		3	1							4	2											1
	5				1		3	4								2							1
	6					5						1			2		3	4					1
	7					1											2	3					1
	8				5			3		4				2		1							1
	9	4								2		3	5							1			1
	10								3												1	2	1
	11			3					1						2								1
	12	5				3				1			4							2			1
	13						1	2								3		4					1
	14	3	4						1		2												1
	15													1	2		3	4					1
	16						3	2								1					4		1
	17	2								1			3										1
	18								1		4										2	3	1
	19		2	1		4						3											1
	20	3									2		4							1			1

Tableau C.14. Matrice d'incidence de l'exemple 14

Dans le quinzième exemple, tiré de [NAG90], il s'agit d'un atelier de 51 produits et 20 machines. Le nombre maximum de machines par cellule est égal à 5. Les routages associés à cet exemple sont présentés dans le tableau C.15.

		Machines																				Volume
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Produits	1						3			2			1									2
	2							3		2			1									2
	3	3					1						2									2
	4	3						1					2									2
	5	1							4		2			3								2
	6	1							4		2			3								2
	7	1					3							2								2
	8	1						3						2								2
	9	2									1			3						4		2
	10	2									1			3					4			2
	11	2									1			3							4	2
	12		3			2	1															2
	13		3			2		1														2
	14					1		4									2			3		2
	15					1		4									2		3			2
	16					1		4									2				3	2
	17					1	4										2			3		2
	18					1	4										2		3			2
	19					1	4										2				3	2
	20		3					2									1					2
	21		3				2										1					2
	22		2					4									1			3		1
	23		2				4										1			3		1
	24		2					4									1		3			1
	25		2				4										1		3			1
	26		2					4									1				3	1
	27		2				4										1				3	1
	28		1			3	4										2					3
	29		1			3		4									2					3
	30			2					1				3							4		3
	31			2					1				3							4		3
	32			2					1				3								4	3
	33			1					2										3			3
	34			1					2											3		3
	35			1					2												3	3
	36			3					2				1						4			4
	37			3					2				1							4		4
	38			3					2				1								4	4
	39										1				3			4	2			2
	40										1				3			4		2		2
	41										1				3			4			2	2
	42										3							2	1			2
	43										3							2		1		2
	44										3							2			1	2
	45										2				3				1			2
	46										2				3					1		2
	47										2				3						1	2
	48										3				2				1			2
	49				2									1		3						3
	50				1										2		3					3
	51				2												1					3

Tableau C.15. Matrice d'incidence de l'exemple 15

Le dernier exemple est dû à [JAY98]. Il concerne un atelier de 40 produits et 25 machines. Le nombre maximum de machines par cellule est égal à 4.

		Machines																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Volume		
Produits	1				5			3			1					4		2				6					1		
	2	2	3														4	4									1	1	
	3			2								3									1							1	
	4												1											2				1	
	5				3								2						1									1	
	6												3												1			1	
	7				3			2			5						2	4		1					1			1	
	8					1											3				2							1	
	9			3									4									1						2	1
	10									2	1																	3	1
	11									2					3								1						1
	12	1		4															3								2	5	1
	13			3									2										1						1
	14			4		1							2										3						1
	15		4			3					5							1			2								1
	16				1				3								2		4										1
	17								1			3								2									1
	18															3	2							1					1
	19									1	3	2																	1
	20																									1			1
	21									1	3	2																	1
	22			3						4	2								1										1
	23					2												3				1							1
	24					1												2											1
	25						1										3							2					1
	26				2									3			4									1			1
	27												1										3	2					1
	28								2	1	3																		1
	29					3	2																1						1
	30				4			2										3		1									1
	31					2													1		3								1
	32														2	1	3							4					1
	33												1									3						2	1
	34													2											1	3			1
	35						2										4						1	3					1
	36	2	3										4						1										1
	37								3					2												1			1
	38									2	3													1					1
	39													1															1
	40						2										3						1						1

Tableau C.16. Matrice d'incidence de l'exemple 16

