

*République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumedienne
Faculté d'Electronique et Informatique*



THESE

présentée pour l'obtention du grade de **Doctorat**
En : **INFORMATIQUE**
Spécialité : Informatique

Par :
Menouar BOULIF

Sujet

**Composition de cellules flexibles dans les Systèmes Cellulaires de Production
statiques et dynamiques : Approche multicritère**

Soutenue publiquement le 23/12/2006 devant le jury composé de :

Pr. Amar	AISSANI	Professeur, USTHB	Président
Dr. Karim	ATIF	Maître de conférence, USTHB	Directeur de Thèse
Pr. Meziane	AIDER	Professeur, USTHB	Examinateur
Pr. Abdelhafid	BERRACHEDI	Professeur, USTHB	Examinateur
Pr. Mohamed Seghir	BOUCHERIT	Professeur, ENP	Examinateur
Dr. Mohamed	BAKALEM	Docteur, ENP	Invité

*A mes chers parents
A mes enfants Abderrahmane et Meriem et la toute petite Afaf
A mes frères et sœurs et leurs enfants
A tous ceux qui ont contribué à mon instruction
Depuis l'école à l'université, sans exception
Je dédie ce manuscrit*

Menouar.

REMERCIEMENTS

En premier lieu, je tiens à exprimer ma reconnaissance envers Allah Le Tout Puissant, Le Miséricordieux. C'est Lui Qui nous a donné la vie puis nous a orienté et instruit. Cependant, quoique je puisse dire, je ne pourrais glorifier Allah comme Il se loue Lui-même¹ :

*Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux.
Louange à Allah, Seigneur de l'univers.
Le Tout Miséricordieux, le Très Miséricordieux,
Maître du Jour de la rétribution.
C'est Toi Seul que nous adorons, et c'est Toi Seul dont nous implorons secours.
Guide-nous dans le droit chemin,
le chemin de ceux que Tu as comblés de faveurs,
non pas de ceux qui ont encouru Ta colère, ni des égarés.*

Ensuite, sachant que «celui qui n'est pas reconnaissant envers les gens n'est pas reconnaissant envers Allah»² :

Je désire exprimer ma profonde reconnaissance à Monsieur Karim Atif, mon Directeur de thèse, pour avoir accepté de m'encadrer pendant toutes ces longues années. J'ai trouvé en lui les qualités du bon encadreur : disponibilité, sens de l'orientation scientifique et amitié. Sans son soutien moral et ses encouragements, lorsque je faillis à plusieurs reprises "laisser tout tomber", cette thèse n'aurait pas vu le jour.

Mes remerciements vont aussi aux membres du jury dont je suis fier de leur participation à la soutenance de cette thèse : A Monsieur Amar AISSANI qui m'a fait l'honneur de présider le jury. A Madame Aicha MOKHTARI - AISSANI dont je n'oublierai pas l'instantanéité de la réponse quant à sa disponibilité à faire partie du jury. A Monsieur Meziane AIDER qui m'a inculqué en graduation l'amour de la recherche dans les modules d'OC et de Modélisation. A Monsieur Abdelhafidh BERRACHEDI avec qui j'ai découvert la Théorie des Graphes. A Monsieur Mourad BOUDHAR qui m'a appris le coté application de la Logique Mathématique.

Je ne pourrai jamais oublier Messieurs Abdelghani SOUILAH (CDTA) et Mohamed-El-Amine CHERGUI (USTHB) qui m'ont suivi durant mes premiers pas dans le domaine des systèmes cellulaires de production. La présente thèse constitue une continuité des travaux réalisés sous leur supervision alors qu'ils soient vivement remerciés.

Ma gratitude va aussi à tous mes collègues au Laboratoire d'Informatique Fondamentale et Appliquée à l'Université de Boumerdès. En particulier : Au Professeur Mohamed MEZGHICHE qui a mis à ma disponibilité un ensemble d'outils de travail. A Monsieur Mohammed KHALIFA qui a eu l'amabilité de vérifier le langage d'un article portant sur une partie de cette thèse. A Monsieur Ali BERRICHI avec lequel j'ai eu des discussions qui m'ont aidées à clarifier mes idées.

Je ne peux aussi oublier Monsieur Abdelhakim AÏTZAI (USTHB) qui m'a beaucoup encouragé, ni le Professeur Patrick SIARRY (Paris XII) qui a eu l'amabilité de me procurer une partie des sources bibliographiques utilisés dans mes recherches.

Je tiens aussi à remercier ma femme pour son soutien moral, sa patience et sa disponibilité à s'occuper des enfants lorsqu'ils voulaient à tout prix chasser ma concentration.

Enfin, pour ne pas vexer ceux que j'ai oubliés, que toute personne qui m'a apporté son aide de près ou de loin trouve ici mes sincères remerciements.

¹ Sourate El-Fatiha, Traduction du Saint Coran, La présidence générale des directions des recherches scientifiques islamiques, de l'Ifta, de la prédication et de l'orientation religieuse, Arabie Saoudite.

² Hadith du Prophète (que le salut soit sur lui) relaté par Ettirmidhi et Abou-Daoud.

SOMMAIRE

INTRODUCTION GENERALE

PREMIERE PARTIE : LES SYSTEMES CELLULAIRES DE PRODUCTION

Chapitre 1 : Le monde de la production	3
1.1 Historique	3
1.2 La fonction production	4
1.3 Les différents modes de production	5
1.3.1 Les principaux modes de production.....	5
1.3.2 Les critères de sélection d'un mode de production	6
1.4 Les objectifs de la fonction production	6
1.5 Entre productivité et flexibilité	7
1.5.1 Les acteurs de la flexibilité	7
1.5.2 Caractère conflictuel des deux objectifs.....	8
1.6 Les systèmes de production traditionnels.....	9
1.6.1 Les systèmes linéaires	9
1.6.2 Les systèmes non linéaires	9
1.7 Rués vers la modernisation	10
Chapitre 2 : Une nouvelle technique de production : le Juste-à-Temps.	12
2.1 Historique	12
2.2 Description	13
2.3 Les acteurs du JAT	14
2.3.1 Le système KanBan	14
2.3.2 Robots industriels et machines-outils à commande numérique	16
2.3.3 Le système de manutention utilisant des Véhicules Guidés Automatisés	17
Chapitre 3 : Les Systèmes Cellulaires de Production.	18
3.1 Historique	18
3.2 Description	19
3.3 Démarches modernes de conception des SCPs	21
3.3.1 Prétraitement des données.....	22
3.3.2 Composition des cellules	22
3.3.3 Agencement intracellulaire	23
3.3.4 Agencement intercellulaire	23
3.4 Avantages et inconvénients des SCPs	23
DEUXIEME PARTIE : OPTIMISATION MULTICRITERE	
Chapitre 4 : Les fondements théoriques de l'optimisation multicritère.	26
4.1 Problème multicritère	26
4.1.1 Définition d'un problème d'optimisation multicritère	27
4.1.2 Optimalité au sens de Pareto	27
4.2 Classification des procédés de résolution multicritères.....	29
4.3 Quelques méthodes d'Optimisation multicritère	30
4.3.1 Méthode de la somme pondérée.....	30
4.3.2 Méthode ϵ -contrainte	31
4.3.3 Méthode de programmation par but.....	32
Chapitre 5 : Les méthodes évolutionnaires multicritères.	34
5.1 Principes de base des Algorithmes Génétiques	34
5.2 Historique	35
5.3 Description de quelques méthodes évolutionnaires	36
5.3.1 La méthode Vector Evaluated Genetic Algorithm.....	36
5.3.2 Niched Pareto Genetic Algorithm.....	37
5.3.3 Nondominated Sorting Genetic Algorithm I.....	37
5.3.4 Strength Pareto Evolutionary Algorithm I	38

5.3.5 Non-dominated Sorting Genetic Algorithm II	39
5.3.6 Strength Pareto Evolutionary Algorithm II.....	40
5.4 Observations sur les méthodes évolutionnaires multicritères	41
5.4.1 Prise en compte des contraintes	41
5.4.2 Incomparabilité due à la relation de dominance	42
5.4.3 Evaluation des performances	43

TROISIEME PARTIE : PROBLEME DE COMPOSITION DE CELLULES STATIQUE

Chapitre 6 : Etat de l'art, position du problème et formulation.	46
6.1 Etat de l'art.....	46
6.1.1 Les précurseurs	46
6.1.2 L'essor	47
6.2 Classification des méthodes existantes.....	48
6.2.1 Classification en méthodes exactes et approchées	48
6.2.2 Les méthodes directes et indirectes.....	48
6.2.3 Les méthodes hiérarchiques et non-hiérarchiques	49
6.2.4 Les méthodes monocritères et multicritères.....	49
6.2.5 Les méthodes basées sur la matrice d'incidence produits/machines.....	49
6.2.6 Les méthodes utilisant les coefficients de similarité	49
6.2.7 Les mét-heuristiques	50
6.3 Description du problème de composition.....	51
6.3.1 De quoi il s'agit ?	51
6.3.2 Les contraintes	53
6.3.3 Les critères	56
6.4 Position du problème.....	63
6.4.1 Hypothèses de résolution	63
6.4.2 Contraintes	63
6.4.3 Objectifs	64
6.4 Le modèle	64
6.4.1 Les inputs	64
6.4.2 Les Contraintes	65
6.4.3 Les Fonctions Objectif.....	65
6.5 Complexité du modèle	66
Chapitre 7 : Les algorithmes génétiques pour la résolution du problème de composition.	68
7.1 L'AG traditionnel	68
7.2 AG binaire	71
7.2.1 Codage binaire	71
7.2.2 Population initiale	73
7.2.3 La fonction d'adaptation.....	74
7.2.4 Les opérateurs génétiques	75
7.2.5 Paramètres de contrôle	76
7.3 La méthode SEP-AG	77
7.3.1 Principe de base des méthodes SEP	77
7.3.2 La méthode SEP cocyclique	77
7.3.3 Conditions d'éligibilité d'une cellule à la scission.....	81
7.3.4 Conditions d'optimalité de la méthode SEP	83
7.3.5 Exploration génétique	84
7.4 Evaluation des performances.....	84
7.4.1 Les données des exemples	85
7.4.2 Analyse des résultats.....	87
Chapitre 8 : Composition de cellules avec routages alternatifs.	89
8.1 Motivation	89
8.2 Identification du problème	89
8.3 Formulation	91
8.3.1 Les inputs	91
8.3.2 Définition d'une solution du problème	91

8.3.3	Contraintes de réalisabilité.....	92
8.3.5	Fonction multi objectifs	93
8.3.6	Exemple illustratif.....	94
8.4	Méthodes de Résolution proposées	95
8.4.1	Résolution par l'AG.....	95
8.4.2	Résolution par hybridation AG-Tabou avec alternance implicite.....	97
8.5	Performances des méthodes proposées	101
8.5.1	Les données des exemples	101
8.5.2	Les résultats obtenus	103
8.5.3	Analyse des résultats.....	106

QUATRIEME PARTIE : PROBLEME DE COMPOSITION DE CELLULES DYNAMIQUE

Chapitre 9 : Etude de l'aspect dynamique.	110	
9.1	Motivations.....	110
9.2	Vue d'ensemble sur l'aspect dynamique	111
9.2.1	Perturbations dans les systèmes dynamiques et approches déterministes et non déterministes	111
9.2.2	La flexibilité revisitée	111
9.3	Influence de l'aspect dynamique sur la conception des SCPs	112
9.3.1	Prétraitement des données.....	112
9.3.2	Composition de cellules	112
9.3.3	Agencement intracellulaire	113
9.3.4	Agencement intercellulaire	113
9.4	Etat de l'art.....	113
9.5	Position du problème.....	115
9.6	Une formulation pour <i>PCAC</i>	116
9.6.1	Les inputs	116
9.6.2	Les Contraintes	117
9.6.3	La fonction multiobjectif	117
9.7	Le coût de reconfiguration	118
9.7.1	Problématique	118
9.7.2	Détermination exacte du coût de reconfiguration	119
9.8	Une formulation pour <i>PCPC</i>	122
9.9	Le problème de décision entre stratégie passive et active	124
Chapitre 10 : Approches de résolution pour l'aspect dynamique.	125	
10.1	Méthodes de résolution pour <i>PCPC</i>	125
10.2	Méthodes de résolution pour <i>PCAC</i>	126
10.2.1	La méthode du plus court chemin	127
10.2.2	Un AG classique	130
10.3	Améliorations floues	133
10.3.1	Recueils de la théorie des sous-ensembles flous	134
10.3.2	Détermination des informations influentes	136
10.3.3	Les règles d'inférence.....	138
10.3.4	L'AG flou	139
10.4	Résultats et performances.....	140
10.4.1	Les données des exemples	141
10.4.2	Embarquement de connaissance dans l'AG	142
10.4.3	Analyse des résultats.....	143

CONCLUSION GENERALE

BIBLIOGRAPHIE

ANNEXES

INTRODUCTION GENERALE

Le travail que nous présentons entre dans le cadre de la conception des *Systèmes Cellulaires de Production* (SCP). Les SCP sont une mise en œuvre de la philosophie Technologie de Groupe qui cherche à exploiter les similitudes entre les produits pour en tirer profit dans la perspective d'amélioration des bénéfices et de réduction des gaspillages dans la production. La mondialisation de la concurrence est l'un des facteurs qui ont incité un très grand nombre d'industriels des pays développés à convertir leurs firmes pour adopter cette "nouvelle" technique. Ceci a commencé par l'industrie automobile, puis ça s'est rapidement propagée à d'autres industries, telles que les composants électroniques, la métallurgie, la plasturgie et même dans l'industrie des textiles et des détergents. La raison de cet engouement n'a pas été dissimulée : il s'avère que les SCP réalisent un très bon compromis entre productivité et flexibilité, garantes de réussite pour toute entreprise qui se veut compétitive.

Les SCP 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 quatre étapes essentielles : (1) Le prétraitement des données, (2) La composition de cellules, (3) L'agencement intracellulaire et (4) l'agencement intercellulaire. Notre travail concerne une partie du module de composition de cellules qui constitue le point d'articulation dans la conception. Dans ce module se fait le groupement des machines en cellules et des produits en familles.

Nous décomposons notre travail en quatre parties : La première est consacrée à la description du domaine d'étude. A cet effet, un premier chapitre décrit des notions relatives au monde de la production. Le deuxième chapitre est consacré à l'exposé d'une nouvelle technique de production qu'est le Juste-à-Temps. Celle-ci est à l'origine du regain de considération voué aux SCP. Le troisième chapitre est consacré à la présentation des SCP ainsi qu'une description des principales étapes de leur conception.

Notre étude du problème de composition adopte une approche multicritère, car, comme tout problème réel, la composition de cellule renferme plusieurs critères. A cet effet, la deuxième partie est dévouée à la présentation de notions générales liées à l'optimisation multicritère. Le chapitre quatre réalise une vue globale sur les fondements de cette théorie, tandis que le chapitre suivant présente des méthodes de résolution appartenant à la famille des approches évolutionnaires. Le choix de ce type d'approches est justifié par les bonnes performances qui ne cessent d'être rapportées concernant leur application sur les problèmes réels multicritères.

Dans la troisième partie nous traitons le problème de composition sous l'aspect statique. C'est à dire que nous supposons inchangé l'état du système. Dans le chapitre six, nous positionnons le problème, nous exposons son état-de-l'art et nous proposons une formulation mathématique basée sur la théorie des graphes. Le modèle mathématique nous permet de classer le problème dans la famille des problèmes NP-Complets. Ce qui justifie l'approche heuristique évolutionnaire présentée dans le septième chapitre. Le chapitre huit traite une version plus difficile du problème de composition qui consiste à considérer des routages alternatifs.

La quatrième partie est consacrée à l'étude du problème sous son aspect dynamique. Ainsi, un premier chapitre ôte le voile sur l'importance de la prise en considération de cet aspect qui a été relativement peu étudié dans la pléthore de publications traitant du problème. Aussi, nous posons le problème sous ce nouvel aspect et montrons que celui-ci peut être abordé suivant deux stratégies : passive et active. Un problème de décision entre ces deux stratégies surgit dans ce cas. Une formulation multicritère est présentée pour tenir compte des stratégies évoquées et du problème de décision. Nous démontrons alors que sous la stratégie passive le problème revient à un problème de composition sous l'aspect statique. Le chapitre dix est consacré à l'exposé des méthodes de résolution sous l'aspect dynamique. Pour résoudre le problème suivant la stratégie active, nous présentons une méthode basée sur l'algorithme de plus court chemin et une méthode évolutionnaire. Pour tenir compte du problème de décision entre stratégie passive et active nous proposons un Algorithme Génétique intelligent.

Nous terminons cette thèse par une conclusion générale qui définit les contributions essentielles proposées tout au long du manuscrit et indique les voies de recherche futures pour l'amélioration du travail présenté.

Enfin, nous avons jugé utile d'ajouter en annexe un lexique de quelques termes utilisés lors de l'exposé des différents développements présentés dans ce manuscrit avec leurs significations et les détails des résultats obtenus.

Chapitre 1

LE MONDE DE LA PRODUCTION

La composition de cellule fait partie du domaine vaste qu'est le monde de la production. Présenter ce domaine et localiser dans celui-ci le champ que nous nous proposons d'étudier serait plus que bénéfique. A cet effet ce chapitre commence par un bref survol historique. Ensuite nous décrivons la fonction production et abordons les différents modes utilisés pour sa mise en œuvre. L'étude est par la suite orientée vers les principales perspectives qui sont liées à la fonction production. L'exposé de ces dernières fait apparaître deux objectifs malheureusement contradictoires : productivité et flexibilité. Le sens du deuxième étant plus important vu les nouvelles tendances du marché, on le décrit de façon plus ou moins détaillée. La section suivante présente les systèmes de production classiques et met en relief leur positionnement par rapport aux deux objectifs évoqués. Ceci permet de rendre palpable la non-conformité des systèmes classiques avec les nouvelles exigences de l'environnement de production, et donc la nécessité d'un nouveau système mettant à profit les nouvelles technologies et les techniques modernes de production. Le chapitre et conclu par l'exposé d'une expérience réelle qui enseigne qu'une telle démarche doit être réalisée avec sérénité et précaution.

1.1 HISTORIQUE

Avant que la production ne devienne ce qu'elle est maintenant, un long parcours a été gravi. Au début, il y a de cela des dizaines de siècles, la production était exclusivement *manuelle*. Vers le milieu du dix-neuvième siècle, la révolution industrielle a engendré une grande métamorphose. L'organisation de la production a été progressivement améliorée. La production, de type artisanal et peu structurée, a cédé la place à la fabrication industrielle en *grandes séries*. La nécessité d'étude des procédés industriels de manière scientifique commençait à se manifester.

Vers mille huit cents quatre vingt quinze (1895), les travaux de FW. Taylor (1856-1915) ont présenté une première tentative d'étude des systèmes de production [Strateg, DEMAp1999]. Ces travaux ont valu à Taylor d'être considéré comme le précurseur de *l'organisation scientifique du travail* (OST) ou *Taylorisme*. Influencé par les tendances du marché de l'époque caractérisée par une forte demande, Taylor développa une philosophie basée sur l'analyse et la quantification du travail : Des experts déterminaient et optimisaient la durée de chaque tâche à chaque poste de travail. La durée d'une tâche était d'autant plus réduite que l'ouvrier l'effectuait de façon répétée. Cette estimation détaillée des temps de travail autorisait une planification précise de la production ainsi que l'équilibrage des chaînes de montage. Les travaux de Taylor contribuèrent au développement de l'analyse des systèmes de production pour devenir, après, une discipline indépendante.

H. Ford (1863-1947), hanté par les travaux des précurseurs de l'OST, a pu bâtir son empire industriel en se basant sur de telles idées [Beranger1987, Chatelet1996, DEMAp1999, Strateg]. Il a assit son prestige dans le domaine de l'industrie automobile en adoptant, pour fabriquer le légendaire modèle Ford T, une stratégie qu'il décrivit fidèlement par une célèbre allocution : «Les clients peuvent avoir la couleur de leur choix, à condition qu'elle soit noire». C'est la stratégie de production par *chaîne de montage* qui permet une grande productivité au détriment d'un degré de flexibilité quasiment nul. Cette organisation a permis à H. Ford d'augmenter la productivité de manière extraordinaire : En 1909, alors qu'en France le plus grand constructeur faisait sortir quatre cents (400) voitures, la Ford T était produite avec la cadence surprenante de dix huit mille (18000) unités par an.

Suite à l'amélioration du niveau de vie après la deuxième guerre mondiale, la balance de l'offre et la demande, caractérisée par une situation de pénurie, commençait à changer jusqu'à subir au début des années soixante-dix un inversement total : c'est le temps de l'offre excédentaire caractérisé par une clientèle de plus en plus exigeante. Les hypothèses de la théorie de Taylor devinrent alors archaïques et le souci unique de productivité maximale se trouvait contraint de se marier avec d'autres soucis tels que qualité, originalité et flexibilité.

Ces nouvelles exigences ont amenés les industriels à se convertir vers les *ateliers flexibles* (en Anglais, *Flexible Manufacturing Systems* qui sont en général organisés en job shop, voir 1.5.2). Les premières implantations sont réalisées aux Etats-Unis en 1967, suivis par le Japon en 1970 puis l'Europe [Jubin1994]. Cependant, les problèmes qui y sont liés et la quête toujours acharnée de nouvelles techniques a donné naissance au concept de *cellule et production cellulaire* (voir chapitre 3).

Ces recueils de l'histoire nous permettent de comprendre l'intérêt que portent les industriels à l'organisation scientifique de la production. Ce chapitre nous permet d'en présenter les notions de base.

1.2 LA FONCTION PRODUCTION

La production est le résultat de la mise en œuvre simultanée de moyens de production et de processus reflétant la manière dont sont conçus les produits à fabriquer.

Le terme production décrit un ensemble de tâches effectuées par un ensemble de ressources (machines) constituant le système de production. Ces ressources sont regroupées de manière à former un système qui permet de réaliser une telle production.

Ce système doit remplir les fonctions suivantes [Souilah1994]:

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 fabrication, de la disponibilité des ressources, de son niveau de priorité, ... etc.

1.3 LES DIFFERENTS MODES DE PRODUCTION

La fonction production peut être réalisée suivant différents modes. Parmi ces modes nous décrivons les suivants.

1.3.1 Les principaux modes de production

Les modes de production peuvent être résumés en quatre types essentiels :

1. la production unitaire.
2. la production par lots.
3. la production par série.
4. la production continue.

L'utilisation de ces différents modes dépend du domaine de production visé. Quelques caractéristiques sont exposées dans la table 1.1 [Geronim].

Table 1.1 Les modes de production traditionnels.

	Mode de production			
	Unitaire	Par lots	En série	Continu
Description	Fabrication sur mesure en fonction de la demande du client	Petites et moyennes séries diversifiées de produits identiques	Grande série de produits identiques et standardisés	Flux continu de produit homogènes
Automatisation	Limitées aux fonctions essentielles	La flexibilité assurée par la présence d'équipements programmables	Forte automatisation à partir d'équipements peu flexibles	Elle s'applique à tous les stades de la production.
Pilotage	par l'aval (sur commande)	par l'aval (sur commande)	par l'amont (Production anticipée)	par l'amont (Production anticipée)
Spécificités	Recherche de qualité et de flexibilité	Juste à temps et ressources flexibles	Recherche d'économie d'échelle mais pas de flexibilité	Economie d'échelle et automatisation complète
Exemple	Industrie navale, aviation	Industrie automobile, électronique	Sous-traitants en fabrication mécanique	Boissons, Sidérurgie

1.3.2 Les critères de sélection d'un mode de production

Bien que chaque entreprise soit libre d'organiser comme elle le souhaite sa fonction de production, le choix du mode de production repose sur l'analyse d'un ensemble de critères [Geronim] qui sont : la quantité produite, la nature du processus de production, le type de gestion et l'organisation des ressources disponibles.

1.3.2.1 Les quantités de biens ou services devant être produits

Le mode de production 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 le mode de production sera standardisé et reposera sur une structure de production formelle et rigide.

1.3.2.2 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 (production à la demande ou fractionnée dans le temps ou dans l'espace).

1.3.2.3 La nature du type de gestion de la production

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

1.3.2.4 L'organisation des ressources

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 tâches nécessaires à la production d'un bien qui circulent 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. Les deux premiers types d'organisation sont détaillés dans la section 1.5, tandis que le chapitre **trois** sera consacré au dernier.

1.4 LES OBJECTIFS DE LA FONCTION PRODUCTION

Le recours à la fonction production renferme plusieurs perspectives. Ceux-ci peuvent être résumés par les objectifs suivants [Geronim] :

1. La productivité : La fonction de production doit permettre à l'entreprise d'atteindre le degré de productivité qui permet de satisfaire la demande qui lui est adressée. Ce qui suppose que l'entreprise adapte sa capacité de production au volume des ventes. Ceci passe par des actions visant à maintenir en l'état les capacités productives ou par la mise au point de plans d'investissement en capacité (acquisition de nouvelle machines, sous-traitance).
2. La qualité : Les biens économiques produits doivent être de bonne qualité. C'est-à-dire, ils doivent permettre de satisfaire les besoins de la clientèle. La production doit être aussi de qualité en terme d'utilisation de ressources afin de respecter le critère d'efficience attaché au système productif. Donc, le système productif doit être économique en ressources et constant en terme de qualité.
3. La réduction des coûts : Le système productif, adopté par l'entreprise, doit proposer les plus faibles coûts de production possibles de manière à garantir la compétitivité de l'entreprise. De plus, les coûts de production calculés doivent aussi être mis en relation avec les coûts de production prévus par le centre opérationnel. Sur la longue période, cet objectif de coût se traduit par la recherche permanente de gains de productivité afin de détenir ou de conserver un avantage compétitif coût pour l'entreprise.
4. Le respect des délais : Le système de production doit certes produire, mais dans des délais raisonnables, c'est-à-dire en conformité avec le niveau de la demande à laquelle doit faire face l'entreprise. Ceci suppose la mise en place d'un mode de production réactif qui permettra soit d'éviter des stocks de biens finaux, soit de ne pas connaître de goulets d'étranglement. En terme de productivité, l'objectif de délai signifie aussi réduire les délais de fabrication.
5. La flexibilité : Le système productif doit être flexible soit pour pouvoir s'adapter aux variations de la demande, soit pour tenir compte des évolutions de l'environnement productif de l'entreprise (innovations technologiques...), soit pour permettre une production simultanée de plusieurs types de produits différents en même temps.

Certains de ces objectifs sont contradictoires (tels que qualité et flexibilité d'une part et délai et coût de l'autre, voir la section suivante). Il revient alors à chaque entreprise de bien prendre en compte les revers liés aux moyens utilisés pour atteindre chaque objectif afin d'adopter la stratégie de compromis adéquate de façon réfléchie.

I.5 ENTRE PRODUCTIVITE ET FLEXIBILITE

Parmi les objectifs qui viennent d'être cités, deux sont très redondants dans les discours des industriels. Il s'agit de la productivité et de la flexibilité dont l'impact sur la capacité de faire face à la concurrence n'est pas à démontrer. Cependant, si la productivité est un terme dont le sens est universellement admis, et qui représente la quantité produite par rapport au travail fourni et aux dépenses engagées, la flexibilité requiert un peu plus de clarification.

La flexibilité peut être définie par la capacité d'adaptation de la production à tout phénomène nouveau qui peut intervenir sur la fonction production. La flexibilité fait donc apparaître une double contrainte liée à l'incertitude des phénomènes et à l'urgence des réactions à entreprendre [Everaere1999]. Ces phénomènes sont généralement la fluctuation des demandes, les pannes de machines, l'apparition de concurrents, la rupture d'approvisionnement des fournisseurs et des sous-traitants, ... etc.

I.5.1 Les acteurs de la flexibilité

Pour qu'un système de production soit flexible, il faut qu'il soit conçu à base d'éléments flexibles. Dans ce qui suit nous décrivons les facettes de la flexibilité pour les différents composants d'un système de production [Jubin1994, DeToni1998, Everaere1999].

I.5.1.1 Flexibilité des ressources

Une ressource est flexible, si elle peut intervenir dans la production de plusieurs types de produits. Autrement dit, si elle peut réaliser plusieurs types d'opérations en changeant rapidement les outils de production. Certains auteurs [Everaere1999] préconisent qu'il n'y a de ressource flexible que l'homme. Pour les machines, il serait plutôt question de polyvalence.

I.5.1.2 Flexibilité des moyens de manutention

Un moyen de manutention est flexible s'il peut transporter plusieurs types de produits sous différentes formes (par lot, par unité, sur des palettes). Le moyen de manutention le plus flexible est l'homme jusqu'à une certaine limite de charge et de dimension. Il existe plusieurs types de moyens de manutention flexibles. Nous pouvons citer comme exemple le Véhicule Guidé Automatique (VGA) ([voir paragraphe 2.1](#)).

I.5.1.3 Flexibilité de l'agencement des ressources

L'organisation des ressources est l'un des points déterminants dans les performances des systèmes de production. Cette organisation est flexible si elle permet plusieurs cheminements possibles entre les différents postes de travail. Parmi les organisations flexibles, nous avons celles orientées opération qui donne naissance à la notion d'ateliers flexibles ([voir paragraphe 1.6.2](#)).

I.5.1.4 Flexibilité du système d'information

Le système d'information sert à véhiculer les informations concernant les produits et les ressources. Il est flexible s'il peut incorporer de façon aisée le lancement de nouveaux produits et/ou l'acquisition de nouvelles ressources. Il doit aussi être en mesure de faire parvenir l'information au poste adéquat, de façon fiable et rapide. Parmi les systèmes d'information utilisés en ce sens, nous avons le système Kanban (voir paragraphe 2.2.1).

I.5.1.5 Flexibilité du système de gestion

Un système de gestion doit réaliser les fonctions de planification, d'exécution et de contrôle de la production. Dans ce cas la flexibilité du système concerne de façon essentielle la prise de décisions. Pour être flexible, cette dernière doit être réalisée de façon décentralisée, d'une part. De l'autre, elle ne doit pas s'articuler sur des constats purement prévisionnels qui consistent à extrapoler le passé sans tenir compte des aléas futurs.

1.5.2 Caractère conflictuel des deux objectifs

Si la productivité était le seul souci des industriels des années d'avant Guerre, la situation s'est complètement inversée à la faveur du critère de flexibilité après la mondialisation de la concurrence. En effet, la production de masse est devenue révolue et il fallait s'orienter vers la production par moyennes et petites séries pour rester compétitif. Cependant, des changements fréquents de séries ne peuvent se faire qu'au détriment de la productivité de l'entreprise à cause des réglages à effectuer sur les ressources, le temps de familiarisation des opérateurs avec les nouvelles tâches, l'éventuel changement dans l'organisation et la planification, ... etc. Le constat qu'on peut tirer de tout ça est qu'il faut réaliser un compromis entre ces deux objectifs. Mais ceci est-il possible ? La réponse est affirmative et consiste à marier les vertus des deux systèmes traditionnels qui sont les systèmes linéaires et non linéaires.

1.6 LES SYSTEMES DE PRODUCTION TRADITIONNELS

L'organisation des systèmes de production dépend de la nature des produits et du type de fabrication associé. Les systèmes traditionnels se ramènent, en ce qui concerne leur organisation, à deux types de base : les systèmes linéaires (ou *flow shop*) et les systèmes non linéaires (ou *job shop*) [Souilah1994].

1.6.1 Les systèmes linéaires

Les systèmes linéaires sont dédiés à la fabrication de masse d'un nombre relativement petit de types de produits ayant des gammes de fabrication semblables. Pour ce genre de système, une organisation orientée

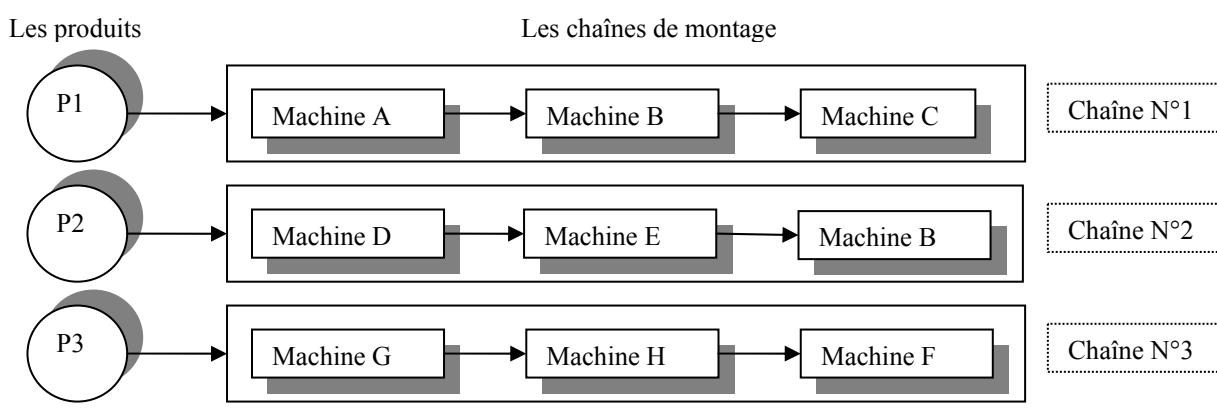


Figure 1.1

Système de fabrication orienté produit.

produit est utilisée : l'organisation des ressources se fait en fonction des exigences du produit, c'est à dire, que les ressources sont groupées de sorte que chaque groupe de ressources fabrique entièrement un ou plusieurs types de produits utilisant les mêmes ressources (fig. 1.1).

Les avantages d'une telle disposition sont le contrôle aisé, le temps de fabrication réduit et la possibilité d'utiliser une main d'œuvre non qualifiée. Cependant, des inconvénients ont été signalés, à savoir : l'investissement important dû à la duplication de machines, le fait que la panne d'une machine bloque la production dans la chaîne associée et enfin, la rigidité du système dû à un coût très élevé de reconfiguration.

Lorsqu'il y eut un changement de balance entre l'offre et la demande, certaines industries étaient contraintes de se convertir à la production en moyennes et petites séries pour lesquelles les systèmes non linéaires étaient tout indiqués.

1.6.2 Les systèmes non linéaires

La production de masse devenue dépassée, des secteurs de la production se sont orientés petit à petit vers des modes où la fabrication est réalisée en quantités d'importance moyenne relative à une grande variété de type de produits ayant des gammes différencierées. Dans ce genre de systèmes, une organisation orientée opération s'est imposée : les machines sont groupées en fonction des opérations qu'elles réalisent. Ainsi, pour une usine de fabrication mécanique, les machines de tournage sont réunies ensemble de même que les machines de fraisage, de perçage, ... etc. Pour subir les différentes séquences d'opérations composant leurs gammes de fabrication, les produits passent d'un groupe à un autre (fig. 1.2).

Les avantages de tels systèmes, qui sont leur flexibilité et le fait que les pannes de machines ne bloquent pas toute la production, les rendent plus fiables aux aléas de la production. Cependant, ce genre de système est caractérisé par une augmentation de la longueur des trajets qui engendre une tâche d'ordonnancement complexe. En plus, il y a la nécessité de mise de stocks tampons entre les machines, d'où un plus grand investissement en matières premières et en *en-cours* ;

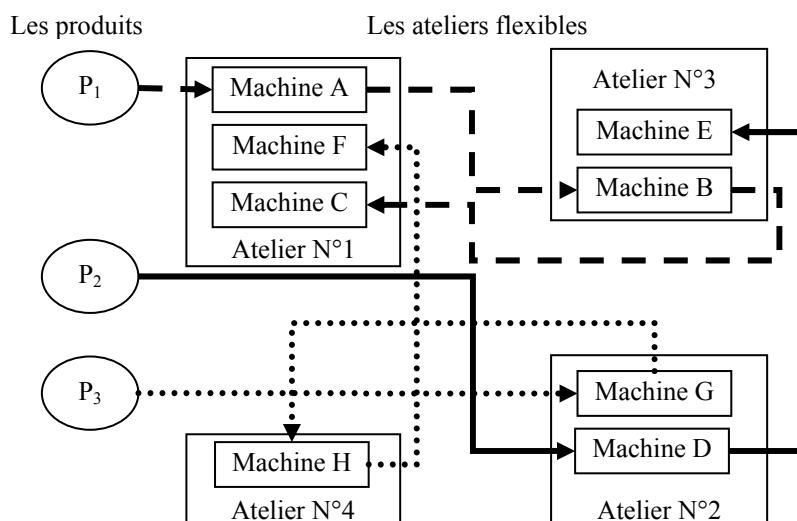


Figure 1.2 Les systèmes de fabrication non linéaires.

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 trajet et attente dans les stocks tampon). La majorité de ce temps est consommée en attente dans des files, le restant est partagé entre la préparation des lots de produits et les traitements [Stone2000].

1.7 RUES VERS LA MODERNISATION

La période de conversion aux systèmes non linéaires a vu la naissance de nouvelles technologies voulant marier les derniers développements de l'outil informatique et de l'électronique pour concevoir des machines flexibles avec un haut degré d'automatisation. Pour rester compétitifs, les insuffisances, que présentaient les systèmes traditionnels, incitaient les dirigeants industriels à investir, avec précipitation, pour moderniser. Cependant, combien de situations ont montré que le recours aux nouvelles techniques n'était qu'un mauvais souvenir. Nous présentons en guise d'argument le cas réel suivant [Béranger1987] : Dans une usine américaine sous-traitante de l'industrie automobile, un projet de modernisation est mis en place en 1984. Il s'agit de l'automatisation complète de la fabrication d'un élément du châssis des automobiles. L'installation comprend une quinzaine de robots, chargés des opérations successives du processus de production de cet élément. L'ensemble du processus doit permettre de réaliser complètement le produit sans intervention humaine. Il représente un investissement de 3,3 millions de dollars.

La mise au point du système s'avère très difficile. Il est impossible de démarrer la production, comme prévu, au printemps 1985. Il est, néanmoins, nécessaire d'approvisionner l'usine cliente à partir de cette date. Il est donc décidé d'installer temporairement une ligne manuelle, équipée de machines traditionnelles, afin d'assurer la production.

Une année plus tard, au printemps 1986, la situation est la suivante : La ligne automatisée est en production depuis plusieurs mois. Elle fonctionne effectivement sans ouvrier. Toutefois, deux techniciens du service entretien et un ingénieur lui sont affectés à plein temps pour remédier au problème de fiabilité. Sur la ligne, les robots ne peuvent saisir les pièces à assembler sur le produit que si ces pièces sont présentées dans des conteneurs spéciaux. Un sous-traitant est donc chargé de mettre dans de tels conteneurs les pièces livrées par les fournisseurs. Le système est arrêté pendant quarante à cinquante pour cent du temps : dans un ensemble intégré aussi complexe, il est fréquent qu'un équipement soit en panne. Mais le volume de production requis nécessite un taux de fonctionnement de quatre-vingts à quatre-vingt-dix pour cent, sur trois équipes. Il a donc été décidé de maintenir la ligne manuelle "temporaire".

La ligne manuelle a coûté 390 000 dollars. Elle produit avec six ouvriers 1100 pièces en huit heures, contre 2000 pièces pour la ligne robotisée. Sa fiabilité est excellente. Sa production a un taux de défectuosité inférieur de moitié à celui de la ligne automatisée. Elle occupe deux fois moins de surface au sol.

En résumé, le taux de fonctionnement de l'ensemble automatisé est trop faible pour rentabiliser l'investissement. Il existait une autre approche, beaucoup moins coûteuse, pour produire mieux, plus vite et pour moins cher.

Cette malencontreuse expérience enseigne que pour se moderniser, il faut adopter une démarche globale qui ne se réduit pas à la simple acquisition de ressources high-tech, mais plutôt une réflexion approfondie qui tient compte

de tous les paramètres qui vont avec, tels que le système de gestion, le système d'information et surtout l'organisation. Le chapitre suivant est consacré à l'exposé d'une nouvelle technique de production adoptant une telle démarche. Cette technique ne cesse de susciter l'intérêt des industriels après le succès qu'elle a réalisé dans l'entreprise mère.

Chapitre 2

UNE NOUVELLE TECHNIQUE DE PRODUCTION : LE JUSTE A TEMPS

Le développement de l'outil informatique et des moyens modernes de production a permis la genèse de nouvelles méthodes et techniques de production. Ces nouvelles techniques sont désormais la clé des actions visant à faire face à la mondialisation de la concurrence. Les perspectives d'utilisation de ces techniques se résument à :

1. augmenter la productivité;
2. réduire les temps de fabrication;
3. faciliter la gestion;
4. améliorer la qualité;
5. accroître la flexibilité.

Afin de décrire les différentes facettes de ces techniques modernes de production, un terme, la *productique*, a été adopté et est largement employé.

Ce chapitre réalise un bref survol sur l'une de ces nouvelles techniques qui a permis d'asseoir le prestige des grandes firmes de production telles que Toyota et General Motors, ... etc. Il s'agit du Juste-à-Temps. Pour ce faire, nous présentons dans la première section des bribes de l'histoire de la technique. La section suivante en décrit le principe général. Enfin, certains points essentiels de la technique, que nous appelons acteurs, sont présentés dans la dernière section.

2.1 HISTORIQUE

Le Jste-à-Temps (JAT ou JIT pour Just In Time) est une philosophie née chez Toyota, le géant mondial de construction d'automobiles [Strateg]. La victoire des Alliés à la deuxième guerre mondiale et l'arsenal du matériel utilisé avaient attirée l'attention des industriels japonais. Ils étudièrent alors les méthodes américaines de production avec une attention particulière aux procédés de Ford.

A Toyota Motor Company, Ohno T., vice-président chargé de la production, et son bras droit Shingo S., commencèrent à combiner essentiellement le Fordisme et la méthode *Statistical Process Control* [Chatelet1996] pour proposer l'approche appelée Toyota Production System ou Juste à Temps.

Les gérants de Toyota ont cependant reconnu que le Fordisme comportait quelques imperfections, en particulier concernant les rapports avec la masse ouvrière : Avec la prolifération des unions syndicales, le Fordisme,

réputé dur envers la masse ouvrière, était inadéquat dans l'environnement du Japon de l'après Guerre. Il était aussi inadéquat au contexte américain. Mais les joies de la victoire ont eu pour effet de retarder ce constat qui allait devenir après quelques années évident.

Toyota a découvert que les ouvriers pouvaient jouer un rôle plus important afin de réaliser l'objectif de qualité, la voie inévitable pour faire face à une concurrence émergente. Une autre découverte concernait la nécessité de variation des produits (flexibilité). Le fordisme s'articulait sur un unique produit, et par conséquent ne tolérait pas le lancement de nouveaux produits.

La réponse à ces exigences revenait à tirer profit des développements théoriques existants, tout en essayant de proposer des solutions nouvelles. La synthèse de toutes ces notions et ces techniques que nous appelons acteurs ([voir paragraphe 2.3](#)) a permis de façonner la philosophie JAT.

Tout cela s'est passé entre 1949 et 1975. Avec le succès chez Toyota, le JAT s'est étendu à d'autres firmes japonaises. Lorsque les gains en productivité et en qualité devinrent évidents à l'extérieur, les américains l'importèrent. Ce qui eut pour effet de le répandre à travers le monde entier.

2.2 DESCRIPTION

Si à l'origine, le JAT décrivait le simple fait que chaque pièce doit être traitée puis transportée pour arriver juste à temps à la prochaine opération, suite à des travaux de recherche, cette notion s'est étendue pour devenir une philosophie englobant les différentes étapes de la production. Selon l'APICS (American Production and Inventory Control Society) le JAT est « une philosophie basée sur l'élimination de tous les gaspillages et sur la mise en œuvre d'une stratégie de progrès permanent en terme de productivité » [Lamouri1999].

Le principe du JAT peut s'exprimer comme suit [Béranger1987] :

Il faut produire et livrer :

- les produits finis juste à temps pour qu'ils soient vendus,
- les sous-ensembles juste à temps pour qu'ils soient montés dans les produits finis,
- les pièces fabriquées juste à temps pour être assemblées en sous-ensembles.

Et il faut acheter :

- les matières premières juste à temps pour être transformées en pièces fabriquées.

Le principe du juste à temps paraît d'une grande simplicité. Ainsi, une personne qui ne connaît pas les entrailles de l'industrie, juge qu'il s'agit d'une notion évidente et ne peut imaginer que les entreprises puissent travailler différemment. Mais en réalité, le JAT est une philosophie industrielle globale et dont l'adoption présuppose l'adhésion aux conditions suivantes [Geronim] :

- La première est que l'existence de stocks, que ce soient de produits finis ou de produits intermédiaires, ne constitue plus une variable classique d'ajustement mais est l'expression d'un dysfonctionnement qu'il faut solutionner.
- La seconde est que l'entreprise ne fonctionne plus comme une entité autonome de production mais doit développer ce nouveau mode de production en développant des relations plus fortes avec ses fournisseurs de manière à ce que ceux-ci adaptent leur mode de production selon les principes du JAT.

- La troisième est que le système de production adopté par l'entreprise doit disposer d'un certain degré de flexibilité lui permettant de s'adapter aux variations de la demande que ce soit en terme de volumes, ou en terme de produits (utilisation des outils industriels modernes, voir [paragraphe 2.3.2](#)).
- La quatrième est que les fonctions opérationnelles doivent pouvoir intervenir rapidement pour garantir le bon fonctionnement de l'outil industriel, soit pour faire face à des problèmes de maintenance, soit pour pouvoir s'adapter aux évolutions des modes de production (plus grande flexibilité des hommes).
- La cinquième réside dans la rapidité de circulation de l'information dans l'entreprise qui se doit d'être à la fois verticale, depuis l'aval, et horizontale pour permettre une gestion optimale des flux physiques reliant les différentes cellules de production (prise de décision d'arrêt de production par l'opérateur de cellule, système Kanban, voir [paragraphe 2.3.1](#)).

2.3 LES ACTEURS DU JAT

Afin d'accomplir ces ambitions, le JAT réalise une synthèse d'outils issus d'idées classiques et nouvelles que nous appelons acteurs. Nous donnons dans ce qui suit une brève description de quelques importants acteurs, étant donné que leur étude détaillée sort du cadre de la thèse. Mais nous consacrerons tout un chapitre pour l'étude de l'acteur principal, qu'est la Technologie de Groupe, car il constitue une partie intégrante dans l'exposé du sujet que nous nous proposons de traiter dans cette thèse.

2.3.1 Le système KanBan

La méthode KanBan est une innovation propre au JAT qui sert à maîtriser les niveaux des flux [Lamouri1999]. KanBan est un mot japonais qui signifie carte ou étiquette. Ces cartes sont utilisées comme moyen de signalisation permettant d'autoriser ou non la fabrication des produits [Stone2000]. Ainsi, il est possible [Béranger1987] :

- d'éviter toute interruption de la production d'un atelier ou d'une personne en raison d'un manque de pièces.
- de ne produire les pièces, sous-ensembles et produits finis, que lorsqu'on en a besoin;
- de contrôler les niveaux des *en-cours* afin d'approcher le but utopique zéro stock.

Nous décrivons dans ce qui suit un système KanBan très simple, mis en place par l'usine Long Beach de la société Toyota Autobody Incorporated of California entre 1981 et 1984 [Béranger1987]. Cette usine comprend deux ateliers : un atelier de fabrication de pièces et un atelier de montage. Le deuxième réalise l'assemblage des pièces produites dans le premier pour concevoir des produits finis.

On s'intéresse au procédé de fabrication des pièces A qui vont être assemblées au poste X de l'atelier de montage.

Dans l'atelier de fabrication, les pièces A, sont transportées dans des conteneurs identiques. Chaque conteneur, lorsqu'il est plein de pièces A, porte une étiquette KanBan dont les informations transcrites sont illustrées dans la figure 2.1.

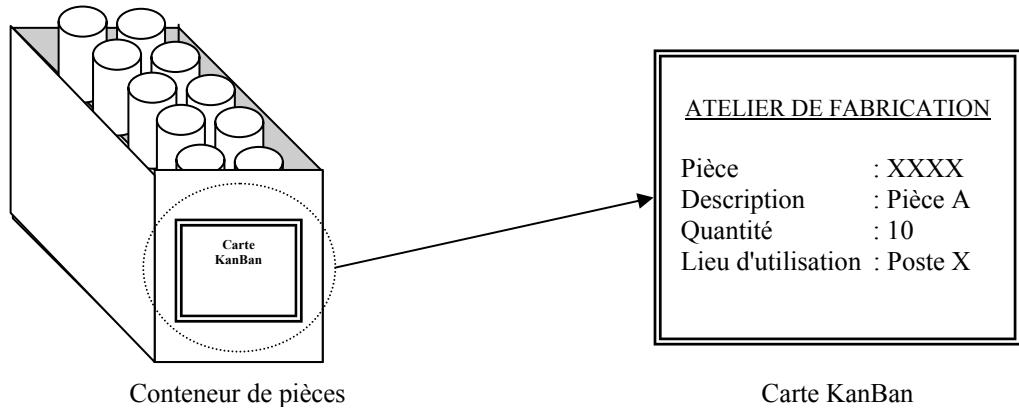


Figure 2.1 Conteneur de pièces et carte KanBan.

Le nombre total de cartes KanBan de la pièce A étant contrôlé, ceci permet de garantir que les *en-cours* de ces pièces ne dépasse pas un niveau déterminé.

Les conteneurs de pièces produites par l'atelier de fabrication sont transportés à l'atelier de montage par un livreur qui dépend de l'atelier de fabrication.

Dans l'atelier de montage, le stock de conteneurs des pièces A est situé au poste de travail X. Lorsque l'opérateur prend la première pièce du premier conteneur, il enlève la carte KanBan de ce dernier et la met de côté dans un petit panier prévu à cet effet. Lorsque le premier conteneur est vide, l'opérateur l'enlève; le conteneur suivant glisse et arrive à portée de main de l'opérateur et ainsi de suite.

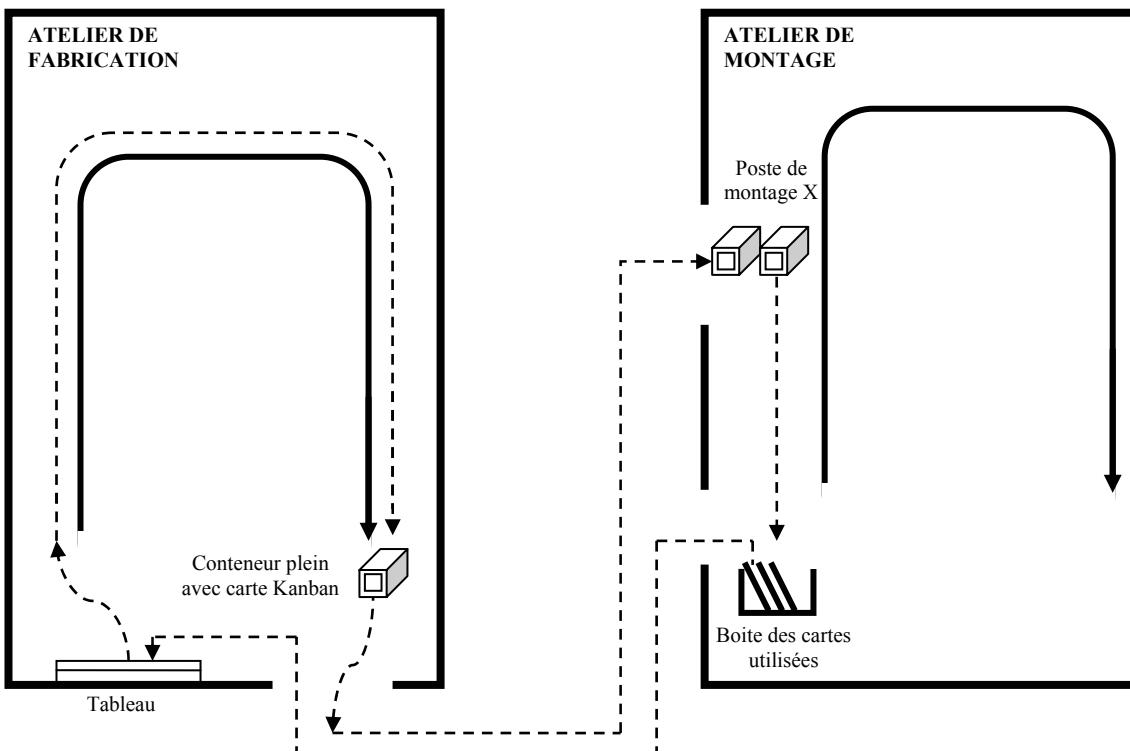


Figure 2.2 circuit fermé des cartes KanBan.

Un agent de manutention de l'atelier de montage récupère tous les KanBan des paniers des postes de montage et les met dans une boîte située dans l'atelier de montage. Chaque fois que le livreur de l'atelier de fabrication vient dans l'atelier de montage, il prend toutes les cartes KanBan de la boîte et les apporte à l'atelier de fabrication pour

les placer dans un grand tableau. Ces cartes deviennent de nouveaux ordres de fabrication pour l'atelier de fabrication (voir figure 2.2 [Béranger1987]).

2.3.2 Robots industriels et machines-outils à commande numérique

Parmi les acteurs modernes du JAT nous avons l'utilisation des machines haute technologie, à savoir les robots industriels et les machines-outils à commande numérique (MOCN).

Un *robot industriel* [Everaere1999] est un appareil automatique programmable se substituant à l'homme grâce à un ou plusieurs bras mécaniques, articulés ou motorisés, et qui doivent s'adapter en étant reprogrammés et/ou grâce à des capteurs, à des tâches diversifiées de manipulation dans l'univers physique.

On distingue trois types de robots :

- les robots transfert (déplacement d'objets) qui sont parfaitement adaptés à des opérations simples et répétitives de manipulation ;
- les robots d'assemblage (soudure) bien adaptés aux grandes séries et aux cadences élevées mais posant des problèmes pour les soudages plus complexes ;
- les robots à trajectoire continue (robots de peinture) conçus pour assurer un mouvement parfaitement uniforme et régulier.

Les vertus de tels équipements automatisés vont de la libération de l'homme d'un certain nombre de tâches pénibles physiquement ou présentant un danger, à la possibilité d'exécuter des mouvements simples, répétitifs avec une grande précision et une parfaite régularité.

À l'instar du robot, la *machine-outil* est également un outil individuel. Mais l'opération effectuée est généralement plus longue (de quelques minutes à quelques heures). On distingue les machines-outils par enlèvement de matière (perceuse, fraiseuse, taraudeuse, ... etc.) de celles par formage (fondeuse, forgeuse, emboutisseur, ... etc.).

Pour ces deux types de machines (robot et machine-outil), les dispositifs de programmation ont incontestablement marqué une étape fondamentale. L'électronique et l'informatique ont permis le passage de l'automatisation rigide, à composante essentiellement mécanique, à l'automatisation flexible. C'est à la programmation que l'on attribue l'acquisition de la propriété de flexibilité.

Deux techniques de programmation s'affrontent et contribuent en même temps à distinguer le robot de la machine-outil. La première est celle dite record/play back (enregistrement et répétition) : le robot enregistre les mouvements effectués par un ouvrier qualifié et les reproduit automatiquement. L'autre technique est plus abstraite et plus indirecte. Elle consiste à transposer en données informatiques et mathématiques, absorbables par un calculateur, les caractéristiques des pièces à transformer (au moyen d'algorithmes) et à connecter le calculateur à la machine-outil [Coriat1990]. C'est cette voie que désigne l'expression « commande numérique » (CN). Cette dernière va elle-même évoluer pour incorporer des calculateurs (commande numérique par calculateur, CNC) ainsi que des microprocesseurs qui vont permettre à la fois de multiplier les mouvements simultanés, de simplifier le travail de programmation et de permettre une autocorrection de la machine par rapport aux données recueillies et traitées pendant le fonctionnement.

2.3.3 Le système de manutention utilisant des Véhicules Guidés Automatisés

Dans les systèmes JAT, une attention particulière est donnée au système de manutention dont la tâche est de réaliser des transports de faible portée. C'est-à-dire, entre les machines de l'atelier de fabrication. Parmi les moyens utilisés, nous avons les Véhicules Guidés Automatisés (VGA).

Un VGA est un véhicule sur roues sans conducteur, guidé à distance (système centralisé) ou auto guidé (système décentralisé). Il est utilisé pour transporter des pièces ou des outils dans l'atelier de fabrication. Dans un système centralisé, les VGAs sont, en général, commandés par signal radio ou par des réseaux de fils électriques implantés au sol. Dans les systèmes décentralisés, les VGA sont dotés d'une certaine intelligence qui leur permet de

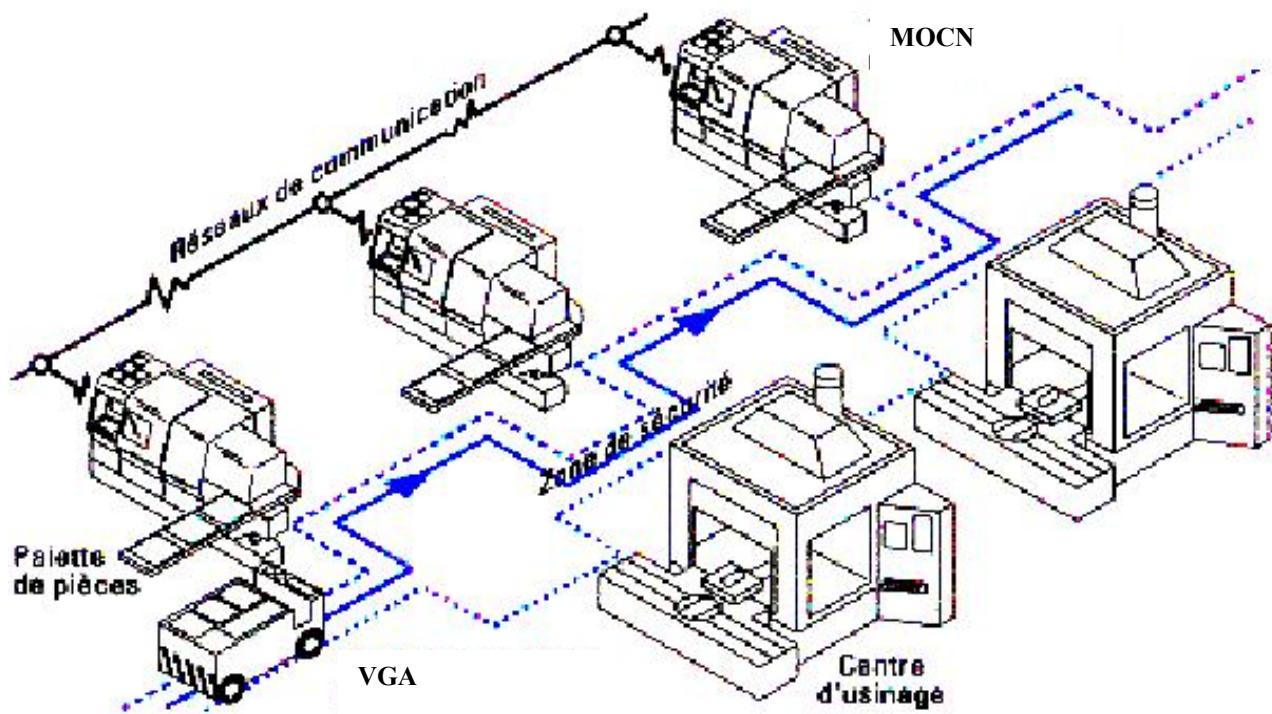


Figure 2.3 Un atelier de production moderne disposant d'un VGA et de plusieurs MOCNs.

circuler de manière autonome. Ils sont, dans ce cas, dotés de caméras qui leur permettent d'éviter les collisions. Les VGAs sont de taille variable allant de quelques centimètres à plus d'un mètre de largeur. La figure 2.3 [Strateg] illustre un atelier de fabrication doté d'un VGA et de plusieurs MOCNs.

Après avoir survolé l'une des principales notions de productique, à savoir le juste-à-temps, en commençant par ça genèse, ses ambitions, nous avons décrit certains de ses acteurs. Il s'agit d'un ensemble d'idées dans lesquels le JAT a pioché pour asseoir sa philosophie. Cependant, si les acteurs qui viennent d'être cités ont contribué au succès qu'a connu le JAT, dans l'environnement industriel, il est admis que le principal atout du JAT est le recours à une ancienne philosophie de la productique. Le chapitre suivant est consacré à cet acteur pour qui le nombre de travaux de recherche publiés et le nombre de firmes qui l'adoptent permet d'en démontrer l'importance.

Chapitre 3

LES SYSTEMES CELLULAIRES DE PRODUCTION

Le constat des chapitres précédents montre que les tendances actuelles du marché vont vers de nouvelles exigences telles que :

- Des livraisons fréquentes par petites quantités ;
- Une bonne qualité des produits et des types différents.

Ces facteurs ont créé un environnement qui a rendu palpable la nécessité de se convertir au mode de production en petites et moyennes séries pour être compétitif. C'est la firme Toyota qui a été présentée comme modèle aux entreprises ayant pu s'adapter à ces nouvelles exigences en adoptant la technique JAT pour la gestion de sa production. Afin de répondre aux attentes de l'approche JAT, une ancienne philosophie jetée aux calendes grecs, « *La Technologie de Groupe* » (TG) était apte à pallier aux dysfonctionnements engendrés par les systèmes de production classiques. La renaissance de l'application du principe de la TG dans les systèmes de production a donné naissance au concept de *Systèmes Cellulaires de Production*.

Ce chapitre présente, donc, une description de cet acteur principal qui sans lequel le JAT n'aurait pas eu le succès qu'il avait eu.

3.1 HISTORIQUE

Bien que des organisations en groupes aient été recensées au début du vingtième siècle [Benders2000], il est communément admis que le concept « Technologie de Groupe » est dû au chercheur russe Mitrofanov S.P. [Mitrofanov1959]. Dès les années 50, Mitrofanov a analysé les moyens et les méthodes de fabrication des pièces pour constater qu'on pouvait améliorer la production en exploitant les analogies des pièces [Souilah1994]. De nombreuses applications industrielles de ses travaux ont été réalisées en tenant compte des formes géométriques des pièces, mais aussi et principalement de l'équipement et de l'outillage nécessaire pour leur usinage. Ainsi, les pièces sont réparties en classes et en sous-classes. En tenant compte de cette classification dans la planification de la fabrication, on parvient à réduire les temps de réglage (setup time) entre deux séries de pièces.

Dans les années soixante, intéressé par ces idées, le professeur Optiz H. [Optiz1970], de l'Université Aachen de l'ex Allemagne de l'ouest, propose une démarche de codification incluant de façon minutieuse tous les stades de la fabrication des pièces. La méthode de d'Optiz devint l'une des méthodes de codification les plus célèbres dans les industries appliquant les principes de la TG [Burbidge1975]. Des applications de la TG ont été recensées en

Allemagne et aux Etats-Unis, mais n'ont pas longtemps survécues [Johnson2004]. Il fallait attendre la fin des années soixante-dix, avec l'avènement du JAT, pour voir la TG renaître de ses cendres. Les premières applications ont été un grand succès. A titre d'exemple, l'usine de General Motors de Dayton (Ohio aux Etat-Unis), en utilisant l'approche TG, a ramené les délais de fabrication de ses pièces de cinq semaines à une demi-heure ! [Béranger1987].

Dans les années quatre-vingt dix, aidée par le développement de l'outil informatique, la TG a pu réaliser des pas sûrs vers le perfectionnement. Ainsi, est né le concept de Technologie de Groupe Assistée par ordinateur (ou TGAO). La TGAO a permis aux entreprises de mieux répondre aux objectifs de coût en simplifiant les études de conception de façon drastique.

Depuis, une fabrication en groupes plus efficace et productive est devenue un besoin indispensable dans les entreprises des pays industrialisés. A titre d'exemple, dans ces dernières années, la production cellulaire constituait entre 43 et 53% de l'activité de production aux Etats-Unis et au Royaume-Uni [Johnson2004].

3.2 DESCRIPTION

La Technologie de Groupe est une philosophie manufacturière qui consiste à identifier puis regrouper dans des familles les produits similaires afin de tirer avantage de leurs similitudes dans les procédés de fabrication [Burbidge1975].

Les premières méthodes de regroupement des produits en familles étaient [Burbidge1975] :

- L'inspection visuelle;
- La classification par codification;
- La méthode Production Flow Analysis (PFA).

La méthode d'*inspection visuelle* est la moins sophistiquée et la moins coûteuse. Elle consiste à classifier les produits dans des familles par une simple détection visuelle de leurs caractéristiques physiques. Ainsi, les produits qui ont des caractéristiques physiques similaires sont groupés ensemble.

Les premiers succès majeurs de l'application de la TG ont été réalisés dans la firme *Langeston Division of Molins Machine Company*, en Cherry Hill aux USA. Cette application a engendré un bouleversement dans la perception de la production, bien qu'elle ait été basée sur l'utilisation de la méthode visuelle [Béranger1987].

La deuxième méthode consiste à classifier les produits par examen de leurs attributs de conception et/ou de fabrication. Ceci permet de donner un numéro de code unique pour chaque produit identifiant ses attributs. Ensuite, les produits sont groupés suivant les similarités des codes, formant ainsi des familles de produits (voir la figure 3.1 [Strateg]).

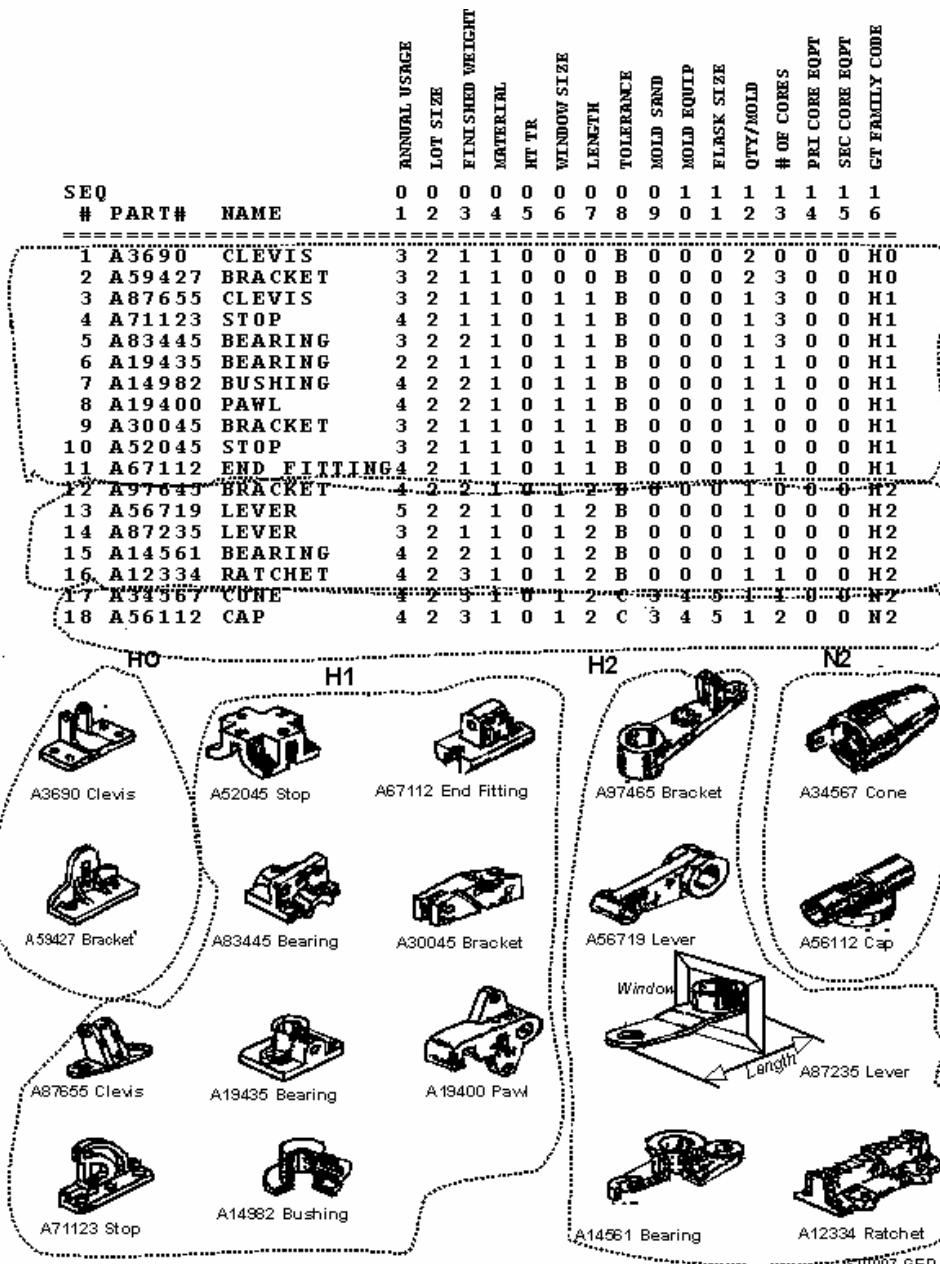


Figure 3.1

Codage et classification.

L'un des pionniers de cette méthode est le professeur Opitz. Dans les années soixante, il a proposé un système de codification des produits [Opitz1970] pour permettre à la TG d'atteindre un stade de perfectionnement avancé.

A cette époque, l'inconvénient principal de la classification, vu la lenteur des procédés d'entrée/sortie des calculateurs disponibles, était dû au fait qu'elle demandait un effort considérable pour pouvoir codifier tous les produits lorsque leur nombre est considérable.

La troisième méthode, Production Flow Analysis (PFA), a été développée par Burbidge [Burbidge1963]. La nouveauté principale de cette méthode est l'incorporation des informations liées à la fabrication du produit dans sa codification. Ainsi, en utilisant les séquences de routage, 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 mise en œuvre de la méthode PFA passait par quatre étapes :

- 1) collecter les données nécessaires tel que le numéro de produit, le code machine, la taille du lot et le taux de production annuel.
- 2) indiquer les routages des produits (séquences de machines qu'un produit visite pendant sa fabrication).
- 3) représenter les produits et leurs routages dans une matrice d'incidence produit-machine (voir figure 3.2.a [Strateg]).
- 4) analyser la matrice ainsi obtenue pour regrouper les produits qui ont des routages similaires dans des familles et les machines dans des cellules (voir figure 3.2.b [Strateg]).

a) Avant		b) Après										
Production Flow Analysis		Production Flow Analysis										
PART#	DESCRIPTION	1001	10148	10210	10233	10211	10233	10148	10210	10233	10096	10214
61354	COVER BEARING			1								
70852	GEAR DRIVEN: 8P 56T RH	1		2	1	1						
52594	SPACER CPLG SHAFT											1
81357-T	IMPELLER	1		3								
50547-D	GLAND MJ 6" DI CLASS B					1						
70935	GEAR DRIVEN: 8P 26T	1		2	1	1						
51171	RETAINER BUSHING											1
81176	BODY VOLUTE		1									1
72246	ELBOW RELIEF VALVE		1									
50763	SPACER BEARING							1				
71972-8	ADAPTER INTAKE 6"		1		1		1					
62575	SHAFT SHIFT								1			
63160	SEAT-SPRING								2			
62966	GENERATOR TACH PULSE		2	1								
71928	HEAD PUMP							1				

Figure 3.2

La matrice d'incidence produit-machine avant et après traitement PFA.

L'application de la philosophie TG dans un atelier de fabrication définit un système de production dit *cellulaire* à la mesure où les machines sont regroupées en cellules dédiées, une par une, à réaliser aussi complètement que possible les opérations de fabrication d'une famille de produits. Les systèmes cellulaires de production (SCP) sont ainsi conçus pour tenter de combiner les avantages des systèmes linéaires et des systèmes non linéaires.

Jusqu'aux travaux de Burbidge [Burbidge1963], les méthodes de mise en œuvre de la philosophie TG étaient basées sur des techniques d'analyse de données. McAuley [McAuley1972] fut le premier à proposer une méthode d'optimisation pour la détermination des familles de produits et des cellules de machines. Il peut être, par conséquent, considéré comme le précurseur de l'approche moderne de conception des SCPs.

3.3 DEMARCHE MODERNES DE CONCEPTION DES SCPs

Le point en commun des approches modernes de conception des SCPs est l'adoption d'une démarche d'optimisation pour mettre en œuvre le principe TG. Cependant, la complexité engendrée par la multitude des paramètres à considérer dans la décision d'implanter de tels systèmes oriente les concepteurs à décomposer le problème en étapes. Parmi les approches courantes, on cite l'approche adoptant les quatre étapes suivantes (voir figure 3.4) [Souilah1994] :

- Le prétraitement de données.

- La composition de cellules de production.
- L'agencement *intracellulaire*.
- L'agencement *intercellulaire*.

3.3.1 Prétraitement des données

Après l'acquisition des données nécessaires de production (exemple : types de produits à fabriquer, leurs gammes de fabrication, leurs routages, leurs demandes, ...etc.) vient l'étape de prétraitement de ces données. Pendant cette étape, le problème multi-gammes initial (certains types de produits disposent de plusieurs gammes de fabrication) se transforme en un problème mono-gamme (attribuer la meilleure gamme à chaque produit). Puis, les routages sont déduits à partir de ces gammes.

Les principales activités de cette étape sont :

- L'allocation des produits aux ressources.
- La détermination des emplacements des stocks et cela à partir des routages relatifs à ces gammes et les discontinuités des flux.

3.3.2 Composition des cellules

La composition de cellules consiste à regrouper les ressources (machines) du système de production en cellules aussi indépendantes les unes des autres que possible, de façon à réduire le coût de production entre les cellules (voir figure 3.3 [Strateg]).

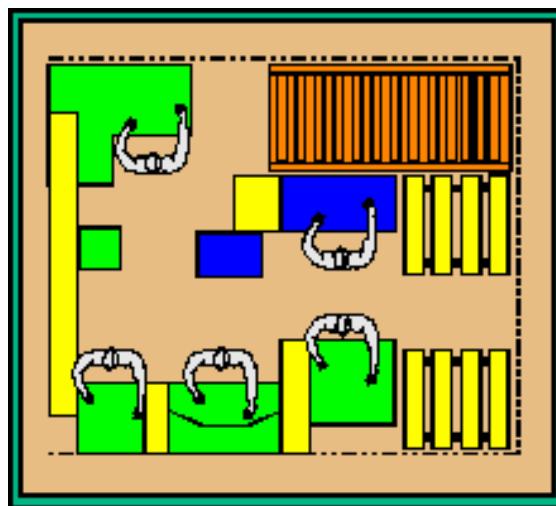


Figure 3.3 Une cellule de cinq opérateurs.

Cette partition de l'ensemble des ressources s'effectue de façon à optimiser plusieurs critères en tenant compte de diverses contraintes. Le but principal est de fabriquer chaque produit dans un nombre minimum de cellules.

L'étude de cette étape constitue l'objet de notre thèse et sera détaillée dans les deux dernières parties du manuscrit.

3.3.3 Agencement intracellulaire

L’agencement des ressources dans une cellule de production consiste à trouver la position de chaque ressource dans la cellule qui minimise le coût de fonctionnement de la cellule. Cette opération peut être réalisée en trois étapes :

- Sélectionner un système de manutention (SDM) à l’aide d’un système expert (SE).
- Déduire, par un deuxième SE, la configuration de base de la cellule à partir du SDM sélectionné.
- Affecter les ressources aux sites de la cellule en tenant compte du moyen de manutention et la configuration de base choisie.

Les contraintes à considérer sont :

- Le chevauchement entre les ressources.
- Le débordement des ressources sur les limites de la cellule.

3.3.4 Agencement intercellulaire

L’agencement des cellules sur le sol de l’atelier de production constitue la dernière étape du processus de conception des SCPs. Le problème consiste à installer les cellules de production, déjà constituées, sur le sol de l’atelier, en minimisant le coût de production du système de production (trafic et distance inter cellules).

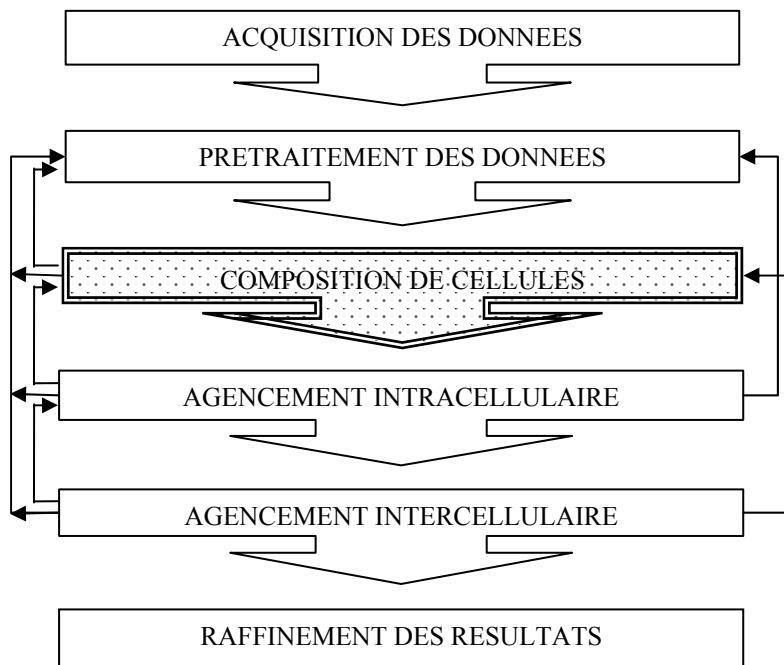


Figure 3.4

Approche à quatre étapes

3.4 AVANTAGES ET INCONVENIENTS DES SCPs

Nous avons vu au chapitre un que les tendances vont vers la recherche de systèmes qui réalisent un compromis entre la flexibilité et la productivité. Deux qualités qui sont en réalité difficiles à concilier. Il s'avère que ce sont les SCPs qui permettent d'atteindre ces objectifs. En effet, une panoplie de chercheurs a réalisé des études qui ont permis de tirer les avantages de tels systèmes moyennant des simulations, des études analytiques et des études de cas réels. Ces avantages peuvent être résumés dans ce qui suit [Béranger1987, Stone2000, Mungwattana2000] :

- 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 : Les temps de mise en marche réduits, il est réalisé une prise en charge plus rapide des produits à être traités. D'où, on obtient une réduction des tailles des stocks tampon.
- Réduction des en-cours : Les deux facteurs précédents entraînent automatiquement la réduction des *en-cours*.
- Les coût 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 aisément d'arrêter le processus de fabrication pour réparer le dysfonctionnement.
- Meilleur contrôle de la production : Dans les *jobshops* (systèmes non linéaires), les produits peuvent circuler dans l'atelier tout entier produisant des cheminement 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.

D'autre part, les inconvénients retenus sont :

- La complexité et la sophistication des machines mises en œuvre.
- La nécessité d'un personnel qualifié.
- Le coût élevé de la réalisation.

Fry et al. [Fry1987] 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 [Choi1992].

Levasseur et al. [Levasseur1995] ont étudié un cas d'implantation de SCP à Steward, Inc. (U.S.A). Les résultats étaient largement favorables à cette décision d'implantation. Chaque critère, dans cette étude de cas, a montré des améliorations surprenantes. Ces critères comprennent les en-cours, les temps de fabrication, les retards de livraison, les défauts de fabrication, le coût de la main d'œuvre et l'espace de fabrication utilisé. La Table 3.1 [Levasseur1995] résume les bénéfices tirés de l'implantation du SCP.

Table 3.1 Bénéfices des SCPs après les deux premiers mois de fonctionnement.

Critère	Job Shop	SCP	Gain
En-cours	\$ 590.000	\$ 116.336	\$ 473.664 (80%)
Produits finis	\$ 880.000	\$ 353.167	\$ 526.833 (60%)
Produits retournés	\$ 8.333 / mois	0	\$ 8.333 (100%)
Temps de fabrication	14 jours	2 jours	12 jours (86%)
Retard de livraison	100	4	96%
Défaux de fabrication	22%	14%	8%
Main d'œuvre	198	145	53 employés (27%)
Espace (pied carré)	45.000	20.000	25.000 (56%)

Hyer [Hyer1984] a réalisé une enquête dans 20 firmes américaines. Un questionnaire détaillé a été utilisé pour collecter des informations sur les coûts et les bénéfices des SCPs. Une large majorité des répondants a estimé que les bénéfices tirés de l'implantation des SCPs égalaient ou dépassaient leurs perspectives. Des économies ont été généralement réalisées dans la réduction des temps de fabrication, des temps d'attente en files, des temps de réglage des machines, des en-cours, des coûts de la main d'œuvre, des coûts de manutention et dans la plus grande facilité de préparation du plan de production.

Wemmerlöv et Hyer [Wemmerlöv1989] ont rapporté des gains d'applications de SCPs dans une étude réalisée sur 32 firmes U.S. Ces 32 firmes ont été choisies da 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 équipement hospitaliers et médicaux, les équipements militaires, les pièces détachés et les engins. La Table 3.2 [Wemmerlöv1989] montre les gains rapportés.

Table 3.2 Gains d'implantation de SCP dans 32 firmes US.

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'oeuvre	15	26.2	5.0	75.0
10. Augmentation de l'utilisation des machines dans les cellules	6	23.3	10.0	40.0

Wemmerlöv et Johnson [Wemmerlöv1997] ont conduit une étude similaire sur les avantages des SCPs recensés dans 46 autres 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, les engins. La Table 3.3 [Wemmerlöv1997] montre les performances recensées.

Table 3.3 Les Performances rapportés par l'étude de 46 firmes US.

Mesure 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

Les résultats présentés montrent l'efficacité d'application de la TG aux Etats-Unis après le Japon. Ils permettent de donner une explication à l'acharnement des dirigeants industriels des pays industrialisés pour la conversion de leurs ateliers en SCPs. Nous estimons que les pays en voie de développement doivent s'impliquer dans l'étude et l'application de tels systèmes afin de faire face à la nouvelle situation qu'engendre la mondialisation, avant qu'il ne soit trop tard.

CONCLUSION

DE LA PREMIERE PARTIE

Avec la mondialisation de la concurrence, investir dans le monde de production est devenu une opération à risques. Afin d'assurer toutes les chances de réussite, le recours aux nouvelles techniques de production est plus que nécessaire.

Le Juste-à-Temps (JAT) est parmi les techniques de production qui ont su pleinement tirer profit de l'outil informatique, à la mesure où le JAT présente une solution aux problèmes que présentent les systèmes traditionnels, en adoptant une vue globale et en usant pleinement des nouvelles technologies. La Technologie de Groupe (TG) est parmi les acteurs du JAT qui en ont assuré le succès. L'application du principe de la TG dans l'atelier de fabrication donne un Système Cellulaire de Production (SCP). La conception des SCPs passe par plusieurs étapes parmi lesquelles le module de composition de cellules constitue le point d'articulation. Cette thèse est consacrée à l'étude de ce module.

Le problème de composition de cellules, comme beaucoup de problèmes réels, est un problème multiobjectif étant donné que leur résolution doit tenir compte d'un certain ensemble de critères, souvent contradictoires. A cet effet, la partie suivante sera consacrée à l'étude des approches de modélisation de tels problèmes dits multicritères. En outre, les démarches empruntées pour leur résolution sont abordées, avec une attention particulière aux méthodes dites évolutionnaires.

Chapitre 4

LES FONDEMENTS THEORIQUES DE L'OPTIMISATION MULTICRITERE

Etant confronté à un problème qui consiste à choisir entre plusieurs alternatives ayant chacune un coût donné, un décideur doit trouver la meilleure solution permettant de minimiser ce coût. On parle dès lors de besoin d'optimisation.

Pour le problème que nous nous proposons de résoudre, le décideur veut trouver la meilleure composition de cellules à appliquer dans son atelier, afin de tirer le plus grand profit possible, tout en restant compétitif, dans un marché de plus en plus difficile et sans merci. Cependant, minimiser le coût global de production peut avoir plusieurs facettes. En effet, ce décideur pourrait songer à minimiser les délais de livraison de ses produits en minimisant les trajets des produits en cours de fabrication. Comme il pourrait aussi bien opter pour le maintien d'un haut degré de qualité de ses produits par l'utilisation de machines performantes, dans la mesure de la disponibilité, assurant des temps de fabrication optimaux. Le constat qu'on peut tirer de ce qui précède est, qu'en réalité, il n'y a pas en général un seul mais plusieurs critères que l'on cherche à optimiser. Dans ces circonstances, l'idéal est de trouver une solution qui est la meilleure vis-à-vis de tous les critères. Cependant, dans beaucoup de cas, une telle situation est utopique. En effet, pour notre problème, l'utilisation de machines sophistiquées réalise certes une diminution des délais mais avec des coûts de fonctionnement relativement significatifs, causés par la nécessité d'une main d'œuvre qualifiée à rémunération élevée et de procédés de maintenance onéreux.

C'est pour essayer d'avoir un modèle rigoureux pour l'étude de ce genre de situations que la théorie de l'optimisation multicritère ou multiobjectif a pris de l'intérêt. Pour avoir un aperçu de l'importance que portent les chercheurs à cette théorie et à ces différentes applications, on ne peut mieux que de noter la multitude de travaux de recherche publiés dans le domaine et dont le nombre est en constante augmentation.

Ce chapitre est consacré à l'exposé de la terminologie ainsi que des principaux résultats théoriques que la communauté des chercheurs a pu mettre en œuvre dans le domaine de l'optimisation multicritère. Cependant, vu l'immensité du sujet à couvrir, nous nous contenterons de présenter les points essentiels de cette théorie. A cet effet, la première section est dévouée aux définitions liées à l'optimisation multicritère. La section suivante donne une classification des approches multicritères. Enfin, la troisième section présente les principales modélisations associées.

4.1 PROBLEME MULTICRITERE

Cette section présente certaines définitions fondamentales de la théorie de l'optimisation multicritère. Pour plus de détails, nous invitons le lecteur à consulter [Collette2002, Barichard2003].

4.1.1 Définition d'un problème d'optimisation multicritère

Un *problème d'optimisation multicritère*¹ (POM) consiste en la recherche d'un vecteur de variables qui vérifie un système de contraintes et qui optimise les différentes composantes d'un vecteur de fonctions. De façon formelle, POM peut être représenté par le programme mathématique suivant :

$$POM = \begin{cases} \text{Minimiser} & \underline{F}(\underline{x}) = (f_1(\underline{x}), f_2(\underline{x}), \dots, f_m(\underline{x})) \\ \text{telle que} \\ \underline{G}(\underline{x}) = (g_1(\underline{x}), g_2(\underline{x}), \dots, g_k(\underline{x})) \leq \underline{0} \\ \underline{x} \in \underline{X}, \underline{F}(\underline{x}) \in \Re^m, \underline{G}(\underline{x}) \in \Re^k \end{cases}$$

où

- m est le nombre de fonctions objectifs ($m \geq 2$) ;
- $\underline{x} = (x_1, \dots, x_n)$ est le vecteur des variables dites *variables de décision* de dimension n ($n > 0$) ;
- $\underline{X} \subseteq \Re^n$;
- $\underline{G}(\underline{x})$ est un vecteur de k ($k \geq 0$) fonctions définissant les contraintes du problème représentées sous forme d'inégalités ;
- $\underline{0}$ est le vecteur nul de dimension k .
- \Re est l'ensemble des réels.

L'ensemble de tous les vecteurs \underline{x} , noté \underline{X} , est désigné par le terme *espace de décision*. Lorsque \underline{X} est fini ou dénombrable, on parle de *problème d'optimisation combinatoire multicritère*.

L'ensemble de tous les vecteurs \underline{x} vérifiant les contraintes du programme mathématique POM, noté \underline{X}_G , est dit *ensemble des solutions réalisables*.

L'image de \underline{X}_G par rapport au vecteur des fonctions objectif $\underline{F}(\cdot)$ est désigné par *l'espace des objectifs*.

Remarque 4.1 :

Bien qu'en général, les problèmes réels présentent des cas où les critères à optimiser consistent à minimiser certains critères tout en minimisant d'autres et les contraintes à satisfaire sont un mélange d'égalités et d'inégalités de différentes formes, il est aisément de se convertir à la forme présentée moyennant certaines transformations.

4.1.2 Optimalité au sens Pareto

Le fait qu'il soit rare de trouver une solution \underline{x}^* minimisant chaque fonction objectif $f_i(\underline{x}^*)$ pour $i=1..m$, rend la notion d'optimalité classique inadéquate. Les chercheurs ont dû alors recourir à d'autres définitions. Parmi celles-ci, la plus fameuse est une définition originale d'anciens travaux effectués à la fin du dix-neuvième siècle. En 1881, F.Y. Edgeworth [Edgeworth1881] fut le premier à introduire la notion d'optimalité en optimisation multiobjectif, généralisée par la suite par l'économiste V. Pareto [Pareto1896] en 1896. Bien que certains auteurs désignent cette notion par les termes *Edgeworth-Pareto optimalité*, c'est le terme *Pareto optimalité* qui est le plus utilisé [Aguirre2003]. Afin de mieux cerner ce concept, quelques notions, qui lui sont liées, sont définies dans ce qui suit :

¹ On utilisera aussi bien le terme multiobjectif que multicritère. Le premier étant plus courant chez la communauté des chercheurs anglo-saxons.

4.1.2.1 Dominance

Une solution \underline{x}_1 domine la solution \underline{x}_2 (on le notera $\underline{x}_1 \succ \underline{x}_2$) si la condition suivante est satisfaite :

$$\forall i \in [1..m] \quad f_i(\underline{x}_1) \leq f_i(\underline{x}_2) \text{ et } (\exists i \in [1..m] \text{ tel que } f_i(\underline{x}_1) < f_i(\underline{x}_2)).$$

Autrement dit, la solution \underline{x}_1 est au moins identique à \underline{x}_2 en regard à tous les critères et elle est meilleure que \underline{x}_2 au moins pour un critère.

Remarque 4.2 :

- Il est bien de noter ici que la dominance décrit une relation d'ordre *partiel* entre les points de l'espace de décision. En d'autres termes, certaines solutions sont incomparables par cette relation. Ceci constitue une source de problèmes pour les approches dites Pareto (voir [paragraphe 5.4.2](#)). En effet, si on prend comme exemple les solutions Pareto optimales, elles sont toutes incomparables. Si leur nombre est consistant, il n'est ni concevable ni utile de les proposer toutes au décideur. Par conséquent, il faut opérer, dans ce cas, une réduction qui doit être faite avec précaution pour ne pas éliminer celles qui l'auraient intéressé.
- Telle qu'elle est définie ici, la relation de dominance présente des insuffisances préjudiciables lorsque des problèmes avec contraintes sont traités. En effet, il se peut qu'on ait deux solutions a et b telles que $a \succ b$. Si, en plus, b est réalisable alors que a ne l'est pas, cette dernière ne doit pas être considérée meilleure. Par conséquent, cette définition doit être revue si on ne veut pas aboutir, au terme d'un procédé d'optimisation, définit autour du principe de dominance, sur des solutions non réalisables. Nous reviendrons à ce point dans le [paragraphe 8.3.1.3](#).

4.1.2.2 Définition formelle de l'optimalité Pareto

Une solution \underline{x}^* est dite *Pareto optimale* si pour toute autre solution \underline{x} , on a :

$$(\forall i \in [1..m]: f_i(\underline{x}^*) = f_i(\underline{x})) \text{ ou } (\exists i \in [1..m]: f_i(\underline{x}^*) < f_i(\underline{x})).$$

En d'autres termes, une solution *Pareto optimale* est une solution qui n'est dominée par aucune autre solution de \underline{X} .

L'ensemble des solutions Pareto optimales est noté P .

4.1.2.3 La frontière Pareto

La *frontière Pareto* ou *front Pareto*, notée PF , désigne l'image de l'ensemble des solutions de P dans l'espace des objectifs.

Remarque 4.3 :

L'utilisation des concepts Pareto dans les méthodes heuristiques peut mettre à confusion. En effet, ce genre de méthodes, comme pour l'optimisation classique, ne garantit pas la détermination exacte de P et PF , mais en donne des ensembles approchés correspondant à toutes les solutions non dominées rencontrées (ou une partie de celles-ci). A cet effet, nous utiliserons, dans le restant du manuscrit, le cas échéant, les notations de [Veldhuizen1999], à savoir P_{known} et PF_{known} pour les valeurs approchées de P et PF respectivement, et P_{true} et PF_{true} pour leurs valeurs exactes.

4.2 CLASSIFICATION DES PROCEDES DE RESOLUTION MULTICRITERES

La résolution des problèmes multicritères peut être réalisée suivant plusieurs procédés. Une manière de classifier ces différents derniers consiste à mettre en relief la relation qui lie les deux acteurs qui interviennent dans cette résolution : le décideur et le solveur. La coopération entre ces deux acteurs peut prendre l'une des trois formes suivantes [Talbi1999, Knowles2005] :

- Procédé de résolution à priori,
- Procédé de résolution à posteriori,
- Procédé de résolution interactif.

Dans le premier, le décideur est consulté avant d'établir un modèle pour exprimer ses préférences. Le solveur procède ensuite à la combinaison des différentes fonctions objectifs en une seule fonction à optimiser, dite *fonction d'utilité*. Cette combinaison utilise en général une pondération des fonctions objectif qui permet de donner un aperçu de l'importance relative de chaque critère, déduite des préférences exprimées par le décideur. La *méthode de la somme pondérée* est parmi les méthodes les plus utilisées dans cette catégorie. Cependant, sa simplicité de mise en œuvre, qui est la cause de sa popularité, cache un revers important : certaines solutions qui peuvent intéresser le décideur peuvent être aisément surclassées par d'autres et donc peuvent être omises par le procédé d'optimisation car jugées, à tort, non performantes. Une fonction d'utilité qui ne présente pas ces insuffisances est en général difficile à élaborer. En outre, il est difficile d'associer les poids adéquats aux critères, surtout lorsque ceux-ci sont non commensurables. Ces insuffisances et les remèdes associés seront plus détaillés au paragraphe suivant.

Dans une résolution *à posteriori*, l'optimisation est lancée en premier, générant plusieurs solutions constituant l'ensemble Pareto optimal. Le décideur choisit ensuite parmi ces solutions, celle qui lui convient le plus. Un problème se pose alors lorsque le nombre de solutions Pareto optimales est consistant : Le décideur n'est pas en mesure de revoir toutes les solutions possibles pour trouver celle qui lui convient le mieux. Afin de surmonter cette difficulté, on procède, en général, à la restriction du nombre de solutions Pareto optimales pendant le procédé d'optimisation, essayant ainsi de procurer au décideur un échantillon de solutions qui soit le plus dispersé que possible sur le front Pareto.

Dans une résolution *interactive*, il y a une coopération progressive entre le décideur et le solveur. A partir des connaissances acquises pendant la résolution du problème, le décideur définit des préférences. Ces préférences sont prises en compte par le solveur dans la résolution du problème. Ce processus est réitéré plusieurs fois (voir figure 4.1 [Talbi1999]).

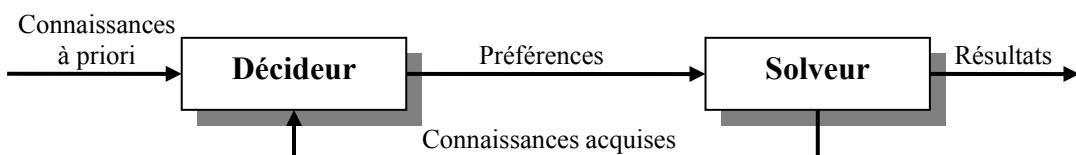


Figure 4.1 Procédé de résolution interactif.

Selon [Knowles2005], cette catégorie constitue la meilleure démarche comparée aux deux précédentes car elle permet de restreindre le nombre de solutions à présenter au décideur et limite la recherche effectuée par la méthode d'optimisation aux seules zones susceptibles de l'intéresser.

Très peu de travaux de recherche ont été effectués dans ce sens. Nous pensons que la cause de cette rareté est le fait que même s'il est possible de concevoir une approche interactive pour un problème spécifique, il sera difficile de généraliser le même procédé de résolution à l'ensemble des problèmes multicritères.

4.3 QUELQUES METHODES D'OPTIMISATION MULTICRITERE

Depuis plus de trois décades, les recherches effectuées dans le domaine de l'optimisation multicritère ont donné naissance à plusieurs méthodes très intéressantes. Cependant, l'exposé exhaustif de celles-ci rendrait la taille du manuscrit irréaliste. Dans ce qui suit, nous présentons une brève description de quelques unes des méthodes qui ont le plus marqué l'itinéraire des chercheurs dans le domaine. Pour un exposé détaillé, nous invitons le lecteur à consulter [Collette2002, Talbi1999, Ehrgott2000, Knowles2005].

4.3.1 Méthode de la somme pondérée

Cette méthode consiste à additionner toutes les fonctions objectif, en associant à chacun un poids indiquant son importance relative procurée par le décideur. Ceci transforme le problème multiobjectif en un problème simple objectif de la forme

$$\text{Min} \sum_{i=1}^m w_i f_i(\underline{x}),$$

avec $w_i > 0$ et $\sum_{i=1}^m w_i = 1$.

Le plus grand avantage de cette technique réside dans sa simplicité de mise en œuvre. Cependant, elle présente plusieurs inconvénients parmi lesquels nous avons [Berro2001] la non prise en compte des zones concaves de l'espace de recherche, la sensibilité à un changement d'échelle et la compensation que nous allons illustrer par les trois exemples suivants :

Exemple 1

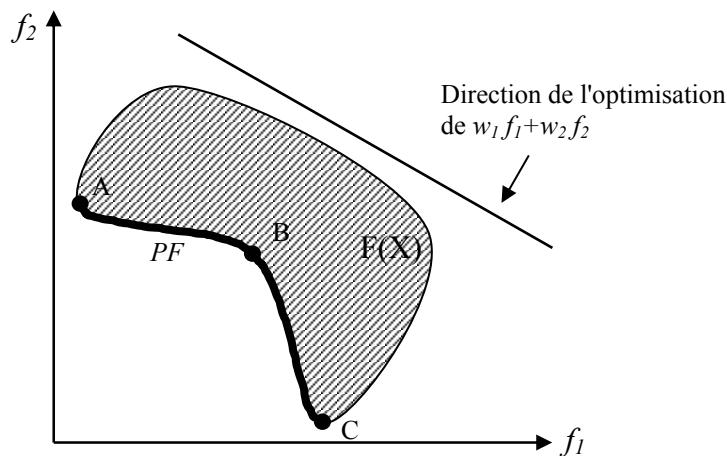


Figure 4.2 Non exploitation des zones concaves de PF.

Supposant que pour un problème bicritère minimisant deux fonctions f_1 et f_2 , le front Pareto soit représenté par la courbe ABC (voir fig. 4.2). Quelles que soient les valeurs données aux poids w_1 et w_2 , la méthode de la somme pondérée ne peut déterminer que les points A et C car le point B se trouve être dans une zone concave du front.

Remarque 4.4 :

De telles solutions qui appartiennent à des zones concaves du front Pareto sont appelées *solutions non supportées*. Celles des zones convexes sont dites *solutions supportées* [Ehrgott2000].

Exemple 2 [Maystre1994]

Considérons trois solutions *a*, *b* et *c* évaluées en fonction d'un critère "coût" de poids $w_1=4/5$ et d'un critère "quantité de déchets" de poids $w_2=1/5$ (voir table 4.1). Sachant que les déchets sont exprimés en tonnes dans la colonne de gauche et en kilogrammes dans celle de droite, on constate que ce changement d'échelle provoque une inversion des rangs des trois solutions.

Table 4.1 Effet d'un changement d'échelle

	Coût	Déchets	Somme		Rang
<i>a</i>	100 000	5 000	5 000 000	81 000	1 080 000
<i>b</i>	80 000	10 000	10 000 000	66 000	2 064 000
<i>c</i>	40 000	20 000	20 000 000	36 000	4 032 000

Exemple 3 [Berro2001]

Considérons une commission de sélection chargée de choisir entre deux candidats. Le premier a un total final de 77 points et le second de 76 (voir table 4.2, le coefficient des épreuves est indiqué entre parenthèses).

Table 4.2 Problème de compensation

Epreuve	A (3)	B (2)	C (1)	Total
Candidat N°1	14	14	7	77
Candidat N°2	13	13	11	76

La commission décide de prendre le premier vu qu'il possède la note la plus élevée. La décision aurait-elle été la même si la commission avait eu connaissance des notes des candidats ?

Les trois exemples illustrent donc trois problèmes qui sont souvent cités comme critiques envers la méthode de la somme pondérée. En guise de réponse à ces critiques, les chercheurs avancent pour le deuxième problème qu'il suffit de normaliser les critères de telle sorte à avoir une même échelle. Pour le troisième, il suffit d'instaurer un seuil à ne pas franchir pour chacun des critères. Enfin, le premier paraît être le plus problématique. Cependant, nous pouvons affirmer que, premièrement, le fait de penser que la méthode de la somme pondérée ne peut trouver des solutions non supportées (voir, par exemple, [Berro2001] p.37 [Barichard2003] p.15) est un propos peu exagéré et qui met à confusion. Il serait plus juste de dire que de telles solutions sont toujours surclassées par d'autres solutions supportées. En effet, le fait d'utiliser une méthode d'optimisation non exacte permet de donner des solutions non supportées, car elle ne garantie pas l'optimalité. Deuxièmement, en prenant le soin de mettre des contraintes sur les valeurs extrêmales des critères pour rendre les points associés non réalisables, on force la méthode à générer des solutions qui étaient à l'origine non supportées, même si c'est d'une méthode exacte qu'il s'agit. Une autre alternative consiste en l'hybridation de la méthode de la somme pondérée avec une autre. La première se charge de déterminer les solutions supportées, pour ensuite les utiliser comme points de démarrage de la deuxième en vue de longer les éventuelles zones concaves intéressantes du front.

Cet exposé présente donc autant de solutions qui poussent à constater que la technique de la somme pondérée n'est pas encore prête à tirer sa révérence. Constat que les publications récentes ne font que consolider.

4.3.2 Méthode ϵ -contrainte

Comme pour la précédente, la méthode ε -contrainte transforme le problème multicritère en monocrâtre. Cependant, au lieu d'agrégier tous les critères en un seul, le critère jugé le plus important est maintenu comme seul critère à optimiser et les autres sont transformées en contraintes (voir fig. 4.3).

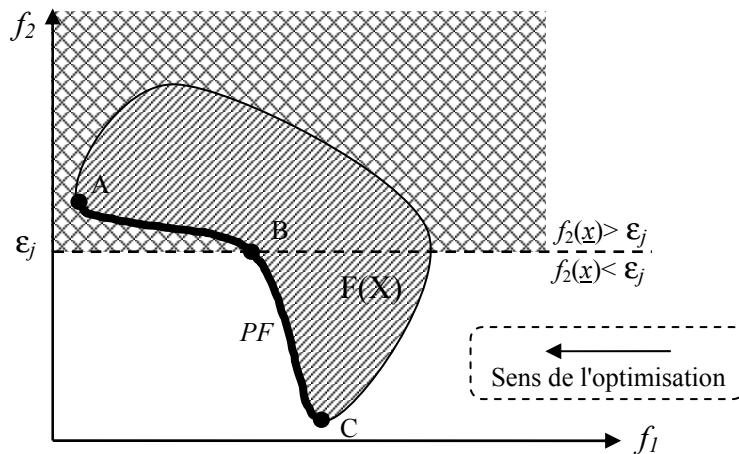


Figure 4.3 principe de la méthode ε -contrainte.

Cette procédure transforme donc le programme mathématique POM du [paragraphe 4.1.1](#) en un programme ayant la forme suivante :

$$P_\varepsilon = \begin{cases} \text{Minimiser} & f_i(\underline{x}) \\ \text{telle que} & \\ G(\underline{x}) = (g_1(\underline{x}), g_2(\underline{x}), \dots, g_k(\underline{x})) \leq \underline{0} \\ f_j(\underline{x}) - \varepsilon_j \leq 0 & j \in \{1, \dots, m\} \text{ et } j \neq i \\ \underline{x} \in X, G(\underline{x}) \in \mathbb{R}^k, \varepsilon_j \in \mathbb{R}. \end{cases}$$

où

- le critère mesuré par la fonction objectif $f_i, i \in \{1, \dots, m\}$ a été choisi pour être optimisé;
- ε_j est la plus grande valeur tolérée du critère associé à la fonction objectif $f_j(\underline{x}), j \in \{1, \dots, i-1, i+1, \dots, m\}$.

Notons enfin que la méthode pourrait être utilisée plusieurs fois en choisissant dans chacune un critère à optimiser différent. Le décideur sera, à la fin, en mesure de choisir la solution qui lui convient le mieux entre toutes les solutions trouvées.

Parmi les critiques retenues contre cette méthode, nous signalons le problème de détermination des intervalles appropriés pour les valeurs ε_j [Berro2001].

4.3.3 Méthode de programmation par but

Cette méthode remonte au début des années soixante [Charnes1961]. Elle consiste à déterminer pour chaque critère mesuré par la fonction objectif f_i un but à atteindre T_i . L'ensemble de ces valeurs définit dans l'espace des objectifs un point qui n'appartient pas en général à l'espace des solutions réalisables et constitue donc un point utopique. La programmation par but consiste donc à chercher un point S qui est le plus proche possible du point utopique. On peut avoir différentes méthodes en jouant soit sur le calcul de cette proximité en utilisant des métriques différentes, ou bien en instaurant une priorité entre les buts à atteindre [Anderson1999]. A titre

d'exemple, en optant pour l'utilisation de la distance euclidienne, le problème revient à résoudre le programme mathématique GP (pour Goal Programming) suivant :

$$GP = \begin{cases} \text{Minimiser} \left(\sum_{i=1}^m (f_i(\underline{x}) - T_i)^2 \right)^{1/2} \\ \text{telle que} \\ G(\underline{x}) = (g_1(\underline{x}), g_2(\underline{x}), \dots, g_k(\underline{x})) \leq \underline{0} \\ \underline{x} \in X, G(\underline{x}) \in \mathfrak{R}^k, T_i \in \mathfrak{R}. \end{cases}$$

Le principe de la méthode est illustré dans la figure 4.5.

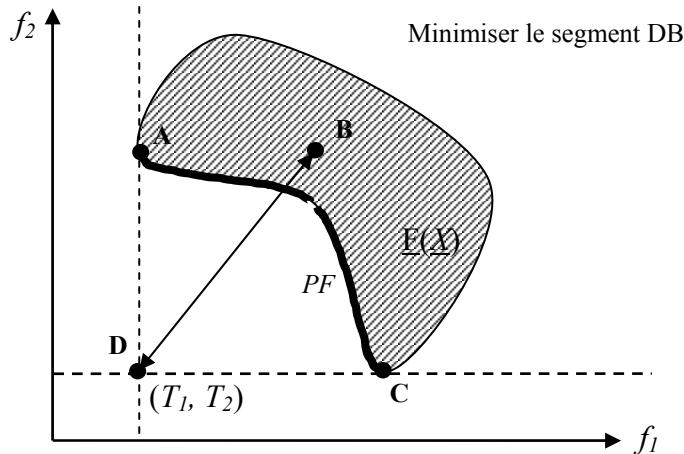


Figure 4.4 principe de la méthode de programmation par but.

Remarque 4.5 :

Les valeurs T_i des buts à atteindre sont, comme pour les valeurs ε_j de la méthode ε -contraintes, des paramètres réels. Cependant, il y a entre eux une grande différence. Une valeur ε_j , pour être utile, doit être au moins incluse dans l'intervalle des valeurs de la fonction objectif $f_j(\cdot)$. Par contre, il est préférable de fixer chaque valeur T_i pour qu'elle soit inférieure ou égale à la plus petite valeur que peut atteindre la fonction objectif $f_i(\cdot)$. Par conséquent, si le point représentatif du vecteur (T_1, \dots, T_m) peut être non réalisable, il est préférable que l'ensemble des points associés aux vecteurs de la forme $(\varepsilon_1, \dots, \varepsilon_{i-1}, f_i(\underline{x}), \varepsilon_{i+1}, \dots, \varepsilon_j)$ soit contenu dans l'ensemble des solutions réalisables.

La principale critique à l'encontre de la programmation par but est sa dépendance des buts à atteindre et dont la détermination n'est pas toujours une chose facile.

Après avoir identifié le caractère multicritère d'un problème donné et opté pour une approche de modélisation, il est alors question de résoudre le modèle par l'une des méthodes de résolution. Parmi ces méthodes, ce sont les approches évolutionnaires qui suscitent, le plus, l'intérêt des chercheurs. Le chapitre suivant présente celles qui sont les plus célèbres.

Chapitre 5

LES METHODES EVOLUTIONNAIRES MULTICRITERES

Parmi les méthodes de résolution des problèmes d'optimisation multicritères, les approches évolutionnaires suscitent de plus en plus l'attention des chercheurs. Ces méthodes s'articulent sur la technique connue sous l'appellation d'Algorithme Génétiques (AGs) qui s'inspirent du principe de *l'évolution naturelle* pour élaborer un procédé efficace de recherche de bonnes solutions par nuage de points et de façon pseudo-aléatoire.

Après une brève description des principes généraux des AGs, ce chapitre sera consacré à l'exposé des méthodes les plus citées dans les travaux de recherche.

5.1 PRINCIPES DE BASE DES ALGORITHMES GENETIQUES

Bien que des applications «génétiques» aient été publiés depuis les années 50 (voir, par exemple, [Fogel1963]), Holland, avec son article «*Adaptation in Natural and Artificial Systems*» [Holland1975], est considéré être le fondateur des AGs. Cependant, l'essor des AGs est dû à l'un de ses élèves, Goldberg, qui publia en 1989 un livre [Goldberg1989] surprenant qui scrute les entrailles des AGs et décrit leurs différentes applications, prouvant leur robustesse dans la résolution d'un vaste éventail de problèmes théoriques et pratiques. Dès lors, les AGs font l'objet d'intenses travaux de recherche pour lesquels des conférences et des revues scientifiques sont consacrées de façon exclusive.

Le principe des AGs est simple : on commence par définir une codification des solutions du problème en une structure dite chromosome qui peut être, par exemple, un tableau de nombres ou une chaîne de caractères. En utilisant cette codification, une population initiale de chromosomes est choisie aléatoirement ou moyennant une heuristique. Ensuite, on applique sur celle-ci des opérateurs génétiques pour construire une nouvelle génération. Pour ce faire, on commence par l'opérateur de sélection qui permet de sélectionner les individus les plus adaptés. Sur ces individus, on applique l'opérateur de croisement qui consiste à apparié les chromosomes sélectionnés deux par deux et de leur faire échanger une partie de leur code génétique. Le dernier opérateur, la mutation, consiste à choisir aléatoirement certains individus de la population et de modifier de façon aléatoire une partie de leur code génétique. L'ensemble des individus résultant de l'application de ces trois opérateurs remplace alors la totalité ou une partie de la population mère et le processus est renouvelé sur la nouvelle génération jusqu'à satisfaction d'un critère d'arrêt. Au long des générations, la population évolue donc selon le principe de la sélection naturelle ou de « la survie du meilleur ».

Pour mettre en œuvre l'algorithme génétique en vue de résoudre un problème donné, nous devons donc déterminer :

1. Une représentation de chaque solution en un chromosome (codage).
2. Une manière (aléatoire ou heuristique) de générer une population de solutions initiales.
3. Une fonction d'évaluation (adaptation ou fitness) qui exprime l'importance relative de chaque individu de la population.
4. Les opérateurs génétiques (sélection, croisement, mutation) qui déterminent la manière par le biais duquel on passe de la population courante à la nouvelle population.

La réputation des AGs s'est de plus en plus établie face aux autres méthodes "rivales" de la famille des métahéuristiques, telles que Recuit Simulé et Recherche Tabou, après l'engouement qu'a connu la recherche dans le domaine de l'optimisation multicritère car les AGs, qui sont des méthodes de recherche par nuage de points, étaient mieux positionnés pour s'adapter aux exigences d'une optimisation qui cherche à déterminer non pas une mais plusieurs solutions réalisant un compromis entre un ensemble de critères. A cet effet, des travaux intéressants ont été réalisés afin d'adapter les AGs à cette nouvelle besogne.

5.2 HISTORIQUE

Le premier travail faisant partie de la famille des méthodes évolutionnaires multicritères remonte au milieu des années quatre-vingt. Plus précisément en l'an 1984, Schaffer JD. [Schaffer1984] propose la méthode *Vector Evaluated Genetic Algorithm*, plus célèbre par ses initiales VEGA. Après la publication du travail de Schaffer, relativement peu de travaux ont été réalisés en optimisation évolutionnaire multiobjectif. Certains chercheurs justifient cette rareté par le fait que la communauté scientifique était dans ses premières tentatives d'extension des approches évolutionnaires monocritères au cas multicritère et par conséquent, les contributions ne s'aventuraient pas dans l'innovation afin de réutiliser la bibliothèque de code fiable disponible de l'optimisation monocritère [Coello2005]. D'autres avancent l'argument de l'outil informatique de l'époque qui ne permettait pas aux chercheurs d'exprimer leurs talents de programmation car contraints par la taille mémoire et la vitesse des processeurs qu'offraient les machines d'autan [Zydallys2003]. Nonobstant les raisons de cette hibernation, ce n'est que vers le milieu des années quatre-vingt-dix que le regain d'intérêt à l'optimisation multicritère s'est fait manifesté. Selon Coello [Coello2005], depuis cette date jusqu'à nos jours, les travaux publiés ont décrit deux générations : une première génération dont le précurseur fut Goldberg avec sa description de la méthode de sélection par *ranking* [Goldberg1989] qui utilise pour la première fois une sélection basée sur la notion de dominance Pareto. Les approches de cette génération étaient influencées par la mise en œuvre de l'opérateur de sélection en s'inspirant du *ranking* de Goldberg. Aussi, afin de surmonter le problème de convergence prématurée pouvant être causé par l'opérateur de sélection, le souci de maintien de la diversité était aussi abondamment abordé et la technique qui revenait le plus dans ces travaux était, sans conteste, celle désignée par le terme *sharing* (voir [le paragraphe 5.3.2.1](#)). Parmi les méthodes les plus référencées dans cette première génération nous avons :

- NPGA : Niched Pareto Genetic Algorithm [Horn1994];

- NSGA I: Nondominated Sorting Genetic Algorithm I [Srinivas1994].

Dans la deuxième génération, l'opérateur d'élitisme attirât l'attention des chercheurs, vu qu'il pouvait améliorer de façon drastique les approches évolutionnaires qui l'implémentaient. Aussi, pour le maintien de la diversité, la technique, connue sous le terme crowding (voir [le paragraphe 5.3.5.1](#)), fut proposée dans cette génération. Parmi les méthodes les plus référencées dans cette deuxième génération, nous avons :

- SPEA I : Strength Pareto Evolutionary Algorithm I [Zitzler1999b];
- NSGA II : Nondominated Sorting Genetic Algorithm II [Deb2000a];
- SPEA II : Strength Pareto Evolutionary Algorithm II [Zitzler2001].

Une brève description des techniques citées est présentée dans ce qui suit.

5.3 DESCRIPTION DE QUELQUES METHODES EVOLUTIONNAIRES

Cette section est consacrée à l'exposé concis de quelques méthodes évolutionnaires multicritère. Pour plus de détail, nous invitons le lecteur à consulter les références des auteurs associés. Nous commencerons par le travail qui est considéré être la première mise en œuvre d'une méthode évolutionnaire multicritère.

5.3.1 La méthode Vector Evaluated Genetic Algorithm

La méthode VEGA [Schaffer1984] réalise une adaptation de l'opérateur de sélection d'un AG simple pour traiter le cas multicritère. L'idée [Berro2001] est que si nous avons m objectifs et une population de k individus, une sélection de k/m individus est effectuée pour chaque objectif. Ainsi, m sous-populations sont, ensuite, mélangées afin d'obtenir une nouvelle population de taille k (voir fig. 5.1 [Berro2001]).

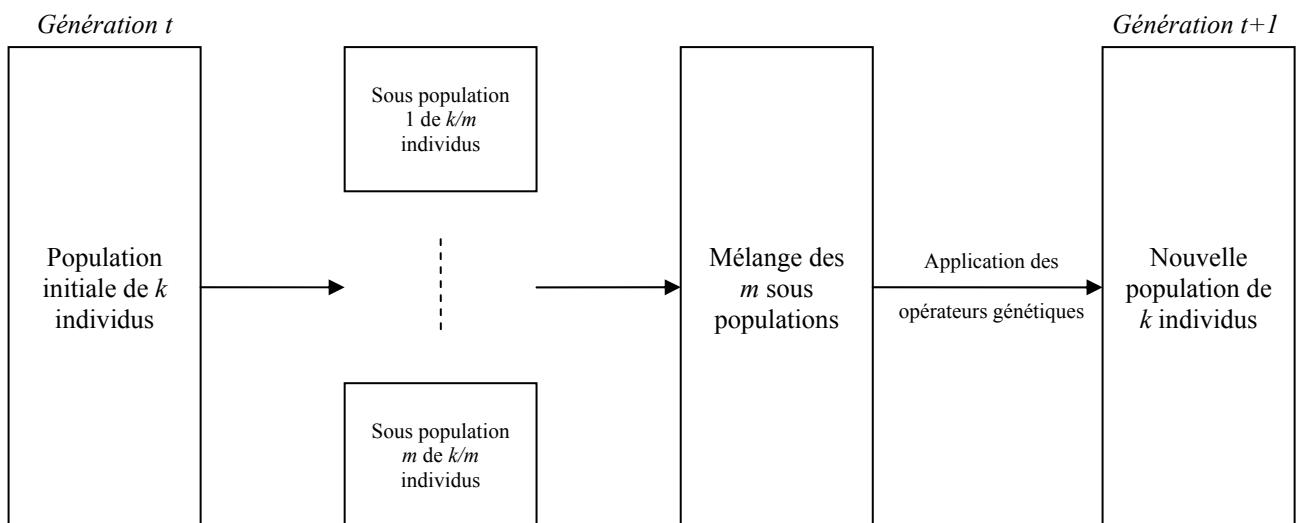


Figure 5.1

Principe de la méthode VEGA.

L'auteur de la méthode constate que VEGA tend à favoriser les solutions spécialisées. En effet, l'opérateur de sélection étant réalisé en tenant compte d'un et un seul critère à la fois, les solutions de compromis sont facilement perdues car elles ne sont bien adaptées par rapport à aucun critère, même si elles étaient Pareto optimales. Pour remédier à ce problème, Schaffer propose deux heuristiques :

- La première est un croisement restreint qui favorise la sélection des individus non dominés;

- La deuxième encourage le croisement entre individus spécialisés sur des objectifs différents.

5.3.2 Niched Pareto Genetic Algorithm

Cette technique est due à Horn, Nafpliotis et Goldberg [Horn1994]. Le principe de base de la méthode *Niched Pareto Genetic Algorithm* (NPGA) est le même que celui d'un AG simple (population unique et application répétitive et ordonnée des opérateurs de sélection, croisement et mutation), mais avec une redéfinition de la sélection pour qu'elle incorpore le principe Pareto.

NPGA utilise la technique de sélection par *tournoi*. En optimisation mono-objectif, cette technique consiste à choisir un sous-ensemble d'individus de la population. Le meilleur d'entre eux est ensuite sélectionné. Dans NPGA, deux individus candidats sont choisis de la population de façon aléatoire, en plus d'un sous-ensemble d'individus de comparaison S [Horn1997]. Si l'un des candidats est dominé par au moins un des individus de comparaison alors que l'autre non, ce dernier est sélectionné. Si les deux candidats ne sont pas dominés ou le sont en même temps, la procédure de *sharing* (voir ci-dessous) est utilisée. La pression de sélection, qui est un terme utilisé pour mesurer l'influence de l'opérateur de sélection sur la vitesse de convergence de l'AG, est contrôlée par le cardinal de S noté t_{dom} : une grande valeur de t_{dom} permet d'accélérer cette vitesse, alors que l'effet inverse est réalisé avec une petite valeur.

Le *sharing* est un opérateur de diversification proposé par Goldberg et al. [Goldberg1987] pour les problèmes d'optimisation multimodaux. Il consiste à dégrader l'adaptation d'une solution suivant la densité de solutions se trouvant dans son voisinage appelé niche. La dimension de la niche, qui est un paramètre très influent sur la performance du *sharing*, est définie par le paramètre σ_{share} qui peut être assimilé au rayon de la sphère définissant la niche (voir fig. 5.2).

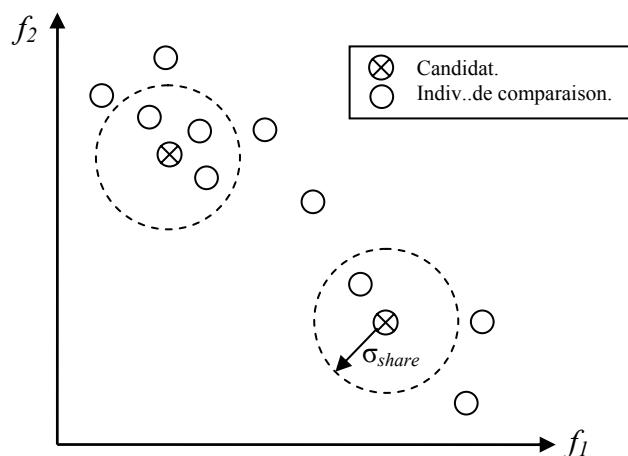


Figure 5.2 En appliquant un sharing, le candidat du bas l'emporte.

NPGA utilise le *sharing* pour choisir un candidat parmi les deux individus sélectionnés lorsque le tournoi donne un ex æquo. Le candidat ayant le moins d'individus dans sa niche appartenant au sous-ensemble de comparaison S l'emporte.

5.3.3 Nondominated Sorting Genetic Algorithm I

La méthode *Nondominated Sorting Genetic Algorithm I* (NSGA I) a été proposée par Srinivas et Deb [Srinivas1994]. Son principe de base est analogue à un AG simple mais toujours avec redéfinition de l'opérateur de

sélection. NSGA fut la première à implémenter la procédure de *ranking* (les auteurs l'appellent *Nondominated Sorting*) pour déterminer l'adaptation de chaque individu.

Le *ranking* consiste à identifier les solutions non dominées de la population qui auront le rang 1 (ce sont les individus de PF_{known} ou premier front). Ensuite, ces individus sont ôtés de la population et les individus non dominés du restant se voient attribuer le rang 2, et ainsi de suite jusqu'à expiration des individus de la population. La figure 5.3 illustre un exemple d'affectation de rangs pour les solutions représentées par les points a, b, \dots, j . Le problème bi-objectif étant à minimiser, la procédure commence par affecter le rang 1 aux solutions a, b et c . Ensuite, on ôte ceux-ci de la population. Les éléments non dominés dans la population réduite seront donc d, e, f, g . Il auront, par conséquent, un rang 2, et ainsi de suite.

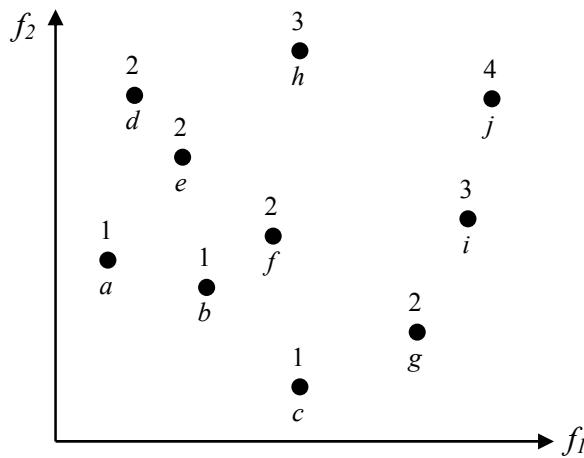


Figure 5.3 Procédure de *ranking*.

L'utilisation de cette procédure pour déterminer l'adaptation d'un individu avec une routine de *sharing* est implantée dans un AG simple utilisant une sélection par restes stochastiques [Goldberg1989]. Cette dernière consiste à combiner le déterminisme et l'aléatoire pour la sélection : par exemple, si l'adaptation d'un individu comparée à la somme de toutes les adaptations donne un rapport de 2,4 sur 100, la partie entière de cette proportion donne la nombre de copies de cet individu à mettre dans la génération suivante (pour une population de taille 100). Les restes des individus (i.e. 0,4 pour le nôtre), divisés par la somme des restes, définissent les probabilités d'insertion de chaque individu, pour le remplissage des places restantes de la population.

5.3.4 Strength Pareto Evolutionary Algorithm I

La technique *Strength Pareto Evolutionary Algorithm I* (SPEA I) est due à E. Zitzler [Zitzler1998]. Le principe général de la méthode est présenté dans la figure 5.4 [Zitzler1998]. SPEA I est, donc, une méthode évolutionnaire qui, contrairement aux méthodes évolutionnaires précédentes, implémente un opérateur d'élitisme. Pour ce faire, elle utilise une archive externe Q pour stocker les individus du premier front Pareto (PF_{known}). En outre, SPEA I utilise une nouvelle méthode pour calculer l'adaptation des individus : La première étape consiste à affecter à chaque individu i du premier front (l'archive Q) une valeur dite force $S(i)$ (du mot anglais *strength*) égale au nombre d'individus qu'il domine dans la population, divisé par la taille de la population. La force d'un individu de l'archive est égale à son adaptation. Ensuite, l'adaptation de chaque individu de la population est donnée par la somme des forces des individus de l'archive qui le dominent plus un. Ainsi, pour l'exemple de la figure 5.3, en

supposant que l'archive est composée par les individus $\{a,b,c\}$ et que le reste forme la population, les adaptations des individus sont : $S(a)=6/7$, $S(b)=4/7$, $S(c)=4/7$, $S(d)=13/7$, $S(e)=13/7$, $S(f)=17/7$, $S(g)=11/7$, $S(h)=21/7$, $S(i)=21/7$, $S(j)=21/7$. Cette définition de la fitness assure que les individus de l'archive possèdent une adaptation comprise entre 0 et 1, tandis que ceux de la population entre 1 et la taille de la population. Ceci afin de privilégier les individus du front lors de la reproduction. Mais en outre, les individus de la population sont classifiés selon le nombre et la force des individus de l'archive qui les dominent. Par conséquent, un individu de la population dominé par un seul individu du front Pareto a plus de chance de se reproduire qu'un autre dominé par tout le front.

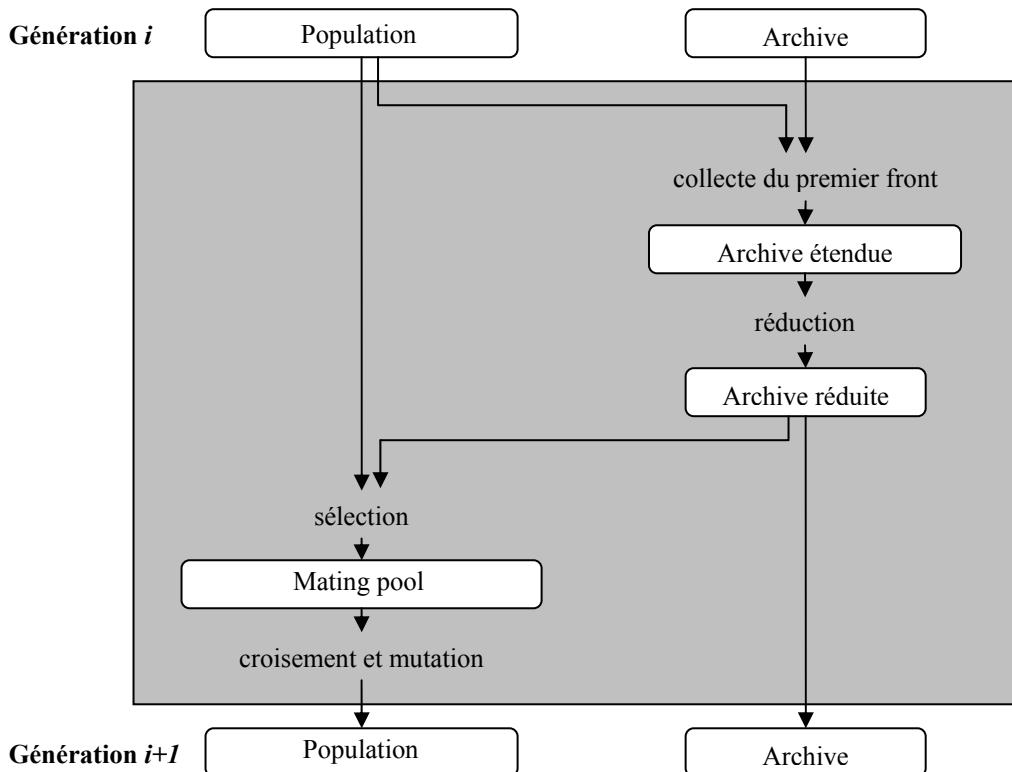


Figure 5.4 Principe de la méthode SPEA I.

Une autre particularité de SPEA I est l'utilisation d'une routine de réduction par groupage hiérarchique. En effet, bien que théoriquement aucune restriction n'est imposée quant à la dimension de l'archive, il est préférable de réduire sa taille afin d'avoir un gain en efficacité. Cependant, cette réduction doit se faire de façon à préserver un sous-ensemble d'individus qui représentent aussi fidèlement que possible le front Pareto PF_{known} . A cet effet, la méthode de groupage par lien moyen (average linkage clustering method) est utilisée. Elle consiste à considérer initialement chaque individu comme groupe. Ensuite, les deux groupes ayant le plus petit lien sont regroupés. Le lien de deux groupes est donné par la moyenne des distances séparant les paires d'individus des deux groupes. Ce procédé est répété jusqu'à l'obtention d'un nombre prédéterminé de groupes. Enfin, l'ensemble des individus centraux de chaque groupe constituent l'archive réduite. Un élément central est celui ayant la valeur minimale de la somme des distances par rapport aux autres éléments du groupe.

5.3.5 Non-dominated Sorting Genetic Algorithm II

Les analyses faites sur la première version de NSGA, par différents chercheurs y compris son auteur, ont montré qu'elle présentait trois points faibles [Deb2000a] :

- Le Temps de calcul prohibitif que nécessite la procédure de *ranking* qui est de l'ordre de $O(mk^3)$ avec m le nombre de critères et k la taille de la population;
- La non-utilisation de l'opérateur d'élitisme qui est le garant, s'il est bien défini, de la préservation des solutions Pareto optimales rencontrées durant le procédé d'optimisation;
- La nécessité de spécification du paramètre σ_{share} pour la procédure de *sharing*.

Ceci a amené Deb et al. [Deb2000a] à proposer une nouvelle version élitiste désignée NSGA II, réalisant le ranking avec un ordre de complexité $O(mk^2)$ et implantant un procédé de diversification sans paramètre, désigné *crowding* et dont l'étude comparative avec le *sharing* suscite toujours l'intérêt des chercheurs.

L'opérateur de diversification *crowding* se base sur le calcul de la *distance de crowding* qui est une estimation de la densité de solutions entourant une solution donnée [Deb2000a]. Pour une solution donnée c (voir fig. 5.5), cette distance est donnée par la taille du plus grand hypercube \mathbf{H} contenant c et ne contenant aucune autre solution de la population (\mathbf{H} est représenté par une ligne en pointillés). La taille de l'hypercube \mathbf{H} est estimée par la moyenne des longueurs de ses côtés. Aux solutions extrémale, a et e , est affectée une distance de crowding relativement grande pour favoriser leur sélection.

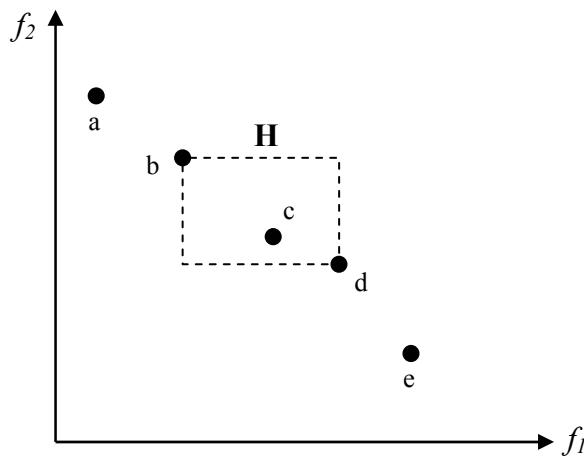


Figure 5.3 Principe du Crowding.

5.3.6 Strength Pareto Evolutionary Algorithm II

La deuxième version de la méthode SPEA fut présentée par son auteur en 2001 [Zitzler2001]. Les modifications apportées, dans cette version améliorée, visaient à pallier aux points faibles suivants:

- Les individus dominés par les mêmes membres de l'archive ont les mêmes valeurs d'adaptation. Par conséquent, dans le cas où l'archive contient un seul individu, les éléments de la population se voient attribuer un même rang, même si les uns dominent les autres. SPEA se comporte, dans ces conditions, comme un algorithme de recherche purement aléatoire.
- Si plusieurs individus de la génération courante sont incomparables (aucun ne domine l'autre), l'ordre partiel instauré par la relation de dominance ne peut être d'une grande utilité dans le procédé d'exploration. En effet, pour guider la recherche de manière efficace, nous avons besoin d'informations concernant la densité des solutions dans l'espace de recherche. La procédure de groupement qui utilise ce genre d'information est, malheureusement, restreinte dans SPEA I aux seuls individus de l'archive et ne concerne pas ceux de la population.

- La méthode de groupement peut facilement éliminer les individus extrêmes du front. Ceux-ci doivent être maintenus si on veut avoir dans l'archive un sous-ensemble de solutions bien réparties sur le front Pareto optimal.

Ce constat a amené les auteurs de SPEA à modifier le procédé de calcul de l'adaptation ainsi que le remplacement de la procédure de groupement par une nouvelle méthode de réduction.

Pour le premier, à savoir le calcul de l'adaptation, l'utilisation de la force d'un individu est étendue à l'ensemble des individus de la population. Ainsi, pour tous les individus i de SPEA II, i.e. la population Pop et l'archive Q , une adaptation brute $R(i)$ est donnée par la somme des forces des solutions qui la dominent. i.e.: $R(i) = \sum_{j \in Pop \cup Q, j > i} S(j)$.

Ensuite, à cette adaptation brute est ajoutée une estimation $D(i)$ de la densité de la zone contenant l'individu i (dans l'espace des objectifs). Cette densité est donnée par la formule : $D(i) = 1/(\sigma_i^k + 2)$, où σ_i^k est la distance qui sépare l'individu i du k -ième individu le plus proche, avec $k = \sqrt{|Pop| + |Q|}$. Enfin, la combinaison de l'adaptation brute avec la densité donne l'adaptation $F(i)$ de l'individu i : $F(i) = R(i) + D(i)$.

Si la modification de la définition de l'adaptation a permis de palier aux deux premiers points faibles de SPEA I, le troisième point a requis l'utilisation d'un nouveau procédé de réduction. Celui-ci consiste à choisir un sous ensemble de représentants pour les individus du premiers front (PF_{known}) à mettre dans l'archive Q , au cas où leur nombre est supérieur à la taille de Q . En effet, parce que SPEA II utilise une archive Q de taille fixe, si le nombre d'individus de PF_{known} est supérieur à $|Q|$, une réduction s'impose. Pour ce faire, on choisit l'individu i de PF_{known} ayant la distance σ_i^1 la plus petite. C'est-à-dire, celui qui a la distance la plus petite à ces compagnons de PF_{known} . En cas d'ex æquo, on choisit celui qui a la valeur σ_i^2 la plus petite. S'il y a quand même ex aequo on passe à σ_i^3 , et ainsi de suite jusqu'à sélection d'un seul individu. Ce dernier est éliminé de PF_{known} . Ensuite on répète la procédure jusqu'à ce que le nombre d'individus restant corresponde exactement à la taille de Q . Ils constitueront alors l'archive. Dans le cas où initialement le nombre d'individus de PF_{known} est inférieur à $|Q|$, ils sont tous insérés dans l'archive. Les $|PF_{known}| - |Q|$ places restantes sont remplies par les individus dominés les plus adaptés.

5.4 OBSERVATIONS SUR LES METHODES EVOLUTIONNAIRES MULTICRITERES

Les méthodes qui viennent d'être citées ont été (et sont toujours) intensivement référencées dans les travaux de recherche récents. Leur applicabilité a été montrée dans plusieurs domaines d'optimisation concrets et pratiques. Cependant, nous avons sur l'étendue de cette applicabilité les observations suivantes.

5.4.1 Prise en compte des contraintes

En optimisation multicritère évolutionnaire, la sélection est parmi les opérateurs les plus critiques. Ceci justifie le fait que la majorité des travaux de recherche dans le domaine consiste à proposer une nouvelle définition de cet opérateur. Rappelons que ce dernier permet de définir les individus qui seront, par la suite, les seuls aptes à avoir une progéniture, par croisement, dans le procédé de création de la nouvelle génération.

La sélection repose sur la fitness, ou adaptation, qui associe à chaque individu une valeur qui déterminera la probabilité de sa sélection. Les procédés de calcul de la fitness, dans les méthodes évolutionnaires monocritères,

étaient basés sur l'utilisation de l'unique fonction objectif du modèle. Pour adapter les AGs aux nouvelles exigences de l'optimisation multicritère, ceci devait être complètement revu lorsqu'il s'agit de tenir compte de plusieurs fonctions objectif. Pour définir cette adaptation, les procédés de sélection s'articulaient sur la relation de dominance (voir [paragraphe 4.1.2.1](#)).

En optimisation avec contraintes, cette méthode est inadéquate du moment qu'elle peut donner à une solution non réalisable une probabilité de sélection plus grande que celle d'une solution réalisable. Ce qui peut conduire à une convergence vers des solutions non réalisables. Dans [Deb2000b], l'auteur propose de modifier la relation de dominance en lui substituant la *dominance contrainte* (l'auteur l'appelle constrained dominance) définie par :

Une solution a domine par contrainte la solution b , si l'une des conditions suivantes est vraie :

1. a est réalisable et b ne l'est pas.
2. a et b sont non réalisables mais a possède moins de contraintes violées.
3. a et b sont réalisables et $a > b$.

Le problème avec cette définition réside dans le fait que deux solutions non réalisables avec le même nombre de contraintes violées sont incomparables.

Selon [Zydallis2003], si deux solutions sont non réalisables, il est difficile de déterminer celle qui doit être sélectionnée, dans le cas où elles ont le même nombre de contraintes violées et que les contraintes violées par ces deux solutions peuvent être différentes. Il préconise, en outre, que l'utilisation du degré de violation de chaque contrainte pour régler le conflit n'est pas évidente. En effet, si deux solutions a et b violent une certaine contrainte par 127,32 et 3,40 respectivement, le cas où a est meilleure que b n'est pas à écarter, car il se peut que la première soit à deux "pas" du domaine réalisable, alors que la deuxième se trouve à cent "pas" de celui-ci. L'auteur entend par "pas" le nombre de solutions qui séparent une solution non réalisable de la frontière du domaine réalisable décrit par la contrainte violée.

Nous estimons que le développement donné par l'auteur n'est pas aussi critique pour plusieurs raisons :

- Une violation d'une contrainte x ou y ne doit pas être vue d'un oeil différent, car elle se traduit par une impossibilité d'implémentation de la solution.
- Si cette notion de "pas" a une explication, quoique intuitive, dans les approches de résolution locale basées sur l'exploration du voisinage, le procédé d'exploration des méthodes évolutionnaires est tout à fait différent. En effet, l'application de l'opérateur principal de prospection qu'est le croisement conduit à des déplacements dans le l'espace de recherche trop complexes pour que la notion de pas puisse les cerner.

En partant de ce constat, nous proposerons, dans le [chapitre 8](#) (voir [paragraphe 8.3.1.3](#)), une petite modification qui permet de combler le "trou" dans la définition de la dominance contrainte.

5.4.2 Incomparabilité due à la relation de dominance

Hormis VEGA, les méthodes présentées sont toutes de nature Pareto. En recourant à la classification présentée au [paragraphe 4.2](#), ces méthodes appartiennent à la classe des approches à posteriori. Le point en commun de telles méthodes est de considérer les critères à pied d'égalité et de différer l'intervention du décideur pour ne faire appel à ses préférences qu'une fois le procédé d'optimisation mené à terme. En outre, ces méthodes se basent sur la relation de dominance qui présente la caractéristique de ne pas être une relation d'ordre total mais partiel. Nous avons

évoqué plus haut que les méthodes récentes proposées dans la littérature tendent à utiliser l'opérateur d'élitisme car il a été montré que les méthodes qui l'implémentent révèlent des performances accrues [Deb2000a]. Ce dernier consiste, essentiellement, à sauvegarder les solutions non dominées (i.e. PF_{known}) pour ne pas les perdre après application des trois principaux opérateurs (sélection, croisement et mutation) de l'AG. Cependant, les éléments de PF_{known} peuvent être très nombreux, ce qui oblige de recourir à une réduction. A ce niveau, la relation de dominance ne peut être d'une grande utilité, car elle perd toutes ses capacités de comparaison. C'est pourquoi on fait appel à un autre critère de comparaison basé sur le souci du degré de représentativité de PF_{known} par rapport à PF_{true} . Ainsi, si deux éléments sont candidats à être insérés dans PF_{known} alors que la limite de taille est atteinte par l'un d'eux, on choisit celui qui améliore la représentativité de PF_{known} , en convenant que cette dernière est d'autant améliorée que la diversité dans PF_{known} est élevée.

Cependant, étant confronté à un problème spécifique où on dispose de certaines informations sur les préférences entre critères, on se sent gêné avec cette attitude Pareto pure. En effet, dans la phase d'exploration, si on rencontre une "cohue" de solutions incomparables, on est conscient que la réduction, pour être efficace, il n'y a point mieux que d'utiliser de telles connaissances.

5.4.3 Evaluation des performances

Après avoir mis en œuvre des méthodes de résolution pour un problème donné, il reste à évaluer les performances de chacune des méthodes afin de déterminer celle qui donne les meilleurs résultats.

Cependant, si en optimisation monocritère (et par voie de conséquence, les méthodes multicritères transformant le problème vers le monocritère) la tâche de comparaison des résultats des méthodes est simple, en analyse multicritère, cette comparaison peut être même plus défiante que la conception de la méthode de résolution ! [Zydallis2003]. En effet, en monocritère, l'évaluation des performances s'articule sur la simple comparaison des valeurs de l'unique fonction objectif à optimiser obtenues par chacune des méthodes. Quelqu'un peut alors affirmer que le temps de calcul est un autre critère d'évaluation à ne pas négliger. Certes, ceci est vrai. Mais la réalité montre que ce dernier vient en général en seconde position car on a tendance à accepter que le critère de temps puisse être amélioré par la suite, en procédant par une analyse de complexité. Cette dernière permettra de déterminer les points de l'algorithme les plus gourmands en terme de temps de calcul afin de les redéfinir, ou bien de recourir aux techniques de parallélisation qui peuvent réduire le temps de calcul de façon drastique.

En optimisation multicritère adoptant une approche Pareto pure, le problème est complètement différent car nous avons, en général, plusieurs solutions par méthode et, de surcroît, chaque solution possède une évaluation multidimensionnelle.

Dans la littérature, pour réaliser une telle étude de performances, les premiers travaux de recherche procédaient par l'application des méthodes proposées sur des exemples de petite taille et/ou en considérant pas plus de deux objectifs. La raison est que, dans de telles situations, il est possible de déterminer la vraie frontière Pareto de façon exacte et/ou de représenter schématiquement les fronts obtenus par chaque méthode. Il suffira, ensuite, d'évaluer les performances par rapport à cette vraie frontière et/ou de comparer les résultats des méthodes de manière visuelle.

Etant donné que cette approche ne peut tenir compte du cas des problèmes difficiles de large taille et ayant plus de deux critères, les tendances actuelles vont vers l'investigation dans les méthodes de comparaison utilisant des métriques. Les métriques de comparaison ont été proposées pour mesurer la conformité des méthodes Pareto par

rapport à un ensemble de critères clés mesurant la qualité de l'ensemble PF_{known} trouvé. Parmi les critères, nous avons ceux décrit par Zitzler et al. [Zitzler1999a] :

- La distance de l'ensemble de solutions Pareto optimales obtenues (PF_{known}) du vrai front (PF_{true}) doit être minimale.
- La distribution des solutions trouvées sur PF_{true} doit être bonne (ce qui signifie, en général, uniforme).
- Les solutions Pareto optimales doivent couvrir, pour chaque objectif, un large spectre de valeurs.

Afin de tenir compte de ces critères, un large éventail de métriques a été présenté dans la littérature (voir [Knowles2002] pour une synthèse plus ou moins exhaustive). Certaines supposent que le vrai front PF_{true} est connu, d'autres non. Parmi les métriques utilisées dans la littérature, nous avons *l'erreur maximale*, le *ratio d'erreur*, *la distance générationnelle* [Veldhuizen1999] qui utilisent le référentiel PF_{true} ; *la métrique des 7-points* [Schott1995], *Hyperarea* [Zitzler1998], *la couverture relative* [Zitzler1999a], *Spacing* [Schott1995] qui ne supposent pas la connaissance du vrai front.

Les métriques serviront à juger lequel entre deux sous-ensembles de solutions A et B¹, issus de la résolution d'un problème multiobjectif par deux méthodes approchées, est meilleur que l'autre ? Les solutions étant évaluées par des vecteurs multidimensionnels, la tâche n'est pas du tout simple. En outre, en adoptant une approche Pareto pure, ce jugement doit s'articuler sur la notion de dominance. Ce qui a pour effet de créer un double problème. D'une part, il y a le fait que la relation de dominance n'est pas un ordre total, ce qui fait que si A et B contiennent des éléments incomparables, il sera difficile de juger qui est le meilleur. De l'autre, il y a le fait que A et B sont multidimensionnels, ce qui fait que si A contient des éléments dominés par des éléments de B et inversement, il en sera de même ardu de trancher entre eux.

Ceci a conduit certains chercheurs [Zitzler2002] à conclure que le problème des métriques est compliqué structurellement car la valeur donnée par une métrique, qui se réduit généralement à un scalaire, ne peut jamais contenir la même quantité d'information incluse dans un sous-ensemble PF_{known} . De ce fait, tout jugement qui s'articule sur une telle métrique peut nécessairement conduire à des verdicts erronés [Zydallis2003, Knowles2005].

Pour toutes les raisons qui viennent d'être citées, nous proposerons dans le restant du manuscrit un certain nombre de méthodes évolutionnaires. Si besoin s'exprime quant à l'utilisation des connaissances spécifiques du problème, y compris les préférences du décideur, nous n'hésiterons pas à enfreindre la "doctrine" d'une recherche purement Pareto. En effet, même en adoptant une approche Pareto pure, le front Pareto, ne va-t-il pas être présenté au décideur pour qu'il choisisse les solutions qui sont en concordance avec ses préférences ? Alors, autant exploiter ces connaissances dès le début.

¹ A et B sont les sous-ensembles PF_{known} associés aux deux méthodes approchées.

CONCLUSIONS

DE LA DEUXIEME PARTIE

Un bref survol a été réalisé sur la théorie de l'optimisation multicritère. L'essentiel des définitions y afférentes a été présenté. Dans un premier temps nous avons exposé les méthodes classiques qui consistent à transformer le problème multicritère en un problème monocritère. Ensuite, des approches de résolution non classiques ont été exposées. Le ton a été mis sur les approches évolutionnaires car ces dernières réalisent les meilleures performances, comparées aux autres méthodes de résolution.

Après avoir abordé, dans la première partie, le domaine d'étude associé à notre problème, d'un point de vue global, ensuite quelques notions liées aux approches de modélisation et de résolution des problèmes multicritères, la partie suivante sera consacrée à l'étude du problème de composition sous l'aspect statique. Cette étude débutera par l'exposé d'un modèle basé sur la théorie des graphes. Ensuite, une approche de résolution multicritère classique est adoptée dans laquelle des méthodes de résolution basées sur l'approche évolutionnaire sont proposées. Enfin, une relaxation de l'une des hypothèses (routages invariables) est apportée au modèle. Une méthode de résolution hybride est alors proposée pour sa résolution.

Chapitre 6

ETAT DE L'ART, POSITION DU PROBLEME ET FORMULATION

Ce chapitre réalise, dans une première partie, un survol sur les différentes approches utilisées pour traiter le problème de composition de cellules. La deuxième partie consiste à décrire le problème en mettant le ton sur les différents critères et contraintes utilisées dans la littérature. Ensuite, nous positionnons le problème et présentons les hypothèses de notre approche de résolution. Enfin, une formulation mathématique et proposée qui nous permettra de déduire la complexité du problème. Cette dernière sert à justifier l'approche de résolution évolutionnaire que nous adopterons dans le restant du manuscrit.

6.1 ETAT DE L'ART

6.1.1 Les précurseurs

Burbidge est le premier à avoir rendu célèbre la philosophie de la Technologie de Groupe ([voir chapitre 3](#)). Sa méthode PFA (Production Flow Analysis, [voir paragraphe 3.2.2](#)) [Burbidge1963] est la première à être largement diffusée dans l'industrie.

Malgré son succès, la méthode PFA n'était pas basée sur des fondements théoriques rigoureux (du moins dans sa première version). En effet, PFA repose sur l'expérience du décideur (celui chargé de la planification de la production) qui doit manipuler manuellement la matrice d'incidence produits-machines ([voir paragraphe 3.2.2](#)), armé de son expérience, afin de déterminer une bonne structure de blocs diagonaux. Le premier à avoir adopté une technique d'optimisation, pour résoudre le problème, est McAuley qui a proposé, en 1972, d'utiliser ce que l'auteur appelle des coefficients de similarité [McAuley1972]. En 1975, Purcheck [Purcheck1975] publie la première approche basée sur la programmation mathématique [AlBadawi2005]. McAuley puis Purcheck ont inspiré les chercheurs hardis, en leur prouvant, par leurs approches, qu'il était possible d'aborder le problème de façon rationnelle et rigoureuse. A partir de cette époque, deux différentes écoles se sont dessinées : 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 [Cantamessa1997]. Il faut noter que, comme le montre l'aveu des adeptes de l'école pragmatique, la majorité des travaux de recherche, qui ont été et qui sont publiés, est réalisée par des chercheurs qui appartiennent à l'*école optimale*.

6.1.2 L'essor

Dans une étude menée récemment [Mahadevan2003], l'auteur affirme que le nombre d'articles parus durant les trente dernières années, a atteint deux cents soixante dix neuf (279). L'étude a concerné dix prestigieux journaux¹ au sein desquels un nombre important de travaux de recherche dans le domaine est publié.

L'évolution du nombre de publications a connu une vitesse modérée à partir du milieu des années soixante-dix, pour connaître, à la fin des années quatre-vingts, une croissance spectaculaire (voir figure 6.1, [Mahadevan2003]). Il paraît que jusqu'à aujourd'hui, l'intérêt à la production cellulaire est en continue augmentation.

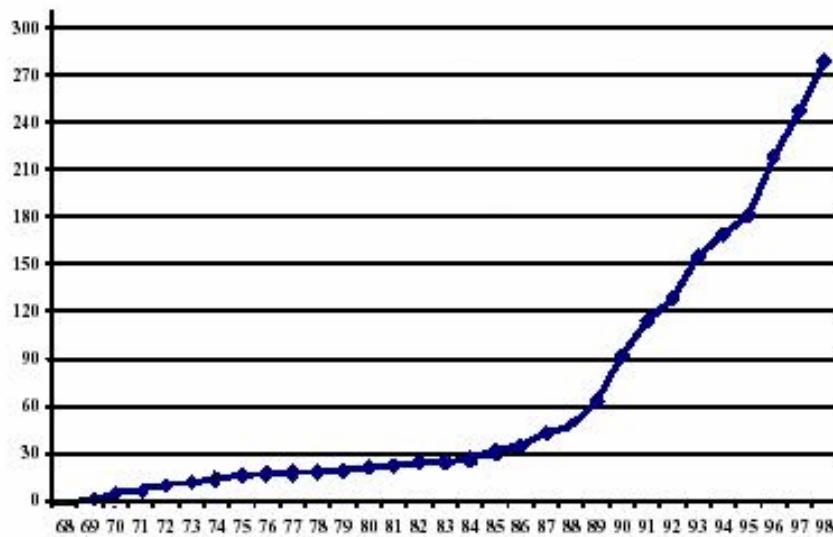


Figure 6.1 Evolution du nombre de publications dans les trente dernières années.

Les premières approches proposaient de concevoir des cellules totalement indépendantes. Ceci est l'idéal, mais le prix à payer est la nécessité de la duplication de certaines machines et/ou de la sous-traitance de certains produits. En effet, si un produit requiert une machine qui n'est pas dans sa cellule, soit on achète une autre, soit on la sous-traite. On cherchera, ensuite, un compromis entre le coût de duplication et celui de sous-traitance [Vanelli1986, Kusiak1987b, Frazier1991]. Le désavantage de cette approche est qu'elle nécessite un très grand investissement (à cause de la duplication), ou qu'elle expose la firme à des aléas qui ne peuvent être maîtrisés (à cause de la sous-traitance). Les applications industrielles de cette approche ont très vite montré leurs limites. Il était inconcevable de supposer une totale indépendance entre les cellules à cause de plusieurs facteurs tels que [Irani1993] :

- les pannes de machines,
- le besoin d'exploiter des machines devant être gardées allumées (car leur mise-en-marche est très coûteuse),
- les changements dans les routages et les volumes de production, ... etc.

Les systèmes modernes de manutention, ainsi que les nouvelles capacités de contrôle, ont rendu cette approche désuète.

¹ Les journaux concernés par l'étude sont : 1. *Comp. & Indus. Engng*, 2. *Comp. & Oper. Res.*, 3. *Decis. Sc.*, 4. *Eur. J. Oper. Res.*, 5. *IIE Trans.*, 6. *Inter. J. Prod. Res.*, 7. *J. of Operations Mngt*, 8. *Prod. Engineer*, 9. *Prod. & Inventory Mngt J.*, and 10. *Prod. Plng & Control*.

Une autre approche propose, pour résoudre le problème des machines requises par des produits de différentes familles, de regrouper ces machines dans une cellule supplémentaire. Cette dernière est placée au centre de l'atelier afin qu'elle puisse servir l'ensemble des familles avec un coût modéré [Murthy1995]. Les partisans de cette approche ne prennent pas en considération le trafic entre la cellule supplémentaire et le restant des cellules. Ce qui est une supposition non justifiée car le procédé d'optimisation aura comme effet de regrouper toutes les machines dans la cellule supplémentaire. Bien évidemment, les adeptes de cette approche procèdent par la limitation de la taille de cette cellule. Mais il reste que cette limite n'est pas simple à déterminer. D'ailleurs, cette approche n'a pas été beaucoup suivie.

La plupart des méthodes publiées appartiennent à la famille d'approches qui consiste à décomposer l'ensemble des produits et celui des ressources de façon biunivoque : on affecte chaque famille de produits à une seule cellule de machines et inversement. Cette démarche s'aligne parfaitement avec les perspectives de la philosophie TG qui cherche à développer chez les opérateurs de chaque cellule un esprit de responsabilité envers la famille de produits qui leur est dédiée. Ces opérateurs pourront développer un certain savoir-faire qui leur permettra de contrôler le procédé de fabrication de ladite famille dans le but d'atteindre l'objectif de qualité.

6.2 CLASSIFICATION DES METHODES EXISTANTES

Les méthodes existantes ont été classifiées de différentes manières. Une première manière, qui se veut exhaustive, met en relief un unique critère et considère donc une décomposition binaire, selon que le critère considéré est pris en compte ou non. Des exemples de telles classifications sont :

- Méthodes exactes et approchées,
- Méthodes directes et indirectes,
- Méthodes hiérarchiques et non-hiéarchiques,
- Méthodes monocritères et multicritères.

6.2.1 Classification en méthodes exactes et approchées

Cette classification met en relief le procédé de résolution adopté par les méthodes. Si celui-ci garantie l'optimalité alors il s'agit d'une méthode exacte, sinon la méthode est dite approchée ou heuristique. A cause de la complexité du problème de composition, la majeure partie des travaux de recherche effectués concerne la deuxième classe (méthodes approchées). Les méthodes exactes proposées dans la littérature [Steudel1987, Kusiak1987a, Chen1999, Boulfif2006a] sont, en général, jointes à des méthodes heuristiques.

6.2.2 Les méthodes directes et indirectes

Les méthodes directes déterminent les cellules de machines et les familles de produits de façon simultanée. La plupart de ces méthodes utilisent ce qu'on appelle la matrice d'incidence produits/machines ([voir figure 3.1](#)) et procèdent à des permutations des lignes et des colonnes afin d'obtenir une bonne structure de blocs diagonaux. A partir de ces blocs, les cellules de machines et les familles de produits sont déduites. Cette classe renferme un très grand éventail d'approches publiées dans les années quatre-vingt [King1980, Garcia1986]. D'autre part, les méthodes indirectes effectuent le groupement en deux étapes. Dans une première étape, le groupement se fait sur

l'ensemble des machines (resp. produits), puis sur celui des produits (resp. machines) [Seifoddini1989, Steudel1987, Ho1996, Sofianopoulou1997, Albadawi2005].

6.2.3 Les méthodes hiérarchiques et non-hiérarchiques

Les méthodes hiérarchiques, d'une part, commençant leurs procédés de résolution en considérant chaque machine comme étant une cellule et/ou chaque produit comme étant une famille [Seifoddini1989, Logendran1995, Verma1995]. D'autre part, les méthodes non hiérarchiques commencent avec une solution initiale générée aléatoirement ou obtenue par un procédé heuristique afin de démarrer avec une bonne solution [Gupta1995, Zhao2000].

6.2.4 Les méthodes monocritères et multicritères

Les premières approches au problème de composition étaient, en général, monocritère. Le critère considéré était soit la maximisation des similarités entre produits d'une même famille [McAuley1972, Kusiak1986] ou la minimisation des mouvements intercellulaires des produits [Klincewics1994, Joines1996, Cheng1996, Sofianopoulou1997]. Au fur et à mesure que la concurrence mondiale s'accentuait, le besoin en approches multicritères commençait à se manifester de plus en plus. Ce qui a donné naissance à plusieurs travaux de recherche pour lesquels on peut trouver un survol dans [Mansouri2000]. Parmi les approches multicritères, nous avons [Sankaran1990, Wei1990, Shafer1991, Shafer1992, Venugopal1992a, Dahel1993, Logendran1993, Min1993, Sankaran1993, Liang1995, Sureh1995, Akturk1996, Boctor1996, Ho1996, Rajamani1996, Lee1997, Logendran1997, Pierreval1998, Su1998, Chen1998, Wicks1999, Zhao2000].

Une autre manière de classifier consiste à décrire des caractéristiques remarquables communes à un ensemble de méthodes qui formeront ainsi une classe. Il est à noter que, dans ce cas, la classification n'est ni exhaustive, ni exclusive (voir figure 6.2). En adoptant une telle démarche, les classes suivantes peuvent être décelées [Boulif2006a] :

6.2.5 Les méthodes basées sur la matrice d'incidence produits/machines

Ces méthodes utilisent la matrice d'incidence produits/machines (MIPM) qui est une matrice booléenne introduite par Burbidge dans sa méthode PFA. Un élément quelconque m_{ij} de cette matrice indique si la machine j fait partie du routage du produit i . Parmi les méthodes appartenant à cette classe, nous avons [King1980, Garcia1986, Askin1991]. Le désavantage de ces méthodes est qu'elles ne prennent pas compte, en général, de la quantité des produits à usiner qui est un facteur très important pour toute approche pragmatique [Vakharia1990].

6.2.6 Les méthodes utilisant les coefficients de similarité

Une grande partie des travaux de recherche traitant du problème de composition fait partie de cette classe. Pour comprendre la raison, il est bien de se rappeler que les systèmes cellulaires cherchent à appliquer la philosophie TG qui consiste à tirer profit des similitudes entre produits pour avoir un bon compromis entre productivité et flexibilité. Par conséquent, ces méthodes "ne vont pas chercher ailleurs" : Elles définissent pour chaque couple de produits (ou machines) un coefficient mesurant leurs similarités qui constituera un paramètre décisif de la fonction objectif à optimiser. Depuis l'approche proposée par McAuley [McAuley1972], qui ne consistait qu'à utiliser la

fonction de Jaccard sur la matrice d'incidence produits/machines, les coefficients de similarité ont connu beaucoup d'amélioration afin d'introduire dans cette mesure des informations telles que quantité produite [Seifoddini1989], temps de fabrication et routages [Gupta1990]. Il est à noter que ces méthodes tendent ainsi à ressembler à des approches multicritères utilisant une fonction d'utilité multi-attribut car chaque information ajoutée met en valeur, en général, un critère distinct. Nous estimons qu'il serait plus intelligent d'aborder le problème directement de cet angle de vue afin de bénéficier des développements réalisés dans le domaine de l'optimisation multicritère. Un survol ainsi qu'une analyse des différents coefficients de similarité proposés dans la littérature peuvent être consultés dans [Yin2001]. Parmi les méthodes utilisant de tels coefficients, en plus de ceux déjà cités, nous trouvons [Al-Sultan1997, Jayakrishna1998, Wang2003, Albadawi2005, Jeon2006].

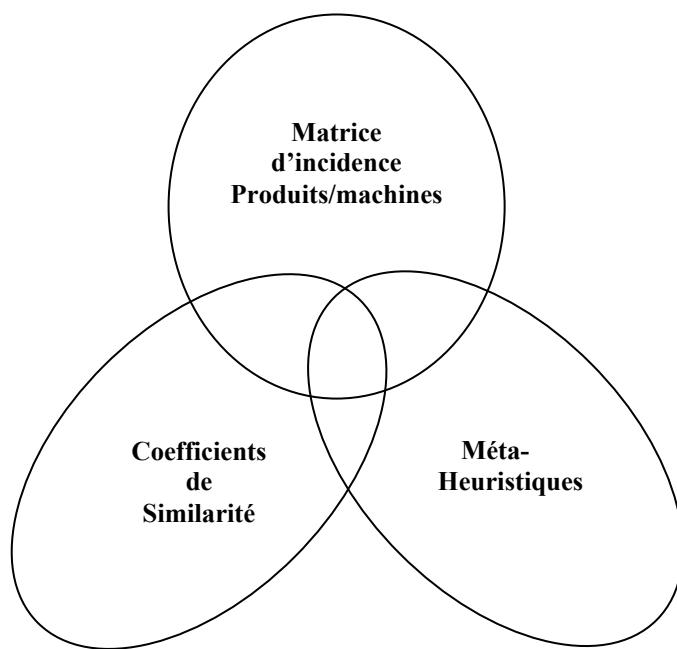


Figure 6.2 Classification non exclusive des méthodes existantes.

6.2.7 Les mét-heuristiques

Depuis l'identification de la NP-complétude du problème de composition, les travaux de recherche se sont dirigés vers les méthodes approchées ou heuristiques. Une partie de ces méthodes, dites mét-heuristiques, a attiré plus d'attention. Les mét-heuristiques sont des méthodes approchées proposées dans un contexte d'optimisation général, à la mesure qu'elles décrivent un procédé de résolution applicable à un large spectre de problèmes. Dans le domaine de la composition de cellules, nous avons l'application du Recuit simulé [Harhalakis1990, Venugopal1992b, Chen1995a], de la Recherche Tabou [Logendran1995], des Réseaux de Neurones [Kaparthi1992, Zolfaghari1997] et des Algorithmes Génétiques (AGs) [Gupta1995, Al-Sultan1997, Pierreval1998, Jeon2006]. La tendance actuelle dans cette classe concerne le développement de méthodes hybrides, eu égard des bonnes performances que cette hybridation peut procurer [Joines1997, Su1998, Gonçalves2002, Boulef2006a].

6.3 DESCRIPTION DU PROBLEME DE COMPOSITION

Dans cette section, nous décrivons le problème, de façon générale, tel que présenté dans la littérature. On insistera sur les critères à optimiser car notre approche au problème se veut multicritère.

6.3.1 De quoi il s'agit ?

Le problème de composition de cellules consiste à décomposer l'ensembles des ressources (en général ce sont des machines) d'un atelier de fabrication en unités de production dites cellules. Chaque cellule sera associée à une famille de produits présentant des caractéristiques similaires. Cette décomposition est réalisée dans le but général d'optimiser la compétitivité de la firme. En effet, la décomposition permet d'améliorer la qualité des produits car la responsabilité liée à l'objectif qualité dans les systèmes cellulaires n'incombe pas à une seule personne, comme dans les systèmes classiques, mais elle est décomposée de façon efficace. La responsabilité de la qualité d'une famille donnée retombe sur l'équipe des opérateurs affectés à la cellule associée. Chaque opérateur a la possibilité d'arrêter le processus de fabrication s'il décèle une quelconque anomalie. D'autre part, le fait de réaliser les produits similaires dans la même cellule permet de réduire le temps de fabrication car le temps de réglage des machines est diminué à cause du faible recours au changement d'outils. Ceci a pour conséquence l'amélioration de l'efficacité de la firme de façon drastique.

Dans la littérature, lorsqu'on envisage résoudre le problème de composition, on entend réaliser l'une des tâches suivantes, qui consistent à trouver :

1. une décomposition des machines en cellules, une décomposition des produits en familles et une affectation des familles aux cellules (simultanément ou par étapes) [King1980, Garcia1986, Sofianopoulou1997, Chen1999]. Si ces trois actions sont réalisées en même temps, on parle d'un problème de composition mixte.
2. une décomposition des produits en familles uniquement [Al-Sultan1997].
3. une décomposition des machines en cellules uniquement [Atmani1995, Pierreval1998, Mungwattana2000, Zhao2000, Vin2003, Boulif2006a].

Pour la première classe de méthodes de décomposition, lorsqu'il s'agit d'une approche mixte, c'est, en général, au détriment de la prise en compte de plusieurs paramètres permettant de définir le problème réel. En effet, il s'agit, en général, d'approches n'utilisant que les informations de la matrice booléenne MIPM. Les deux autres approches essayent de pallier aux insuffisances de la première en essayant de prendre en considération des paramètres décisifs tels que les séquences opératoires, les temps de fabrication, les volumes de production, . . . , etc.

Le choix entre les deux dernières approches de décomposition (ou pour la première, lorsqu'il s'agit d'une approche par étapes, le choix de la première action à entreprendre) est régi par l'aspect le plus important au regard du décideur ou du concepteur du système. En effet, comme nous allons voir par la suite (voir [paragraphe 6.3.3](#)), la conception est réalisée en tenant compte de plusieurs critères. Certains sont liés à la décomposition des machines, tandis que d'autres sont liés à celle des produits. Les critères les plus importants vont définir l'approche appropriée.

Dans certaines approches, on associe au problème de composition une ou plusieurs des tâches supplémentaires suivantes :

1. La détermination du nombre de machines devant réaliser les produits planifiés [Logendran1997, Su1998, Wicks1999, Mungwattana2000].
2. L'affectation des opérations aux machines disponibles [Atmani1995, Vin2003].
3. La disposition des machines à l'intérieur des cellules [Akturk1996].
4. L'affectation des machines à des sites sur l'atelier [Irani1993].
5. L'affectation des opérateurs aux cellules [Min1993].

L'approche à quatre étapes, que nous avons présentée au [chapitre 3](#), associe les deux premières tâches à l'étape de prétraitement de données. En effet, dans l'étape de composition, on suppose que le parc-machines disponible suffit pour réaliser la totalité du plan de fabrication (et donc la détermination et l'acquisition des machines nécessaires à la réalisation des produits ont été effectuées) et que les routages des produits sont connus (et donc l'affectation des opérations aux machines a été réalisée). La première supposition est justifiée par le fait que si le décideur connaît les quantités à fabriquer de chaque produit, il peut aisément déterminer le nombre minimum de machines à acquérir en considérant les capacités de celles-ci. Il est, en général, raisonnable d'opter pour cette valeur minimale, vu le coût élevé des machines et le problème de contournement intracellulaires engendré par la duplication de machines. La deuxième supposition est introduite pour des raisons de simplification en conformité avec l'approche adoptée qui traite le problème de flexibilité des routages dans le module de prétraitement de données. Cependant, pour des raisons que nous présenterons en temps opportun, cette supposition sera relaxée [au chapitre 8](#).

La troisième tâche est, en réalité, une action réalisée dans le module d'agencement intracellulaire. La prise en compte de cette tâche de façon simultanée avec la composition de cellules a été essayée par certains chercheurs [Akturk1996]. Vu la complexité du modèle généré par cette combinaison, des suppositions simplificatrices sont ajoutées, telles que :

- le nombre fixe de cellules qui est une supposition trop contraignante dans le module de composition ;
- l'omission des contraintes de la taille de l'atelier, l'omission de l'orientation des machines, ... , qui sont des paramètres décisifs quant à l'applicabilité de toute solution dans le module d'agencement.

Pour la quatrième tâche, elle peut être considérée comme une fusion des deux derniers modules de l'approche à quatre étapes que nous avons adoptée. Elle est en général réalisée dans les approches de *composition de cellules virtuelles*. Dans cette dernière, les cellules n'ont pas de sens physique, mais sont vues en tant que tel par le logiciel de contrôle du système de production [Irani1993]. Peu de travaux de recherche ont suivi une approche virtuelle. Nous estimons que la cause de cette rareté est liée au fait qu'une approche virtuelle ôte à l'organisation cellulaire l'une de ses inestimables vertus, qui est la cohésion des opérateurs et le travail d'équipe, garant de l'objectif qualité de la philosophie TG.

Enfin, la cinquième tâche a été citée car nous cherchions par notre exposé l'exhaustivité. En effet, il paraît qu'il n'y a qu'un seul article [Min1993] qui l'incorpore avec la composition de cellule. Le travail présenté par les auteurs est original à la mesure qu'il présente un modèle traitant du facteur humain et de son influence sur la composition de cellule. Cependant, il procède pour le résoudre par étapes : La première étape pour composer les cellules de

machines indépendamment des considérations liées au facteur humain. La deuxième pour la composition des équipes d'opérateurs et leur affectation aux cellules de machines déjà réalisées.

L'amélioration du travail présenté par les auteurs constitue un bon sujet à développer. Les défis à relever concernent, à notre sens, l'identification des paramètres influant du facteur humain tels que la satisfaction des opérateurs, les unions syndicales, les mœurs et la culture, ... etc. Autant de paramètres qui ne se prêtent pas facilement à la modélisation et dont la démonstration de la rigueur, par la suite, n'est pas moins difficile.

6.3.2 Les contraintes

Pour qu'une solution au problème de composition soit retenue, il faut qu'elle satisfasse certaines conditions dites contraintes. La plupart du temps, les contraintes sont soit imposées par des considérations pratiques, soit pour des raisons de simplification. Dans le premier cas, on est obligé d'en tenir compte si on ne veut pas que la résolution du problème n'aboutisse à une solution dont l'implantation dans l'atelier de fabrication relèverait de l'impossible. Dans le deuxième, il n'est guère conseillé de considérer ce genre de contrainte car, par excès de simplification, on peut aboutir à des modèles "de laboratoire", très éloignés des circonstances réelles et presque sans utilité dans la pratique. A ces deux types de contraintes, on peut ajouter deux autres, à savoir, les contraintes critères et les contraintes de cohérence.

Dans ce qui suit, nous présentons les différentes contraintes prises en compte dans la littérature en commençant par le premier type, à savoir les contraintes pratiques.

6.3.2.1 Contraintes pratiques

Parmi les contraintes imposées pour des considérations pratiques, nous avons :

a) La taille des familles : La taille d'une famille est donnée par le nombre de produits qui la composent. Il est préférable de modérer cette taille pour bien maîtriser l'objectif de qualité en équilibrant la charge de la responsabilité des produits entre les différentes équipes de l'atelier. Parmi les travaux ayant considéré ce genre de contraintes, nous trouvons [Moon1999]. Les auteurs citent comme prétexte la quête de simplicité du contrôle et de la planification dans la cellule. D'autres auteurs, tels que [Taboun1998, Wicks1999], imposent une borne inférieure à toute famille de produits créée.

b) La taille des cellules : La contrainte de la taille des cellules est de loin la contrainte la plus considérée dans la littérature. On entend par taille d'une cellule le nombre de machines que celle-ci peut abriter. Ce nombre doit être modéré (et donc borné) pour différentes raisons citées avec générosité dans la littérature. Par exemple, [Vakharia1990] argumente que l'espace disponible dans l'atelier peut restreindre le nombre de machines à mettre dans chaque cellule. En plus, la taille de cellule ne doit pas être trop large au risque de compromettre l'environnement sociologique dans la cellule ou de rendre difficile le contrôle visuel dans celle-ci. Les auteurs de [Rajagopalan1975] avancent d'autres considérations, à savoir, la possibilité de négligence des coûts de manutention intracellulaire et le renforcement des gains de la TG (réduction des en-cours, des temps de réglage des machines, ... etc.) car avec des tailles de cellules larges, ces gains peuvent disparaître. [Nagi1990] soutient que la restriction est liée aux capacités des robots manipulateurs et des tampons intracellulaires qui ne peuvent approvisionner qu'un nombre modéré de machines.

Parmi ceux qui ont opté pour ce type de contraintes nous trouvons [Rajagopalan1975, Askin1990, Harhalakis1990, Nagi1990, Boctor1991, Min1993, Harhalakis1994, Liang1995, Logendran1995, Murthy1995, Sofianopoulou1997, Chen1998, Su1998, Chen1999, Zhao2000].

c) Nombre de cellules : Il est préférable d'avoir un nombre modéré de cellules pour faciliter la gestion de l'atelier. Pour ce faire, deux bornes peuvent être utilisées : une borne inférieure et une autre supérieure. Parmi les travaux de recherche ayant opté pour ces contraintes, nous trouvons [Adil1996, Chen1999, Boulef2006a]. Notons l'interférence entre cette contrainte et celle de la taille des cellules. En effet, l'utilisation d'une borne supérieur 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. Il faut alors vérifier si ce nombre n'excède pas la borne supérieure du nombre de cellules, auquel cas l'espace de recherche ne contiendrait aucune solution réalisable. Par conséquent, si la contrainte de taille des cellules est considérée, on peut omettre la contrainte de la borne inférieure du nombre de cellules pour les raisons précédentes ainsi que celle de la borne supérieure car une telle borne existe toujours (nombre de machines).

d) Cohabitation entre machines : Cette classe de contraintes fait intervenir deux types de contraintes : les contraintes de cohabitation et les contraintes de non cohabitation. Dans le premier type, il est question de considérer le cas de machines qu'on veut mettre proches les unes des autres. La raison est que, par exemple, ces machines requièrent des sources d'énergie avec des installations onéreuses ou encombrantes [Boulef2006a]. Pour le deuxième type, il est question de vouloir éloigner certaines machines les unes des autres. Par exemple, écarter les machines de haute précision, des machines générant de grandes vibrations. Parmi les auteurs incluant dans leurs modèles cette classe de contraintes nous trouvons [Souilah1994, Chen1999, Boulef2006a].

e) Capacité des machines : Ce genre de contraintes assure que la charge des machines ne dépasse pas sa capacité. Une condition nécessaire pour que la prise en compte de cette contrainte soit plausible est la considération d'une certaine flexibilité des routages telle que la considération de plusieurs machines de même type, la possibilité de réalisation d'une opération sur différentes machines ou, de façon générale, la possibilité de réaliser chaque produit suivant plusieurs gammes opératoires. Cette contrainte est prise en compte dans [Askin1990, Nagi1990, Harhalakis1994, Logendran1994, Atmani1995, Logendran1995, Taboun1998, Wicks1999, Vin2003].

6.3.2.2 Contraintes de simplification

Il est erroné de croire que l'ajout de contraintes à un problème est toujours synonyme de difficulté accrue. La cause de cet état de pensée est peut être le surcoût de calcul engendré par la vérification de la réalisabilité des solutions dans les méthodes de résolution s'appuyant sur l'énumération totale ou partielle. Ceci fait que les logiciels commerciaux de résolution des problèmes formulés sous forme de programmes mathématiques s'attellent à évaluer l'efficacité de leurs produits par la somme du nombre de variables et de contraintes qu'ils peuvent prendre en charge. Il ne faut pas oublier qu'une contrainte engendre la réduction de l'espace de recherche, et que dans certains cas [Boulef2006a], il est possible de résoudre un problème NP-dur, à l'origine, de façon efficace, si des contraintes sont en plus considérées. Parmi les contraintes de simplification que nous avons rencontrées, nous présentons les suivantes.

a) Fixation de la taille des familles : Au lieu de borner la taille des familles, Al-Sulatan [Al-Sultan1997] considère des tailles fixes pour chaque famille. Les auteurs ne donnent aucune explication pratique pour leur

restriction. Certes, il semble que le souci est de restreindre la responsabilité des opérateurs de cellules à un nombre modéré de produits. Cependant, dans ce cas, la restriction par l'imposition d'une borne supérieure aurait suffit.

b) Fixation du nombre de cellules : Une bonne partie des travaux de recherche publiés (par exemple, [Atmani1995, Akturk1996, Su1998, Taboun1998, Zhao2000, Mungwattana2000, Mansouri2002]) suppose que le nombre de cellules (et par conséquent, le cas échéant, celui des familles) est fixé au préalable. Certains avancent que ceci est dû à des considérations pratiques telles que les recommandations du décideur qui veut fixer ce nombre pour simplifier la gestion de son atelier. D'autres [Mungwattana2000] ont eu le courage d'affirmer qu'il s'agit d'une supposition simplificatrice car elle permet de réduire l'espace de recherche de façon significative. Il est évident que les considérations pratiques avancées par les premiers peuvent être *perçues* par la détermination de deux bornes au nombre de cellules : une borne inférieure et une autre supérieure (voir [paragraphe 6.3.2.1.c](#), un peu plus haut). Parmi les auteurs qui adhèrent, avec force, à la maladresse de la fixation préalable du nombre de cellules, nous avons [Taboun1991, Liang1995, Sofianopoulou1997, Jayakrishna1998].

6.3.2.3 Contraintes critères

Nous avons vu, au [chapitre 4](#), qu'il était possible de tenir compte de certains critères, non pas en les incluant dans l'objectif, mais en leur associant des contraintes. Certains auteurs ont opté pour cette technique dans leurs approches multicritères. Nous avons recensé pour une telle démarche les applications suivantes :

- Les contraintes de budget,
- Les contraintes de seuil de similarité.

a) Contrainte de budget : En supposant que le budget est restreint, plusieurs travaux imposent à ce que le coût associé à un (ou plusieurs) critère(s) donné(s) ne dépasse pas l'enveloppe allouée. Une application d'une telle démarche considère la somme des coûts de duplication de machines avec celui de sous-traitance de produits dans [Logendran1995].

b) Seuil de similarité : Au lieu de considérer la similarité entre produits comme un critère à maximiser, les auteurs dans [Taboun1991, Taboun1998] considèrent pour chaque famille de produit un seuil à ne pas franchir pour que la solution associée soit acceptée.

6.3.2.4 Contraintes de cohérence

En plus des précédents types de contraintes, on peut trouver dans la littérature des contraintes que nous appelons *contraintes de cohérence*. Ces dernières sont utilisées dans les modèles mathématiques (le plus souvent dans les approches de programmation mathématique) et servent à préserver la cohérence des variables du modèle. A titre illustratif, si une variable indique l'affectation d'une machine à une cellule, alors on ne doit pas avoir de variables indiquant qu'une machine est affectée à plusieurs cellules en même temps. Le fait que ce genre de contraintes dépend du modèle utilisé et qu'il ne doit avoir aucune influence sur l'énoncé non formel du problème (car les restrictions associées sont sous-entendues), nous n'avons pas jugé utile d'exposer les différentes réalisations associées.

6.3.3 Les critères

Le choix de l'expression "optimiser la compétitivité" utilisée un peu plus haut n'était pas fortuite. Il fallait utiliser une expression aussi flexible que possible pour incorporer la pléthore de fonctions objectif présentes dans la littérature. En effet, comme pour tout problème réel, il ne s'agit pas d'un seul objectif à atteindre, mais de plusieurs. Pour être convaincu de cette réalité, jetons un coup d'œil sur les critères à optimiser proposés dans la littérature.

6.3.3.1 Survol des critères

Loin de prétendre l'exhaustivité, nous avons pu recensé les critères suivants sur plus d'une trentaine d'articles publiés, en plus d'une autre vingtaine étudiée par Mansouri [Mansouri2000]. Contrairement à l'étude de ce dernier qui est restreinte aux approches multicritères, notre survol concerne les critères associés aux approches monocritères et multicritères.

a) La minimisation des éléments spéciaux : Ce critère est calculé par la comptabilisation des 1 à l'extérieur des blocs diagonaux de MIPM. Ils sont appelés éléments spéciaux à la mesure où ils désignent des opérations associées à des produits fabriqués à l'extérieur de leurs cellules, ou de façon dual, aux machines réalisant des tâches de produits externes. Le cas idéal de zéro éléments spéciaux équivaut à des cellules totalement indépendantes. Parmi les chercheurs ayant opté pour ce critère, nous trouvons [Boctor1991, Murty1995, Adil1996].

b) La minimisation des mouvements intercellulaires : Ce critère cherche à minimiser les déplacements qu'effectuent les produits entre les cellules. Ce critère a été proposé pour pallier à une insuffisance liée au critère de minimisation des éléments spéciaux. En effet, ce dernier ne donne pas le nombre approprié de déplacements lorsqu'un produit réalise des opérations non consécutives sur une machine externe à sa cellule. Parmi les articles ayant utilisé ce critère, nous avons [Venugopal1992, Schafer1992, Dahel1993, Gupta1995, Logendran1995, Sureh1995, Verma1995, Sofianopoulou1997, Zhao2000, Gonçalves2002, Mansouri2002, AlBadawi2005].

c) La minimisation des mouvements intracellulaires : Ce critère est considéré par certains auteurs afin d'éviter les mouvements des produits en contre sens ou de contournement de machines à l'intérieur des cellules. Ce genre de mouvements induit des complications dans le système de manutention et un manque de sécurité (risque de collision). En général, ce critère est pris en considération avec celui de minimisation des mouvements intercellulaires, les deux combinés par une pondération favorisant ce dernier, dans une fonction objectif agrégative. Parmi les articles proposant une telle combinaison, nous avons [Logendran1991, Logendran1995, Gupta1995, Verma1995, Gupta1996, Zhao2000].

d) La minimisation du coût de duplication de (ou d'investissement en) machines : Dupliquer les machines permet d'éviter les déplacements intercellulaires, en mettant à la disposition des produits requérant une opération ne pouvant être réalisée à l'intérieur de leur cellule, une machine du même type que celle requise. Le prix à payer pour cette duplication est un coût supplémentaire d'investissement en nouvelles machines. Le critère en question sert justement à minimiser le coût de cet investissement. Parmi les travaux incluant ce critère, nous avons [Askin1990, Schafer1992, Logendran1994, Sureh1995, Logendran1997, Su1998, Wicks1999, Mansouri2002].

e) La minimisation du coût de sous-traitance de produits : Une autre alternative qui permet d'éviter les déplacements intercellulaires est la sous-traitance des produits requérant de tels déplacements. Cette sous-traitance a un coût et ce critère cherche à le minimiser. Notons qu'en général, ce critère est combiné avec celui de duplication

de machines afin d'en trouver un compromis. Parmi les approches qui ont opté pour ce critère, nous trouvons [Wei1990, Shafer1992, Logendran1997, Taboun1998, Mansouri2002].

f) La minimisation du trafic intercellulaire : Ce critère est proche du critère de minimisation des mouvements intercellulaires. Cependant, au lieu de considérer les flux entre machines à égalité, il tient en compte le volume de produits transportés ou leur coût. Parmi les travaux utilisant ce critère nous avons [Askin1990, Harhalakis1990, Nagi1990, Venugopal1992a, Harhalakis1994, Atmani1995, Pierreval1998, Su1998, Taboun1998, Chen1999, Wicks1999].

g) La maximisation de l'utilisation (ou la minimisation de la sous-utilisation) des machines à l'intérieur des cellules : Ce critère cherche à intensifier l'utilisation des machines de chaque cellule par les produits de la famille associée. La majorité des travaux ayant opté pour ce critère procèdent par la maximisation des 1 (ou la minimisation des 0) à l'intérieur des blocs diagonaux de la matrice d'incidence produits/machines MIPM. Ce critère est, en général, combiné de façon pondérée avec la minimisation des 1 à l'extérieur des blocs diagonaux (voir les critères : minimisation des éléments spéciaux et minimisation des mouvements intercellulaires). Cette combinaison donne une mesure de performance qui a été très utilisées dans les premiers travaux, connue sous l'appellation 'Efficacité de groupement' ou GE (pour Grouping Efficacy ou Efficiency) [Kumar1990]. Le critère en question a été choisi dans les articles [Chen1999, Mansouri2002, Vin2003]. D'autre part, parmi les articles qui ont utilisé la combinaison GE, nous avons [Chen1995b, Mukhopadhyay1995, Joines1997, Gonçalves2002].

h) La minimisation du coût de fabrication des produits : Ce critère entre en général en lice lorsqu'on considère la possibilité de réalisation d'une opération sur des machines de différents types, chacune avec un coût de fabrication différent. On cherche alors à affecter les produits aux machines disponibles de telle sorte à minimiser le coût global de production tout en respectant les contraintes de capacité. Parmi les articles utilisant ce critère : [Sankaran1993, Atmani1995, Ho1996, Rajamani1996].

i) La minimisation des écarts des charges intracellulaires : La charge d'une machine est donnée par la somme des temps de fabrication des produits sur ladite machine. Ce critère cherche donc à équilibrer les charges des machines afin que les opérateurs de cellules puissent avoir une charge de travail similaire, d'une part. De l'autre, pour assurer une certaine régularité des flux à l'intérieur des cellules afin de minimiser les en-cours [Venugopal1992a]. Parmi les auteurs ayant opté pour ce critère, nous avons [Wei1990, Venugopal1992a, Akturk1996, Gupta1996, Su1998, Zhao2000].

j) La minimisation des écarts des charges intercellulaires : Ce critère concerne les charges des cellules qui sont calculées par la somme des charges des machines qui les composent. On veut, donc, par ce critère avoir des cellules avec des charges de travail similaires car l'équilibrage de ces charges permet de réduire la taille des stocks tampons des composants intermédiaires [Akturk1996] et d'éviter plusieurs problèmes liés à la gestion de l'atelier [Mansouri2002]. Parmi les articles utilisant ce critère, nous trouvons [Wei1990, Akturk1996, Lee1997, Pierreval1998, Su1998, Mansouri2002].

k) La minimisation du coût d'évitement intracellulaire : Lorsqu'un produit n'utilise pas toutes les machines de la cellule à laquelle il est affecté, un coût d'évitement est généré par le besoin d'utiliser des moyens de manutention spéciaux, l'extension des délais de fabrication et éventuellement l'augmentation des en-cours [Verma1995]. Il est donc judicieux de minimiser ce coût pour se conformer avec les perspectives de la philosophie TG. Parmi les

travaux qui ont choisi ce critère, nous discernons [Verma1995, Akturk1996]. Pour tenir compte de ce critère, nous avons besoin d'informations sur la disposition des machines à l'intérieur des cellules.

Si les critères quantitatifs ont été apanagés par la majorité des chercheurs, certains ont essayé d'inclure dans leur modèle des critères qualitatifs. La particularité des critères qualitatifs est, peut être, la difficulté d'en donner une définition claire et qui ne peut être controversée. Parmi les critères considérés appartenant à cette catégorie, nous avons :

l) La maximisation des similarités (ou la minimisation des dissimilarités) entre produits de même famille :

Ce critère adhère à la l'esprit de la philosophie TG. Pour mesurer la similarité (recip. la dissimilitude) entre deux produits, on utilise, en général, une fonction qui calcule les similitudes (recip. Les dissimilarités), entre les vecteurs lignes associés, de la matrice d'incidence MIPM. Le pionnier de cette technique est McAuley [McAuley1972] qui a utilisé le coefficient de similarité de Jaccard (JSC pour Jaccard's Similarity Coefficient) défini par la formule suivante :

$$JSC = \frac{\text{Nombre de paires dont les composants sont égaux à un}}{\text{Nombre de paires dont un composant, au moins, est égal à un}}$$

Quatorze coefficients de similarités sont analysés dans [Anderberg1973]. Un survol plus récent est réalisé dans [Yin2001]. Les approches relativement récentes, ayant opté pour les coefficients de similarité, utilisent des formules plus élaborées incluant des informations concernant les volumes de production, les temps de fabrication et les séquences d'opérations. Parmi les travaux qui ont choisis ce critère, nous citons [Seifoddini1987, Gupta1990, Taboun1991, Vakharia1990, Liang1995, Al-Sultan1997, Jayakrishnan1998, Vin2003, AlBadawi2005]. Pour tenir compte de ce critère, nous avons besoin d'informations sur l'affectation des produits aux familles.

m) La maximisation de la flexibilité : Bien que nous avons essayé de donner une définition à la flexibilité dans la section 1.4, le passage suivant permet de se rendre compte de la difficulté de cette tâche : « Ten or 15 years ago, quality was much like flexibility is to day: vague and difficult to improve yet critical to competitiveness ... Flexibility is only beginning to be explored ... It means different things to different people » [Upton1995]. Nous invitons le lecteur, désireux de découvrir les différentes facettes de ce concept, à consulter [DeToni1998]. Ce qui nous intéresse dans cette partie c'est la manière avec laquelle ce critère a été pris en compte par les travaux de recherche effectués dans le domaine. Nous avons pu recensé deux applications :

La première [Liang1995] consiste à mesurer la flexibilité par le nombre de produits pouvant êtres directement fabriqué par le système cellulaire. En effet, les auteurs proposent une approche qui consiste à convertir un système *jobshop* en un système cellulaire. Certains produits du système d'origine, qui n'ont pu être assignés à des familles pendant le procédé d'optimisation, sont amenés à être sous-traités. Mais au lieu de minimiser le coût de sous-traitance, les auteurs procèdent à la maximisation du nombre de produits retenus. Ceci nous permet de tirer le constat que le critère ainsi défini est dual au critère de minimisation du coût de sous-traitance.

Dans la deuxième application [Vin2003], les auteurs partent du fait que si une machine tombe en panne, il est utile de pouvoir réaffecter sa charge sans pour autant produire un grand retard de production. Pour ce faire, les auteurs proposent de définir pour chaque machine un seuil d'utilisation inférieur à la capacité réelle de la machine en question (En toute honnêteté, une proposition similaire, non référencée par les auteurs, est donnée dans

[Vakharia1990], mais sans faire allusion à son utilisation pour mesurer la flexibilité). Ce seuil d'utilisation n'est pas imposé comme une contrainte, mais il est préférable de le respecter sinon une pénalisation est calculée mesurant le degré de violation du seuil d'utilisation par rapport à la capacité de la machine. Ainsi, à chaque solution est affecté un degré de conformité, avec ce souhait, égal à la moyenne des pénalisations associées à l'ensemble des machines. Il est évident que ce procédé ne peut être réalisé avec des approches classiques qui ne considèrent pas des routages alternatifs (la considération de la duplication de machine est un cas particulier de la prise en compte des routages alternatifs) car les utilisations des machines sont dans ce cas constantes.

n) La maximisation de l'efficacité de production : Dans [Liang1995], les auteurs affirment que l'efficacité d'un système de production est le résultat de la simplification des flux de production, de la réduction des tâches de manutention et des temps de fabrication. Par conséquent, ils jugent que la mesure de l'efficacité passe par l'évaluation de ces trois paramètres. Or, ils estiment que la détermination de ceux-ci dans l'étape de composition est difficile, voire impossible, car certaines informations qui y sont liées telles que les distances des trajets (dépendant de l'emplacement des cellules et des machines sur le sol de l'atelier) et les temps de fabrication (influencés par les décisions d'ordonnancement) sont inconnues. Les auteurs finissent alors par utiliser les coefficients de similarité en guise de mesure indirecte de l'efficacité.

6.3.3.2 Observations sur les critères

Après avoir survolé une bonne partie des différents critères proposés dans la littérature, quelqu'un serait tenté par l'élaboration d'un modèle incluant toute cette pléthore de critères. Cependant, une longue réflexion doit précéder l'adoption d'une telle démarche car la considération d'un nombre élevé de critères peut conduire à des difficultés dans l'étape de résolution, notamment lorsque celle-ci se conforme à l'esprit de l'optimisation Pareto pure. Ceci est dû au fait que la taille de PF_{known} et PF_{true} (voir [remarque 4.3](#)) deviennent trop grandes à la mesure que le nombre de critères croît [Fonseca1998, Deb2001]. Une importante première étape pour aborder des problèmes multiobjectifs est de ne considérer que deux critères [Venugopal1992a].

Si on se trouve amené à opérer une réduction du nombre d'objectifs, il est nécessaire que ça soit avec une démarche rigoureuse et pragmatique. Parmi les efforts que nous jugeons être intéressants dans ce contexte, il y a le travail réalisé dans [Purshouse2003]. L'auteur essaye d'apporter une approche formelle à ce que fait de façon non formelle tout chercheur confronté à un problème comportant un nombre consistant de critères. Il classe les relations pouvant lier les critères en trois : indépendance, harmonie et conflit (voir figure 6.3). A la lumière de ces relations, les critères sont analysés pour n'en garder en fin qu'un sous-ensemble restreint.

Deux critères sont en conflit si l'amélioration de l'un cause la dégradation de l'autre. Ils sont en harmonie si l'amélioration de l'un entraîne l'amélioration de l'autre. Enfin, l'indépendance décrit le fait qu'une amélioration sur l'un des critères n'a aucune influence sur l'autre. Cependant, l'auteur exploite ces différentes relations en se fiant à un genre d'échantillonnage, donné par la population de solutions que procure l'approche évolutionnaire qu'il adopte. Or, une telle démarche peut conduire à des déductions erronées, à la mesure où l'échantillon peut ne pas être représentatif de l'espace de recherche tout entier. En outre, le caractère conflictuel ou harmonique de deux critères donnés, peut dépendre des instances du problème : pour les unes ils sont en conflit, pour les autres ils sont en harmonie.

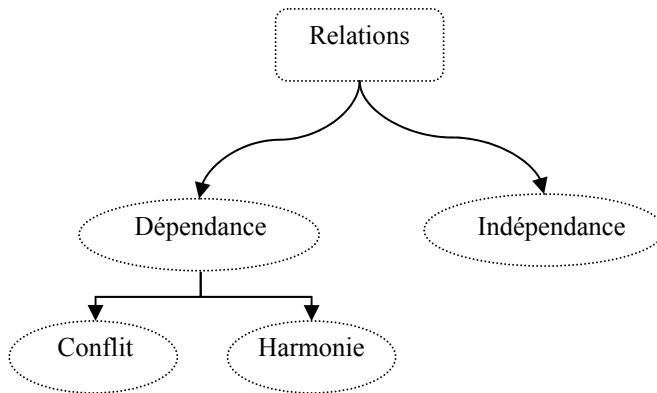


Figure 6.3 Relations entre critères [Purshouse2003]

Nous estimons qu'il est préférable d'essayer d'exploiter ces relations en se fiant à une représentation analytique des critères, les uns par rapport aux autres. Une illustration de cette proposition va être donnée dans ce qui suit, pour lier le critère de mouvements intercellulaires et intracellulaires. Ce qui permettra la suppression de l'un des critères. Cette représentation analytique n'étant pas toujours simple à élaborer, surtout lorsque les critères utilisent différents inputs, on peut recourir par la suite aux préférences entre les critères.

Une démarche de réduction est d'autant plus pragmatique que si elle parvient à détecter :

- Une redondance dans les critères.
- Une incohérence dans la prise en compte simultanée de certains critères.

Pour notre problème, le premier point se révèle dans les cas suivants :

- Le recours à une définition indirecte pour le critère d'efficacité l'a rendu redondant, du moment qu'il est devenu avec la définition précédente équivalent au critère de similarité.
- Le critère de minimisation des éléments spéciaux est redondant à la mesure qu'il est pris en considération par le critère des mouvements intercellulaires. Ce dernier est à son tour redondant, car il est pris en compte par le critère de trafic intercellulaire.

D'autre part, un cas de figure du deuxième point est donné par la considération simultanée des critères de minimisation des mouvements inter et intracellulaires. En effet, les mouvements des produits peuvent être déduits directement des routages. Du moment qu'un mouvement est soit intercellulaire soit intracellulaire, en considérant des routages fixes pendant la totalité de l'horizon de planification, la somme de ces mouvements est toujours constante. Certains chercheurs [Gupta1995] ont opté pour la combinaison de ces deux critères dans un modèle multicritère en utilisant la technique de la somme pondérée (voir [paragraphe 4.3.1](#)). On peut affirmer qu'en réalité, cette combinaison ne permet d'optimiser qu'un seul critère. En effet, la somme des mouvements inter et intracellulaires étant constante, en supposant f_1 et f_2 les fonctions objectif mesurant les mouvements inter et intracellulaires respectivement, nous aurons :

$$\begin{aligned}
 \text{Min } Z = w_1 f_1 + w_2 f_2 &\Leftrightarrow \text{Min } Z = w_1 f_1 + w_2 \cdot (c - f_1) && \text{ceci en posant } c = f_1 + f_2. \\
 &\Leftrightarrow \text{Min } Z = (w_1 - w_2) \cdot f_1 + w_2 \cdot c \\
 &\Leftrightarrow \text{Min } Z' = (w_1 - w_2) \cdot f_1 && \text{car le terme } w_2 \cdot c \text{ est constant.}
 \end{aligned}$$

Or, w_1 est pris supérieur à w_2 car il est communément admis que le critère de minimisation du trafic intercellulaire est, de loin, plus important que la minimisation du trafic intracellulaire. Par conséquent,

$$\text{Min } Z = w_1 f_1 + w_2 f_2 \Leftrightarrow \text{Min } Z'' = f_1$$

Ce développement permet donc de déduire que la minimisation de la somme pondérée de ces critères revient à minimiser l'un des critères uniquement. Notons que les articles [Gupta1995, Logendran1995] ne considèrent que les deux critères évoqués².

Pour argumenter le choix des critères que nous allons retenir dans notre étude, nous analysons les différentes combinaisons d'objectifs adoptées dans la littérature. Dans [Mansouri2000], le survol réalisé par l'auteur sur dix-neuf articles lui a permis de conclure que les critères les plus utilisés par les chercheurs sont : La minimisation du coût de duplication de machines, la minimisation du coût des trafics intercellulaires, la minimisation du nombre des mouvements intercellulaires et la minimisation des écarts de charge intracellulaires. Le survol que nous avons réalisé, ajoutant plus d'une trentaine d'articles aux travaux analysés dans [Mansouri2000], montre que parmi les trois critères cités, le critère de minimisation des trafics intercellulaires, qui est une généralisation du critère du nombre de mouvements intercellulaires, est de loin le plus populaire (voir la table 6.1).

L'analyse de ces critères nous permet de retenir les critères de minimisation des trafics intercellulaires, et la minimisation des écarts de charges intercellulaires pour les raisons suivantes :

- Le critère de minimisation des trafics intercellulaires étant le plus populaire, il est retenu comme objectif principal. Il est pris en considération dans 58 % des articles survolés. 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 liées à l'indépendance des cellules. Par conséquent, ils sont perdus si les produits passent la plupart de leur temps à se trimballer entre les cellules.
- La minimisation des écarts de charge est un critère qui se veut palliatif aux effets de bord de la minimisation des trafics sur l'équilibrage des charges de travail entre les cellules.

Cette réduction n'est pas trop contraignante pour les raisons suivantes :

- Les critères de minimisation des éléments spéciaux et des mouvements intercellulaires sont inutiles car le premier critère retenu en présente une généralisation palliant à leurs insuffisances.
- Le critère de minimisation des mouvements intracellulaires a été omis pour impossibilité de conciliation avec le critère principal.
- La duplication de machines n'est pas considérée car elle n'est pas toujours plausible à cause du coût élevé des machines. Ceci a pour effet de rendre négligeable l'apport de certains critères qui ont été justement ajoutés pour pallier aux effets de bord de la duplication, à savoir la minimisation des coûts de duplication, d'évitement et de sous-utilisation de machines. En effet, c'est la duplication qui engendre des problèmes de contournement de

² L'auteur du survol multicritère [Mansouri2000] a quand même inclus la référence [Gupta1995] dans son travail restreint aux approches multicritères !

machines à l'intérieur des cellules [Akturk1996]. C'est elle aussi qui est à l'origine de la sous-utilisation des machines engendrée par le surplus de capacité.

- Le critère de charge intracellulaire est considéré de façon implicite. En effet, dans les SCPs, les opérateurs de cellules sont, en général, amenés à assurer le fonctionnement de plusieurs machines dans leurs cellules [Min1993]. Certains chercheurs suppose même que les opérateurs ont été entraînés sur la totalité des machines du système [Askin1990], flexibilité oblige. Par conséquent, il est préférable d'équilibrer plutôt les charges intercellulaires, pour ensuite les départager entre les opérateurs. Ainsi, nous réaliserons l'équilibrage des charges inter et intracellulaires en un seul coup.
- Les critères restants n'ont pas été considérés car ils ne sont pas uniquement liés à la composition de machines. En effet, les coûts de sous-traitance et d'évitement ainsi que les similarités et la flexibilité version [Liang1995] (voir [paragraphe 6.3.3.1.m](#)) requièrent décider, en plus, de l'affectation des produits aux familles. Le coût de fabrication ainsi que l'autre version de la flexibilité [Vin2003] requièrent décider de l'affectation des opérations aux machines.

Table 6.1 Recensement des critères.

Les articles	<i>Les critères</i>												
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>
[Adil1996]	*												
[Akturk1996]		*							*	*	*		
[AlBadawi2005]													*
[Al-Sultan1997]													*
[Askin1990]				*			*						
[Atmani1995]							*				*		
[Boctor1991]	*												
[Cao2004]				*			*						
[Chen1998]				*			*						
[Chen1999]							*	*					
[ChenSJ1995]								*					
[Dahel1993]		*											
[Gonçalves2002]		*							*				
[Gupta1995]	*		*										
[Gupta1996]		*									*		
[Harhalakis1990]							*						
[Harhalakis1994]							*						
[Ho1996]									*				
[Irani1991]	*						*						
[Jayakrishnan1998]												*	
[Joines1997]								*					
[Lee1997]											*		
[Liang1995]												*	*
[Logendran1991]			*										
[Logendran1993]	*		*										
[Logendran1995]	*		*										
[Logendran1997]				*	*								
[Mansouri2002]	*			*				*			*		
[Min1993]													
[Mukhopadhyay1995]								*					
[Murthy1995]	*												
[Moon1999]				*					*				
[Nagi1990]							*						
[Pierreval1998]							*				*		
[Rajagopalan1975]													
[Rajamani1996]									*			*	
[Sankaran1993]									*				
[Schafer1992]	*						*						
[Seifoddini1987]												*	

[Sofianopoulou1997]	*												
[Su1998]		*											
[Sureh1995]	*		*										
[Taboun1991]											*		
[Taboun1998]				*	*								
[Vakharia1990]											*		
[Venugopal1992a]						*							
[Venugopal1992b]							*				*		
[Verma1995]	*	*									*		
[Vin2003]							*				*	*	
[Wei1990]				*							*	*	
[Wicks1999]			*			*							
[Zhao2000]	*	*									*		
[Zolfaghari1997]							*						
Pourcentage	05.77	25.00	11.54	17.31	07.70	26.92	15.38	09.62	13.46	11.54	03.85	17.31	03.85

Par conséquent, nous proposons dans ce qui suit un modèle bi-objectif, pour la composition de cellules de machines, tenant compte des objectifs et des contraintes pratiques précitées. Dans le restant de la thèse, ce modèle va être revu de façon graduelle afin de s'approcher beaucoup plus des circonstances réelles du monde de la production.

6.4 POSITION DU PROBLEME

Le but de cette section est de présenter de façon claire la version préliminaire du problème de composition que nous allons résoudre. Pour ce faire, nous présentons les hypothèses de résolution, les contraintes ainsi que le but à atteindre qui est bi-critère.

6.4.1 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 est constant (cette supposition sera relaxée [dans le chapitre 9](#)).
2. Les ressources (machines) nécessaires pour fabriquer ces produits sont connues.
3. Les machines sont disponibles pendant tout l'horizon de planification (cette supposition sera relaxée [dans le chapitre 9](#)).
4. Les machines disponibles permettent de réaliser la totalité de la commande en produits.
5. Chaque produit est fabriqué suivant un unique routage (cette supposition sera relaxée dans le [chapitre 8](#)).
6. Le système est soumis à un unique scénario Sc_i pendant tout l'horizon de planification (cette supposition sera relaxée [dans la quatrième partie](#)).

Le but étant de trouver une meilleure composition de cellules (partition de machines) vérifiant les contraintes suivantes.

6.4.2 Contraintes

Toute composition, pour être réalisable, doit respecter les contraintes pratiques suivantes :

1. La taille des cellules est bornée supérieurement.
2. Le nombre de cellules est borné supérieurement et inférieurement.

3. Certains couples de machines doivent être placées dans la même cellule.
4. Certains couples de machines doivent être séparés dans des cellules différentes.

6.4.3 Objectifs

La composition retenue doit minimiser le trafic et les écarts de charge intercellulaires du système de production.

6.4 LE MODELE

Rajagopalan [Rajagopalan1975] est le premier à avoir modélisé le problème de composition en utilisant un graphe dont les sommets sont les machines et les arcs relient les machines ayant un degré de similarité dépassant un seuil préétabli. Le modèle que nous proposons utilise le même concept graphique, mais avec quelque différence quant à la définition des arcs et de la fonction objectif.

6.4.1 Les inputs

(1) Considérons $M = \{M_1, M_2, \dots, M_m\}$ l'ensemble des m machines disponibles et $P = \{P_1, P_2, \dots, P_p\}$ l'ensemble des 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 produit P_k ($k = 1, 2, \dots, p$), on suppose donné :

(i) une séquence de machines, dite routage et noté R_k , à être visitées par le produit P_k :

$R_k = \langle M_{k,1} \rangle, \langle M_{k,2} \rangle, \dots, \langle M_{k,s_k} \rangle$, où: $\langle M_{k,t} \rangle \in M$ ($t = 1, 2, \dots, s_k$) et s_k est le nombre de machines dans le routage R_k ;

(ii) tf_{kt} , $t = 1, 2, \dots, s_k$: le temps de fabrication d'une unité de P_k par l'opération t du routage R_k .

(iii) r_k : le ratio de production du produit P_k par unité de temps.

(3) Pour chaque machine M_i , $i=1,\dots,m$, on note $e_{i,k,t}$ le prédicat indiquant si la t -ième opération du routage R_k est réalisé sur la machine M_i . C'est-à-dire : $e_{i,k,t}=1$ si $\langle M_{k,t} \rangle = M_i$, 0 sinon.

(4) On note la charge de la machine M_i égale à la somme des temps de fabrication des opérations réalisées sur M_i

$$\text{par : } ch_i = \sum_{k=1}^p \sum_{t=1}^{s_k} e_{i,k,t} \cdot r_k \cdot tf_{kt} .$$

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

(6) 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 :

$$t_{iq} = \sum_{k=1}^p r_k \cdot v_{k,i,q}$$

On définit alors 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 de machines reliées par un flux positif. C'est-à-dire : $E = \{e_{i,q} / (M_i, M_q) \in M \times M, i,q = 1, \dots, m ; i \neq q \text{ et } t_{iq} \neq 0\}$

(7) Soit $C = \{C_1, C_2, \dots, C_J\}$ une partition de M en J cellules. C'est à dire :

$$\bigcup_{j=1}^J C_j = M \text{ and } C_j \cap C_g = \emptyset \quad \forall j, g \in \{1, 2, \dots, J\}, \quad j \neq g.$$

(8) Une fonction de pondération des sommets, V , est définie pour indiquer la charge des machines : $V(M_i) = ch_i$.

(9) On définit la charge d'une cellule C_j par la somme des charges des machines qui la composent. C'est-à-dire :

$$Ch(C_j) = \sum_{M_i \in C_j} V(M_i).$$

(10) La variance des charges intercellulaires est donnée par : $CH_{out}(C) = \frac{1}{J} \sum_{j=1}^J (Ch(C_j) - \overline{Ch})^2$ avec

$$\overline{Ch} = \sum_{j=1}^J Ch(C_j) / J.$$

(11) On définit $E(C)$ comme étant le sous-ensemble des arêtes intercellulaires associés à la partition C :

$$E(C) = \{ e_{iq} \in E / (M_i, M_q) \in C_j \times C_g ; j, g = 1, 2, \dots, J ; j \neq g \text{ et } i, q = 1, 2, \dots, m \}.$$

(12) Une fonction de pondération des arêtes, W , est définie pour indiquer le coût du trafic entre les deux machines associées à ses extrémités : $W(e_{iq}) = t_{iq}$. où $i, q = 1, \dots, m$.

(13) Le coût du trafic intercellulaire de la partition C est donné par : $T(C) = \sum_{e_{iq} \in E(C)} W(e_{iq})$

6.4.2 Contraintes

Soit C une partition de M qui respecte les contraintes citées dans la description du problème (voir [paragraphe 6.4.2](#)). Ceci se traduit formellement par :

(1) Etant donné le nombre maximum de machines permis dans une cellule, N , il faut que :

$$\forall C_j \in C \quad (j = 1, 2, \dots, J), \quad \text{Card}(C_j) \leq N.$$

(2) Etant donné une borne supérieure (J_{max}) et inférieure (J_{min}) pour le nombre de cellules autorisé, J doit vérifier :

$$J_{min} \leq J \leq J_{max}.$$

(3) Etant donné un ensemble de couples de machines à cohabiter, E_{coh} , il faut que :

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

(4) Etant donné un ensemble de couples de machines à ne pas cohabiter, E_{ncoh} , il faut que:

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

Remarque 6.2 : Pour la convenance avec les approches de résolution que nous allons proposer, E est étendue à $(E \cup E_{coh} \cup E_{ncoh})$. Cette extension consiste à ajouter dans le graphe de flux des arêtes fictives entre les machines qui doivent ou ne doivent pas cohabiter lorsqu'elles ne sont pas reliées par un mouvement de produits. Ces arêtes sont alors caractérisées par des pondérations nulles.

6.4.3 Fonctions Objectif

Soit S l'ensemble des partitions qui respectent les contraintes précédentes. Le problème est de trouver une partition $C^* \in S$ de M telle que :

$$\underline{Z}(C^*) = \min_{C \in S} (T(C), CH_{out}(C))$$

C'est-à-dire qu'on cherche une partition des sommets du graphe de flux qui respecte toutes les contraintes et qui réalise le meilleur compromis entre le coût du trafic intracellulaire et la variance de charge intercellulaire.

6.5 COMPLEXITE DU MODELE

Si l'étude de la complexité en optimisation classique a été beaucoup étudiée, l'étude de la complexité des problèmes multiobjectifs n'est pas encore arrivée au stade de maturité. Toutefois, avant de se lancer dans l'étude de la complexité du modèle, il faut avant tout bien définir le sens de l'optimisation utilisée dans celui-ci.

En général, lorsqu'on cherche à optimiser un vecteur de fonctions objectif (dans le modèle proposé, le vecteur est défini par le couple de fonctions (T, CH_{out})), on entend optimiser selon le sens de Pareto [Ehrgott2000]. Les rares travaux qui font référence à la complexité des problèmes multicritères affirment qu'il est communément admis qu'un problème multiobjectif à optimiser au sens Pareto est au moins aussi difficile que la plus difficile des versions monocritère du problème lorsqu'on considère l'un de ses critères. On peut penser qu'il est facile pour le prouver de supposer l'inverse. C'est à dire, on suppose qu'il est moins facile et donc, il suffit de résoudre le problème multicritère, de prendre ensuite le front Pareto pour déduire par simple énumération les solutions ayant la valeur la plus petite par rapport au critère de son choix. Le problème, avec ce raisonnement, est que la taille du front peut être exponentielle par rapport à la taille du problème et donc, cette énumération, n'est pas en général polynomiale.

En réalité, ce qui laisse sentir qu'un problème multicritère est plus difficile que les versions monocritères associées est peut être le fait que la résolution optimale de celles-ci ne permet de donner qu'une partie des solutions du front en omettant la partie cachée de l'iceberg représentée par les solutions de compromis. En outre, dans certains travaux on a pu démontrer que des problèmes monocritères polynomiaux (par exemple, le problème de l'arbre recouvrant de poids minimum) deviennent NP-durs en version multicritère [Ehrgott2000].

En partant de ce constat, nous allons montrer que la version monocritère de notre problème, mettant en valeur le critère de minimisation du trafic intercellulaire, fait partie de la fameuse classe des problème NP-complets.

En supposant qu'un problème est dans NP, il y a plusieurs manières qui permettent de montrer qu'il est NP-complet. L'une de ces techniques est la restriction [Garey1979] qui signifie que lorsqu'un problème P a un cas particulier P' connu pour être NP-complet, il peut être déduit alors que P est également NP-complet.

Il est évident que le modèle proposé, en tenant compte du critère de trafic intercellulaire uniquement, est dans NP. En effet, pour une instance donnée du problème, on peut répondre par "oui" ou par "non" au problème de décision associé au modèle dans un temps polynomial, en parcourant les arêtes intercellulaires pour en calculer la somme des poids. En outre, sans l'ajout des contraintes de cohabitation et de non cohabitation, un cas particulier du modèle proposé, connu dans le jargon de l'optimisation combinatoire par l'appellation "*problème de partitionnement de graphe*", est défini comme suit [Garey1979] :

« Etant donnés un graphe $G=(V,E)$, des poids $w(v) \in Z^+$ pour chaque sommet $v \in V$ et $l(e) \in Z^+$ pour chaque arête $e \in E$, deux nombres entiers positifs K et J; V peut-il être divisé en m sous-ensembles disjoints V_1, V_2, \dots, V_m ,

tels que $\sum_{v \in V_i} w(v) \leq K$ pour $1 \leq i \leq m$ et tel que si $E' \subseteq E$ est l'ensemble des arêtes qui ont leurs deux extrémités dans deux sous-ensembles V_i , alors $\sum_{e \in E'} l(e) \leq J$? »

La correspondance entre ce modèle et le problème de composition de cellules avec un nombre fixe de cellules est évidente. Ce modèle est NP-complet à condition que K (représentant le nombre maximum de machines par cellule, N dans notre modèle) soit supérieur à deux. A Noter que si K est égal à deux, le problème de partitionnement est équivaut à un problème de couplage, pouvant être résolu de façon exacte dans un temps polynomial [Garey1979].

Afin de tenir compte des contraintes de cohabitation et de non cohabitation sans changer la définition du modèle de partitionnement dans les graphes, les modifications suivantes peuvent être apportées au graphe de flux G :

1. Le poids de chaque arête, reliant deux machines non cohabitantes, est porté à une valeur élevée (un multiple de la somme des poids des arêtes, par exemple).
2. Pour chaque couple de machines cohabitantes, les sommets correspondants sont fusionnés dans un seul sommet. On redéfinit alors la fonction de pondération des sommets, w (qui valait un pour tous les sommets) pour indiquer le nombre de machines fusionnées dans le sommet en question. Les arêtes reliant des sommets fusionnés sont alors supprimés de E

Pour un nombre de cellules variable, le problème demeure NP-complet puisque le problème avec un nombre de cellules fixe peut être résolu en plaçant J_{max} égale à J_{min} (voir [paragraphe 6.4.2](#)) dans le problème de composition à nombre variable de cellules. Cette restriction fournit également une transformation polynomiale du problème de partitionnement de graphe vers le modèle proposé (dans sa version monocritère).

Nous avons identifié dans ce chapitre le problème de composition statique et les différentes décisions devant être prise pour sa résolution. Nous avons ensuite présenté une modélisation biobjectif mettant en valeur le critère le plus important avec un autre qui essaye d'en atténuer les effets de bord. La complexité de la version monocritère du problème informe sur la difficulté de la tâche de résolution du problème. Par ce fait, nous avons opté, pour le résoudre, pour une approche évolutionnaire à cause de deux raisons. La première est que cette technique se prête bien à la résolution des problèmes multicritères. La deuxième tient à la facilité d'incorporation de critères sans aucune restriction quant aux caractéristiques des fonctions objectif qui les mesurent. Cette résolution fera l'objet du chapitre suivant.

Chapitre 7

LES AGs POUR LA RESOLUTION DU PROBLEME DE COMPOSITION STATIQUE

La complexité du problème de composition a amené les chercheurs à orienter leurs efforts vers les méthodes approximatives qui permettent de résoudre le problème de façon efficace, mais au détriment de la perte de la garantie d'optimalité. Notre travail ne fera pas exception du moment que les approches de résolution que nous proposons, tout au long du manuscrit, vont s'articuler sur l'approche évolutionnaire.

Les AGs ont été utilisés pour la résolution du problème de composition par [Venugopal1992a, Joines1995, Gupta1995, Al-Sultan1997, Pierreval1998, Zhao2000, Gonçalves2002, Jeon2006]. Dans la plupart de ces travaux, les auteurs affirment que les AGs donnent les meilleurs résultats comparés à un large spectre d'approches publiées dans la littérature.

Dans ce chapitre, nous insisterons sur un aspect dont l'impact sur les performances de l'AG a été souvent négligé par la communauté des chercheurs traitant le problème de composition. Il s'agit du codage utilisé. Dans une première partie nous présentons le codage adopté par la plupart des travaux publiés. Ensuite, nous proposerons notre codage. Lorsque l'AG est confronté à des instances de grande taille, la partie suivante présente une amélioration basée sur la technique d'exploration par séparation et évaluation progressives. Enfin, nous présentons une application des méthodes présentées sur un ensemble d'exemples.

On adoptera dans ce chapitre une approche multicritère ε -contrainte. Le modèle est supposé optimiser le critère de trafic intercellulaire, le critère de charge étant transformé en contrainte. Ce choix est justifié par le souci de rendre l'évaluation des performances plus distinctive.

7.1 L'AG TRADITIONNEL

La plupart des travaux, ayant appliqué l'AG pour résoudre le problème de composition, utilise un codage entier pour représenter les solutions. Pour le problème de composition de cellules uniquement (PCC), ce codage définit une chaîne de m allèles (recip. p allèles pour le problème de formation de familles de produits, PFF). Chaque allèle

est associé à une machine (recip. produit) et indique la cellule (recip. famille) hôtesse. Par exemple, la chaîne suivante représente le chromosome représentatif de la composition de la figure 7.1:

M ₁	M ₂	M ₃	M ₄	M ₅	M ₆
1	1	2	2	2	3

La première cellule contient les machines 1 et 2; la deuxième contient 3, 4 et 5; et la dernière contient la machine 6 uniquement.

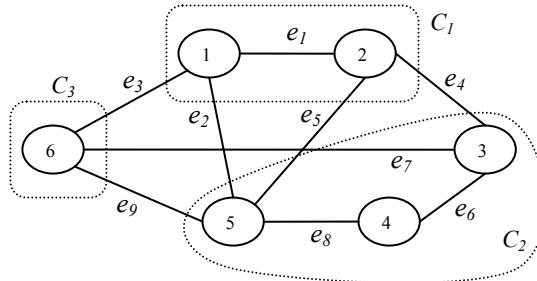


Figure 7.1 Une solution à trois cellules

Ce codage est utilisé dans le plupart des approches publiées. Par exemple, par Venugopal et al. [Venugopal1992a], Gupta et al. [Gupta1995], Pierreval et al. [Pierreval1998] et Zhao et al. [Zhao2000] pour PCC, Al-sultan et al. [Al-Sultan1997] et Jeon [Jeon2006] pour PFF, et Joines et al. [Joines1995] pour PFF/PCC simultanément. Gonçalves et al. [Gonçalves2002] utilisent aussi cette codification pour PCC, quoique indirectement. Ils définissent une chaîne de $(m+1)$ valeurs fractionnaires dans laquelle les m premiers allèles sont dédiés aux machines et la dernière est utilisée pour déterminer le nombre de cellules. La composition de cellules est déduite comme suit : la valeur du dernier allèle, combiné avec la borne supérieure du nombre de cellules, donne le nombre cellules. Ce nombre est à son tour combiné avec chacune des m valeurs des premiers allèles (associés aux machines) donne le numéro de la cellule hôtesse. A titre d'exemple, lorsque J_{max} (voir [paragraphe 6.4.2](#)) est égale à 4, la précédente solution de six machines peut être codée par la chaîne suivante :

0.32	0.23	0.47	0.35	0.65	0.81	0.59
------	------	------	------	------	------	------

Le nombre de cellules est égal à la borne entière supérieure du produit $0.59 \times J_{max}$ qui donne trois cellules. Ensuite, la borne entière supérieure du produit de chaque valeur d'allèle multipliée par $J=3$ donne le numéro de la cellule hôtesse. Par exemple, en notant par $\lceil \cdot \rceil$ l'opérateur donnant la partie entière supérieure, le numéro de la cellule hôtesse de la première machine est : $\lceil 0.32 \times 3 \rceil = 1$.

Des codifications similaires ont été utilisées, directement ou indirectement, dans d'autres approches mét-heuristiques telles que l'approche Réseaux de Neurones de Zolfaghari et al. [Zolfaghari1997], Tabou de Logendran [Logendran1995], celle utilisant le Recuit Simulé de Murthy et al. [Murthy1995] pour ne citer que quelques uns. Cette codification entière est similaire à la modélisation utilisant la variable de décision X_{ij} définie comme suit :

$$X_{ij} = \begin{cases} 1 & \text{si la machine (ou le produit) } i \text{ est affectée à la cellule } j \\ 0 & \text{autrement} \end{cases}$$

Cette modélisation est largement utilisée dans les articles publiés. Parmi ceux-ci, nous avons [Askin1990, Vakharia1990, Boctor1991, Liang1995, Logendran1995, Murthy1995, Al-Sultan1997, Joines1997, Taboun1998, Chen1999] et la liste est loin d'être exhaustive.

Une telle codification renferme une anomalie significative qui entrave les performances des approches de résolution qui l'adoptent. En effet, la représentation entière des cellules induit une redondance qui croît de façon exponentielle par rapport au nombre de cellules. Par exemple, la solution à trois cellules de la figure 7.1 aura six chaînes représentantes possibles, à savoir

<table border="1"><tr><td>1</td><td>1</td><td>2</td><td>2</td><td>2</td><td>3</td></tr></table>	1	1	2	2	2	3	<table border="1"><tr><td>1</td><td>1</td><td>3</td><td>3</td><td>3</td><td>2</td></tr></table>	1	1	3	3	3	2	<table border="1"><tr><td>2</td><td>2</td><td>1</td><td>1</td><td>1</td><td>3</td></tr></table>	2	2	1	1	1	3
1	1	2	2	2	3															
1	1	3	3	3	2															
2	2	1	1	1	3															
<table border="1"><tr><td>2</td><td>2</td><td>3</td><td>3</td><td>3</td><td>1</td></tr></table>	2	2	3	3	3	1	<table border="1"><tr><td>3</td><td>3</td><td>2</td><td>2</td><td>2</td><td>1</td></tr></table>	3	3	2	2	2	1	<table border="1"><tr><td>3</td><td>3</td><td>1</td><td>1</td><td>1</td><td>2</td></tr></table>	3	3	1	1	1	2
2	2	3	3	3	1															
3	3	2	2	2	1															
3	3	1	1	1	2															

obtenues par les trois factorielles ($3!$) permutations des indices de cellules. Cette redondance élargit considérablement l'espace de recherche à explorer, rendant ainsi la tâche de la méthode de résolution très difficile.

Ce problème devient de plus en plus critique pour les approches considérant le nombre de cellules comme un paramètre à optimiser car celle-ci ne va pas seulement élargir l'espace de recherche mais elle va l'encombrer par des solutions médiocres. En effet, plus le nombre de cellules d'une solution réalisable est important, plus elle possède de chaînes représentantes et puisque l'objectif le plus important dans notre problème est la minimisation du trafic intercellulaire (ce qui va probablement être vérifié pour une solution à nombre de cellules modéré), les solutions médiocres vont infester les lieux puisque possédant beaucoup de représentants. A titre d'exemple, pour l'atelier de six machines de la figure 7.1, la solution la plus mauvaise en terme de trafic intercellulaire (c'est la composition dont toutes les cellules sont réduites à une seule machine) possède $6! = 3.628.800$ chaînes représentantes.

Cette redondance génère, pour les approches évolutionnaires, d'autres problèmes. En effet, les opérateurs assurant la diversité dans la population sont déroutées par le degré élevé de chaînes différentes qui, malheureusement, ne représentent qu'un ensemble très restreint de solutions. En outre, l'opérateur d'exploitation, le croisement, devient un opérateur d'exploration, i.e. une mutation, car les caractéristiques des parents peuvent être facilement non reconduites dans leur progéniture, même si les parents représentaient exactement la même composition.

Le problème de la redondance a valu aux AGs d'être considérés, à tort, par certains chercheurs¹ non adéquats aux problèmes de partitionnement.

Afin de pallier à ces insuffisances, les remèdes proposés dans la littérature oeuvrent sur différents niveaux : Dans [Boulif2000], l'auteur propose d'intervenir au niveau du codage afin d'éliminer (ou presque) la redondance. Ils suggèrent d'établir une contrainte sur l'affectation des indices aux cellules : la cellule 1 est celle qui contient la machine 1, la cellule 2 est celle qui contient la première machine (en terme d'indice) qui n'est pas dans la première cellule, et ainsi de suite.

¹ Dans [Vin2003] on peut lire :" Falknauer [Falknauer1998] a montré la faiblesse des AGs standards quand ils sont appliqués à des problèmes de groupement."

L'auteur de [Falkenauer1998] propose d'intervenir au niveau de l'opérateur de croisement. Il définit un crossover par groupement de machines. Notons que ce dernier procédé, contrairement au précédent, n'élimine pas la redondance mais essaye d'en atténuer les conséquences sur la tâche d'exploitation du code génétique.

Les résultats fournis par ces interventions ont permis d'améliorer considérablement les performances de l'AG traditionnel. Cependant, les problèmes liés à cet AG, et qui ont pour origine une anomalie structurelle dans un composant essentiel de la méthode qui est le codage, nous incitent à revoir ce point de façon catégorique. C'est cette réflexion qui nous a menée à une conclusion logique : si le problème est lié au codage alors il faut le changer.

7.2 AG BINAIRE

Nous pouvons remarquer, dans le codage entier, que le fait de connaître l'indice de la cellule pour chaque machine était inutile. Il était même la cause de la redondance, source de tous les maux. Ce qui est nécessaire et suffisant de savoir, pour définir une composition de cellules (une partition dans le graphe des flux), est plutôt le fait qu'un groupe de machines est dans la même cellule ou non et peu importe son indice. Ceci nous amène à dire qu'il est juste suffisant de savoir si pour tout couple de machines, celles-ci sont ou ne sont pas dans la même cellule. Ce qui est parfaitement représentable par une variable binaire. Ce raisonnement nous a permis d'élaborer le codage suivant :

7.2.1 Codage binaire

La méthode que nous proposons adopte un codage binaire qui s'articule sur le modèle basé sur la théorie des graphes (voir [paragraphe 6.4](#)). Chaque solution est codée par une chaîne d'allèles binaires de la longueur $\text{Card}(E)$ (voir [remarque 6.1](#)). Chaque allèle représente une arête du graphe de flux $G=(M,E)$ et indique si 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). Par convention, si une arête est intercellulaire, elle est notée un (1), sinon elle est notée zéro (0). Avec cette codification, la solution à trois cellules de la figure 7.1 sera codée par la chaîne suivante :

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
0	1	1	1	1	0	1	0	1

L'avantage le plus important du codage binaire est que les AGs sont très sensibles aux alphabets réduits (0 et 1 dans ce cas-ci). Avec un alphabet binaire, il est plus facile pour l'AG de détecter les *bons blocs* des codes des individus. Un autre avantage vient du gain relatif de l'espace mémoire requis pour mettre en œuvre l'algorithme puisque des variables booléennes peuvent être utilisées pour représenter les chromosomes de la population.

Le codage binaire semble avoir deux inconvénients potentiels. Cependant, le développement ci-après montrera, que ce qui paraît être inconvénient est, en fait, un avantage.

7.2.1.1 Redondance du codage Binaire

La figure 7.2 dévoile une situation incohérente à la mesure où les arêtes e_k , e_{k+1} et e_{k+3} sont intracellulaires mais e_{k+2} est intercellulaire. Autrement dit, les extrémités d'une arête intercellulaire (codée 1) sont reliées par une chaîne d'arêtes intracellulaires.

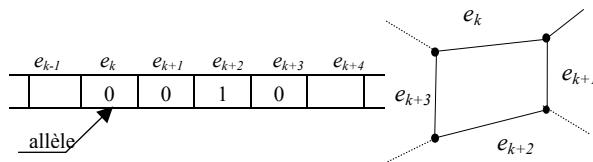


Figure 7.2 Incohérence de certaines chaînes binaires.

Cette incohérence est en réalité une redondance. En effet, pour un chromosome donné, on peut considérer que les allèles de valeur nulle définissent un sous-graphe du graphe de flux dont les composantes connexes déterminent les cellules de la solution associée (voir fig. 7.3).

Cette redondance, contrairement à celle du codage traditionnel, a un bon effet sur l'exploration de l'AG. En effet, plus le nombre d'arêtes intracellulaires dans une solution est grand, plus le nombre de chaînes représentantes est grand. Ainsi, les solutions avec des mouvements intracellulaires intensifs, et donc celles pauvres en mouvement intercellulaires, possèdent relativement le plus de représentants.

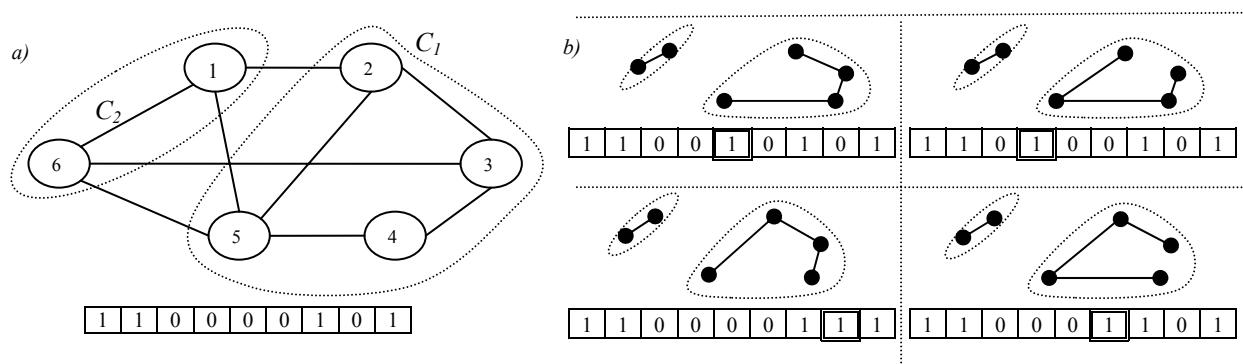


Figure 7.3 La redondance du codage binaire.

7.2.1.2 Cécité du codage binaire

L'exemple suivant illustre ce qui semble être un autre inconvénient du codage binaire. Considérons le graphe de flux de la figure 7.4 comportant quatre machines $\{M_1, \dots, M_4\}$ et trois arêtes. Les deux premières arêtes, chacune avec un poids a , lient M_1 à M_3 et M_2 à M_4 , respectivement. La troisième, de poids b , lie M_3 à M_4 . Supposons que b est plus grand que $2a$.

Lorsque le nombre maximum de machines par cellule est égal à deux ($N=2$), la valeur de la solution optimale minimisant le trafic intercellulaire est égale à $2a$. Cette évaluation peut être obtenue à partir de l'une des compositions suivantes : $C_1 = \{M_1\}$, $C_2 = \{M_2\}$ et $C_3 = \{M_3, M_4\}$ ou $C_1 = \{M_1, M_2\}$ et $C_2 = \{M_3, M_4\}$. Bien que le code binaire puisse facilement représenter la première composition, il est dans l'incapacité de représenter la seconde composition parce qu'il n'y a aucun chemin intracellulaire reliant M_1 et M_2 . Heureusement, dans ce cas-ci,

le code binaire a pu représenter une solution optimale. Ce qui est considéré comme suffisant. Cependant, si le nombre maximum de cellules est égal à deux ($J_{max} = 2$), la situation devient critique car la seule solution optimale que le codage binaire peut représenter devient non réalisable.

Cette cécité de l'AG, qui ne peut "voir" certaines solutions réalisables, semble être le grand handicap du codage binaire. Fort heureusement, elle ne l'est pas. Au contraire, cette cécité peut être même considérée comme étant l'une des vertus du codage binaire : Les solutions non vues par l'AG présentent des cellules avec des groupes indépendants de machines. Or, l'AG considère que chaque groupe indépendant définit une cellule différente. Ainsi, si le nombre de cellules n'est pas borné supérieurement, l'AG essayera de définir un nombre de cellules qui est en total harmonie avec l'esprit de la philosophie TG et engendré par la nature de l'instance du problème.

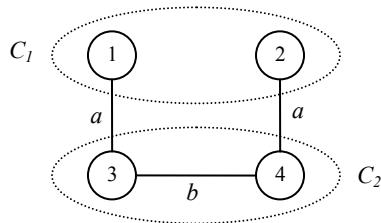


Figure 7.4 Une solution invisible.

Remarque 7.1 : La longueur des chaînes peut être réduite par les contraintes de cohabitation. En effet, quand deux machines sont obligées de faire partie de la même cellule, l'allèle correspondant à l'arête reliant les sommets associés à ces machines est mis à zéro. Du moment que sa valeur doit rester constante, cet allèle peut être enlevé du code génétique. Les valeurs des allèles correspondant aux machines de cohabitation sont sauvegardées dans un vecteur indépendant qui peut être utilisé pour évaluer l'adaptation de la solution.

Il semble que la même procédure peut être appliquée aux allèles correspondant aux machines non-cohabitantes. Cependant, le problème d'incohérence nous oblige de vérifier si les machines non-cohabitantes ne sont pas liées par une chaîne intracellulaire.

7.2.2 Population initiale

Pour un AG binaire ayant montré son aptitude à fournir de bonnes solutions, même avec une population initiale aléatoire, nous avons choisi de générer la population initiale aléatoirement dans nos expériences. Cependant, étant donné que $(m-1)$ arêtes intracellulaires peuvent donner une seule composante connexe dans le cas où l'ensemble des allèles nuls constitue un arbre recouvrant, il est préférable de mettre la probabilité P_1 de produire des allèles nuls (arêtes intracellulaires) à une valeur inférieure à la probabilité P_2 de produire des allèles non nuls (arêtes intercellulaires) afin d'éviter les solutions unicellulaires. A cet effet, quand le graphe de flux n'est pas faiblement relié (c'est-à-dire, le nombre d'arêtes est plus grand que celui des sommets), les probabilités P_1 et P_2 peuvent être définies par la formule suivante :

$$P_1 = (m-1)/\text{card}(E) \quad \text{et} \quad P_2 = 1 - P_1$$

où m est le nombre de sommets (machines) et $\text{card}(E)$ est le nombre d'arêtes dans le graphe de flux.

7.2.3 La fonction d'adaptation

La fonction d'adaptation, "fitness", est un paramètre essentiel dans la conception de l'AG. La fitness fournit une évaluation pour chaque individu, permettant d'avoir une relation d'ordre total dans la population de l'AG qui sera ensuite utilisée dans la sélection. Dans la littérature, une telle évaluation est généralement obtenue par l'utilisation directe de la fonction objectif. Cependant, pour des problèmes d'optimisation avec contraintes, comme le nôtre, l'utilisation simple de la fonction objectif n'est pas suffisante. En effet, en raison de la génération aléatoire d'individus, ou à cause de l'application des opérateurs génétiques, des solutions non réalisables sont produites. Dans ce cas, la relation d'ordre peut évaluer les solutions non réalisables comme étant meilleures que certaines solutions réalisables. Ce qui a pour effet de multiplier les mauvais individus par l'opérateur de sélection au détriment des meilleurs. Pour remédier à ce phénomène, les chercheurs recourent en général à deux techniques. La première consiste à ajuster le code génétique de toute solution non réalisable pour la rendre réalisable (ce que font [Venugopal1992a, Al-Sultan1997, Zhao2000] à titre indicatif). La deuxième technique, largement utilisée en optimisation évolutionnaire [Aguire2003], consiste à pénaliser sévèrement les solutions non réalisables pour éviter leur sélection. Etant donné que la nature présente plusieurs situations dans lesquelles de bons individus peuvent être extraits à partir d'autres mauvais (les travaux dans le génie génétique confirment une telle vérité), un AG capable de tirer profit des bons blocs, enfouis dans le code génétique des solutions non réalisables, serait tout à fait indiqué. En se basant sur cette réflexion, il semble préférable de définir la fonction fitness F comme suit :

Premièrement, le problème de minimisation est transformé en un problème de maximisation par l'intermédiaire de la formule,

$$T'(C) = B - T(C),$$

où $T(C)$ est la valeur de la fonction objectif, indiquant le trafic intercellulaire pour une solution donnée C , et B est une borne supérieure de $T(C)$. Ainsi, la maximisation de T' équivaut à la minimisation de T .

Ensuite, la valeur obtenue est translatée en utilisant la formule,

$$Z(C) = T'(C) + (u - v(C)) \times B,$$

où u est le nombre de contraintes et $v(C)$ est le nombre de contraintes violées par C .

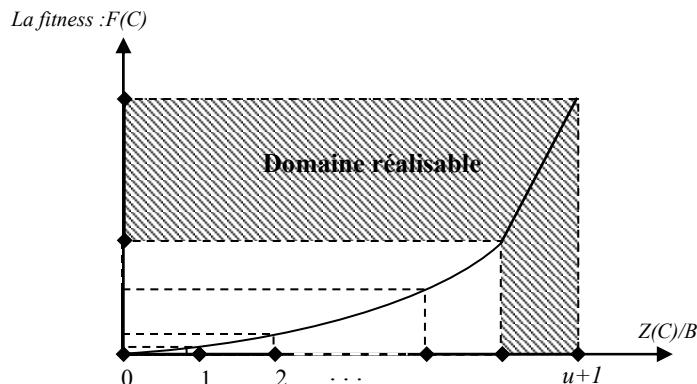


Figure 7.5 Procédure d'ajustement de la fitness F.

Cette translation permet de trier les solutions dans $u+1$ intervalles consécutifs de longueur B : Le premier intervalle étant dédié aux solutions violant toutes les contraintes. Le suivant pour celles ne vérifiant pas $u-1$

contraintes, et ainsi de suite, jusqu'à ce que les solutions réalisables soient mises dans le $(u + 1)^{\text{ième}}$ et dernier intervalle. La valeur de $Z(C)$, ainsi obtenue, peut alors être ajustée en utilisant une fonction qui permet d'étendre le domaine des solutions réalisables (voir fig. 7.5).

L'interprétation du procédé de translation est directe : il aide l'AG à se déplacer du domaine de solutions non réalisables vers celui des solutions réalisables. Cependant, cette translation peut avoir un mauvais effet venant du fait que la capacité de l'AG à distinguer entre bonnes et mauvaises solutions réalisables est affaiblie. L'application du procédé d'ajustement permet à l'AG de maintenir cette capacité en l'incitant non seulement à quitter l'espace non réalisable, mais aussi à converger dans l'espace réalisable vers les solutions les mieux adaptées.

En outre, la fonction d'ajustement jouera un autre rôle très important : celui de normaliser la ou les valeurs de la fonction d'adaptation. Nous y reviendrons dans le [paragraphe 8.3.1.3](#).

7.2.4 Les opérateurs génétiques

Les opérations suivantes sont appliquées, dans l'ordre, à la population initiale après que l'individu le plus adapté ait été sauvegardé.

7.2.4.1 Sélection

Le procédé aléatoire de la "roue de loterie biaisée" est utilisé pour choisir un individu [Goldberg1989]. Sur cette roue, chaque individu de la population possède une portion proportionnelle à son adaptation, calculée en utilisant la formule,

$$P(C_k) = \frac{F(C_k)}{\sum_{t=1}^s F(C_t)}$$

où s est la taille de population de l'AG, C_k est la k -ième solution de la population ($k \in \{1, 2, \dots, s\}$) et $F(C_k)$ est la fitness (adaptation) de la solution C_k .

Un nombre entre 0 et 1 est choisi aléatoirement $s \times P_3$ fois où P_3 est un nombre réel entre 0 et 1. A chaque fois, l'individu lié à la section contenant le nombre généré est choisi. Les individus moins adaptés laissent ainsi leurs places (dans la population) à ceux qui sont mieux adaptés et les $s \times (1 - P_3)$ emplacements restants sont remplis par de nouveaux individus générés aléatoirement.

7.2.4.2 Croisement

Une paire d'individus, les parents, est choisie aléatoirement parmi les individus de la population. Les parents

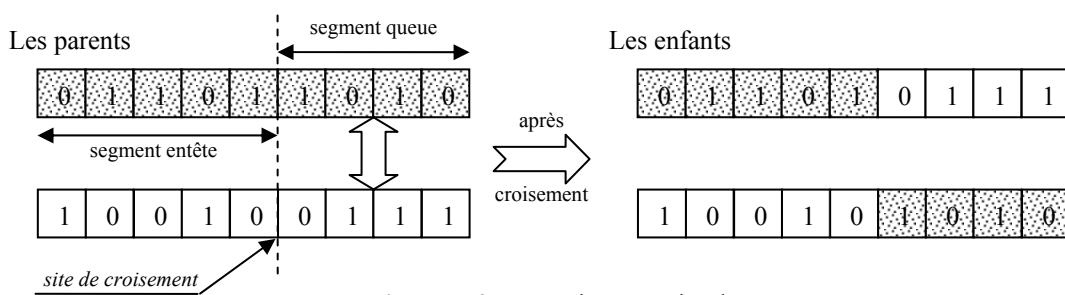


Figure 7.6 Un croisement simple

sont alors divisés à un point aléatoire mais identique pour les deux parents, donnant ainsi deux segments entêtes et deux segments queues (voir fig. 7.6). Deux enfants sont alors créés en échangeant les segments queues des parents. Les parents sont ensuite remplacés par leurs enfants. Ce procédé est répété avec une autre paire jusqu'à ce que $s \times P_4$ individus soient remplacés où P_4 est une valeur entre 0 et 1.

7.2.4.3 Mutation

Des individus, aléatoirement choisis, sont modifiés en changeant un allèle de façon aléatoire dans leurs chaînes. Le nombre des individus qui subissent cet opérateur est déterminé par la valeur du produit $s \times P_5$ où P_5 est une valeur entre 0 et 1.

Après application de ces opérations, le premier individu sauvegardé est réinséré dans la population à la place d'un individu moins adapté, choisi aléatoirement. La population résultante devient la nouvelle génération et le processus est répété selon un nombre prédéterminé d'itérations i_{max} .

7.2.5 Paramètres de contrôle

Les paramètres de contrôle, cités tout au long des paragraphes précédents, sont récapitulés ci-dessous:

- s le nombre d'individus dans la population.
- P_3 le pourcentage d'individus à sélectionner.
- P_4 le pourcentage de la population qui subit l'opérateur de croisement.
- P_5 le pourcentage de la population qui subit l'opérateur de mutation.
- i_{max} le nombre d'itérations (générations) de l'AG.

Ces paramètres ont un rôle influant dans le perfectionnement des résultats de l'AG. De ce fait, plus d'attention doit être prêtée à leur ajustement. Le premier paramètre (la taille de population) doit être suffisamment grand pour assurer une bonne exploration de l'espace de recherche, sans être trop grand à cause du coût cpu que ceci nécessiterait. Le second (le pourcentage des individus sélectionnés) doit être mis à une valeur qui permet la duplication facile des bons individus susceptible de produire les individus les plus adaptés pendant l'étape de reproduction. Le troisième paramètre (le pourcentage des individus subissant le croisement) doit être assez grand pour permettre une bonne exploitation du code génétique de la population. Le pourcentage des individus mutants doit être assez petit pour éviter de compromettre le processus de convergence. Enfin, le nombre d'itérations doit être porté à une valeur qui permet à l'AG de mener le processus de convergence à terme. Malheureusement, quelles que soient les précautions prêtées à l'ajustement théorique de ces paramètres, l'expérimentation est généralement nécessaire pour choisir les valeurs des paramètres qui pousseront l'AG à donner le meilleur de lui-même.

Bien que l'AG binaire donne des résultats satisfaisants, comparé à l'AG traditionnel, il serait utile de pouvoir diriger l'exploration génétique vers les régions les plus prometteuses, notamment en traitant des problèmes de taille réelle. Dans ce qui suit, nous proposons une amélioration de l'AG binaire consistant en une hybridation avec une méthode d'exploration par séparation et évaluation.

7.3 LA METHODE SEP-AG

Les méthodes d'exploration par Séparation et Evaluation Progressive (SEP), ou Branch and Bound (B&B) en anglais, sont des méthodes qui réalisent une énumération intelligente de l'espace de solutions à la quête des meilleures solutions. Les méthodes SEP font partie des méthodes exactes, à la mesure où si certaines conditions sont respectées, elles permettent de résoudre le problème de façon optimale. Les méthodes exactes, ayant été déjà appliquées au problème de composition, sont généralement basées sur l'énumération implicite des valeurs des variables de décision du modèle considéré. Ce modèle étant en général un programme mathématique souffrant du problème de redondance généré par une variable semblable à celle définie dans le [paragraphe 7.1](#) (voir, par exemple, [Kusiak1993, Heragu1997]). Cela va sans dire qu'un problème de redondance, de surcroît exponentiel, est plus critique pour une méthode exacte à cause du coût excessif, en temps de calcul et en espace mémoire, que de telles méthodes énumératives nécessitent.

Dans ce paragraphe, le principe de base des méthodes SEP est brièvement présenté, suivi par une description de la méthode proposée qui est une hybridation entre une méthode SEP cocyclique et l'AG binaire. Pour la première, nous montrerons qu'il est possible d'exploiter la prise en compte du critère de charge par l'approche ε -contrainte pour améliorer les procédures d'exploration et d'évaluation de la première étape de la méthode proposée.

7.3.1 Principe de base des méthodes SEP

Les énumérations par séparation et évaluation progressives consistent à diviser l'espace de recherche en sous-ensembles qui seront à leur tour décomposés en d'autres sous-ensembles. C'est le principe de séparation qui donne une structure d'arborescence. Une fonction d'évaluation, établissant une borne inférieure (dans le cas d'un problème de minimisation) des fonctions objectif des solutions associées à chaque sous-ensemble, est à chaque fois calculée. Le qualificatif d'énumération "intelligente", ou implicite des méthodes SEP, vient du fait que si une solution réalisable est atteinte avec une fonction objectif égale à v , toutes les branches dont la borne inférieure est plus grande que v sont inhibées (non explorées). Ce procédé est alors répété afin de localiser une solution bonne ou optimale. Pour de plus amples détails sur les méthodes SEP et leurs principes, voir [Papadimitriou1982] et [Gondran1989].

7.3.2 La méthode SEP cocyclique

7.3.2.1 L'idée générale

Les AG sont réputés être des méthodes d'optimisation aveugles. Ce qualificatif fait référence au principe d'exploration des AG qui n'utilisent, dans sa forme simple, que la valeur de l'adaptation pour guider la recherche. Cette caractéristique est à l'origine de la robustesse des AGs qui ont pu être appliqués avec succès sur un large spectre de problèmes car les AGs ne requièrent aucune conditions telles que linéarité, dérivabilité, ... etc., pour qu'ils se mettent en oeuvre. Cependant, il est communément admis que pour être performante dans la résolution d'un problème donné, une méthode peut (voire doit) tirer profit des informations spécifiques à ce problème. Le théorème NFL [Wolpert1997], largement admis, s'attelle à le prouver. En effet, ces informations peuvent être exploitées afin de guider la recherche de façon implicite. C'est dans cet esprit que nous présentons la méthode SEP-AG.

Dans le but de prévenir l'AG contre la réalisation d'explorations peu ou non prometteuses, une procédure SEP est utilisée pour déterminer les régions susceptibles de cacher de bonnes solutions. Ces régions sont ensuite explorées par l'AG (voir fig. 7.7).

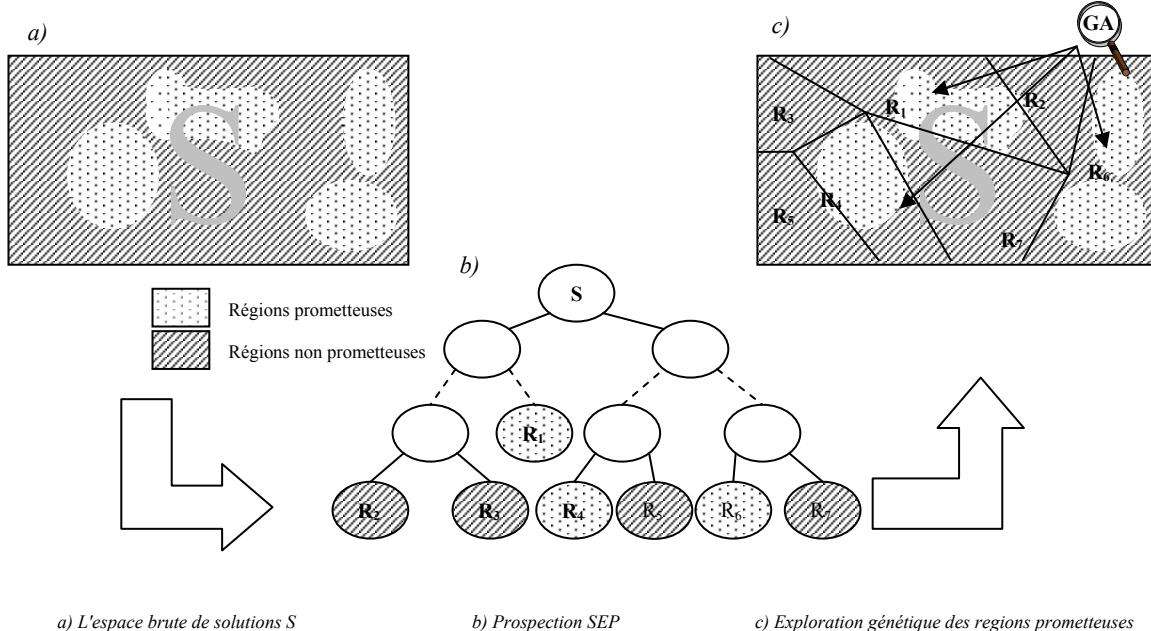


Figure 7.7 L'idée générale de la méthode hybride SEP-AG

Enonçons ce qui précède formellement : considérons le modèle basé sur la théorie des graphes (voir [paragraphe 6.4](#)). Puisqu'une solution est une partition des sommets du graphes de flux, elle peut, *dans le cas d'un graphe connexe*, être représentée par une somme, en utilisant l'opérateur booléen OU dénoté \oplus , de co-cycles (coupes). La solution de la figure 7.8 peut être définie par la somme de deux des co-cycles w_1, w_2, w_3 .

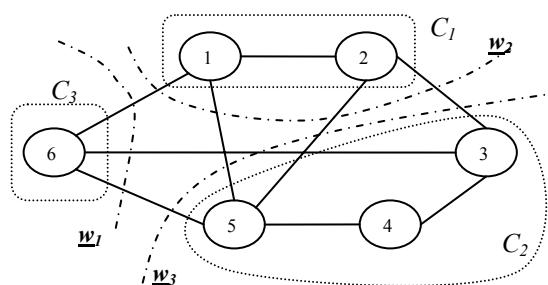


Figure 7.8 Décomposition en co-cycles.

Par exemple, la somme de $w_1=(0,0,1,0,0,0,1,0,1)$ et $w_2=(0,1,1,1,1,0,0,0,0)$ donne $w_1 \oplus w_2=(0,1,1,1,1,0,1,0,1)$. Ce qui n'est autre que le code génétique de la solution. Pour mettre en œuvre la méthode, nous devons spécifier les règles de séparation et la manière avec laquelle nous calculons la borne inférieure des sous-ensembles explorés.

Remarque 7.2 : Si le graphe de flux n'est pas connexe, il doit être connecté en ajoutant des arêtes fictives de poids nuls. Cette procédure, que nous appelons prétraitement du graphe de flux, nous permet de supposer que le graphe de flux est connexe dans tout le développement qui suit.

7.3.2.2 Règles de séparation

Soit $A = \{w_1, w_2, \dots, w_n\}$ l'ensemble des n co-cycles associés au graphe de flux G et soit S l'ensemble de toutes les solutions possibles. L'ensemble S est d'abord divisée 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 à partir de w_1). Puis, chacun des sous-ensembles S_0 et S_1 est divisé en deux sous-ensembles enfants (S_0 donne S_{01} et 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.

a) La procédure de séparation : La procédure de séparation est défini en associant à chaque co-cycle w_i , $i \in \{1, 2, \dots, n\}$ une variable décision x_i (voir fig. 7.9) définie comme suit:

$$x_i = \begin{cases} 1 & \text{si } w_i \text{ construit les solutions du sous - ensemble enfant.} \\ 0 & \text{sinon.} \end{cases}$$

Par conséquent, le sous-ensemble S_{00101} , par exemple, désigne le sous-ensemble atteint en suivant la séquence des 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. 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 d'autres co-cycles à cette somme.

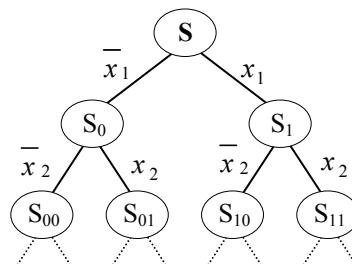


Figure 7.9 Représentation de la règle de séparation

b) Construction de la liste des co-cycles : Le nombre de co-cycles n croît de manière exponentielle par rapport au nombre de sommets et donc devient colossal pour des instances réelles. Pour cette raison, la procédure de séparation est réalisée sur un sous-ensemble de co-cycles choisis pour cet effet, que nous appelons la *Liste de co-cycles* (L). L peut être déterminée soit de façon aléatoire ou par un procédé d'optimisation.

La *Liste de co-cycles* L peut influencer l'efficacité de la méthode proposée et ses éléments doivent ainsi être choisis avec soin. La liste L doit être assez grande pour permettre une bonne exploration de l'espace des solutions, sans être aussi longue, au risque de générer un coût prohibitif en espace mémoire et en temps d'exécution. En outre, il est important de choisir des co-cycles de faible poids (le poids d'un co-cycle est la somme des poids de ses arêtes) afin de détecter les régions prometteuses de l'espace de solutions.

Etant donné ces considérations, un algorithme génétique peut être employé pour déterminer la majorité des co-cycles de L et le restant peut être déterminé aléatoirement. Puisque l'ensemble des co-cycles, muni de l'opérateur booléen *XOR*, décrit un sous-espace vectoriel, chaque co-cycle peut être écrit sous forme d'une combinaison linéaire unique des co-cycles d'une base donnée de cet espace. Le codage de cet AG définit une chaîne de longueur $m-1$ (c'est-à-dire la dimension de la base, égale au nombre de sommets moins un) dans laquelle chaque allèle est associé à un co-cycle de base et indique si le co-cycle associé est employé pour générer le co-cycle défini par la chaîne. La base de l'espace vectoriel de co-cycles peut être déterminée en utilisant la méthode de l'arbre recouvrant [Gondran1989].

c. L'évaluation de la borne inférieure : Une bonne borne inférieure est caractérisée par deux critères qui sont malheureusement contradictoires : la rapidité de calcul et l'adhérence de valeur. Le premier fait allusion à la capacité d'évaluer la borne inférieure de chaque sous-ensemble aussi rapidement que possible afin d'accélérer la vitesse d'exploration. Ceci est généralement le fruit d'une procédure d'évaluation peu complexe. Le second entend dire que cette évaluation doit être aussi proche que possible de la valeur de la fonction objectif associé à la meilleure solution du sous-ensemble. Cette deuxième caractéristique, qui permet d'accentuer le recours au procédé d'inhibition (et par voie de conséquence, la vitesse de la recherche SEP), est néanmoins synonyme de complexité.

Un compromis entre ces deux caractéristiques peut être obtenu en utilisant le procédé d'évaluation suivant : La fonction objectif de la solution (nous l'appellerons solution basique), définie par la somme des co-cycles ayant des variables décision positives, est utilisée pour établir une évaluation de base de la borne inférieure d'un sous-ensemble.

Quand un sous-ensemble est subdivisé en deux sous-ensembles selon un co-cycle w_i , l'évaluation de base de l'enfant, correspondant à la valeur positive de la variable décision x_i , est obtenue à partir de l'évaluation du parent à laquelle est ajouté le poids des arêtes de w_i qui n'appartiennent pas aux co-cycles qui composent le sous-ensemble parent. Pour le deuxième enfant, correspondant à la valeur nulle de x_i , l'évaluation est exactement la même.

Cette évaluation de base peut être serrée si la solution liée au sous-ensemble courant est non réalisable. En effet, la non réalisabilité, qu'elle soit due à

1. la taille des cellules,
2. au nombre de cellules,
3. aux contraintes de non cohabitation ou
4. à la contrainte de charge,

exige la scission d'au moins une cellule pour obtenir une solution réalisable. Nous proposons, par ce fait, de considérer la contrainte de taille de cellules d'abord : si la solution contient des cellules dont la taille excède le nombre de machines permis dans une cellule, la valeur du plus petit poids des arêtes intracellulaires de chacune de ces cellules est déterminée et leur somme est ajoutée à l'évaluation de base. Autrement, le nombre minimum de cellules permis est considéré : si cette valeur n'est pas respectée, le poids de l'arête de plus petit poids parmi les arêtes intracellulaires est ajouté. Ensuite, si toutes les deux contraintes précédentes sont respectées, les contraintes de non cohabitation sont considérées : la somme des poids des arêtes intracellulaires dont les extrémités sont des

machines non cohabitantes, sont ajoutées à l'évaluation de base. Enfin, on passe à la contrainte liée au critère de charge. Le poids de l'arête intracellulaire ayant le poids minimum parmi les arêtes des cellules vérifiant les conditions d'éligibilité à une scission réductrice (voir [proposition 7.1](#)) est ajouté à l'évaluation de base.

Notons que le choix de l'ordre de considération des contraintes n'est pas fortuit. En effet, les solutions basiques des sous-ensembles des premiers niveaux de l'arborescence sont susceptibles de violer les contraintes de la taille et du nombre minimum de cellules car on utilise peu de co-cycles pour leur construction. En outre, entre ces deux contraintes, la première a été favorisée car, en général, elle permet de mieux serrer l'évaluation en faisant intervenir les poids de plusieurs arêtes. L'ordre figé qui vient d'être donné est basé sur des fondements intuitifs. L'élaboration d'une méthode adaptative qui change l'ordre de manière dynamique pendant l'exécution ferait l'objet d'une question intéressante à développer.

Pour l'évaluation par défaut des sous-ensembles dont les variables globales de décision sont toutes nulles, la somme des poids des K premières arêtes de poids minimum est utilisée. K étant le nombre minimum d'arêtes pouvant définir un co-cycle.

Une fois qu'une solution réalisable est définie pour un sous-ensemble, elle cesse d'être explorée du moment que l'ajout de tout autre co-cycle ne fait que dégrader la solution soit en élevant son trafic intracellulaire ou son nombre de cellules.

7.3.3 Conditions d'éligibilité d'une cellule à la scission

Dans les méthodes SEP serrer l'évaluation n'est pas une mince affaire. En effet, une petite amélioration peut dégager des avantages considérables en terme d'inhibition de branches qui vont se répercuter positivement sur l'espace mémoire requis et la rapidité d'exécution de la méthode. Dans ce qui suit nous montrons qu'il est possible d'exploiter le critère de charge, inclus sous forme de contrainte, pour l'accomplissement d'une telle amélioration.

Un co-cycle appliqué à une composition permet de scinder certaines de ses cellules pour en créer d'autres. Cette section est consacrée à l'étude de l'effet de la scission sur le critère de charge. Pour ce faire, nous énonçons la proposition suivante qui permet de donner une condition nécessaire pour que la scission d'une cellule conduit nécessairement à l'amélioration (diminution) de la valeur du critère de charge. On utilise dans ce qui suit les notations suivantes :

Soit $C = \{C_1, \dots, C_J\}$ une composition de J cellules et soient ch_1, \dots, ch_J les charges cellulaires associées. Supposons que l'application d'un co-cycle sur la solution C conduise à la scission de la cellule C_k en deux cellules : A et $C_k - A$ (où $-$ désigne l'opérateur de soustraction ensembliste). Soit C' la composition obtenue, $C'_k = \{C_1, \dots, A, C_k - A, \dots, C_J\}$.

Proposition 7.1 : Une condition nécessaire pour que $CH(C') < CH(C)$ est que

$$(ch_k)^2 > 2 \left[S_1^2 \frac{(2J+1)}{J^2(J+1)} - S_2 \frac{1}{J} \right]$$

avec S_1 est la somme des charges cellulaires et S_2 la somme de leurs carrés.

Preuve :

On étudie la différence de charge $\Delta CH = CH(C') - CH(C)$ en fonction de l'une des charges obtenues après scission. Notons x une telle charge et, sans perdre la généralité, on considère x la charge de A . Nous avons alors :

$$CH(C) = \frac{1}{J} \cdot \sum_{j=1}^J (ch_j - \bar{ch})^2 = \frac{1}{J} \cdot S_2 - \frac{1}{J^2} \cdot S_1^2.$$

D'autre part,

$$CH(C') = \frac{1}{J+1} \cdot S'_2 - \frac{1}{(J+1)^2} \cdot S_1^2$$

où $S'_2 = ch_1^2 + \dots + x^2 + (ch_k - x)^2 + \dots + ch_J^2$. Puis après des développements simples, nous aurons :

$$CH(C') = \frac{2}{J+1} \cdot x^2 - \frac{2 \cdot ch_k}{(J+1)^2} \cdot x + \left[\frac{1}{J+1} \cdot S_2 - \frac{1}{(J+1)^2} S_1^2 \right].$$

Ceci nous permet de déduire que :

$$\Delta CH = \frac{2}{J+1} \cdot x^2 - \frac{2 \cdot ch_k}{(J+1)^2} \cdot x + \left[\frac{2 \cdot J+1}{J^2 \cdot (J+1)^2} \cdot S_1^2 - \frac{1}{J \cdot (J+1)} S_2 \right]$$

La valeur du coefficient $\frac{2}{J+1}$ étant positive, la valeur de ΔCH ne peut être négative que si l'équation $\Delta CH=0$

admet deux racines. Par conséquent, en notant c le coefficient du membre inférieur,

$$\Delta' = \left(\frac{-ch_k}{J+1} \right)^2 - \frac{2}{J+1} \cdot c > 0 \Rightarrow (ch_k)^2 > 2 \left[S_1^2 \frac{(2J+1)}{J^2(J+1)} - S_2 \frac{1}{J} \right] \text{ CQFD. } \square$$

La condition précédente procure à la méthode SEP une aide considérable en lui allouant de serrer d'avantage la valeur de l'évaluation des sous-ensembles explorés. En poussant un peu plus loin le développement précédent, on peut déduire une condition nécessaire et suffisante pour que la scission soit améliorante.

Proposition 7.2 : Une condition nécessaire est suffisante pour que $CH(C') < CH(C)$ est que :

1. $(ch_k)^2 > \Psi$ avec $\Psi = 2 \left[S_1^2 \frac{(2J+1)}{J^2(J+1)} - S_2 \frac{1}{J} \right]$
2. $x \in \left[\frac{ch_k}{2} - \sqrt{\left(\frac{ch_k}{2} \right)^2 - \frac{\Psi}{4}}, \frac{ch_k}{2} + \sqrt{\left(\frac{ch_k}{2} \right)^2 - \frac{\Psi}{4}} \right]$

Preuve : En reconduisant le développement de la preuve de la proposition précédente, on déduit que ΔCH est négative si et seulement si $(ch_k)^2 > \Psi$ et x appartient à l'intervalle compris entre les deux racines réelles de l'équation $\Delta CH=0$ CQFD \square

L'exploitation de cette condition nécessaire et suffisante dans la phase d'exploration par séparation et évaluation consiste à déterminer pour une composition basique non réalisable donnée (pour violation de la contrainte de charge) les cellules dont les charges vérifient la condition n°1. Ces charges qui vérifient la condition nécessaire sont éligibles à une scission améliorante. Ensuite, étant donné les co-cycles non encore utilisés dans la séparation,

on considère ceux qui réalisent une scission d'une cellule éligible de telle sorte à donner une nouvelle cellule dont la charge x vérifie la condition n°2. Si un tel co-cycle n'existe pas alors la branche en question est inhibée.

Lorsque la liste de co-cycles L est grande, cette procédure devient très coûteuse en terme de calcul. C'est la raison pour laquelle nous avons décidé de n'exploiter que la condition nécessaire.

7.3.4 Conditions d'optimalité de la méthode SEP

Il est vrai que nous n'envisageons pas résoudre le problème de composition de façon optimale car il s'avère que ce problème est NP-complet. Donc, sa résolution de façon optimale serait prohibitive pour des instances réelles. Cependant, explorer les conditions d'optimalité peut être intéressante à la mesure où elle permettra d'évaluer la qualité des approches approximatives sur des instances de taille modérée. A cet effet, nous nous proposons dans ce qui suit de discuter les conditions pouvant garantir une telle optimalité

L'optimalité de la méthode SEP serait garantie si certaines précautions sont prises au départ, bien qu'évidemment, il n'y aura aucun besoin de faire une exploration par l'AG dans ce cas-ci.

En général, l'optimalité d'une approche SEP dépend de la complétude de son procédé de séparation. En d'autres termes, l'énumération couvre-t-elle toutes les solutions possibles ? Dans notre approche proposée, une réponse affirmative à cette question dépend de deux choses : la représentativité du codage binaire et la composition de la *liste de co-cycles*.

7.3.4.1 Représentativité du codage binaire

Le [paragraphe 7.2.1.2](#) a montré que quelques compositions ne pouvaient être représentées en binaire parce que le codage ne peut pas regrouper deux machines dans une même cellule si elles ne sont pas liées par une chaîne intracellulaire. Un cas spécial a été alors présenté exhibant la situation où une solution non représentée était optimale ! Du moment que notre méthode SEP énumère les solutions en se basant sur cette même représentation binaire, elle ne peut pas couvrir la totalité des solutions possibles. Cependant, une modification du graphe peut assurer la représentativité totale. En effet, si des arêtes fictives de poids nuls sont ajoutées au graphe de flux jusqu'à ce qu'il soit complet, toutes les solutions possibles peuvent alors être représentées par le codage binaire. Par exemple, en utilisant un graphe étendu complet, la solution optimale absente dans la figure 7.4 peut être représentée en mettant à zéro les allèles correspondant aux quatre arêtes intercellulaires, et à un les allèles des arêtes intracellulaires liant M1 à M2 et M3 à M4. Notez que si le nombre maximum des cellules n'est pas borné par J_{max} ², cette modification n'est plus nécessaire puisque pour chaque solution non représentée, il y a une solution codée en binaire avec une valeur de fonction objectif équivalente.

7.3.4.2 La composition de la liste de co-cycles

Bien qu'il semble à priori que la liste de co-cycles L doit contenir la totalité des co-cycles pour assurer la complétude de l'énumération, L peut être réduite de façon significative tout en maintenant la complétude. D'abord, les co-cycles non élémentaires sont inutiles puisque tout co-cycle non élémentaire peut être représenté par une somme de co-cycles élémentaires. En outre, L peut être raffinée en se servant de l'information fournie par certaines

² Le nombre de cellules est évidemment toujours borné par m .

contraintes. Par exemple, nous pouvons ne considérer que les co-cycles dont la cardinalité de l'un des deux sous-ensembles de sommets associés³ est inférieure au nombre maximum de machines permis dans une cellule. Cette réduction n'altère pas la complétude de l'énumération puisque chaque solution réalisable peut être représentée par une somme de tels co-cycles si le graphe de flux est étendu. En outre, il n'y a aucun besoin de considérer des co-cycles avec une arête reliant deux machines non cohabitantes. En effet, toute solution qui résulte d'une combinaison utilisant un tel co-cycle est non réalisable.

7.3.5 Exploration génétique

Après avoir atteint une certaine profondeur dans l'arborescence de la méthode SEP, les meilleures branches sont choisies pour une exploration des sous-ensembles associés en utilisant l'AG binaire. L'exploration peut être restreinte à un sous-ensemble d'une branche donnée en fixant à 1 certains allèles des chromosomes de l'AG. Ceci est réalisé en ajoutant une contrainte de non-cohabitation pour chaque couple de machines reliées par une arête intercellulaire dans la solution basique de la branche en question. Notons que la taille du code génétique est ainsi réduite parce que les allèles fixes sont enlevés de la chaîne génétique (voir *remarque 7.1*).

Illustrons l'étape d'exploration génétique par un exemple. Considérons le graphe de flux représenté dans la figure 7.1. Supposons que L est défini par les co-cycles $w_1 = (0,0,1,0,0,0,1,0,1)$ et $w_2 = (0,1,1,1,1,0,0,0,0)$. L'exploration SEP génère une arborescence ayant quatre feuilles dont les solutions basiques associées sont définies comme suit :

- $S_{00} = (0,0,0,0,0,0,0,0,0)$ est atteint en empruntant le chemin défini par $x_1 = 0$ et $x_2 = 0$ (aucun co-cycle n'est utilisé).
- $S_{01} = (0,1,1,1,1,0,0,0,0)$ est atteint en empruntant le chemin défini par $x_1 = 0$ et $x_2 = 1$ (w_2 est utilisé mais pas w_1).
- $S_{10} = (0,0,1,0,0,0,1,0,1)$ est atteint en empruntant le chemin défini par $x_1 = 1$ et $x_2 = 0$ (w_1 est utilisé mais pas w_2).
- $S_{11} = (0,1,1,1,1,0,1,0,1)$ est atteint en empruntant le chemin défini par $x_1 = 1$ et $x_2 = 1$ (les deux co-cycles sont utilisés).

Si le deuxième sous-ensemble S_{01} est choisi pour l'exploration génétique, quatre contraintes de non cohabitation sont ajoutées, définies par l'ensemble $E_{ncoh} = \{(M_1, M_5), (M_1, M_6), (M_2, M_3), (M_2, M_5)\}$. Si le troisième sous-ensemble est choisi, l'ensemble de non cohabitation est défini par $E_{ncoh} = \{(M_1, M_6), (M_3, M_6), (M_5, M_6)\}$, et ainsi de suite. Notons que le choix du premier sous-ensemble pour une exploration génétique revient à exécuter l'AG sans aucune restriction dans l'espace de recherche.

Enfin de compte, le nombre de sous-ensembles devant subir l'exploration génétique dépend de la taille du problème et est de ce fait déterminé de manière empirique.

7.4 EVALUATION DES PERFORMANCES

Dans ce paragraphe, nous présentons les résultats des méthodes proposées appliquées sur des exemples tirés de la littérature. Une comparaison directe des performances de ces méthodes avec celles proposées dans la littérature s'est avérée impossible pour plusieurs raisons : ces méthodes utilisent des fonctions objectif différentes ou des informations supplémentaires parfois non justifiées (telles que le nombre préétabli de cellules) ou omettent de tenir

³ Un co-cycle w est associé à deux sous-ensemble complémentaires de sommets. Chaque arête de ce co-cycle a une de ses extrémités dans le premier sous-ensemble et la deuxième dans le second.

compte de certaines informations utiles (telles que les temps de fabrication, les machines à, ou à ne pas, cohabiter). Cependant, pour illustrer les avantages des méthodes proposées, nous avons implémenté l'AG en utilisant le codage entier classique, notre codage binaire et la méthode hybride. Ces méthodes seront notées AGT, AGB et SEPAG respectivement. Les trois applications ont été implémentées sur un PC Cyrix MII 300 à 233 MHz et 32 Mo de RAM. Nous avons utilisé un compilateur C++. Les paramètres (voir [paragraphe 7.2.5](#)) par défaut utilisés dans AGT et GAB sont : $P_3=90\%$, $P_4=80\%$ et $P_5=01\%$. SEPAG utilise une stratégie d'exploration "largeur d'abords" [Gondran1979]. Ensuite, l'AG explore les branches ayant les meilleures évaluations en utilisant les mêmes opérateurs par défaut que ceux de AGT et GAB.

7.4.1 Les données des exemples

Nous avons choisi de la littérature trois exemples de différentes tailles parmi les travaux de recherche qui tiennent compte des routages. Les exemples sont triés suivant leur taille supposée égale au produit $m \times p$ (nombre de machines \times nombre de produits). Il est bien de noter qu'aucun des travaux sélectionnés ne tient compte des contraintes de cohabitation. Par conséquent, toutes les informations les concernant ont été ajoutées. De même, nous avons ajouté les informations concernant les temps de fabrications pour les travaux qui ne tiennent pas compte de ce genre de données, à savoir [Jayakrishna1998]. Nous supposerons, en outre, que le décideur estime que la valeur 10% est le seuil tolérable pour le critère d'équilibrage de charge.

Le premier exemple est tiré de [Jayakrishna1998]. Il s'agit d'un atelier de 20 produits et 8 machines. La table 7.1 illustre la matrice d'incidence produit/machine avec les informations des séquences d'opérations. Le nombre maximum de machines par cellule est égal à 5.

	Machines								Ratios
	1	2	3	4	5	6	7	8	
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

Dans le deuxième exemple, tiré de [Su1998], il s'agit d'un atelier de 35 produits et 18 machines ($p=35$ et $m=18$). 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 ($N=6$). Les routages des produits sont mentionnés dans le [tableau 7.2](#).

Table 7.2 Matrice d'incidence de l'exemple 2

	Machines										Ratios									
	1	2	3	4	5	6	7	8	9	0	1	1	1	1	1	1	1	1	1	
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	.	2	.	1	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
30	2	.	.	4	1	.	.	3	.	5	1
31	5	.	.	3	2	.	.	1	.	4	1
32	.	.	3	5	.	2,4	.	.	.	6	.	1	.	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

Le troisième exemple est dû à [Nagi1990]. Il concerne 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 la [table 7.3](#).

Table 7.3. Matrice d'incidence de l'exemple 3

	Machines										Ratios									
	1	2	3	4	5	6	7	8	9	0	1	1	1	1	1	1	1	1	1	2
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	.	.	5
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	

7.4.2 Analyse des résultats

Les résultats de l'application des trois méthodes sur les trois exemples précédents sont portés sur la [table 7.4](#). Les trois méthodes ont été exécutées plusieurs fois pour chaque exemple à la quête des valeurs de paramètres qui rendent les résultats stables. Nous convenons qu'un résultat est stable si ses meilleures performances sont toujours réalisées dans au moins deux parmi cinq exécutions. La meilleure solution est alors indiquée avec son propre temps d'exécution.

Table 7.4 Résultats des exemples

Exemple N°	AGT			AGB			SEPAG		
	trafic	charge	cpu (sec.)	trafic	charge	cpu (sec.)	trafic	charge	cpu (sec.)
1	0.317	0.002	11.21	0.317	0.002	22.85	0.317	0.002	30.87
2	0.557	0.061	134.65	0.407	0.052	237.54	0.414	0.055	320.31
3	0.496	0.072	46.91	0.310	0.051	96.84	0.303	0.051	305.54

Dans le premier exemple, les trois méthodes donnent la même solution avec un trafic intercellulaire normalisé égal à 0.317. A titre indicatif, le trafic intercellulaire de la solution trouvée par [Jayakrishna1998] est 0.415. Cependant, SEPAG a atteint cette performance dans la phase d'exploration SEP et il n'était pas utile de lancer l'AG.

Dans l'exemple 2, il n'était pas possible de comparer notre solution avec celle des auteurs de l'exemple car ils considèrent la duplication de machines. Pour le dernier exemple, une solution différente de celle trouvée par les auteurs de l'exemple a été atteinte par la méthode SEPAG mais les deux ont le même trafic intercellulaire égal à 0.303 (voir la [table 7.5](#) pour les détails des solutions trouvées par SEPAG).

Table 7.5 Les meilleures compositions de SEPAG

Exemple N°	Composition de cellules
1	{1,3,5,6} {2,4,7,8}
2	{1,5,11,15} {2,4,7,17,13,14} {3,8,9,10,16,18} {6,12}
3	{1,9,12} {2,5,6,7,16} {3,8,11,19} {4,13,15} {10,14,17,18,20}

La [table 7.6](#) donne certains paramètres utilisés par AGT et AGB. Ces paramètres montrent que l'AG binaire requiert moins de ressources (en ce qui concerne la taille de la population et le nombre d'itérations) pour stabiliser ses meilleures performances. Cette économie est due aux qualités de l'AG binaire qui permettent de faciliter la détection des *schèmes* prometteurs enfouis dans le code génétique des solutions. A cause de la redondance et de la nature du codage entier à compacter l'information, la détection de ces schèmes devient plus ardue pour l'AG

classique. Ce dernier essaye de pallier à ces insuffisances à travers une taille de population plus grande et/ou un nombre d'itérations plus élevé. Dans la méthode SEPAG, l'apport de l'exploration SEP a permis de réduire la taille de la population et le nombre de générations de l'AG binaire à 20 et 100 respectivement.

Table 7.6 Valeurs de certains paramètres génétiques

Exemple	GAI		GAB	
	Population	Générations	Population	Générations
1	200	500	100	500
2	250	2000	200	2000
3	250	3000	250	2000

Le résultat identique fourni par les trois méthodes pour le premier exemple est dû à la taille relativement petite de l'instance considérée. Dans ces circonstances, SEPAG peut effectuer une exploration cocyclique couvrant la totalité des co-cycles. Ceci a permis d'affirmer que la solution trouvée par les trois méthodes est une solution optimale. Lorsque la taille des exemples croît, le mauvais effet de la redondance de l'AG classique commence à ce manifester. L'élargissement de la taille de la population et l'augmentation du nombre d'itérations se sont avérés infructueux.

En tout, SEPAG donne les meilleures performances en ce qui concerne le trafic intercellulaire. Ce qui permet de montrer l'avantage de l'amélioration procurée à l'AG par l'exploration SEP. Comme a permis de le montrer la [table 7.4](#), SEPAG réalise les mauvaises performances en ce qui concerne le temps de calcul. Cependant, le temps d'exécution peut être considérablement amélioré en lançant l'exploration génétique des régions prometteuses en même temps.

Ce chapitre a traité le problème de composition de cellules en supposant des routages statiques procurés par la phase antérieure de prétraitement de données. Cependant, l'influence de cette décision sur le module que nous étudions va nous pousser à revoir cette stratégie dans le chapitre suivant.

Chapitre 8

COMPOSITION DE CELLULES AVEC ROUTAGES ALTERNATIFS

La prise en compte des routages alternatifs constitue l'une des voies de recherche, relativement récentes, dans le domaine de la composition de cellules. Cet aspect considère la possibilité de fabrication des produits suivant plusieurs routages. Nous présentons, dans ce chapitre, un bref survol sur les méthodes publiées, suivi d'une description approfondie du problème, puis d'une formulation mathématique traitant l'aspect évoqué. Enfin, nous concluons le chapitre par des méthodes proposées pour la résolution du problème, suivies des résultats obtenus.

8.1 MOTIVATION

Jusqu'ici notre approche au problème de composition considérait que chaque produit était associé à un unique routage ([voir chapitre 6](#)). La restriction adoptée par notre approche est justifiée par le fait que le problème de détermination des routages était réalisé dans l'étape précédente (module de prétraitement de données). Cependant, il n'est pas rare qu'un produit donné puisse avoir plusieurs routages possibles, ne serait ce que pour le fait que l'atelier puisse avoir des machines identiques ou pouvant réaliser des opérations similaires. Fusionner le problème de détermination des routages avec celui de composition peut s'avérer très bénéfique pour ce dernier. En effet, en permettant une flexibilité des routages, on pourra jouer simultanément sur les charges des machines et les flux inter-machines. Ce qui permettra de mieux concilier entre les critères retenus, à savoir le trafic intercellulaire (fonction des flux inter-machines) et l'équilibrage de charge (fonction des charges des machines).

Parmi les chercheurs qui ont opté pour la résolution de ce problème d'affectation des routages aux produits en même temps que celui de composition, nous trouvons [Atmani1995, Vin2003].

Dans ce qui suit nous décrivons le problème de façon profonde. Ensuite, nous proposons une nouvelle approche au problème en commençant par une formulation qui étend celle [du chapitre 6](#) au cas évoqué.

8.2 IDENTIFICATION DU PROBLEME

Le problème des routages alternatifs possède plusieurs facettes que nous pouvons résumer en trois cas qui consistent à considérer :

1. ***Version 1*** : Des machines identiques [Askin1990, Nagi1990, Harhalakis1994, Logendran1995, Taboun1998, Wicks1999, Zhao2000].
2. ***Version 2*** : Des machines pouvant réaliser des opérations similaires [Vin2003].

3. **Version 3** : Plusieurs gammes ou séquences opératoires par produits [Logendran1994, Atmani1995].

Le premier cas est le plus restrictif, tandis que le dernier est le plus général mais le plus difficile. En effet, dans le premier cas, on considère qu'il y'a plusieurs machines de mêmes types (le terme "identiques" aurait été beaucoup plus clair, mais nous avons opté pour le terme confus car très utilisé dans la littérature). A chaque produit est affecté une unique séquence de machines. La flexibilité des routages est restreinte donc à la possibilité d'affectation d'une opération à une autre machine de même type. Dans la deuxième version, on considère pour chaque produit une unique séquence opératoire. Chaque opération peut être réalisée sur un ensemble de machines non nécessairement équivalentes. Dans le troisième cas, on considère pour chaque produit plusieurs séquences opératoires et chaque opération peut être réalisée sur différentes machines. [Logendran1994] a opté pour cette dernière version. Mais il traite le problème d'affectation des opérations seul sans celui de composition. Il démontre que ce problème est NP-dur. Atmani a eu le courage d'aborder le problème sous sa version générale simultanément avec la composition de cellule. Il affirme "Clearly, the trend has been to solve cell formation and operation allocation as separate problems" [Atmani1995]. Il propose un programme mathématique avec variables bivalentes qu'il résout en utilisant le logiciel LINDO. Bien que le travail de Atmani soit intéressant, ne serait ce que pour être l'un des précurseurs ayant opté pour une prise sérieuse du problème, il présente quelques limites. En effet, il suppose donné le nombre de cellules. En outre, il expose l'efficacité de résolution de son modèle sur des exemples de six machines, quatre produits et deux séquences par produits au maximum. En outre, il impose un nombre de cellules très réduit : il considère deux cellules, voire une. Ce qui est une relaxation cachée du problème de composition.

En plus des travaux qui viennent d'être cités, les travaux de [Zhao2000, Vin2003] nous intéressent à la mesure qu'ils adoptent une approche évolutionnaire.

Dans [Zhao2000], les auteurs présentent une approche multicritère évolutionnaire agrégative (somme pondérée) pour le problème mixte de composition de familles et de cellules, basée sur un algorithme génétique à codage entier. Le codage génétique utilisé permet de définir l'affectation des machines aux cellules uniquement (codage entier classique), la détermination du routage se basant sur un algorithme spécifique. Ce dernier sert à déterminer le routage minimisant la somme des mouvements intercellulaires et la variance des charges cellulaires. Les inconvénients de cette approche sont la considération du cas le plus restrictif du problème des routages alternatifs, la fixation du nombre de cellules et l'utilisation du codage entier redondant.

Dans [Vin2003], les auteurs affirment : «Pour éviter les inconvénients de la résolution séquentielle des deux sous problèmes [allocation des opérations aux machines et affectation des machines aux cellules], la méthode proposée dans cet article groupe simultanément les opérations des produits sur les machines et les machines en cellule». Pour ce faire, ils utilisent une approche évolutionnaire intégrant la méthode d'aide à la décision multicritère PROMETHE II [Brans1985]. Deux algorithmes génétiques multicritères sont utilisés. Le premier, disons AG1, utilise un codage entier qui détermine l'affectation des opérations aux machines. Le deuxième, AG2, utilise un codage entier classique pour l'affectation des machines aux cellules en utilisant les meilleures solutions de la population de AG1 déterminées par la méthode PROMETHE II. Les auteurs ont le mérite d'avoir essayé d'aborder le problème sous sa deuxième version simultanément avec le problème de composition en considérant un nombre de cellules variable. Cependant, l'efficacité de la méthode doit être vérifiée car les auteurs n'ont pas jugé

utile d'incorporer dans leur travail les résultats de l'application de leur méthode sur des exemples. Ils ne présentent qu'une instance de quatre machines et une seule séquence opératoire avec pas plus de trois machines par opération. L'instance n'est pas résolue mais elle sert à illustrer le calcul des critères considérés. Une chose est sûre : L'efficacité de la méthode va se plaindre de deux problèmes majeurs. D'une part, la première partie posera des problèmes de réalisabilité car on doit vérifier si l'affectation de l'opération concerne une machine qui peut la réaliser. D'autre part, nous avons la redondance du codage entier utilisé.

Dans ce qui suit, nous nous proposons de résoudre simultanément le problème des routages alternatifs, sous sa deuxième version, et celui de la composition de cellule, en ayant dans l'esprit le souci d'efficacité. Notons que la résolution, de façon efficace, du problème de composition avec un nombre de cellules variable, en tenant compte de la troisième version reste un sujet de recherche ouvert à développer.

8.3 FORMULATION

Dans le chapitre 6, nous avons présenté une formulation du problème de composition basée sur la théorie des graphes. Cette formulation modélisait le problème de composition en un problème de partitionnement dans un graphe de flux ayant des pondérations fixes. Nous montrons, dans ce qui suit, que le même modèle peut être adapté pour tenir compte de l'aspect routages alternatifs, mais sur un graphe de flux paramétré. Vu que cette formulation est une extension de la précédente modélisation, afin d'éviter la répétition, nous nous restreignons aux points distincts de la précédente.

8.3.1 Les inputs

Nous commençons par relaxer l'hypothèse d'unicité des routages. Donc au lieu d'associer un routage unique à un produit, nous associons à chaque produit :

(1) Une séquence unique d'opérations G_k appelée *gamme opératoire*, que le produit P_k doit subir :

$G_k = (Op_{k1}, Op_{k2}, \dots, Op_{ks_k})$ où : Op_{kt} est la t-ième opération de la gamme G_k avec $t = 1, 2, \dots, s_k$ et s_k étant la longueur de la gamme G_k .

(2) On suppose donné pour chaque opération Op_{kt} :

(i) l'ensemble des machines $M_{Op_{kt}} \subseteq M$ pouvant réaliser l'opération Op_{kt} :

$$M_{Op_{kt}} = \{<M_{kt1}>, <M_{kt2}>, \dots, <M_{ktq_{kt}}>\} \text{ avec } <M_{ktu}> \in M \text{ pour } u=1, 2, \dots, q_{kt}.$$

(ii) un temps $t f_{kti}$ désignant le temps nécessaire pour la réalisation de l'opération Op_{kt} sur une machine $M_i \in M_{Op_{kt}}$ pouvant réaliser cette opération. On mettra $t f_{kti} = 0$ si $M_i \notin M_{Op_{kt}}$.

8.3.2 Définition d'une solution du problème

(3) Soit $C = \{C_1, C_2, \dots, C_J\}$ une partition de M en J cellules.

Pour définir une solution au problème de composition avec routages alternatifs, nous avons, en plus, besoin de la décision d'affectation des opérations aux machines (ou de façon duale des machines aux opérations). Cette affectation est décrite par :

(4) la variable tridimensionnelle $\alpha = (\alpha_{kti})_{k=1..p, t=1..s_k, i=1..m}$ définie comme suit :

$$\alpha_{kti} = \begin{cases} 1 & \text{si la } t\text{-ième opération du produit } P_k \text{ est réalisée sur la machine } M_i \\ 0 & \text{sinon} \end{cases}$$

8.3.3 Contraintes de réalisabilité

En plus de celles définies dans le [chapitre 6](#), trois contraintes liées à l'introduction de l'aspect multi routage dans le modèle sont données dans ce qui suit. Les deux premières contraintes veillent sur la cohérence du modèle, tandis que la dernière permet de tenir compte de considérations pratiques liées au fait que sous le nouvel aspect, les charges des machines deviennent variables.

(5) Une opération doit être réalisée sur une seule machine :

$$\sum_{i=1}^m \alpha_{kti} = 1$$

pour $k = 1, 2, \dots, p$ et $t = 1, 2, \dots, s_k$.

(6) La machine désignée pour accomplir une opération doit pouvoir la réaliser :

$$\sum_{i=1}^m \alpha_{kti} \cdot tf_{kti} > 0$$

pour $k = 1, 2, \dots, p$ et $t = 1, 2, \dots, s_k$.

(7) Les machines pouvant avoir des durées opératoires qui varient selon l'affectation des opérations, la capacité d'opérabilité limite des machines doit être respectée :

$$\sum_{k=1}^p \sum_{t=1}^{s_k} \alpha_{kti} \cdot r_k \cdot tf_{kti} \leq c_i,$$

pour $i=1..m$, où c_i est la charge maximale (ou capacité d'opérabilité) de la machine M_i .

8.3.4 Traitements intermédiaires

(8) Pour chaque $(P_k, M_i, M_q) \in P \times M \times M$, le nombre de fois que M_i suit M_q et inversement ($i, q = 1, \dots, m$) dans le routage de P_k , noté $v_{k,i,q}$, va dépendre de l'affectation des opérations comme suit :

$$v_{k,i,q} = \sum_{t=1}^{s_k-1} ((\alpha_{kti} \cdot \alpha_{k,t+1,q}) + (\alpha_{ktq} \cdot \alpha_{k,t+1,i}))$$

(9) On définit alors 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 pouvant être reliées par un flux positif :

$$E = \{e_{i,q} / (M_i, M_q) \in M \times M, i, q = 1, \dots, m ; i \neq q \text{ et } (\exists k \in \{1, \dots, p\}, \exists t \in \{1, \dots, s_k\} \text{ tels que } M_i \in MOp_{kt} \text{ et } M_q \in MOp_{k,t+1})\}$$

Sur le graphe de flux G, on définit une pondération des sommets et des arêtes qui dépendra de l'affectation des machines aux opérations comme suit :

(10) La fonction de pondération des sommets, V , est donc définie comme suit :

$$V(M_i) = \sum_{k=1}^p \sum_{t=1}^{s_k} \alpha_{kti} \cdot r_k \cdot tf_{kti}$$

où $i = 1, \dots, m$.

(11) La fonction de pondération des arêtes, W , est d'autre part définie comme suit :

$$W(e_{iq}) = \sum_{k=1}^p r_k \cdot v_{kiq} \quad \forall e_{iq} \in E,$$

i et $q \in \{1, \dots, m\}$.

(12) On définit $E(C)$ comme étant le sous-ensemble des arêtes intercellulaires associés à la partition C :

$$E(C) = \{ e_{iq} \in E / (M_i, M_q) \in C_j \times C_g ; j, g = 1, 2, \dots, J ; g \neq j \}$$

8.3.5 Fonction multi objectifs

Les fonctions objectif sont calculées à la base des fonctions de pondération des sommets et des arêtes de façon analogue à celle décrite dans le paragraphe 6.4, et sont donc les suivantes :

(13) Le trafic intercellulaire de la composition C , selon l'affectation α , est donné par la somme des poids des arêtes intercellulaires :

$$T(C, \alpha) = \sum_{e_{kl} \in E(C)} W(e_{kl})$$

(14) La variance des charges des cellules d'une composition C donnée est donc :

$$CH(C, \alpha) = \sum_{j=1}^J (Ch_j - \overline{Ch})^2 / J$$

où Ch_j est donnée par $Ch_j = \sum_{M_i \in C_j} V(M_i)$ et \overline{Ch} est la moyenne des Ch_j pour $j=1..J$.

Soit S l'ensemble des partitions C qui respectent les contraintes précédentes y afférentes, et soit A l'ensemble de toutes les affectations α respectant les contraintes associées (5,6,7). Le problème est de trouver l'ensemble des solutions $(C, \alpha)^*$ optimales au sens Pareto :

$$\underline{Z}((C, \alpha)^*) = \min_{C \in S, \alpha \in A} (T(C, \alpha), CH(C, \alpha))$$

C'est-à-dire une partition des machines et une affectation des opérations réalisant un meilleur compromis entre trafic et équilibrage de charge intercellulaires.

8.3.6 Exemple illustratif

L'exemple suivant permet d'illustrer une instance du problème qui sera utilisée au fil du chapitre pour expliquer le procédé des méthodes de résolution proposées. Soit un atelier de 5 machines et 4 produits. Les gammes opératoires, les temps d'exécution des opérations (supposés indépendants des machines pour alléger la présentation) et le ratio moyen de fabrication de chaque produit sont définis dans la table 8.1.

Table 8.1 Informations sur les produits

Numéro produit	Séquence d'opérations				Temps opératoires	Ratio
	Op1	Op2	Op3	Op4		
1	M ₂	M ₅	M ₃ , M ₄	M ₅	2,1,3,1	2
2	M ₁	M ₃ , M ₄	M ₁	M ₅	1,3,2,5	3
3	M ₃ , M ₄	M ₁	M ₃ , M ₄	M ₅	4,1,3,2	1
4	M ₃ , M ₄	M ₂	M ₃ , M ₄	-	1,3,3	2

Le graphe de flux paramétré associé à cet exemple est présenté dans la figure 8.1.

Les sommets de ce graphe sont les cinq machines de l'atelier. Deux machines sont reliées par une arête si et

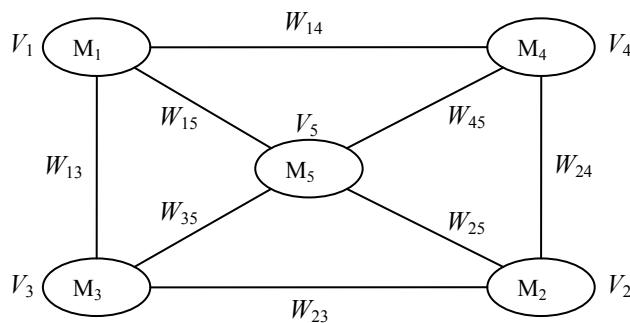


Figure 8.1 Graphe de flux paramétré.

seulement si il existe deux opérations consécutives d'un produit pour lesquelles une machine peut réaliser la première opération et l'autre machine la deuxième. A titre d'exemple, l'arête (M₂,M₃) est portée sur le graphe car il existe un produit, P₄, dont la première opération peut être réalisée par M₃ et la deuxième par M₂. Par contre, M₁ et M₂ ne sont pas reliées car il n'y a pas d'opérations consécutives pouvant être réalisées par ces deux machines.

Les poids des sommets $V(M_i)$, que nous avons notés V_i , sont définis par la formule (10) (du paragraphe 8.3.4) qui donne :

$$V(M_1) = 10.$$

$$V(M_2) = 10.$$

$$V(M_3) = 6 \alpha_{133} + 9 \alpha_{223} + 4 \alpha_{313} + 3 \alpha_{333} + 2 \alpha_{413} + 2 \alpha_{433}. \quad V(M_4) = 6 \alpha_{134} + 9 \alpha_{224} + 4 \alpha_{314} + 3 \alpha_{334} + 2 \alpha_{414} + 2 \alpha_{434}.$$

$$V(M_5) = 9.$$

D'autre part, les poids des arêtes $W(M_i, M_q)$, que nous avons notés W_{iq} , sont définis par la formule (11) (du paragraphe 8.3.4) qui donne :

$$W(M_4, M_5) = 4 \alpha_{143} + \alpha_{343}$$

$$W(M_2, M_4) = 2 \alpha_{441} + 2 \alpha_{443}$$

$$W(M_2, M_3) = 2 \alpha_{431} + 2 \alpha_{433}$$

$$W(M_1, M_3) = 6 \alpha_{232} + \alpha_{331} + \alpha_{333}$$

$$W(M_3, M_5) = 2 \alpha_{133} + \alpha_{333}$$

$$W(M_1, M_4) = 6 \alpha_{242} + \alpha_{343}$$

$$W(M_1, M_5) = 3$$

$$W(M_2, M_5) = 2$$

Vu que les valeurs des pondérations comportent des inconnus, nous convenons, dans ce qui suit, d'appeler un tel graphe par *graphe de flux paramétré*.

8.4 METHODES DE RESOLUTION PROPOSEES

Dans une première étape, nous proposons une approche basée sur les AGs. Puis, afin d'améliorer l'efficacité de celui-ci, nous proposons une hybridation entre l'AG et la méthode Tabou. Ces deux méthodes ont des principes différents mais des propriétés complémentaires: L'AG tend à réaliser une exploration dispersée de l'espace de recherche et Tabou est une méthode de recherche locale. Le but de l'hybridation est de marier leurs avantages pour que chacun puisse pallier aux inconvénients de l'autre.

8.4.1 Résolution par l'AG

Bien que l'AG ait été utilisé pour la résolution du problème de composition en tenant compte de l'aspect multi routage par Zhao et Wu [Zhao2000], ces dernier n'incorporent pas l'aspect évoqué directement dans l'AG. En effet, le codage proposé par les auteurs ne tient compte que de l'affectation des machines aux cellules et non pas des opérations aux machines. L'affectation des machines aux cellules étant définie par le codage génétique, les auteurs implémentent dans leur AG une heuristique basée sur un algorithme de plus court chemin qui permet de définir l'affectation optimale des opérations (par rapport à un seul critère) qui dépend, bien sûr, de la composition de cellules définie par le code génétique. Par conséquent, l'espace de recherche exploré n'englobe pas toutes les solutions possibles et une telle cécité peut compromettre la recherche des meilleures solutions.

Ayant peut être senti le bien-fondé d'étaler l'espace de recherche tout entier à l'AG, [Vin2003] ont opté pour une codification en deux parties : une partie pour l'affectation des opérations et l'autre pour les machines. Cependant, ils utilisent le codage entier classique, source de tous les maux.

Etant donné ce constat, nous proposons, dans ce qui suit, une nouvelle codification qui se veut palliative aux problèmes évoqués.

8.4.1.1 Codage

Le codage que nous proposons est constitué de deux parties. La première partie est bivalente de longueur $|E|$ (nombre d'arêtes du graphe de flux) et est semblable à celle décrite dans le [paragraphe 7.2.1](#). Un allèle est donc associé à une arête et indique si elle est *intra* ou *intercellulaire*. La deuxième partie est entière de longueur égale au nombre d'opérations pouvant être réalisées sur plusieurs machines. De façon plus formelle, la longueur de la deuxième partie est égale à $nop = \sum_{k=1}^p \sum_{t=1}^{s_k} f(MOp_{kt})$ où f est une fonction retournant un si le cardinal de l'ensemble qu'elle admet comme argument est supérieur strictement à un, elle est nulle sinon. Un allèle de cette partie est associé à une opération et indique le rang de la machine désignée pour sa réalisation. Cette représentation indirecte permet d'éviter le problème de non réalisabilité de l'affectation des opérations évoqué dans le [paragraphe 8.2](#).

Exemple

Pour l'exemple précédent, considérons la solution définie par la composition suivante de cellules : $C_1 = \{M_1, M_3, M_5\}$ et $C_2 = \{M_2, M_4\}$. Pour l'affectation des opérations, on suppose que les trois premières sont réalisées

par M_3 et les trois dernières par M_4 (l'ordre de référence est celui établi en parcourant les gammes de la première vers la dernière et de gauche à droite). Cette solution est donc représentée par la chaîne génétique suivante :

e_{13}	e_{14}	e_{15}	e_{23}	e_{24}	e_{25}	e_{35}	e_{45}	Op_{13}	Op_{22}	Op_{31}	Op_{33}	Op_{41}	Op_{43}
0	1	0	1	0	1	0	1	0	0	0	1	1	1

L'interprétation de la première partie du codage est directe : les arêtes e_{14} , e_{23} , e_{25} et e_{45} sont intercellulaires et le reste est intracellulaire. Pour la deuxième partie, on prend comme valeur de l'allèle Op_{kt} le rang de la machine désignée pour réaliser cette opération dans MOp_{kt} en respectant l'ordre des indices de machines. Ainsi, pour notre exemple, les ensemble MOp_{kt} associés aux six opérations avec alternance sont tous égaux à $\{M_3, M_4\}$. Par conséquent, tout choix de M_3 est représenté par 0. Par contre, le choix de M_4 est représenté par 1.

8.4.1.2 Générateur de population initiale

La population initiale est générée de façon aléatoire sans redondance. Cependant, comme nous l'avons décrit dans le [paragraphe 7.2.2](#), la probabilité de génération d'une arête intercellulaire est prise plus grande que celle d'une arête intracellulaire afin de prévenir la construction de compositions unicellulaires.

8.4.1.3 Sélection

Le problème que nous traitant étant constraint, utiliser la fonction objectif comme seul critère de sélection n'est pas du tout adéquat car des solutions non réalisables peuvent être favorisées au détriment d'autres qui sont réalisables. Dans [le paragraphe 5.4.1](#), nous avons décrit une alternative consistant en la modification de la relation de dominance. La relation qui en résulte, à savoir la dominance contrainte, ne permettait pas de comparer des solutions non réalisables. Nous estimons qu'il serait judicieux d'étendre la troisième condition de la définition de cette relation à tout couple de solutions ayant le même nombre de contraintes violées. Cette extension permet d'utiliser la définition classique de la dominance mais au lieu de comparer les fonctions objectif, on utilise à leur place les fitness calculées par la procédure décrite dans le [paragraphe 7.2.3](#). Ceci à pour effet de réaliser un ranking à plusieurs niveaux. C'est-à-dire, au lieu d'avoir un ranking de la population toute entière, nous la classifions, d'abord, en classes de solutions ayant le même nombre de contraintes violées. On applique ensuite la procédure de ranking sur chaque classe. Ainsi, si pour la dernière classe (éventuellement les solutions violent toutes les contraintes) les rangs des individus varieront entre 1 et k, le rang des individus non dominés de la classe précédente commencera à partir de $k+1$ et ainsi de suite. Ceci permet d'avoir des rangs ordonnés suivant la réalisabilité puis la non dominance.

La valeur d'adaptation déterminée, nous pouvons utiliser une sélection par roue de loterie ou par tournoi¹.

8.4.1.4 Croisement

Nous avons opté pour un croisement pouvant avoir deux points de coupure. Un site de croisement aléatoire est défini dans la partie machines avec une probabilité Pc_1 et un autre dans la partie opérations avec une probabilité Pc_2 . Le taux d'individus devant subir cet opérateur est défini par le paramètre Pc_3 .

8.4.1.5 Mutation

¹ Avec le procédé de ranking par classes, les solutions réalisables auront les rangs les plus élevés. Ceci permettra d'utiliser ces rangs directement pour une sélection par roue de loterie. Avec la technique du tournoi, il faut utiliser la relation de dominance en supposant des fonctions à maximiser.

Pour cet opérateur, on choisit une partie parmi les deux avec une probabilité Pm_1 et $1-Pm_1$. Puis un allèle du code génétique de la partie choisie est pris aléatoirement et modifié. Dans la partie machines, une modification rend l'arête correspondant à l'allèle choisi intercellulaire après qu'elle ait été intracellulaire ou inversement. Dans l'autre partie, une modification permet de changer l'affectation de l'opération correspondant à l'allèle.

8.4.2 Résolution par hybridation AG-Tabou avec alternance implicite

8.4.2.1 Motivations

La précédente approche génétique présente un inconvénient majeur : lorsque le nombre d'opérations pouvant être réalisées sur plusieurs machines est grand, la taille du code génétique devient très importante et le temps d'exécution nécessaire à l'AG pour atteindre de bonnes solutions devient significatif.

Il serait alors judicieux de prendre en compte l'affectation des machines aux opérations mais de façon implicite. Cette idée fera l'objet de la méthode hybride AG-Tabou.



Figure 8.2 Les deux étapes de la méthode hybride AG-Tabou : a) L'AG explore l'espace de recherche.
b) Tabou explore le voisinage des solutions trouvées par l'AG.

Cette méthode consiste à résoudre le problème de composition de cellules en utilisant dans une première étape l'algorithme génétique pour fournir une affectation des machines aux cellules. Ensuite, la méthode Tabou affecte les opérations aux machines. Mais, afin que cette dernière affectation ne soit pas lésée par une mauvaise décision réalisée lors de la première affectation, cette décision doit être réalisée en tenant compte de son impact sur l'affectation des opérations.

On sait que la variable régissant cette affectation n'est autre que la variable α qui est à valeurs bivalentes (0 ou 1). En affectant à cette variable des valeurs moyennes, on définit dans le graphe, des valeurs de trafic inter machines et des valeurs d'utilisation de machines moyennes. Par ces valeurs moyennes l'AG va tenter d'optimiser les fonctions objectif. Ainsi, le code génétique est réduit à la première partie uniquement (machines).

Dans un deuxième temps, Tabou va essayer d'améliorer les meilleures solutions fournies par l'AG en déterminant le meilleur compromis entre affectation de machines et d'opérations. Le choix de la méthode Tabou est motivé par le fait qu'il nous faut, après que l'AG ait fait le gros du travail d'exploration, une méthode d'exploration locale. Tabou a été, de ce fait, choisie pour son efficacité démontrée dans bon nombre de problèmes d'optimisation.

8.4.2.2 Mise en œuvre

a) L'AG traite le graphe de flux non paramétré : Dans cette étape de la méthode hybride, l'AG, seul ou épaulé par un pré-raffinage de l'espace de recherche par la méthode SEP (voir [paragraphe 7.3](#)), fournit une décomposition

des machines en cellules en utilisant le graphe de flux non paramétré. Ce graphe est obtenu en calculant pour chaque pondération (d'arêtes ou de sommets) ses valeurs maximale et minimale. On mettra, alors, la pondération égale à leur moyenne. Sauf que pour les sommets, il faut vérifier que cette valeur moyenne n'est pas supérieure à la capacité de la machine, auquel cas il faut mettre la pondération égale à la valeur de cette capacité.

Cette manière de faire a pour effet de réduire le temps d'exécution de l'AG de façon drastique car la partie opérations du codage n'entre pas en lice et les contraintes liées à l'aspect routage alternatif sont négligées.

Exemple

Si, pour l'exemple précédent, on affecte la valeur moyenne à chaque pondération, nous aurons les poids définis sur le graphe de flux non paramétré de la figure 8.3.

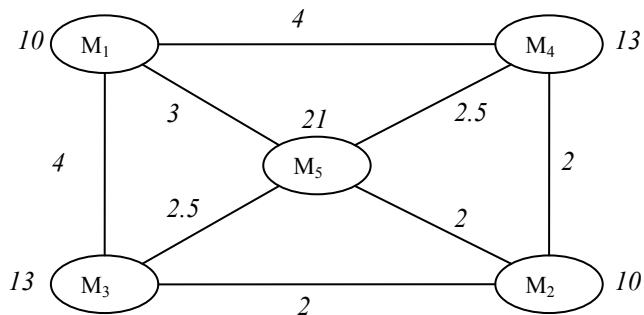


Figure 8.3 Graphe de flux non paramétré.

La mise en œuvre de l'algorithme génétique dans la première étape de l'hybridation est similaire à celle décrite au [paragraphe 8.4.1](#). Cependant, l'AG ne manipulant que la partie du code associée à la composition de cellules, il faut omettre toute manipulation portant sur la partie affectation des opérations. A la fin de cette étape, l'AG va nous fournir une affectation des machines aux cellules. Ce qui nous permet de passer à la deuxième étape qui consiste à faire appel à la méthode Tabou.

b) La deuxième étape, la méthode Tabou : Dans cette étape, la méthode Tabou prend les meilleures solutions trouvées par l'AG et affecte les opérations aux machines pour avoir une solution finale.

Dans la littérature, Tabou a été utilisée pour résoudre le problème de composition monocritère par Logendran [Logendran1995]. Il s'agit d'une approche Tabou classique point par point. Cependant, dès que l'aspect multicritère est abordé, une difficulté surgit lorsqu'une stratégie d'optimisation Pareto est utilisée. En effet, en substituant à la notion d'optimalité classique celle de Pareto basée sur la dominance, il est question de choisir, dans le voisinage générée, entre plusieurs individus incomparables (ceux qui sont non dominés). Parmi les alternatives proposées en ce sens, certaines approches telles que MOTS (Multiobjective Optimization Tabu Search) [Hansen1997] proposent de procéder par une exploration par nuage de points. Le recours à une telle approche, dans notre cas, est justifié par le fait qu'elle convient le plus à l'hybridation du moment que Tabou est sensée démarrer dès le début par le nuage de points des meilleures solutions trouvées par l'AG.

A cet effet, au lieu d'une seule solution courante, nous considérons un ensemble de solutions courantes à chaque itération. Initialement, cet ensemble est constitué des meilleures solutions trouvées par l'AG avec une

extension de code aléatoire ou déterminé par une heuristique pour l'affectation des opérations ([voir paragraphe 8.4.2.2.b.2](#)). Une archive est, en plus, utilisée pour sauvegarder toutes les solutions optimales rencontrées. A chaque itération, le voisinage de chaque solution courante est exploré à la quête d'une solution dominant cette dernière. Les nouvelles solutions courantes sont ensuite comparées à celles de l'archive pour la mettre à jour. La mise à jour de l'archive est réalisée en respectant un critère de diversification des solutions et une contrainte de borne supérieure de la taille de l'archive qui tient compte du fait que le critère de trafic intercellulaire est plus important que l'équilibrage des charges entre les cellules.

La mise en œuvre de la méthode Tabou nécessite la spécification des éléments suivants :

- le codage utilisé,
- les solutions initiales,
- le voisinage d'une solution,
- la liste Tabou,
- le critère d'aspiration,
- le critère d'arrêt.

b.1 Codage utilisé

Une solution au problème est représentée par une chaîne de taille égale au nombre de machines plus celui des opérations pouvant être réalisés sur différentes machines. En d'autres termes, pour la partie composition de cellule, on utilise un codage entier classique. Le choix du codage entier classique pour l'affectation des machines au lieu du binaire est justifié par :

- la quête d'efficacité dans l'implémentation des transformations locales à réaliser pour définir le voisinage ([voir paragraphe b.4](#)),
- le fait que le plus gros du travail d'exploration des compositions a été réalisé par l'AG binaire,
- l'atténuation significative de l'effet de la redondance en veillant au respect des contraintes sur les indices de cellules ([voir paragraphe 7.1](#)) après chaque transformation de type T2, T3 et T4 ([voir paragraphe 8.4.2.2.b.3](#)).

b.2 Solutions initiales

Nous avons déjà mentionné que Tabou va être lancée en utilisant les meilleures solutions de la dernière génération de l'AG. Cependant, vu que celui-ci utilise un codage binaire et une prise en compte de l'alternance de façon implicite, il est nécessaire de transformer le code binaire en son équivalent en entier et de lui ajouter l'extension qui permet de définir l'affectation des opérations. Cette extension peut être réalisée par affectation des machines aux opérations de façon aléatoire ou heuristique.

b.3 Définition du voisinage

Le voisinage d'une solution est généré en utilisant aléatoirement pendant un certain nombre d'itérations (couvrant la totalité ou une partie de la taille du voisinage), essentiellement la transformation suivante :

- T1 : Changement d'affectation d'une opération donnée.

Cette unique transformation peut s'avérer insuffisante car l'exploration génétique est réalisée en se fiant à des valeurs moyennes de pondérations des sommets qui peuvent être non exactes. En effet, si ces valeurs moyennes dépassent la capacité maximale, le surplus est ignoré. Par ce fait, des erreurs de déduction peuvent conduire l'AG vers des compositions de cellules qui peuvent sensiblement perdre leurs performances après extension du code. Ce constat nous conduit à donner à Tabou la possibilité de corriger ces éventuelles erreurs en effectuant des perturbations locales sur les compositions avec de faibles probabilités. Ceci revient à étendre les transformations permises aux trois suivantes :

- T2 : Transfert d'une machine de sa cellule vers une autre cellule.
- T3 : Permutation entre deux machines appartenant à des cellules différentes.
- T4 : Transfert d'une machine vers une cellule nouvellement créée.

Ceci donne, en tout, quatre transformations, la probabilité de la première étant la plus grande comparée aux autres qui sont proches les unes des autres. La grandeur de la première dépend de l'ampleur des erreurs de déduction estimées par la somme des surplus ignorés.

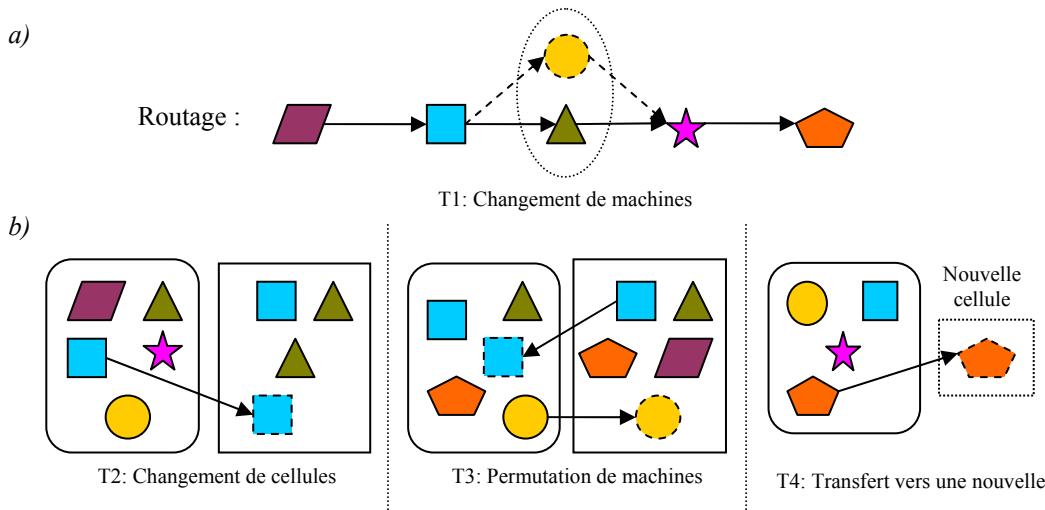


Figure 8.4 Transformations élémentaires : a) perturbations de l'affectation des opérations
b) perturbation des compositions

b.4 Liste Tabou

Pour chaque solution courante, on utilise trois listes taboues attributives. La première, qui concerne le changement de la machine réalisant une opération donnée, permet de sauvegarder l'opération choisie pour subir cette transformation. La deuxième, associée au transfert d'une machine à une autre cellule existante ou nouvellement créée, sauvegarde l'indice de la machine et de la cellule vers laquelle elle a été transférée. La dernière liste, associée à la permutation de machines, sauvegarde les indices des machines permutées.

b.5 Critère d'aspiration

Au fur et à mesure que Tabou explore les solutions de l'espace de recherche, une archive des solution non dominées est mise à jour. Chaque fois qu'une transformation générée s'avère être tabou, on vérifie si celle-ci domine au moins une des solution de l'archive. Si tel est le cas, cette solution est aspirée.

b.6 Critère d'arrêt

Deux critères d'arrêt sont considérés. Dans le premier critère, on se donne un nombre maximum d'itérations. Dans le second critère, on considère un nombre maximum d'itérations sans modification de l'archive. Si l'un des critères est rempli, l'algorithme prend fin.

8.5 PERFORMANCES DES METHODES PROPOSEES

Afin d'évaluer les performances des approches proposées nous avons implémenté trois méthodes : La méthode SPEA II, l'AG binaire avec prise en compte de l'alternance de façon explicite, et la méthode hybrideant l'AG basé sur l'alternance implicite avec la recherche locale Tabou. Nous les noterons respectivement **SPEA**, **AGS** et **AGT**. Pour la première, nous avons été obligés de modifier la fitness pour tenir compte du fait que le problème de composition est contraint. A cet effet, si une solution est réalisable, sa force (voir paragraphe 5.7) est calculée à la base des solutions dominées réalisables, sinon elle est sévèrement pénalisée. Par contre, le principe de diversification de SPEA, fidèle à l'esprit de l'optimisation Pareto pure, a été implémenté tel que décrit par son auteur [Zitzler2001]. Nous avons utilisé le même matériel ayant servi pour implémenter les méthodes du chapitre précédent.

Nous avons choisi, parmi les différents exemples sur lesquels nous avons effectué nos tests, un échantillon de trois exemples. Le premier exemple est celui qui nous a servi de moyen pour décrire la modélisation proposée au début de ce chapitre (voir [paragraphe 8.2.6](#)). Les deux autres exemples sont tirés de [Mungwatanna2000]².

8.5.1 Les données des exemples

Nous illustrons dans le listing suivant les données concernant les trois exemples.

```
*****
***** Exemple 1 *****
***** Les données *****
*****
Le nombre de machines :5
Le nombre de produits :4

• Les gammes de produits3
Produit 1:2-5-3/4-5
Produit 2:1-3/4-1-5
Produit 3:3/4-1-3/4-5
Produit 4:3/4-2-3/4

• Temps opératoires:
Produit 1:2.00-1.00-3.00-1.00 Ratio:2.00
Produit 2:1.00-3.00-2.00-5.00 Ratio:3.00
Produit 3:4.00-1.00-3.00-2.00 Ratio:1.00
Produit 4:1.00-3.00-1.00 Ratio:2.00
*****
***** Les Contraintes *****
*****
Le nombre maximum de machines par cellule: 3
Le nombre maximum de cellules: 5
```

² Nous étions obligés de modifier certaines valeurs incohérentes liées probablement à des fautes de frappe. Ces modifications ont concerné les charges maximales de l'exemple 2 qui rendaient l'espace de recherche vide, ainsi que le temps de fabrication de la cinquième opération du deuxième produit que l'auteur a oublié de porter sur son manuscrit.

³ Les machines sont représentées par leurs indices. Un slash (/) sépare les machines pouvant réaliser une même opération.

Le nombre minimum de cellules: 1
 Une contrainte de non cohabitation entre les machines: M3 et M4
 Les contraintes de charge : M1:10.00 M2:10.00 M3:20.00 M4:20.00
 M5:25.00

 ***** Exemple 2 *****
 ***** Les données *****

Le nombre de machines :9
 Le nombre de produits :6

- Les gammes de produits

 Produit 1:5/6-5/9-3/7-4/6
 Produit 2:7/9-3/4
 Produit 3:2/7-2/7
 Produit 4:4/9-1/9-6/7
 Produit 5:1/3-1/4-2/6-5/8
 Produit 6:2/3-3/8-4/6

- Temps opératoires :

Produit 1:1.23-4.13-4.69-3.65	Ratio:0.94
Produit 2:1.23-3.55	Ratio:0.90
Produit 3:3.85-2.90	Ratio:1.07
Produit 4:3.35-3.07-2.68	Ratio:0.80
Produit 5:4.78-14.10-2.14-3.11	Ratio:0.91
Produit 6:1.91-3.81-1.95	Ratio:0.84

 ***** Les Contraintes *****

Le nombre maximum de machines par cellule: 3
 Le nombre maximum de cellules : 9
 Le nombre minimum de cellules : 1
 Une contrainte de non cohabitation entre les machines : M2 et M7
 Les contraintes de charge : M1:15.00 M2:7.50 M3:15.00
 M4:20.50 M5:7.50 M6:7.50 M7:10.50 M8:6.50 M9:7.50

 ***** Exemple 3 *****
 ***** Les données *****

Le nombre de machines :10
 Le nombre de produits :12

- Les gammes de produits

Produit 1:2/3-9-7/8	
Produit 2:1/9-7-6/7-5/9-10	
Produit 3:10-3/8-1/2	
Produit 4:7-3/7-9/10-1/5	
Produit 5:10-1-3-7	
Produit 6:1-5/7-4-5/8-8	
Produit 7:7-6/10-9-4/10-2	
Produit 8:5/8-3-9/10	
Produit 9:5/9-5-1	
Produit 10:2-6-1/5-9/10	
Produit 11:7-8-4	
Produit 12:9-3/10-4/7	

- Temps opératoires :

Produit 1:3.52-5.40-5.50	Ratio:0.88
Produit 2:3.53-5.44-3.03-4.85-4.18	Ratio:0.89
Produit 3:4.18-4.64-4.63	Ratio:0.81
Produit 4:3.68-2.26-3.17-3.66	Ratio:0.92
Produit 5:5.82-2.35-4.48-5.17	Ratio:0.87
Produit 6:2.25-4.13-3.20-5.39-5.12	Ratio:0.98
Produit 7:5.78-5.71-2.22-2.89-5.91	Ratio:0.93

```

Produit 8:3.58-2.67-3.61 Ratio:0.91
Produit 9:5.92-5.57-3.93 Ratio:0.94
Produit 10:5.04-5.67-2.45-3.13 Ratio:0.80
Produit 11:2.51-3.68-5.94 Ratio:0.93
Produit 12:2.42-4.52-4.33 Ratio:0.82
*****
***** Les Contraintes *****
*****
Le nombre maximum de machines par cellule: 4
Le nombre maximum de cellules : 10
Le nombre minimum de cellules : 1
Une contrainte de cohabitation entre les machines : M1 et M10
Une contrainte de non cohabitation entre les machines : M5 et M7
Les contraintes de charge : M1:83.10 M2:88.80 M3:95.80 M4:73.90
M5:94.10 M6:78.80 M7:91.90 M8:84.90 M9:100.00 M10:71.50
*****

```

Ces exemples sont triés suivant la taille qui est supposée égale au produit $p \times m \times nop$. C'est-à-dire, le nombre de produits fois le nombre de machines fois le nombre d'opérations pouvant être réalisées sur plusieurs machines. Ceci nous donne les tailles 120, 972 et 2400 pour les exemples 1,2 et 3 respectivement.

8.5.2 Les résultats obtenus

La taille relativement petite du premier exemple a permis de déterminer, par une méthode énumérative simple, l'ensemble Pareto optimal P_{true} . Ce dernier est composé des 8 solutions suivantes :

P_{true} :

Sol 0:1221110100 TRAFIC:0.136364	CHARGE:0.214079
Sol 1:12121001011 TRAFIC:0.136364	CHARGE:0.214079
Sol 2:12122100011 TRAFIC:0.181818	CHARGE:0.050123
Sol 3:12212011100 TRAFIC:0.181818	CHARGE:0.050123
Sol 4:11122100100 TRAFIC:0.272727	CHARGE:0.010916
Sol 5:11212011011 TRAFIC:0.272727	CHARGE:0.010916
Sol 6:11122101100 TRAFIC:0.318182	CHARGE:0.000223
Sol 7:11212010011 TRAFIC:0.318182	CHARGE:0.000223

L'exécution des trois méthodes sur l'exemple 1 a donné les résultats présentés dans la table 8.2.

Table 8.2 Performances des méthodes pour l'exemple 1

	AGS	SPEA	AGT
Nombre de solutions	6	3	7
Temps d'exécution	94.37	178.70	6.45
Distance Générationnelle	0	0	0
Hyperarea	0.151299	0.209370	0.151299

Les ensembles P_{known} trouvés sont comme suit :

AGS :

Sol 0: 12121001011	TRAFIC:0.136364	CHARGE:0.214079
Sol 1: 1221110100	TRAFIC:0.136364	CHARGE:0.214079
Sol 2: 12122100011	TRAFIC:0.181818	CHARGE:0.050123
Sol 3: 12212011100	TRAFIC:0.181818	CHARGE:0.050123
Sol 4: 11122100100	TRAFIC:0.272727	CHARGE:0.010916
Sol 5: 11212010011	TRAFIC:0.318182	CHARGE:0.000223

AGT :

Sol 0:1221110100	TRAFIC:0.136364	CHARGE:0.214079
Sol 1:12122100011	TRAFIC:0.181818	CHARGE:0.050123
Sol 2:12212011100	TRAFIC:0.181818	CHARGE:0.050123
Sol 3:11212011011	TRAFIC:0.272727	CHARGE:0.010916
Sol 4:11122100100	TRAFIC:0.272727	CHARGE:0.010916

Sol 5:11212010011	TRAFIG:0.318182	CHARGE:0.000223
Sol 6:11122101100	TRAFIG:0.318182	CHARGE:0.000223

SPEA :

Sol 0:12212011000	TRAFIG:0.181818	CHARGE:0.098240
Sol 1:12213011100	TRAFIG:0.454545	CHARGE:0.006906
Sol 2:12213011000	TRAFIG:0.500000	CHARGE:0.000891

Ces résultats ont été obtenus grâce au paramétrage présenté dans la table 8.3.

Table 8.3 Paramètres des méthodes pour l'exemple 1.

	AGS	SPEA	AGT
Taille population	120	100 (+ 20 archive)	40
Nombre générations	600	100	100 (+ 50 Tabou)
Taux de sélection	0.8	0.8	0.8
Taux de croisement	0.8	0.8	0.8
Taux de mutation	0.02	0.02	0.02

L'exécution des trois méthodes sur l'exemple 2 a donné les résultats illustrés par la table 8.4.

Table 8.4 Performances des méthodes pour l'exemple 2

	AGS	SPEA	AGT
Nombre de solutions	9	4	14
Temps d'exécution	122.76	124.99	29.94
Hyperarea	0.249859	0.251654	0.000463

Les ensembles Pareto trouvés sont comme suit :

AGS :

Sol 0:1122333120000111011001100	TRAFIG:0.245344	CHARGE:0.042290
sol 1:11223331200001111111001100	TRAFIG:0.245344	CHARGE:0.042290
sol 2:11223331200001111111001011	TRAFIG:0.322894	CHARGE:0.010002
sol 3:11223331200011111111001011	TRAFIG:0.409617	CHARGE:0.003842
sol 4:11232132300001111111001011	TRAFIG:0.410451	CHARGE:0.003827
sol 5:11232233100011111111001111	TRAFIG:0.481794	CHARGE:0.001308
sol 6:11223331200011111111001111	TRAFIG:0.487168	CHARGE:0.000401
sol 7:11221332300011111111001111	TRAFIG:0.503382	CHARGE:0.000300
sol 8:11231233200001111111001100	TRAFIG:0.577689	CHARGE:0.000131

AGT :

Sol 0:11232231310010111110001101	TRAFIG:0.000000	CHARGE:0.002371
sol 1:112322313100101110110001101	TRAFIG:0.000000	CHARGE:0.002371
sol 2:11232231310011110110000100	TRAFIG:0.161864	CHARGE:0.000498
sol 3:112322313000101110110000100	TRAFIG:0.161864	CHARGE:0.000498
sol 4:11232231300010111110000100	TRAFIG:0.161864	CHARGE:0.000498
sol 5:112322313000111111110000100	TRAFIG:0.161864	CHARGE:0.000498
sol 6:112322313000111110110010110	TRAFIG:0.239414	CHARGE:0.000424
sol 7:112322313100001111011001101	TRAFIG:0.319652	CHARGE:0.000028
sol 8:11232231310001110011001101	TRAFIG:0.319652	CHARGE:0.000028
sol 9:11232231300001110011001101	TRAFIG:0.319652	CHARGE:0.000028
sol10:11232231310001111011001101	TRAFIG:0.319652	CHARGE:0.000028
sol11:112322313000001110011001101	TRAFIG:0.319652	CHARGE:0.000028
sol12:112322313100001110011001101	TRAFIG:0.319652	CHARGE:0.000028
sol13:122322313100001110100001101	TRAFIG:0.329658	CHARGE:0.000009

SPEA :

Sol 0:122344324001001001100001100	TRAFIG:0.248587	CHARGE:0.006198
sol 1:122344324011001001100001100	TRAFIG:0.248587	CHARGE:0.006198
sol 2:12234432400101011100001100	TRAFIG:0.332345	CHARGE:0.005547
sol 3:122344324011011011100001100	TRAFIG:0.431761	CHARGE:0.003513

Le paramétrage utilisé est décrit dans la table 8.5.

Table 8.5 Paramètres des méthodes pour l'exemple 2

	AGS	SPEA	AGT
Taille population	120	100 (+ 20 archive)	40
Nombre générations	1100	100	100 (+ 50 Tabou)
Taux de sélection	0.8	0.8	0.8
Taux de croisement	0.8	0.8	0.8
Taux de mutation	0.02	0.02	0.02

Pour le dernier exemple, l'exécution des trois méthodes a donné les résultats présentés dans la table 8.6.

Table 8.6 Performances des méthodes pour l'exemple 3

	AGS	SPEA	AGT
Nombre de solutions	18	2	13
Temps d'exécution	302.75	889.96	36.45
Hyperarea	0.344757	0.397586	0.335223

Les ensembles Pareto trouvés sont comme suit :

AGS :

sol 0:12132213210101100011011001101011
 sol 1:12123312310101100011011001101011
 sol 2:12123312310101100011011001111011
 sol 3:12132213210101100011011001101010
 sol 4:12132213210101100011011001111000
 sol 5:12132213210101000001011001101000
 sol 6:12132213210101100011011001101000
 sol 7:12123312310101100011011001111000
 sol 8:12123312310101100011011001101000
 sol 9:12121233210011000011001001101011
 sol10:12121233210011010011001001101011
 sol11:12121233210011010111001001101000
 sol12:12121233210011010111001001101011
 sol13:12121233210111100011001001101011
 sol14:12321233210011010111001001100010
 sol15:12321233210011010011001000001100
 sol16:1212123321001101011111001001011
 sol17:12122133210011000011011001101011

TRAFIG: 0.339574 CHARGE: 0.068945
 TRAFIC: 0.365068 CHARGE: 0.062091
 TRAFIC: 0.365068 CHARGE: 0.062091
 TRAFIC: 0.367268 CHARGE: 0.051286
 TRAFIC: 0.392762 CHARGE: 0.049232
 TRAFIC: 0.392762 CHARGE: 0.049232
 TRAFIC: 0.420320 CHARGE: 0.012974
 TRAFIC: 0.420320 CHARGE: 0.012974
 TRAFIC: 0.447879 CHARGE: 0.008285
 TRAFIC: 0.447879 CHARGE: 0.006373
 TRAFIC: 0.450554 CHARGE: 0.005650
 TRAFIC: 0.475438 CHARGE: 0.000186
 TRAFIC: 0.479229 CHARGE: 0.000793
 TRAFIC: 0.511900 CHARGE: 0.000200
 TRAFIC: 0.546873 CHARGE: 0.000168

AGT :

sol 0:12321232311011001101101000101
 sol 1:12321232310011001101101001101
 sol 2:12321232310011001101101000101
 sol 3:12321232310011001111101101001101
 sol 4:12321232310011011011011000101
 sol 5:12321232310011011111101101001101
 sol 6:12321232310011001101101101001100
 sol 7:1232123231001101101101101001100
 sol 8:12321232311011001111001011100100
 sol 9:1232123231101100111101011100100
 sol10:12321232311011001111001011101100
 sol11:12321232310010010101000001100000
 sol12:12321232310010010101100001100000

TRAFIG: 0.335071 CHARGE: 0.003872
 TRAFIC: 0.364864 CHARGE: 0.000915
 TRAFIC: 0.392558 CHARGE: 0.000060
 TRAFIC: 0.392558 CHARGE: 0.000060
 TRAFIC: 0.425094 CHARGE: 0.000032
 TRAFIC: 0.425094 CHARGE: 0.000032
 TRAFIC: 0.425094 CHARGE: 0.000032
 TRAFIC: 0.574601 CHARGE: 0.000009
 TRAFIC: 0.574601 CHARGE: 0.000009

SPEA :

sol 0:1232113233110011100101110001101
 sol 1:12321132441110011100101110001101

TRAFIG: 0.388699 CHARGE: 0.047328
 TRAFIC: 0.569760 CHARGE: 0.000740

Ces résultats ont été obtenus en appliquant le paramétrage défini dans la table 8.7.

Table 8.7 Paramètres des méthodes pour l'exemple 3

	AGS	SPEA	AGT
Taille population	120	120 (+ 20 archive)	40
Nombre générations	600	1100	600 (+ 50 Tabou)
Taux de sélection	0.8	0.8	0.8
Taux de croisement	0.8	0.8	0.8
Taux de mutation	0.02	0.02	0.02

8.5.3 Analyse des résultats

Pour évaluer les performances des méthodes, deux critères sont utilisés : l'efficacité et la qualité. Pour le premier on se fiera au temps d'exécution. Pour le deuxième, on a utilisé une métrique similaire à Hyperarea [Zitzler1998]. La métrique Hyperarea décrit la grandeur de la superficie⁴ couverte par l'ensemble des solutions du front PF_{known} . Une zone A de l'espace des objectifs est couverte par une solution a si A ne peut contenir que des solutions dominées par a . Du moment que les fonctions objectif sont normalisées de telle sorte que leurs valeurs sont toujours comprises dans l'intervalle $[0,1]$, la plus grande superficie de l'espace des objectifs ne peut dépasser 1. A cet effet, pour avoir une métrique qui décroît chaque fois que le front est meilleur, nous avons pris comme métrique la superficie de l'espace non couvert. C'est-à-dire, 1 moins la valeur donnée par Hyperarea. Cette transformation est motivée par le souhait de rendre la mesure de la qualité en concordance avec celle de l'efficacité : une diminution décrit une performance accrue. Lorsque la taille de l'exemple le permet (c'est notamment le cas de l'exemple 1), on utilisera, en plus, la métrique de la distance générationnelle [VanVeldhuizen1998] qui nécessite la connaissance de PF_{true} .

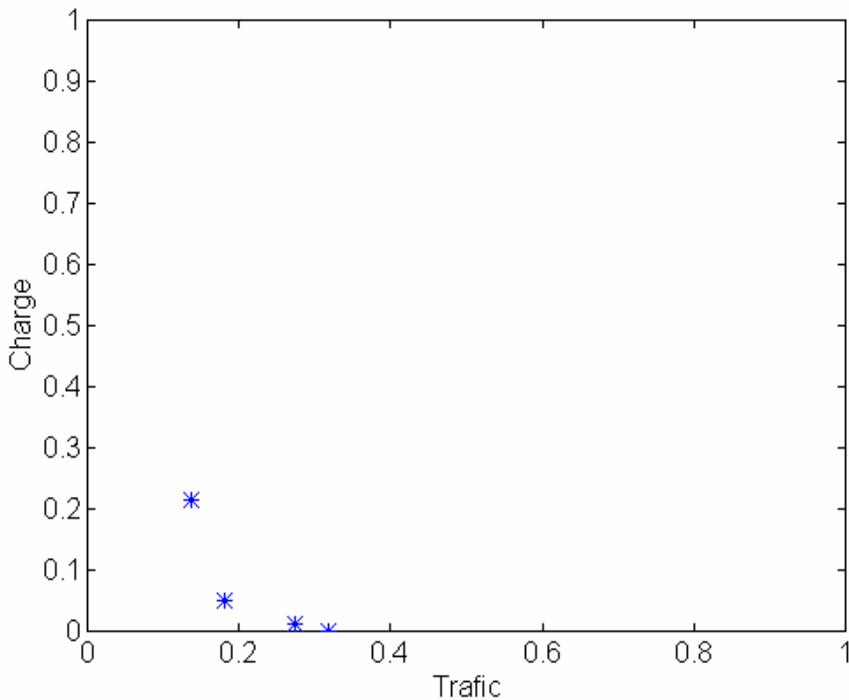


Figure 8.5 Le vrai front Pareto de l'exemple 1.

Les résultats qui viennent d'être exposés montrent que la méthode AGT donne, dans l'ensemble, les meilleures performances, suivie par AGS, puis par SPEA. En effet, dans le premier exemple, bien qu'aucune méthode n'ait été en mesure de déterminer la totalité de l'ensemble Pareto, elles sont toutes parvenues à des solutions Pareto optimales comme permet de le montrer la valeur nulle de la distance générationnelle (voir table 8.2). Cependant, seule AGT a pu déterminer la totalité du front ce qui s'est traduit par son obtention de la plus petite valeur possible de la métrique Hyperarea (voir figure 8.5). Pour ce qui est du temps de calcul, c'est AGT qui affiche le meilleur résultat. Le paramétrage utilisé pour chaque méthode (voir table 8.3) révèle que AGT a pu déterminer le vrai front avec le moins d'effort en ce qui concerne le nombre d'itérations et la taille de la population.

⁴ Ceci en bidimensionnel. Il s'agira de volume puis d'hyper volume pour des dimensions plus importantes.

Pour le deuxième exemple, AGS donne des résultats un petit peu meilleures que ceux de SPEA, comme permet de le constater la valeur de Hyperarea. Cependant, les solutions trouvées par AGT sont nettement meilleures que

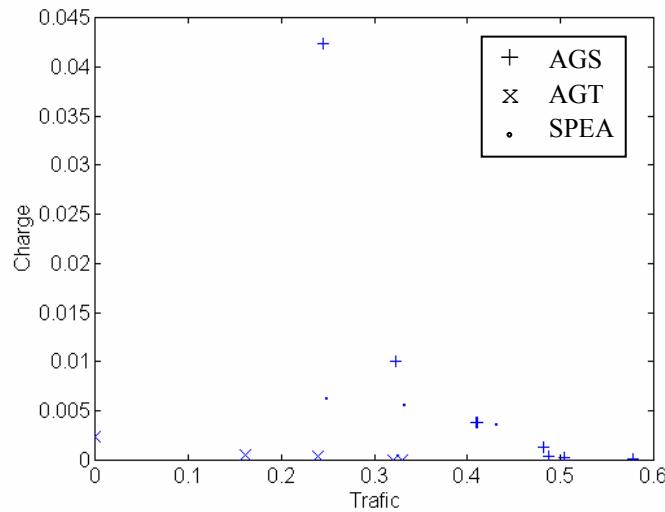


Figure 8.6 Les fronts Pareto des méthodes pour l'exemple 2

celles de ses deux acolytes (voir table 8.4). En effet, les solutions trouvées par AGS et SPEA sont toutes dominées par au moins une solution de AGT (voir figure 8.6). L'efficacité de AGT est démontrée par le temps de calcul

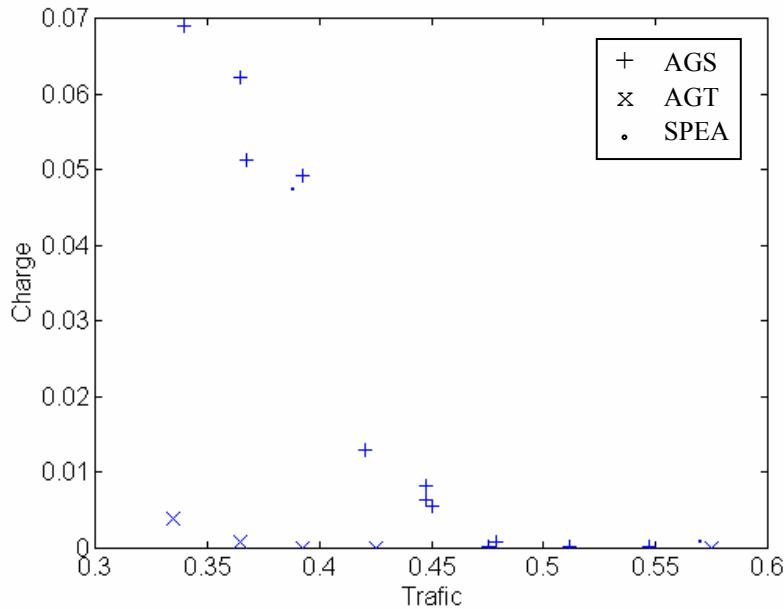


Figure 8.7 Les fronts Pareto des méthodes pour l'exemple 3

requis pour atteindre ses performances qui est loin des temps d'exécution de SPEA et AGS.

Dans le troisième exemple, AGS donne le front le plus consistant en terme de taille : 18 solutions face aux 13 et 2 solutions de AGT et SPEA respectivement. Cependant, la position du front PF_{known} de AGT montre que chaque solution trouvées par AGS est dominée par au moins une solution de AGT (voir fig. 8.7).

Dans l'ensemble, la méthode hybride AGT a donné les meilleures performances en ce qui concerne la qualité et l'efficacité. AGS vient dans la deuxième position avant SPEA. Cependant, un constat négatif peut être fait par rapport à la stabilité : pour des tailles moyennes, aucune méthode n'a pu maintenir les mêmes performances. En d'autres termes, le front Pareto trouvé par chaque méthode diffère d'une exécution à une autre bien que les valeurs de Hyperarea soient proches. Pour SPEA, nous avons constaté que cette instabilité s'étend au temps de calcul. Le temps d'exécution diffère, considérablement, d'une exécution à une autre, même en maintenant les mêmes valeurs pour les paramètres de taille et de nombre d'itérations. Ceci est dû à la procédure de diversification qui peut avoir des temps de calcul très sensibles à la répartition des solutions dans l'espace des objectifs.

CONCLUSIONS

DE LA TROISIEME PARTIE

Notre approche au problème de composition se veut graduelle. Le but étant d'approcher petit à petit les circonstances pratiques d'un problème réel. Dans cette perspective, nous avons présenté une modélisation basée sur la théorie des graphes. Les vertus de cette théorie ne sont pas à démontrer : Elle utilise des outils d'une très grande simplicité mais qui permettent de concevoir des modèles complexes et d'aboutir à des résultats d'analyses fiables et rigoureux. Le modèle nous a permis de proposer une méthode spécifique, tirant profit de la bibliothèque des résultats théoriques présents dans la théorie des graphes. Plusieurs pas ont été réalisés pour approcher les circonstances réelles. En effet, le modèle permet à la fois

- de tenir compte des routages et de la possibilité de réaliser des opérations non consécutives dans une même machine, des volumes de production, des contraintes pratiques telles que la taille des cellules et les contraintes de cohabitation.
- d'éviter certaines suppositions de simplification telles que le nombre statique de cellules.

A la fin de cette partie, nous avons franchi un autre pas dans la direction du but évoqué en considérant des routages alternatifs.

La partie qui suit est consacrée à la réalisation d'un autre pas, traitant d'un aspect relativement peu abordé par la communauté des chercheurs. Il s'agit de l'aspect dynamique pour lequel une étude approfondie est de rigueur vu son implication dans la prise en compte des mutations fréquentes subies dans un marché hostile et sans merci.

Chapitre 9

ETUDE DE L'ASPECT DYNAMIQUE

En guise d'introduction, ce chapitre commence par l'exposé des différents facteurs ayant rendu l'étude de l'aspect dynamique primordiale. Ensuite, la définition de celui-ci est donnée avec un peu de détails. Ce qui nous permettra de voir, dans le paragraphe qui suit, l'impact de la prise en compte de cet aspect sur les différents modules de la conception. Le restant du chapitre est consacré à l'étude de l'aspect dynamique en ce qui concerne le module de composition. A cet effet, des formulations liées au problème sont proposées, tenant compte des stratégies qui peuvent être exigées par le décideur.

9.1 MOTIVATIONS

Dans la première partie, nous avons introduit le problème de conception des systèmes cellulaires de production. Nous avons mentionné, alors, que ces derniers ont été conçus afin de remédier à la carence en flexibilité des systèmes traditionnels de production (les systèmes linéaires orientés produits) et en rapidité des systèmes non linéaires orientés opérations, en se basant sur une décomposition intelligente de l'atelier de fabrication.

Néanmoins, pour des raisons de simplification, les modèles, proposés dans la littérature, considéraient que les demandes en produits étaient constantes¹ durant la totalité de l'horizon de planification. Pour cette raison, nous parlons de systèmes statiques de production, du fait que les commandes en produits sont supposées inchangées, en genre et en quantités, durant tout l'horizon de planification.

Cette supposition, très restrictive, qui est la caractéristique de toute étude préliminaire d'un domaine complexe, immense et méconnu, met en cause toute approche de résolution adoptant une telle supposition. L'évolution de l'environnement de production, caractérisé par des perturbations de plus en plus fréquentes et significatives, a engendré un manque d'adaptation de telles approches : toute modification de la demande, qui n'est pas accompagnée d'une réévaluation des décisions de gestion de la production, se traduirait par une désadaptation du système considéré. Ceci entraîne une éventuelle dégradation, voire un écroulement des performances.

Face à ces exigences, le besoin d'étude des *systèmes dynamiques de production* apparut, ayant pour vocation de concevoir des systèmes pouvant garder un certain nombre de performances en tenant compte des variations survenues dans leur environnement.

¹ A vrai dire, on considère la moyenne des demandes pour chaque produit ou bien la demande maximale durant l'horizon de planification.

9.2 VUE D'ENSEMBLE SUR L'ASPECT DYNAMIQUE

Ce paragraphe est consacré à l'élucidation de certaines notions liées à la définition de l'aspect dynamique et aux stratégies pouvant être adoptées pour leur résolution.

9.2.1 Perturbations dans les systèmes dynamiques et approches déterministes et non déterministes

Etudier un système sous l'aspect dynamique sous-entend la prise en compte des changements ou des perturbations y afférentes dans l'élaboration du modèle et de l'approche de résolution. D'un point de vue générale, les perturbations peuvent être internes ou externes [Chetouane1998]. Les premières sont liées à des changements survenus sur l'état des éléments qui constituent le système. Dans un système de production, il s'agit, en général, des variations liées aux pannes de machines, la non disponibilité d'un ou de plusieurs opérateurs, ..., etc. D'autre part, les perturbations externes sont liées aux changements dans les inputs du système. Dans un système de production, il s'agit, en général, des modifications dans les produits à fabriquer, en ce qui concerne le type et la quantité.

La prise en compte de ces perturbations peut être réalisée suivant deux approches : déterministe et non déterministe. Dans la première approche, les perturbations sont supposées prévisibles. On procède alors dans ces circonstances à décomposer l'horizon de planification en plusieurs périodes. Les périodes sont définies de manière à indiquer les intervalles de temps pendant lesquels le système et son environnement sont stables. Les pannes de machines sont prises en compte dans ce cas, de façon indirecte, par la déduction des temps de maintenance des capacités des machines dans chaque période. Dans une approche non déterministe, la situation se complique à la mesure où les perturbations sont supposées aléatoires.

Pour faire face aux perturbations auxquelles est éventuellement soumis le système, des décisions et des actions doivent être effectuées sur celui-ci afin d'assurer le maintien de ses performances. Ces actions, qui dépendent de la stratégie adoptée, peuvent se décomposer en deux niveaux : prévisionnel et opérationnel [Chetouane1998]. Pour le premier niveau, Il s'agit des actions effectuées hors ligne (i.e. hors horizon de planification) ayant pour but de faire face aux perturbations intégrées dans le modèle. Dans ce cas, il est, en général, question de perturbations du type prévisible et dont l'apparition a été envisagée, au préalable, lors de l'élaboration des décisions de gestion. La résolution des problèmes liés aux systèmes cellulaires de fabrication est, dans ce cas, abordée en adoptant ce qu'on appelle «une approche préactive». D'autre part, le niveau opérationnel concerne les actions effectuées en temps réel ayant pour but d'éliminer les conséquences des perturbations non prévues. Il découle de ce type d'actions ce que l'on appelle «une approche réactive» qui caractérise la capacité du système à maintenir ses performances face aux perturbations en modifiant les décisions de gestion en temps réel. Le recours à ce type de gestion est, en général, dû au fait que les perturbations sont du type non prévisible ou bien que le coût de reconfiguration (voir [paragraphe 9.7](#)) est suffisamment petit pour permettre une telle attitude.

9.2.2 La flexibilité revisitée

Pour discuter des stratégies de confrontation des perturbations auxquelles est soumis le système, nous avons besoin de revenir à la notion de flexibilité. Face à ces perturbations, deux stratégies sont possibles : la stratégie

active et la stratégie passive. La première stratégie consiste à modifier l'organisation du système de telle sorte à s'adapter aux nouvelles données. On parle dans ce cas de *flexibilité d'action*. D'autre part, la stratégie passive consiste à maintenir l'organisation du système inchangée pendant tout l'horizon de planification. Dans ce cas de figure, on parle de *flexibilité d'état* [DeToni1998].

Le choix entre ces deux stratégies est une question très intéressante à développer.

Notons que les stratégies active et passive et les approches préactive et réactive peuvent être combinées (voir fig. 9.1) vu que la conception des Systèmes Cellulaires de Production (SCPs) est, en général, réalisée par étapes, et

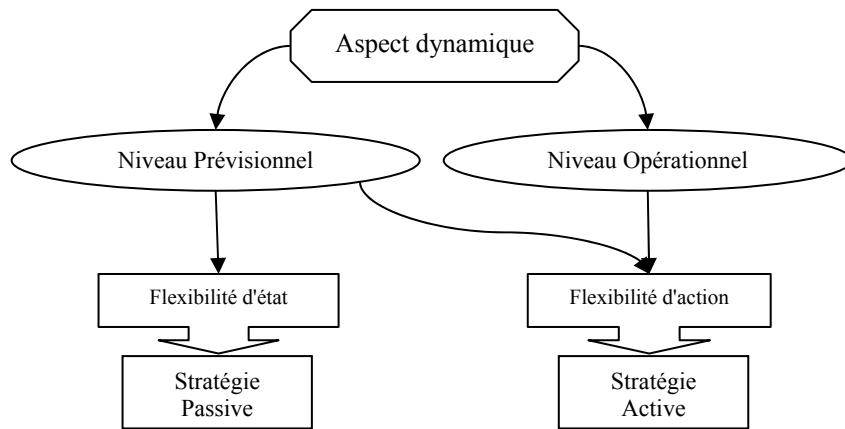


Figure 9.1 Niveaux d'actions et stratégies associées

que les deux types de perturbation peuvent coexister. On peut alors opter pour des stratégies différentes dans des étapes différentes, comme on peut opter pour une approche préactive pour les perturbations prévisibles, tout en surveillant les perturbations non prévisibles pour les accueillir de manière réactive. La figure 9.1 donne une synthèse des notions qui viennent d'être abordées.

9.3 INFLUENCE DE L'ASPECT DYNAMIQUE SUR LA CONCEPTION DES SCPS

Nous avons évoqué, dans la première partie, que l'approche à la conception des SCPs, que nous avons adoptée, passait par quatre modules. Dans ce qui suit, nous montrons, brièvement, l'influence de la prise en compte du caractère très important que constitue le dynamisme sur chacun des quatre modules.

9.3.1 Prétraitement des données

Dans ce module, nous devons déterminer le nombre ainsi que les débuts des périodes constituant l'horizon de planification. Pour chaque période, il sera affecté un unique scénario de demande qui engendrera une unique matrice de flux. C'est à dire, définir les portions de temps dans lesquelles les demandes en produits peuvent être supposées stables. Ceci fait, le problème de détermination des routages des produits, et ce pour chaque période, est semblable à celui traité dans le cas des SCPs statiques. Parmi les travaux traitant certains points de ce module sous l'aspect dynamique, nous citons [Lee1998] [Riezbos2001].

9.3.2 Composition de cellules

La résolution du problème de composition de cellules de fabrication peut être effectuée suivant deux stratégies : passive ou active. Dans la première stratégie, il s'agit de concevoir une composition et de la maintenir inchangée durant tout l'horizon de planification. Dans la seconde stratégie, il sera question de déterminer, non pas

une seule composition, mais une séquence de compositions qui seront reparties sur les H périodes de l'horizon de planification. Ceci va engendrer des coûts de reconfiguration que nous devons prendre en considération car la faisabilité du système va en dépendre de façon significative.

Nous n'allons pas insister ici sur ce module car il sera traité tout au long de cette partie de la thèse.

9.3.3 Agencement intracellulaire

Dans ce module, nous devons répartir les ressources sur le sol de chaque cellule. Si une stratégie passive est adoptée pendant l'étape précédente, c'est-à-dire que les cellules ne changent pas de composants, le problème peut être résolu, à son tour, suivant une stratégie statique ou dynamique. Le choix, entre telle ou telle stratégie, sera basé sur l'importance du coût de transition caractérisé par le temps improductif et par le coût de réagencement intracellulaire.

D'autre part, si une stratégie active est adoptée dans la phase antérieure, l'agencement intracellulaire doit se faire en tenant compte de l'éventuel changement des composants des cellules. Par conséquent, et au contraire de la stratégie précédente, une vue globale de l'ensemble des cellules de l'atelier doit être faite afin de résoudre le problème. Parmi les travaux ayant traité ce module sous l'aspect dynamique, nous citons [Aitzai1998].

9.3.4 Agencement intercellulaire

L'agencement intercellulaire procède, comme dans le cas des SCPs statiques, à la localisation des emplacements des cellules sur le sol de l'atelier pour chaque période de l'horizon de planification. Or, ceci doit se faire en tenant compte de l'éventuel changement du nombre de cellules et de leurs surfaces à cause de la modification des composants de celles-ci.

9.4 ETAT DE L'ART

La nécessité d'une approche tenant en compte l'aspect dynamique du problème se faisait entendre depuis le début des années quatre-vingt-dix. Dans leur article paru en 1991 [Irani1991], les auteurs affirment qu'aucune publication ne traite du problème de composition de cellules en tenant compte des variations éventuelles des demandes en produits : « No cell formation methods exist today capable of designing cells based on the part mixes and demands from several production periods ».

En 1995, Wicks [Wicks1995] propose, dans sa thèse de Doctorat, une formulation multipériodique au problème de composition mixte. La fonction objectif, de son modèle, réalise l'agrégation de trois critères, à savoir, le coût de manutention intercellulaire, l'investissement en machines additionnelles et le coût de reconfiguration. Elle propose, pour le résoudre, un algorithme génétique utilisant un codage qui définit une composition de cellules et qui inclut une heuristique spécifique pour l'affectation des produits aux cellules. Le travail de Wicks sera, par la suite, publié en 1999 dans [Wicks1999].

En 1998, Chen [Chen1998] publie une approche basée sur la programmation dynamique. Son modèle cherche à minimiser les mouvements intercellulaires, le coût de duplication de machines et celui de reconfiguration. Premièrement, il résout une version relaxée de son modèle qui ignore le coût de reconfiguration pour déterminer, pour chaque période, la meilleure composition. Ensuite, il utilise la programmation dynamique pour résoudre le problème initial, en considérant, pour chaque période, l'ensemble des meilleures compositions déterminées dans la première phase. Le procédé de Chen presuppose, donc, qu'une meilleure solution du problème multipériodique est

une séquence de compositions qui sont les meilleures dans au moins une des périodes, ce qui n'est évidemment pas toujours le cas.

Dans la même année, Taboun et al. [Tabou1998] publient un article traitant du problème de composition multipériodique mixte. Ils supposent que seules les familles de produits peuvent changer pendant l'horizon de planification. Leur fonction objectif est une somme pondérée de six critères : Le coût de configuration des cellules, l'investissement en machines, la maintenance de machines, le temps improductif, les mouvements intercellulaires et la sous-traitance de produits. La composition de cellules étant statique, les deux premiers critères ne sont considérés que pour la première période. En raison de la taille du modèle, une heuristique à deux étapes est utilisée pour le résoudre. Dans la première étape, les familles de produits et une partie du contenu de quelques cellules sont déterminés en utilisant une technique Max-Min (maximum de similarités entre produits et minimum de machines). Dans la deuxième étape, le reste des machines est affecté aux cellules en utilisant un modèle relaxé puisque les familles de produits étant déjà créées dans la première étape, les contraintes correspondantes sont omises. Comme nous l'avons déjà évoqué, les auteurs cherchent une composition de cellules qui sera maintenue durant tout l'horizon de planification. On peut affirmer que cette stratégie dite *passive* n'est pas adéquate dans toutes les situations car il se peut qu'une firme soit dans l'obligation de modifier sa composition en guise de réponse à une modification de la demande pour rester compétitive.

En 2000, Mungwattana propose, dans sa thèse de Doctorat [Mungwattana2000], un modèle pour la composition dynamique de cellules en tenant compte de l'aspect multi routage dans sa version la plus simple (i.e.: des machines identiques). Il utilise, pour le résoudre, une approche basée sur la méthode du Recuit Simulé.

Dans tous ces travaux qui viennent d'être cités, ceux qui tiennent compte du coût de reconfiguration [Wicks1995, Chen1998, Wicks1999, Mungwattana2000] supposent que le nombre de cellules est statique et qu'il est préalablement connu. Mungwattana justifie cette supposition trop restrictive par le désir de réduire la taille de l'espace de recherche et, par conséquent, la complexité du modèle. Cependant, on peut constater qu'un nombre de cellules dynamique peut être un paramètre important d'optimisation, voire une obligation. A titre d'exemple, si pour une période donnée, toutes les cellules sont saturées, toute nouvelle acquisition de machines doit être suivie par une augmentation du nombre de cellules. En outre, nous avons remarqué une autre chose plus sérieuse et qui paraît être la raison voilée de la restriction : cette dernière a procuré une aide considérable pour la simplification de la définition du coût de reconfiguration qui doit être revue aussitôt le nombre de cellules est considéré variable. Une discussion concernant le coût de reconfiguration est présentée dans le [paragraphe 9.7](#).

Ce tour d'horizon sur les travaux réalisé traitant de l'aspect dynamique montre que nonobstant l'abondance de travaux de recherche dans le domaine de la composition de cellules, le nombre d'articles traitant de l'aspect dynamique reste pauvre. En effet, une approche non déterministe multi-périodique, est une question ouverte à développer, à la mesure où aucune publication l'adoptant n'existe jusqu'à ce jour, en notre connaissance. En outre, aucun travail, ne traite du problème de décision entre les stratégies active et passive.

Notre travail adoptera une approche préactive déterministe, du fait que nous supposerons que les demandes en produits, quoique variables, sont connues au préalable, et ce pour toute la durée de l'horizon de planification. En outre, un essai de contribution au problème de décision entre les stratégies évoquées est présenté [au chapitre 10](#).

9.5 POSITION DU PROBLEME

Le problème de composition consiste à regrouper les machines en cellules et les produits en familles. Chaque famille est associée à une cellule de telle sorte que les produits de cette famille soient essentiellement fabriqués dans la cellule associée.

Dans le problème statique de composition, les données d'entrée (séquences d'opérations, demandes en produits) sont supposées constantes durant l'horizon de planification. Par contre dans le problème dynamique de composition, ces données peuvent varier de façon significative. Parmi les approches retenues pour le cas dynamique, l'approche multipériodique est parmi les axes qui suscitent l'intérêt des chercheurs actuellement. Dans une approche multipériodique, l'horizon de planification est subdivisé en un ensemble de périodes, chacune avec des données d'entrée distinctes. Les données d'entrée d'une période, qui sont supposées constantes, forment ce que l'on appelle *scénario*. Etant donnée ces considérations, le problème de composition multipériodique peut être abordé selon les deux stratégies *active* et *passive* décrites comme suit :

Nous disposons d'un système de production vérifiant les hypothèses suivantes :

1. L'horizon de planification est composé d'un ensemble de périodes de durée connue.
2. Dans chaque période, le système subit un unique scénario de demande.
3. Chaque type de produit possède un unique routage.

Pour le *Problème de Composition Active de Cellules (PCAC)*, nous cherchons une séquence de compositions de cellules, une composition par période, qui optimise la compétitivité. D'autre part, pour le *Problème de Composition Passive de Cellules (PCPC)*, nous cherchons une unique composition pour arriver au même but.

En outre, comme pour le cas statique, une composition doit respecter les contraintes suivantes pour être acceptée :

1. La taille des cellules ne doit pas dépasser une certaine borne.
2. Le nombre de cellules peut être borné inférieurement et supérieurement.
3. Des contraintes de cohabitation voulant que des machines soient placées dans une même cellule.
4. Des contraintes de non cohabitation voulant que des machines soient éloignées les unes des autres.

Pour des raisons de simplification, nous nous restreignons à deux critères. Nous avons, premièrement, le critère de trafic intercellulaire qui est le plus prescrit. D'autre part, traiter le problème sous l'aspect dynamique fait intervenir un autre critère qui n'est pas, en général, de moindre importance : Il s'agit du coût de reconfiguration qui permet d'évaluer le coût de passage d'une composition à une autre dans les phases de transition de l'horizon de planification, incluant le temps improductif et les éventuels mouvements de machines.

Tenant compte de ces deux critères, le paragraphe suivant présente une formulation bi-objectif au problème PCAC. Pour PCPC, on verra par la suite (voir [paragraphe 9.8](#)) que ce dernier en est, simplement, un cas particulier.

9.6 UNE FORMULATION POUR PCAC

La formulation mathématique du problème de composition statique, présentée [au chapitre 6](#), peut être étendue au cas multipériodique de la façon suivante :

9.6.1 Les inputs

- (1) Considérons $M = \{M_1, M_2, \dots, M_m\}$ l'ensemble des m machines disponibles et $P = \{P_1, P_2, \dots, P_p\}$ l'ensemble des p types de produits à fabriquer. Les produits de P sont, entièrement, fabriqués sur les machines de M sur un horizon de planification de H périodes, chacune avec une durée d_h exprimée en unités de temps.
- (2) Pour chaque type de produit P_k ($k = 1, 2, \dots, p$), on suppose donné :
 - (i) R_k : une unique séquence (ou routage) de machines visitées par le produit, $R_k = \langle M_{k,1}, M_{k,2}, \dots, M_{k,s_k} \rangle$ où $\langle M_{k,j} \rangle \in M$ ($j = 1, 2, \dots, s_k$) et s_k est le nombre de machines dans le routage de P_k .
 - (ii) r_{kh} : le ratio moyen de production du produit de type P_k par unité de temps durant la période h ($h=1..H$).
- (3) Pour chaque triplet $(P_k, M_i, M_j) \in P \times M \times M$, on note $v_{k,i,j}$ le nombre de fois où M_i suit M_j ou inversement ($i, j = 1, \dots, m$) dans le routage de P_k .
- (4) Pour chaque couple $(M_i, M_j) \in M \times M$, on note t_{ijh} le trafic entre les machines (M_i, M_j) ($i, j = 1, \dots, m$) durant la période h ($h=1..H$), défini par :

$$t_{ijh} = \sum_{k=1}^p r_{kh} \times v_{k,i,j}$$

- (5) Pour chaque période h , $h=1..H$, on définit le réseau $N_h = (M, E, W_h)$ où :
 - (i) M est l'ensemble des machines disponibles durant l'horizon de planification,
 - (ii) $E = \{e_{ij} / (M_i, M_j) \in M \times M \text{ et } \exists h=1..H : t_{ijh} \neq 0\}$ est l'ensemble des arêtes reliant chaque couple de machines ayant un flux de produits dans une période quelconque et
 - (iii) W_h est une fonction de pondération des arêtes qui évalue le trafic entre les machines extrémités durant la période h (i.e.: $W_h(e_{ij}) = t_{ijh}$).
- (6) Pour chaque période h , on définit :
 - (i) une partition C_h de M en J_h cellules, $J_h \in \{1, \dots, m\}$ telle que $C_h = \{C_{h1}, C_{h2}, \dots, C_{hJ_h}\}$ et $\forall h=1..H$:

$$\bigcup_{i=1}^{J_h} C_{hi} = M \text{ et } C_{hi} \cap C_{hj} = \emptyset, \forall i, j \in \{1, \dots, J_h\}, i \neq j;$$
 - (ii) le sous-ensemble $E(C_h)$ des arêtes intercellulaires associées à la partition C_h :

$$E(C_h) = \{e_{kl} \in E / (M_k, M_l) \in C_{hi} \times C_{hj}; i, j = 1, 2, \dots, J_h; i \neq j \text{ et } k, l = 1, 2, \dots, m\}$$
 - (iii) le trafic intercellulaire normalisé $T(C_h)$ de la partition C_h :

$$T(C_h) = d_h \cdot \sum_{e_{kl} \in E(C_h)} W_h(e_{kl}) \Bigg/ \left(\sum_h d_h \cdot \sum_{e_{kl} \in E} W_h(e_{kl}) \right)$$

- (7) pour chaque couple de compositions consécutives (C_h, C_{h+1}) , $h=1..H-1$, on définit le coût de reconfiguration (recomposition) normalisé, $R(C_h, C_{h+1})$, qui évalue le coût des changements devant être appliqués sur C_h pour

avoir la composition suivante, C_{h+1} . La définition formelle de ce coût fera l'objet d'une discussion dans le [paragraphe 9.7](#).

(8) Pour chaque H compositions consécutives S , $S=(C_1, \dots, C_H)$, couvrant la totalité de l'horizon de planification, on définit :

(i) le trafic intercellulaire entier normalisé $TE(S)$:

$$TE(S) = \sum_h T(C_h)$$

(ii) le coût de reconfiguration entier normalisé $RE(S)$:

$$RE(S) = \sum_{h=1}^{H-1} R(C_h, C_{h+1}) / (H-1)$$

9.6.2 Les Contraintes

Soit $S=(C_1, \dots, C_H)$ une séquence de H compositions qui respecte les contraintes citées dans la position du problème (voir [paragraphe 9.5](#)). Ceci peut être, formellement, reproduit par :

(1) Etant donné le nombre maximum de machines permis dans une cellule durant la période h , $Nmax_h$, nous devons avoir

$$\forall C_{hi} \in C_h \ (i = 1, 2, \dots, J_h), \text{Card}(C_{hi}) \leq Nmax_h$$

où $\text{Card}(C_{hi})$ est le cardinal de la cellule C_{hi} (nombre de machines affectées à C_{hi}).

(2) Etant donné une borne supérieure ($Jmax_h$) et inférieure ($Jmin_h$) pour le nombre de cellules permis dans la période h , J_h doit vérifier :

$$Jmin_h \leq J_h \leq Jmax_h$$

(3) Etant donné un ensemble de couples de machines devant être mises dans la même cellule durant la période h , $Ecoh_h$, on doit respecter :

$$\forall (M_k, M_l) \in Ecoh_h, \exists C_{hi} \in C_h : M_k, M_l \in C_{hi} \text{ où } k, l \in \{1, \dots, m\} \text{ et } i \in \{1, \dots, J_h\}$$

(4) Etant donné un ensemble de couples de machines devant être mises dans des cellules différentes durant la période h , $Encoh_h$, on doit respecter :

$$\forall (M_k, M_l) \in Encoh_h, \exists C_{hi}, C_{hj} \in C_h : M_k \in C_{hi} \text{ et } M_l \in C_{hj} \text{ où } k, l \in \{1, \dots, m\} \text{ et } i, j \in \{1, \dots, J_h\}, i \neq j$$

Remarque 9.1 : Par convention, on étend E à $E \cup \left(\bigcup_h (Ecoh_h \cup Encoh_h) \right)$. Cette extension consiste à ajouter des arêtes fictives entre les machines cohabitantes ou non cohabitantes lorsqu'elles ne sont reliées par aucun flux de produits durant toutes les périodes.

9.6.3 La fonction Multi objectif

Soit CS l'ensemble des séquences de compositions qui respectent les contraintes précédentes. Le problème est de trouver une séquence de compositions $S^* \in CS$ telle que :

$$Z(S^*) = \min_{S \in CS} (TE(S), RE(S)) \quad (5)$$

C'est à dire, on cherche une séquence qui vérifie toutes les contraintes et qui optimise le trafic intracellulaire et le coût de reconfiguration.

9.7 LE COUT DE RECONFIGURATION

9.7.1 Problématique

Lorsque nous considérons l'aspect dynamique du problème de composition, un coût de transition doit être considéré entre chaque couple de périodes consécutives due à la reconfiguration du système de production au début de chaque période succédée. Ce *coût de reconfiguration* est un paramètre déterminant de la stratégie devant être adoptée. En effet, plus important il est, plus le recours à une séquence statique (stratégie passive) est plausible. Ce coût de reconfiguration mesure l'importance des changements devant être appliqués sur la structure du système et leurs effets, consistant essentiellement en : les déplacements de machines (changements) et le temps improductif (effets). Par conséquent, il est évident qu'une évaluation exacte de ce coût doit utiliser les informations concernant l'emplacement physique des machines. Cependant, aborder le problème avec de telles informations requiert traiter, en même temps, le problème d'agencement, conduisant à un modèle de grande taille, dont la résolution pour des instances de taille réelle n'est pas aisée.

Dans le but d'avoir un bon compromis entre la négligence de ce coût et le recours à une évaluation exacte prohibitive de celui ci, nous pouvons recourir à une estimation basée exclusivement sur les informations de la décomposition des ressources en cellules, indépendamment de leurs positions (machines et cellules) dans le sol de l'atelier.

Les recherches précédentes [Wicks1995, Chen1998, Wicks1999, Mungwattana2000] définissent ce coût en considérant un nombre statique de cellules qui permet d'avoir la même indexation de cellule dans toutes les périodes de l'horizon de planification. Ensuite, pour chaque couple de compositions consécutives, les machines déplacées sont identifiées en considérant les changements survenus dans les cellules ayant les mêmes indices dans les deux périodes. Cependant, cette méthode présente une insuffisance significative due à la redondance de l'indexation des cellules. En effet, si l'indexation n'est pas adéquate, un grand coût de reconfiguration peut être obtenu même si la composition de cellules est exactement la même dans toutes les périodes de l'horizon. En outre, une autre insuffisance vient du fait que cette méthode ne tient pas compte du cas du nombre variable de cellules.

L'exemple suivant est utilisé pour deux fins. D'une part, il nous permet d'expliquer et de montrer la maladresse de la méthode utilisée dans la littérature. De l'autre, il nous servira de moyen d'illustration des méthodes que nous proposerons par la suite.

Un système est composé de cinq machines et est soumis à deux scénarios de demandes. Les compositions adoptées dans ces dernières sont :

- $C_1 = \{C_{11}, C_{12}\}$ avec $C_{11} = \{M_1, M_2, M_3\}$ et $C_{12} = \{M_4, M_5\}$.
- $C_2 = \{C_{21}, C_{22}\}$ avec $C_{21} = \{M_1, M_4, M_5\}$ et $C_{22} = \{M_2, M_3\}$.

Chaque machine M_i est supposée avoir un coût de déplacement δ_i . Pour simplifier l'exposé, supposons qu'il est le même pour toutes les machines (i.e. : $\forall i \in \{1, \dots, m\} : \delta_i = \delta$).

Le calcul du coût de reconfiguration utilisé dans la littérature évalue ce dernier à $R(C_1, C_2) = \delta \cdot (2+2) = 4 \cdot \delta$. C'est-à-dire, en comparant C_{11} à C_{21} , on trouve que deux machines qui étaient dans la première ne sont pas dans la deuxième (donc déplacées). D'autre part, le même nombre de déplacements est trouvé entre C_{12} et C_{22} . On peut remarquer alors qu'une correspondance entre C_{11} et C_{22} , d'une part, et entre C_{12} à C_{21} d'autre part, aurait donné un coût de déplacement de δ uniquement. Cet exemple simple nous permet de voir qu'il ne faut pas se fier à

l'indexation des cellules pour établir la correspondance qui nous permettra de calculer le coût de déplacement. Il faut, par contre, chercher la correspondance qui l'optimise. Dans le paragraphe suivant, nous proposons un théorème qui donne une méthode qui permet de trouver une telle correspondance.

9.7.2 Détermination exacte du coût de reconfiguration

Ce paragraphe permet de montrer comment obtenir une évaluation exacte du coût de reconfiguration [Bouli2006b]. Rappelons que ce dernier est considéré à la base des informations définissant les compositions de cellules en faisant abstraction des emplacements de machines sur le sol de l'atelier. A cet effet, nous entendons par le terme déplacement tout changement d'affectation d'une machine à une cellule entre les deux périodes considérées. Pour ce faire, nous utilisons les notations suivantes :

h : indice de périodes ou de transitions. Dans le premier cas, il est dans $\{1, \dots, H\}$. Dans le second cas, il est dans $\{1, \dots, H-1\}$.

i, q : indices de machines.

j, g : indices de cellules.

δ_{hi} : le coût de déplacement de la machine M_i durant la transition h supposé indépendant de son emplacement.

Δ_h : la somme des coûts de déplacement des machines durant la transition h (i.e: $\Delta_h = \sum_{i=1}^m \delta_{hi}$).

C_h, C_{h+1} : deux composition de cellules consécutives.

Md_h : sous-ensemble de M , représentant les machines à déplacer durant la transition h , pour passer de C_h à C_{h+1} .

u_{hi} : indicateur booléen associé à la transition h , égal à un si $M_i \in Md_h$, zéro sinon.

J_h : taille de la composition adoptée durant la période h .

C_{hj} : la j -ième cellule de C_h .

b_{hijg} : indicateur booléen pour la transition h (i.e.: entre C_h et C_{h+1}), égal à un si $M_i \in C_{hj} \cap C_{(h+1)g}$, zéro sinon.

Etant donnée une composition de cellule C_h adoptée durant la période h , nous avons besoin de déterminer un sous-ensemble $Md_h \subseteq M$ de machines à déplacer durant la transition h pour passer à la composition C_{h+1} .

Proposition 9.1 : Un ensemble de machines à déplacer Md_h permet de réaliser une transition d'une composition C_h à C_{h+1} données si et seulement si il vérifie les deux conditions (Cd_1 et Cd_2) suivantes :

1. **Cd_1** : $\forall M_i \notin Md_h, \forall M_q \notin Md_h$:

$$(\exists j \in \{1, \dots, J_h\} : (M_i \in C_{hj}) \wedge (M_q \in C_{hj})) \Leftrightarrow (\exists j \in \{1, \dots, J_{h+1}\} : (M_i \in C_{(h+1)j}) \wedge (M_q \in C_{(h+1)j}))$$

C'est-à-dire que pour tout couple de machines non déplacées, si elles sont dans une même cellule dans une période, elles resteront ensemble dans l'autre.

2. **Cd_2** : $\forall M_i \in Md_h, \forall M_q \in Md_h$:

$$(\exists j \in \{1, \dots, J_h\} : (M_i \in C_{hj}) \wedge (M_q \in C_{hj})) \Rightarrow \neg (\exists j \in \{1, \dots, J_{h+1}\} : (M_i \in C_{(h+1)j}) \wedge (M_q \in C_{(h+1)j}))$$

et

$$(\exists j \in \{1, \dots, J_{h+1}\} : (M_i \in C_{(h+1)j}) \wedge (M_q \in C_{(h+1)j})) \Rightarrow \neg (\exists j \in \{1, \dots, J_h\} : (M_i \in C_{hj}) \wedge (M_q \in C_{hj}))$$

C'est-à-dire que pour tout couple de machines dont l'une est déplacée tandis que l'autre ne l'est pas, si elles sont dans une même cellule dans une période, elles seront séparées dans l'autre.

Preuve : Il est évident que les deux conditions sont nécessaires. En effet, si deux machines habitant une même cellule n'ont pas vu leur affectation changer pendant une transition donnée, elles resteront ensemble dans la deuxième période associée à ladite transition. D'autre part, elle vont se séparer si l'une est déplacée et l'autre non.

Sans perte de généralité, montrons que la vérification des deux conditions par un sous-ensemble Md_h est suffisante pour que ce dernier puisse assurer la transition de C_h à C_{h+1} . En effet, en procédant par construction, on détermine les cellules de C_h contenant au moins une machine non déplacée, i.e. n'appartenant pas à Md_h . A chacune de ces cellules, il correspond une unique cellule de C_{h+1} car à cause de Cd_1 , si plusieurs machines sont ensemble dans C_h elles resteront ensemble dans une cellule de C_{h+1} . Nous obtenons ainsi un sous-ensemble de cellules de C_{h+1} , éventuellement incomplètes, car elles ne contiennent que des machines non déplacées. Cd_2 nous permet alors d'assurer que les seules machines candidates à la complétion sont des machines déplacées (i.e. appartenant à Md_h). En complétant l'affectation du restant des machines, en comblant les cellules non complètes de C_{h+1} et en créant, si nécessité il y a, d'autres, nous obtenons la composition C_{h+1} . \square

Illustrons la proposition et la démarche de la preuve sur l'exemple précédent (voir [paragraphe 9.7.1](#)). Supposons que $Md_1 = \{M_1, M_4, M_5\}$. En ne déplaçant que ces machines, peut-on passer de C_1 à C_2 ? La proposition 9.1 en donne une réponse affirmative car d'une part Cd_1 est vérifiée. En effet, pour tous les couples (non ordonnés) de machines non déplacées, qui sont ensemble dans C_1 , à savoir (M_2, M_3) dans C_{11} (réciproquement, dans C_2 , à savoir (M_2, M_3) dans C_{22}), il existe une cellule qui les regroupe dans C_2 , à savoir C_{22} (réciproquement, dans C_1 à savoir C_{11}). D'autre part, pour tout couple de machines où l'une est déplacée et l'autre non et qui sont dans C_1 , à savoir, (M_1, M_2) et (M_1, M_3) (réciproquement, dans C_2 mais il n'y a aucun), ils sont séparés dans C_2 . La possibilité de construction de C_2 à partir de C_1 en utilisant les machines de Md_1 , formulée dans la preuve de suffisance, consiste à associer à chaque cellule de C_1 contenant au moins une machine non déplacée, l'unique cellule de C_2 qui contient les mêmes machines non déplacées. Ainsi, nous associons à C_{11} la cellule C_{22} . La deuxième étape consiste en l'affectation des machines déplacées pour compléter la composition C_2 . A cet effet, nous créons une nouvelle cellule et nous y ramenons les machines M_1, M_4, M_5 . Ce qui nous donne la cellule C_{21} .

Un autre sous-ensemble, non conforme aux exigences de la proposition 9.1, rendrait les choses plus claires. Supposons que $Md_1 = \{M_1, M_2, M_4\}$. Ce sous-ensemble vérifie Cd_1 car il n'y a pas de couple de machines non déplacées et qui appartiennent à une même cellule dans les deux compositions. Cependant, il ne vérifie pas Cd_2 car les machines M_2 et M_3 qui sont dans une même cellule dans C_1 , à savoir C_{11} , n'ont pas été séparées dans C_2 . On peut vérifier qu'aucun déplacement des machines M_1, M_2, M_4 ne permet de passer de C_1 à C_2 .

Définition 9.1 : Nous appelons **transitaire** associé à la transition h (ou transitaire tout court lorsqu'il n'y a pas d'ambiguïté), tout sous-ensemble de machines $Md_h \subset M$ ($Md_h \neq M$) respectant les conditions de la proposition 9.1.

Si la proposition précédente nous permet de savoir si un sous-ensemble de machines permet de transiter entre deux compositions données, pouvant avoir des cardinalités différentes, il est intéressant de déterminer celui qui

réalise la transition avec un coût minimum car, pour une transition donnée, il peut y avoir plusieurs transitaires possibles. Le théorème suivant permet de donner une solution à ce problème.

Théorème 9.1

La détermination du coût de reconfiguration minimum revient à résoudre un problème de couplage de poids maximum dans un graphe biparti.

Preuve

Le coût de reconfiguration $R(C_h, C_{h+1})$ est donné par la somme des coûts de déplacement des machines du transitaire désigné à faire passer l'état du système de la composition C_h à C_{h+1} . C'est-à-dire :

$$R(C_h, C_{h+1}) = \sum_{i=1}^m u_{hi} \cdot \delta_i .$$

Le problème est donc de trouver le sous-ensemble Md_h minimisant la somme des coûts de déplacement des machines déplacées. Notons, pour cela, la dualité entre cet objectif et celui de la maximisation du coût de déplacement des machines non déplacées (i.e.: $M - Md_h$). Ceci est directement déduit du fait que la somme des coûts de déplacement des machines est constante pour chaque transition.

Nous construisons le graphe biparti valué $G_h = (X_h \cup X_{h+1}, E_h)$ comme suit :

- (1) Le sous-ensemble de sommets X_h (recip. X_{h+1}) associe un sommet x_{hj} (récip. $X_{(h+1)j}$) à chaque cellule C_{hj} (récip. $C_{(h+1)j}$).
- (2) L'ensemble des arêtes E_h est défini comme suit : $E_h = \{e_{jg} = (x_{hj}, x_{(h+1)g}) / j \in \{1, \dots, J_h\} \text{ et } g \in \{1, \dots, J_{h+1}\} \text{ et } C_{hj} \cap C_{(h+1)g} \neq \emptyset\}$. C'est à dire, une arête est définie pour chaque couple de sommets représentatifs de deux cellules ayant au moins une machine en commun.

- (3) La fonction de pondération des arêtes Y_h est donnée par la formule : $Y_h(e_{jg}) = \sum_{i=1}^m \delta_{hi} \cdot b_{hijg}$. Cette fonction

associe à chaque arête une valeur égale à la somme des coûts de déplacement des machines communes aux cellules associées à ses extrémités.

Il existe une bijection entre l'ensemble des couplages de G_h et l'ensemble des transitaires associés à la transition h . Autrement dit, à chaque couplage de G_h est associé un unique sous-ensemble Md_h vérifiant la proposition 9.1 et inversement. En effet, pour chaque couplage $Cp = \{\langle e_1 \rangle, \langle e_2 \rangle, \dots, \langle e_{|Cp|} \rangle\}$ de G_h où $\langle e_k \rangle \in E_h$ et $k \in \{1, \dots, |Cp|\}$, le transitaire associé est défini comme suit : $Md_h = M - \{M_i \in M / (\exists j \in \{1, \dots, J_h\}, \exists g \in \{1, \dots, J_{h+1}\} : e_{jg} \in Cp \text{ et } b_{hijg} = 1)\}$. De façon plus claire, le transitaire est le complémentaire, par rapport à M , de l'ensemble de machines construit par la réunion des intersections des cellules associées aux arêtes du couplage Cp . Un tel ensemble vérifie les conditions de la proposition 9.1 car, d'une part, deux machines non déplacées qui étaient dans une même cellule dans une période sont issues d'une seule intersection associée à une seule arête de Cp et, donc, Cd_1 est vérifiée. D'autre part, supposons que Cd_2 n'est pas vérifiée par un Md_h issu d'un couplage Cp . Ceci implique l'existence de deux machines, l'une M_i déplacée et l'autre M_q non ($i, q \in \{1, \dots, m\}, i \neq q$), qui étaient dans une même cellule dans une période et elles ne se sont pas séparées dans l'autre période. M_q étant non déplacée (i.e. $\notin Md_h$), il existe une arête $\langle e_k \rangle$ de Cp dont l'intersection de cellules associée contient M_q . Le fait de supposer que M_i et M_q ne se sont pas

séparées implique que M_i appartient aussi à l'intersection des cellules associée à $\langle e_k \rangle$. Or, par définition du transitaire associé à Cp , $M_i \notin Md_h$. Contradiction. Par conséquent, tout couplage nous donne un transitaire.

Il nous reste de montrer que tout transitaire définit un couplage dans G_h . Supposons, par l'absurde, que nous avons un transitaire Md_h qui nous définit un sous-ensemble Cp qui n'est pas un couplage. Deux cas sont alors possibles : Deux arêtes $\langle e_k \rangle$ et $\langle e_{k+1} \rangle$ de Cp ont une même extrémité ou bien Cp est vide. Dans le premier cas, nous devons, donc, avoir deux machines M_i et M_q non déplacées appartenant à l'intersection des deux cellules associées à $\langle e_k \rangle$ et à $\langle e_{k+1} \rangle$ respectivement. $\langle e_k \rangle$ et $\langle e_{k+1} \rangle$ ayant une même extrémité, ces deux machines auraient été dans une même cellule dans une période et se sont séparées dans l'autre, sans qu'elles ne soient des machines déplacées. Ceci est en contradiction avec le fait que Md_h est un transitaire. Dans le deuxième cas, le transitaire consiste à réaliser le déplacement de toutes les machines (i.e.: $Md_h=M$), ce qui est impossible par définition. Notons enfin que cette restriction de la définition des transitaires est liée au fait qu'on n'est jamais obligé à réaliser le déplacement de toutes les machines pour assurer une transition entre n'importe quel couple de compositions. Par conséquent, une telle manœuvre, qui consiste à déplacer toutes les machines (même si ceci engendre un $Md_h=M$ qui vérifie les conditions de la proposition 9.1), peut être omise dans le procédé d'optimisation car il y a toujours une meilleure solution (en toute rigueur, lorsqu'au moins une machine a un coût de déplacement non nul). \square

Remarque 9.2

Le problème de couplage de poids maximum dans les graphes bipartis possède une autre appellation très célèbre, à savoir, le problème d'affectation.

Le théorème précédent nous permet de déterminer le coût de reconfiguration de façon optimale en résolvant un problème de couplage de poids maximum (*pcm*) dans un graphe biparti G_h . Ce problème a fait l'objet de plusieurs travaux de recherche au sein de la communauté des chercheurs opérationnels. L'algorithme le plus efficace de *pcm* pour les graphes bipartis est dû à Hopcroft [Hopcroft1973] et est en $O(n^{2.5})$ pour un graphe d'ordre n . Pour notre cas, la détermination du coût de reconfiguration est donc de l'ordre de $O(H \cdot m^{2.5})$ car nous devons résoudre le problème de couplage ($H-1$) fois pour des instances de graphes ayant au maximum $2 \cdot m$ sommets. A cela, il faut ajouter le coût des traitements associant à chaque couple de compositions consécutives le graphe biparti adéquat. Ce volume de calcul peut être compromettant pour une approche de résolution qui se veut efficace, surtout lorsqu'on envisage aborder des problèmes de grande taille. Nous donnerons, dans le chapitre suivant, des propositions pour contourner ces obstacles.

9.8 UNE FORMULATION POUR PCPC

PCPC survient lorsque le décideur impose une stratégie passive. En d'autres termes, que la même composition doit être maintenue durant la totalité de l'horizon de planification. Cette attitude est due généralement au fait que le coût de réagencement inter- et/ou intracellulaire est si important que le décideur ne veut pas courir le risque d'effondrement en adoptant une stratégie active.

Considérons la formulation de *PCAC* (voir paragraphe 9.6). Une formulation de *PCPC* peut être déduite, directement, de celle de *PCAC* en imposant que la séquence S soit composée d'une unique composition. Autrement

dit, $S=(C_1, \dots, C_H)$ telle que $C_1=\dots=C_H=C$. Notons CS' l'ensemble de telles séquences. Cette restriction a un effet direct sur la fonction bi-objectif $Z(S^*)$ qui devient définie comme suit : $Z(S^*) = \min_{S \in CS'} TE(S)$ puisque le coût de reconfiguration est, dans ce cas, nul quelle que soit la composition adoptée.

En s'inspirant des travaux de Batta [Batta1987], qui a proposé un théorème montrant que le problème d'agencement dynamique passif revient à un problème d'agencement statique (mono-periodique), nous montrons, dans ce qui suit, qu'un résultat similaire peut être trouvé pour le problème de composition passive de cellules.

Proposition 9.2

Le problème de composition dynamique passive de cellules, *PCPC*, revient à un problème statique de composition (mono-periodique).

Preuve

La fonction objectif de PCPC peut être développée comme suit :

$$\begin{aligned} Z(S^*) &= \min_{S \in CS'} TE(S) \\ &= \min_{S \in CS'} \sum_h T(C_h) \\ &= \min_{S \in CS'} \sum_h \left(d_h \cdot \sum_{e_{kl} \in E(C_h)} W_h(e_{kl}) \right) / \left(\sum_h d_h \cdot \sum_{e_{kl} \in E} W_h(e_{kl}) \right) \\ &= \min_{S \in SP} \sum_h \sum_{e_{kl} \in E(C_h)} \left(d_h / \left(\sum_h d_h \cdot \sum_{e_{kl} \in E} W_h(e_{kl}) \right) \right) \cdot W_h(e_{kl}) \end{aligned}$$

Cependant, puisque la même partition est maintenue durant les H périodes (i.e.: $E(C_1)=\dots=E(C_h)=E(C)$), nous pouvons remplacer C_h par C et inverser les deux signes de sommation, l'ensemble des arêtes intercellulaires $E(C_h)$ étant indépendant de h . Ces modifications nous donnent :

$$= \min_{C \in CP} \sum_{e_{kl} \in E(C)} \sum_h \left(d_h / \left(\sum_h d_h \cdot \sum_{e_{kl} \in E} W_h(e_{kl}) \right) \right) \cdot W_h(e_{kl})$$

où CP est l'ensemble des compositions vérifiant les contraintes associées à chaque période (voir [paragraphe 9.6.2](#)).

Par cette formulation, nous définissons un problème de partitionnement dans le réseau $N=(M, E, W)$ où la fonction définissant les poids des arêtes, W , est comme suit : $\forall e_{kl} \in E : W(e_{kl}) = \sum_h \left(d_h / \sum_h d_h \cdot \sum_{e_{kl} \in E} W_h(e_{kl}) \right) \cdot W_h(e_{kl})$ \square

Remarque 9.3

La proposition qui vient d'être énoncée et qui a permis de montrer que la résolution de *PCPC* revenait à la résolution d'un problème statique n'est pas valable pour n'importe quel formulation du problème de composition passive dynamique. En effet, la linéarité de la fonction objectif a été d'un grand apport pour permettre une telle assertion. Cependant, si on considère, dans le modèle, le critère de minimisation des écarts de charge intercellulaires dont la formulation est réalisée par une fonction non linéaire (moyenne des carrés des écarts à la moyenne ou variance, voir [paragraphe 6.4.1](#)), on peut montrer que le précédent procédé de démonstration n'est plus

valable. Déterminer les conditions générales et les critères pouvant être formulés de façon à respecter ces conditions, pour que le problème de composition passive dynamique revienne à un problème statique, est une question ouverte à développer.

9.9 LE PROBLEME DE DECISION ENTRE STRATEGIE PASSIVE ET ACTIVE

Si, pour *PCAC*, nous sommes à la quête de la meilleure séquence de compositions, pour *PCPC* nous cherchons une unique composition de cellules qui peut supporter au mieux les changements de l'environnement du système. Discuter du problème de décision entre stratégie passive et active cherche à aider le décideur qui veut déterminer laquelle parmi les deux stratégies est la plus adéquate. Certes, nous pouvons affirmer qu'une stratégie passive serait vraisemblablement plus adéquate lorsque la reconfiguration du système engendre un coût important. Cependant, à partir de quel seuil ceci peut être garanti ? (voir fig.9.4).

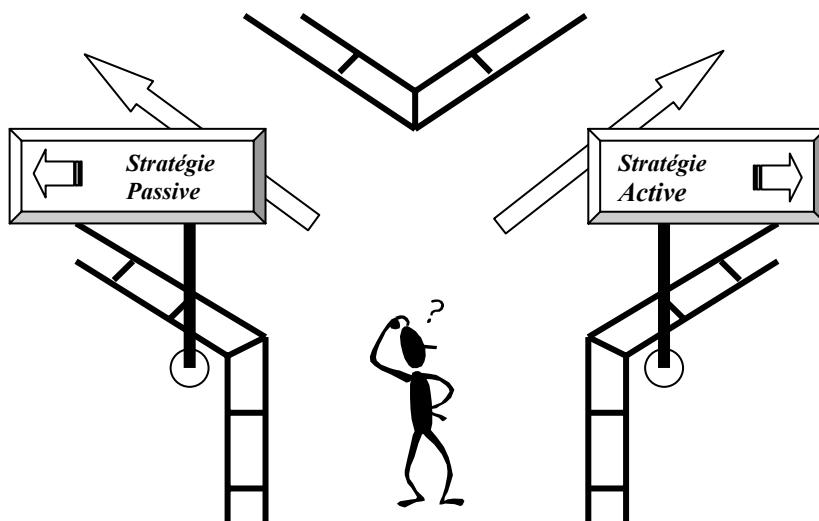


Figure 9.4 Stratégie passive ou active ?

Il est important de constater, ici, que *PCPC* est un cas particulier de *PCAC*. En effet, pour *PCAC*, une solution réalisable peut être composée d'une séquence de compositions identiques. Par conséquent, l'espace des solutions de *PCPC* est inclus dans celui de *PCAC*. Donc, théoriquement, le problème de décision entre les deux stratégies ne se pose pas du moment que résoudre *PCAC* permet de mettre fin aux tourments du décideur.

Nous avons bien dit théoriquement car *PCAC* est au moins aussi difficile que la version statique du problème de composition que nous avons démontré être NP-complet. En effet, le problème statique est un cas particulier de *PCAC* qui consiste à mettre $H=1$. Par conséquent, résoudre *PCAC* de façon optimale peut relever de l'irréalisme pour des tailles larges du problème.

Après identification de plusieurs concepts et problèmes liés à l'aspect dynamique, il est temps d'essayer de les résoudre. Le chapitre suivant se doit donc de donner une réponse aux différentes questions soulevées dans l'étude de l'aspect dynamique à savoir la composition avec la stratégie passive ou active et le problème de décision entre ces deux stratégies.

Chapitre 10

APPROCHES DE RESOLUTION POUR L'ASPECT DYNAMIQUE

Dans ce chapitre, des essais de contribution à la résolution des problèmes décrits dans le chapitre précédent et qui sont liés à l'aspect dynamique, sont présentés. Pour la stratégie passive, nous illustrons le procédé de conversion qui permet d'utiliser les méthodes proposées dans la troisième partie consacrée à l'aspect statique. Pour la stratégie active, deux approches de résolution sont proposées. Enfin, pour le problème de décision, un Algorithme Génétique, piloté de façon floue, permet de faciliter la tâche au décideur lorsque celui-ci n'est pas sûr de la meilleure stratégie à adopter pour son environnement. Le chapitre est conclu par l'exposé des différents résultats obtenus après implémentation des méthodes proposées sur des exemples de différentes tailles.

10.1 METHODES DE RESOLUTION POUR PCPC

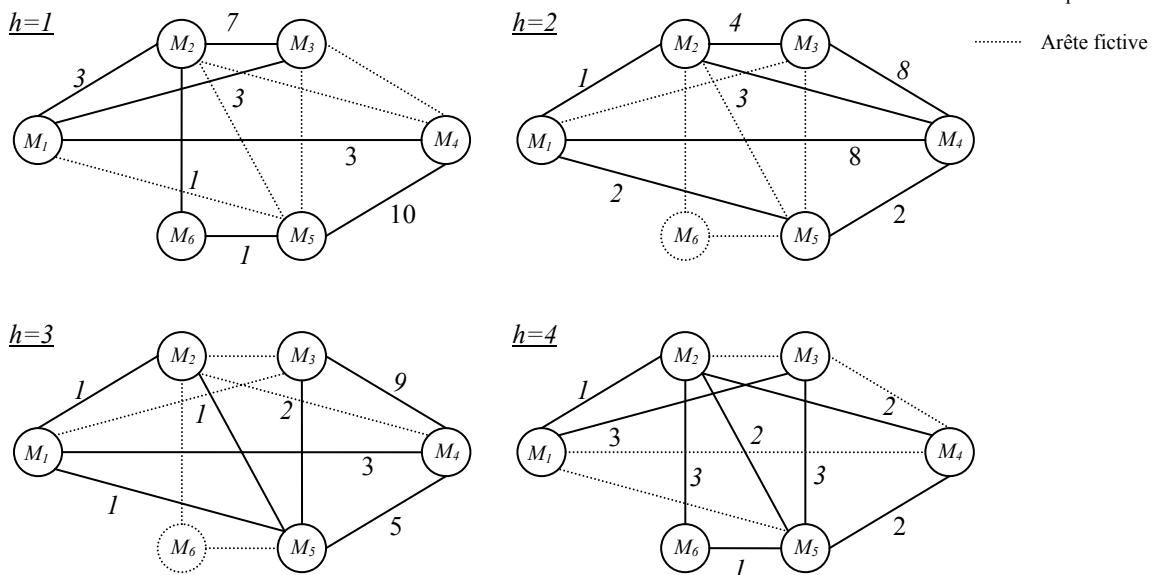
La proposition 9.2 (voir [paragraphe 9.8](#)) possède un résultat direct : il est possible de résoudre *PCPC* en lui appliquant n'importe quelle méthode de résolution du problème de composition statique. Pour ce faire, on construit une instance de graphe de flux déduite des H graphes de flux de l'horizon de planification.

Par exemple, la figure 10.1 illustre quatre graphes de flux associés à un système de production dont l'horizon de planification est sujet à quatre scénarios de demande avec des périodes de durées 4,2,4,2 mois respectivement (voir fig. 10.1.a. pour les graphes de flux associés). Quand le décideur impose une stratégies passive, le problème revient à résoudre une instance de problème statique dont le graphe de flux est déduit des précédents graphes en considérant pour chaque arête e_{kl} , joignant la machine M_k à M_l , le poids défini par l'équation

$$W(e_{kl}) = \sum_h \left(d_h / \sum_h d_h \cdot \sum_{e_{kl} \in E} W_h(e_{kl}) \right) \cdot W_h(e_{kl}).$$

C'est à dire, égal à la somme des flux normalisés reliant ces machines dans les quatre graphes, pondérés par les durées de périodes (voir fig. 10.1.b.).

a) Les graphes de flux



b) La matrice et le graphe de flux du problème statique

	M_1	M_2	M_3	M_4	M_5	M_6
M_1	.	0.0482	0.0542	0.1325	0.0301	0
M_2	.	.	0.0904	0.2169	0.0301	0.0422
M_3	.	.	.	0.1506	0.0482	0
M_4	.	.	.	0.1386	0	
M_5	0.0181	
M_6	

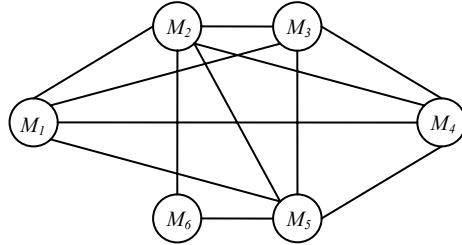


Figure 10.1

Procédure de conversion de PCPC en un problème statique.

Dans le cas où on utilise pour PCPC une formulation qui ne satisfait pas la [proposition 9.2 \(voir remarque 9.3\)](#), bien qu'on puisse construire des méthodes spécifiques à ce problème, il est possible d'utiliser les méthodes de résolution, plus générales, de PCAC et de les contraindre pour qu'elles donnent des séquences de compositions statiques. C'est pour cette raison que, dans ce qui suit, on ne s'intéressera qu'au problème PCAC et celui de décision.

10.2 METHODES DE RESOLUTION POUR PCAC

Nous adopterons pour la définition de PCAC une approche qui consiste à agréger les fonctions objectif des deux critères considérés ([voir paragraphe 9.6.3](#)) dans une somme pondérée. Ceci a pour effet de modifier l'objectif du modèle pour l'énoncer comme suit :

Soit CS l'ensemble des séquences de compositions qui respectent les contraintes décrites dans le [paragraphe 9.6.2](#). Le problème est de trouver une séquence de compositions $S^* \in CS$ telle que :

$$Z(S^*) = \min_{S \in CS} (w_1 \cdot TE(S) + w_2 \cdot RE(S))$$

où w_1 et w_2 sont deux nombres fractionnaires définis par le décideur et dont la somme est égale à un. En d'autre termes, on cherche une séquence qui vérifie toutes les contraintes et qui réalise le minimum de la somme pondérée du trafic intracellulaire et du coût de reconfiguration. Notons que les paramètres fractionnaires de pondération permettent d'attribuer à chaque critère l'importance relative qu'il mérite. Le choix de l'approche de la somme pondérée est justifié par son efficacité et la simplicité d'implémentation. En plus, nous proposerons, dans le [paragraphe 10.5](#), un procédé qui permet d'atténuer l'une des principales critiques à l'encontre de l'approche évoquée, à savoir, la difficulté de détermination des poids et la sensibilité de la méthode à leur modification. A cet effet, l'information de l'importance de l'un des critères par rapport à l'autre, véhiculée par ces poids, sera prise en compte de façon floue.

En considérant cette approche, nous présentons deux méthodes de résolution : la première s'inspire des travaux de Rosenblatt réalisés dans la cadre du problème d'agencement dynamique d'ateliers (dynamic plant layout problem) [Rosenblatt1986], la deuxième est une généralisation de l'AG qui a donné des résultats satisfaisants pour le problème de composition statique ([voir chapitre 7](#)).

10.2.1 La méthode du plus court chemin

10.2.1.1 Description

Une instance du problème peut être représentée par un graphe H -partite $G_{\text{par}} = (\bigcup_h XS_h, US)$ dans lequel chaque sous-ensemble XS_h , $h=1..H$, de sommets est constitué des compositions réalisables possibles C_{ih} (C_{ih} est la $i^{\text{ème}}$ composition réalisable possible durant la $h^{\text{ème}}$ période, $i \in \{1, \dots, T_h\}$ et T_h est le nombre des compositions considérées dans la période h). L'ensemble des arcs US est défini pour tous les couples ordonnés de sommets de la forme $(C_{ih}, C_{j(h+1)})$, c'est-à-dire les sommets représentatifs de compositions réalisées dans des périodes consécutives.

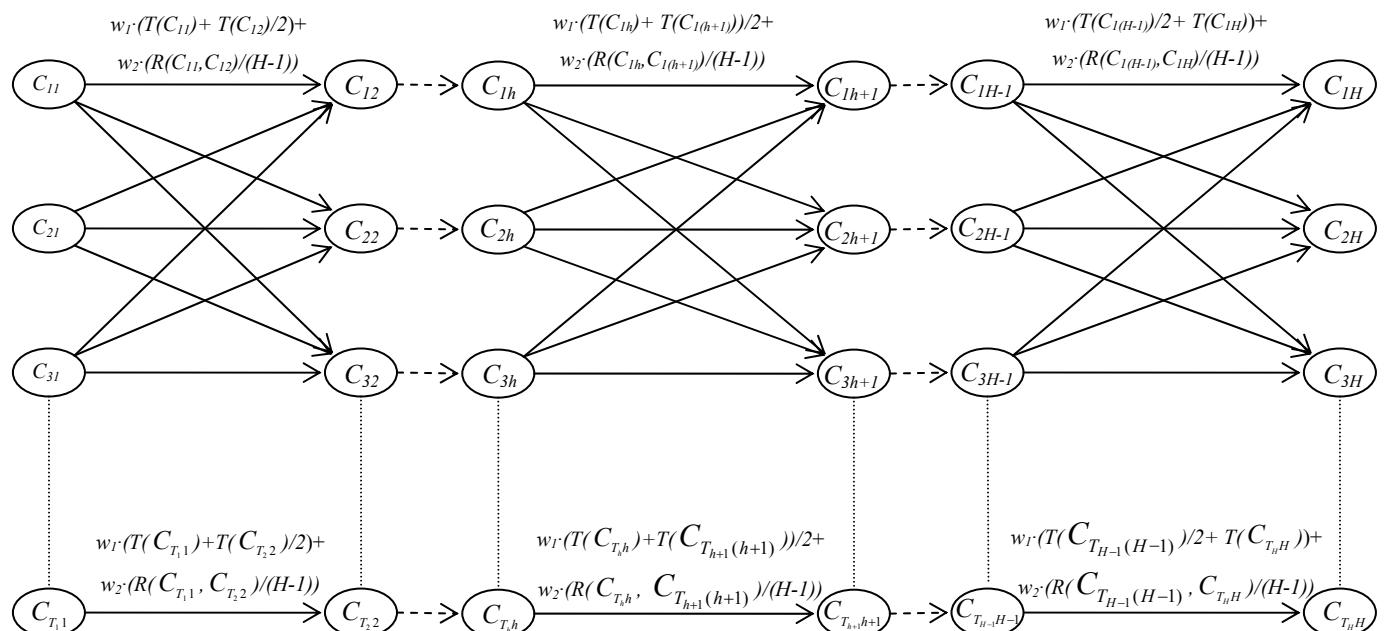


Figure 10.2 Le graphe H -partite de la méthode du plus court chemin.

En outre, une pondération des arcs de G_{par} est définie comme suit : Pour les arcs non incidents à un sommet du premier et du dernier ($H^{\text{ème}}$) sous-ensemble, on associe une valeur égale à la somme pondérée du coût de transition entre les deux compositions, $R(C_{ih}, C_{j(h+1)})$ (voir [paragraphe 9.7](#)), et la moyenne de leurs trafics intercellulaires, i.e. $w_2 \cdot R(C_{ih}, C_{j(h+1)})/(H-1) + w_1 \cdot (T(C_{ih}) + T(C_{j(h+1)}))/2$. Pour les arcs incidents à un sommet du premier et du dernier sous-ensemble, ces valeur sont définies par $w_2 \cdot R(C_{i1}, C_{j2})/(H-1) + w_1 \cdot T(C_{i1}) + w_1 \cdot T(C_{j2})/2$ et $w_2 \cdot R(C_{i(H-1)}, C_{jH})/(H-1) + w_1 \cdot T(C_{i(H-1)})/2 + w_1 \cdot T(C_{jH})$ respectivement (voir [fig. 10.2](#)).

Cette définition des poids des arcs implique que la longueur d'un chemin démarrant d'un sommet du premier sous-ensemble et aboutissant à un sommet du sous-ensemble H est égale à la fonction objectif associée à la solution construite par les compositions des sommets parcourus. Ainsi, déterminer une solution optimale pour PCAC revient à déterminer un plus court chemin, allant d'un sommet du premier sous-ensemble et arrivant à un sommet du dernier sous-ensemble, dans le graphe H -partite.

Cependant, pour des problèmes de grande taille (voire de taille moyenne), le nombre compositions devant être considérées dans chaque période est si important qu'il devient très coûteux d'énumérer toutes les solutions possibles. En effet, dans [Venugopal1992b] les auteurs montrent que le nombre de manières avec lesquelles on peut affecter m machines à J cellules est donné par le nombre de Stirling du second ordre, à savoir,

$$S^{(J)} = \frac{1}{J!} \sum_{j=1}^J (-1)^{J-j} C_J^j j^m .$$

Pour simplement 19 machines, ce nombre donne approximativement 11.259.666.000 compositions possibles [Mungwattana2000]. Ceci pour un nombre J fixe de cellules. Pour un nombre de cellules variables, il faut compter $\sum_{J=J_{\min}}^{J_{\max}} S^{(J)}$ compositions (J_{\min} et J_{\max} sont les bornes inférieure et supérieure au nombre de cellules).

En exploitant un théorème dû à Sweeney [Sweeney1976] traitant le problème dynamique de localisation d'entrepôts (dynamic multiple warehouse, or facility, location problem), Rosenblatt affirme qu'il n'est pas nécessaire d'énumérer tous les sommets possibles. Par conséquent, il est possible de faire la recherche d'un plus court chemin sur un nombre restreint de sommets. Pour exposer le théorème, dans un contexte "composition de cellules", nous avons à définir les notions suivantes : Soit Z_{hr} la valeur de la fonction objectif de la r -ième meilleure solution du problème statique associé à la période h , et soit $Z^{inf} = \sum_h Z_{h1}$, la somme des coûts des solutions optimales de chaque période h . Du fait qu'aucun coût de transition n'est considéré, il est clair que Z^{inf} est une borne inférieure pour la valeur de la solution optimale du problème PCAC. Et soit Z^{sup} , une borne supérieure de la valeur de la solution optimale de PCAC. Dans ce cas le théorème évoqué s'énonce comme suit :

Théorème 10.1 [Sweeney1976]

Si $K = Z^{sup} - Z^{inf}$ et T_h est donnée par $(Z_{hT_h} - Z_{h1} \leq K)$ et $(Z_{h(T_h+1)} - Z_{h1} > K)$, alors dans n'importe quelle période, aucune solution (composition) de rang n , $n > T_h$, ne peut appartenir à la solution optimale de PCAC. \square

La preuve de ce théorème peut être consultée dans [Sweeney1976].

Afin de tirer profit du théorème, nous devons effectuer les deux tâches suivantes :

- Déterminer une bonne borne supérieure Z^{sup} pour la valeur de Z .
- Déterminer les T_h meilleures solutions du problème de composition de cellules dans les SCPs statiques pour chaque période h ($h=1..H$).

a) Détermination d'une borne supérieure Z^{sup} : La borne supérieure peut être déterminée de plusieurs façons.

Ainsi, une première possibilité consiste à considérer les meilleures compositions associées à chaque période en leur ajoutant les coûts de reconfiguration. Une autre possibilité consiste à maintenir inchangée, durant les H périodes, une solution optimale associée à l'une quelconque des périodes (suggestion de Rosenblatt pour le problème du layout [Rosenblatt1986]). Le plus intéressant est d'utiliser le résultat de la [proposition 9.2](#) pour déterminer la solution optimale du problème dynamique avec stratégie passive *PCPC* (suggestion de Batta pour le layout [Batta1987]). La valeur de celle-ci est une borne supérieure pour la solution optimale de *PCAC*, car, comme nous l'avons souligné dans le [paragraphe 9.9](#), *PCPC* en est un cas particulier.

b) Détermination des T_h meilleures solutions : Les T_h meilleures solutions sont déterminées à partir de la formule donnée dans le théorème précédent : $(Z_{hT_h} - Z_{h1} \leq K)$ et $(Z_{h(T_h+1)} - Z_{h1} > K)$ où $K = Z^{sup} - Z^{inf}$.

L'amélioration de ces deux bornes constitue un bon sujet de recherche à développer si on veut investir dans le domaine des méthodes exactes. On peut, à cet effet, songer à utiliser les informations spécifiques du problème de composition. Un travail [Hormozi1996] a été réalisé dans ce sens pour l'amélioration des bornes de Sweeney, en ce qui concerne le problème dynamique de location d'entrepôts.

10.2.1.2 Remarques et critiques

Comme nous l'avons signalé, l'application de cette méthode requiert la résolution du problème statique de composition de cellules pour chaque période de façon optimale. Or, tenant compte de la complexité du modèle, la méthode présentée, pour être concrétisée pour des problèmes de grande taille, ne peut faire l'objet que d'une approche heuristique. Par conséquent, il sera question de déterminer pour chaque période h , non pas les T_h meilleures solutions, mais un sous-ensemble de bonnes solutions dont le nombre T_h devient un paramètre réglable à définir de façon empirique.

Notons que la méthode de Chen [Chen1998] est similaire à la méthode du plus court chemin que nous proposons à la différence que l'auteur définit, comme pour Sweeney [Sweeney1976] et Rosenblatt [Rosenblatt1986], un modèle en utilisant la programmation dynamique. Aussi, Chen restreint T_h à H en considérant uniquement les H compositions optimales déduites en résolvant un problème statique dans chaque période et ce en utilisant son scénario séparément.

Pour des instances de grande taille, le nombre limité de compositions à considérer pour chaque période devient un handicap, conduisant à une exploration pauvre de l'espace de recherche. Par conséquent, il serait plus intéressant d'investir plus d'efforts dans la catégorie des métaheuristiques qui ont prouvé leur capacité à explorer efficacement des espaces complexes et de grande taille. Ceci sera accompli dans le paragraphe suivant en utilisant une approche basée sur les algorithmes génétiques.

10.2.2 Un AG classique

Les algorithmes génétiques sont des méthodes d'optimisation basées sur une simulation du phénomène de la sélection naturelle. Le principe de base de ces méthodes a été décrit au [chapitre 5 \(voir paragraphe 5.1\)](#). Ensuite, des applications de celles-ci au cas statique ont été proposées [aux chapitres 7 et 8](#).

Dans ce qui suit, nous décrivons les différents points de mise en application des AGs pour le traitement de PCAC, en insistant sur les aspects qui sont différents de ceux déjà présentés dans [le chapitre 7 \(voir paragraphe 7.2\)](#).

10.2.2.1 Codage binaire

Une séquence de compositions peut être représentée par un vecteur chromosome de dimension $|E| \times H$, E étant l'ensemble des arêtes d'un réseau N_h (i.e. graphe de flux associé à la période h , [voir paragraphe 9.6.1](#)) et H le nombre de périodes. Le vecteur est ainsi composé de H parties de longueur $|E|$ où chaque partie décrit la composition de cellules adoptée dans la période associée. Chaque allèle est, donc, associé à une arête du graphe de flux associé à la période en question. Un allèle d'une partie h , $h=1..H$, du vecteur indique si l'arête correspondante est intracellulaire ou non durant la $h^{\text{ème}}$ période. Un exemple d'une telle codification est présenté dans la figure 10.3.

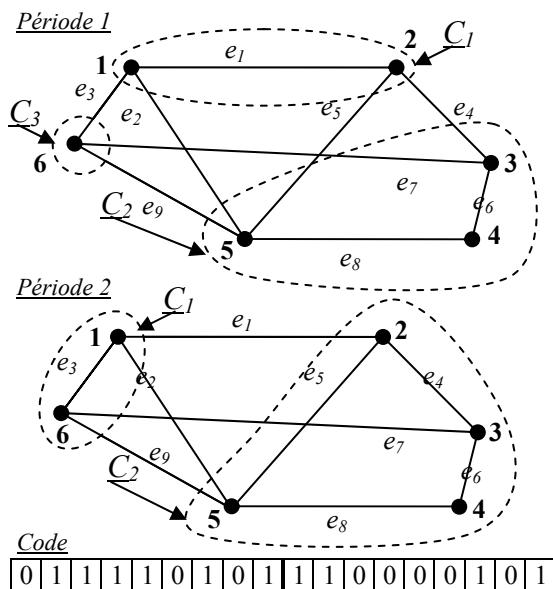


Fig. 10.3 Une instance avec son codage binaire.

Dans cet exemple, un système de production, dont l'horizon de planification est subdivisé en deux périodes, est considéré. La composition adoptée, durant la première période, possède trois cellules tandis que celle de la deuxième période en a deux. Du moment que le nombre d'arêtes est 9, la longueur des chromosomes est $9 \times 2 = 18$. La première partie du code, qui concerne la première période, montre qu'il y a trois arêtes intracellulaires, à savoir les arêtes associées aux allèles nuls e_1, e_6, e_8 , et six arêtes intercellulaires, i.e. les arêtes des allèles non nuls $e_2, e_3, e_4, e_5, e_7, e_9$. D'autre part, la partie associée à la deuxième période montre cinq arêtes intracellulaires, i.e. e_3, e_4, e_5, e_6, e_8 , et quatre arêtes intercellulaires, i.e. e_1, e_2, e_7, e_9 . La composition de cellules de chaque période est définie par les composantes connexes du graphe de flux induites par les arêtes intracellulaires.

10.2.2.2 Population initiale

Nous avons opté pour une génération aléatoire de la population initiale. Cependant, la probabilité P_1 de génération des allèles non nuls (arêtes *intercellulaires*) est mise à une valeur plus grande que celle de la génération des allèles nuls (arêtes *intracellulaires*). Ceci est justifié par le souci de vouloir prévenir les compositions unicellulaires car m arêtes intercellulaires suffisent pour connecter tout le graphe.

10.2.2.3 La fitness

Pour les raisons décrites dans le [paragraphe 7.2.3](#), il est préférable de ne pas utiliser directement la fonction objectif du problème. Mieux vaut la transformer moyennant un procédé qui permet de tenir compte des solutions non réalisables, afin de profiter de l'éventuel bon code génétique qu'elles peuvent cacher, sans pour autant les rendre meilleures que n'importe quelle solution réalisable.

Cependant, le recours à la définition exacte du coût de reconfiguration (voir [paragraphe 9.7](#)) n'est pas en accord avec l'objectif d'efficacité. En effet, l'AG étant une méthode par nuage de points, nous aurons besoin, pour lancer l'opérateur de sélection, de mener $K \cdot (H-1)$ résolutions de problèmes de couplage de poids maximum (pcm), K étant la taille de la population et H le nombre de périodes (voir [paragraphe 9.7.2](#)). Pour exécuter l'algorithme de résolution de pcm , qui est en $O(m^{2.5})$, nous avons besoin de réaliser des traitements sur le code génétique binaire afin de déterminer le graphe biparti associé à pcm . Ces traitements consistent à

- déterminer les compositions de cellules relatives aux H portions du code binaire. Asymptotiquement, chaque composition est déterminée avec une complexité de calcul en $O(m^3)$.
- déterminer les poids des arêtes des $(H-1)$ graphes bipartis associés aux transitions. Asymptotiquement, chaque graphe requiert un $O(m)$ de calculs.

Ceci donne en tout une complexité en $O(K \cdot H \cdot m^3)$.

Il serait, par ce fait, préférable d'opter pour une estimation rapide du coût de reconfiguration pour l'utiliser dans le procédé d'optimisation génétique et de ne recourir à une évaluation exacte que de façon exceptionnelle. Par exemple, lorsqu'une nouvelle bonne solution est rencontrée. Notons qu'une autre raison valable pour cette démarche est le fait qu'une évaluation du coût de reconfiguration, quoique exacte, n'est en réalité qu'approchée car l'exactitude, au sens propre du terme, requiert des informations concernant l'emplacement des machines qui ne peuvent être disponibles que dans l'étape d'agencement intracellulaire.

A cet effet, nous proposons de substituer un coefficient de dissimilitude au coût de déplacement exact. Pour ce faire, on codifie les compositions, d'une manière adéquate, pour utiliser ensuite une fonction qui mesure la dissimilitude entre les codes des compositions. Par exemple, on peut considérer pour chaque composition C_h , $h=1..H$, la matrice B_h d'ordre $m \times m$, définie comme suit :

$$B_{hij} = \begin{cases} 1 & \text{si } M_i \text{ et } M_j \text{ sont dans la même cellule durant la période } h \\ 0 & \text{autrement} \end{cases}$$

Cette codification est appropriée car elle permet de décrire fidèlement les groupements de machines sans ajouter d'information d'indexation qui est à l'origine de la redondance déroutante. Ensuite, en utilisant une distance qui évalue la dissimilitude entre les codes des compositions consécutives, C_h et C_{h+1} , nous obtenons le coût de reconfiguration. Pour cette fin, la formule suivante, qui est en $O(m^2)$, utilise une distance de Hamming normalisée :

$$R(C_h, C_{h+1}) = \frac{2}{(m-1) \cdot m} \cdot \sum_{i=1}^{m-1} \sum_{j=i+1}^m |B_{hij} - B_{(h+1)ij}|.$$

Dans le but de donner au déplacement de certaines machines une plus grande influence dans la nouvelle définition du coût de reconfiguration, nous pouvons utiliser une codification qui considère essentiellement les couples de machines reliées par un flux de produits car c'est le déplacement de ces machines qui influe sur la performance du système de production. Une telle codification peut être procurée par le codage binaire des compositions. En effet, au lieu de considérer tous les couples de machines de la matrice B pour la codification d'une composition donnée, nous pouvons considérer celles qui sont reliées par un trafic positif dans au moins une période. Ceci permet de donner plus d'importance à ce genre de machines dans la définition du coût de reconfiguration puisque leur jumelage (ou leur séparation) a une plus grande influence sur la fonction objectif. Nous estimons que ces considérations peuvent être respectées par la formule suivante :

$$R(C_h, C_{h+1}) = \left(\frac{1}{\text{card}(E)} \cdot \sum_{l=1}^{\text{card}(E)} |ev_{hl} - ev_{(h+1)l}| \right),$$

où ev_{hl} est la valeur du l -ième allèle de la h -ième portion du code génétique de la solution en cours d'évaluation. Cette formule peut être calculée en $O(m^2)$.

Notons que cette alternative ne tient pas compte des coûts de déplacement de machines. Dans le cas où ces coûts sont suffisamment différents, inclure l'information de ces derniers dans l'évaluation du coût de reconfiguration, de façon suffisamment efficace pour justifier le non recours à une évaluation exacte, est une question intéressante à développer.

10.2.2.4 Les opérateurs génétiques

a) Croisement

Nous utilisons un croisement multi-points qui donne deux individus fils à partir de deux individus parents selon l'algorithme suivant :

étape 1 :

Choisir aléatoirement :

1. deux individus (les parents);
2. un nombre nc de points de coupure ($1 \leq nc \leq H$);
3. nc points de coupure c_1, \dots, c_{nc} , au plus un dans chaque période.

étape 2 :

Le premier fils est construit par la concaténation des c_1 premiers allèles du premier parent avec les allèles de rang c_1+1 à c_2 du second, puis les allèles de rang c_2+1 à c_3 du second, et ainsi de suite. Le deuxième fils est construit par la concaténation du restant du code des parents.

Cette procédure est répétée avec une autre paire jusqu'à ce que $s \times P_3$ individus soient remplacés où P_3 est une valeur entre 0 et 1.

b) La mutation

Le nombre d'individus, qui subissent cet opérateur, est déterminé par la valeur du produit $s \times P_4$ où P_4 est une valeur entre 0 et 1. Deux parties de code, associées à deux périodes différentes de chaque individu choisi, sont échangées.

10.2.2.5 Les paramètres de contrôle

Les paramètres de contrôle, qui sont

- s définissant la taille de la population,
- i_{max} donnant le nombre de génération,
- P_1 et P_2 déterminant les probabilités de génération aléatoire des allèles,
- P_3 définissant le taux de sélection des individus à croiser et enfin
- P_4 décrivant le taux de mutation,

dépendent de l'instance du problème considéré. Par Conséquent, quelle que soit l'attention prêtée pour leur détermination, l'expérimentation est généralement nécessaire pour choisir les valeurs de ces paramètres qui pousseront l'AG à donner le meilleur de lui-même.

10.3 AMELIORATIONS FLOUES

Nous avons évoqué que *PCPC* est un cas particulier de *PCAC*. Par conséquent, résoudre *PCAC*, de façon optimale, résoudrait le problème de décision entre stratégies passive et active. Cependant, le souhait de résoudre *PCAC*, de façon optimale, est un objectif trop optimiste à cause de la complexité du problème qui s'avère être NP-complet (voir paragraphes 6.5 et 9.9). En outre, plus grande est la taille du problème, plus grande est la proportion de solutions actives comparées à celles qui sont passives¹. Par conséquent, la recherche d'une bonne solution, dans le cas où celle-ci venait d'être passive ou pseudo-passive², ressemble à la recherche d'une aiguille dans une botte foin. Cette réalité nous mène à essayer de contrôler l'AG de façon implicite. Cependant, dans le souci de maintenir pour l'AG le principe d'exploration aléatoire, ce contrôle doit être assez souple. Par conséquent, nous proposons d'utiliser un pilotage par logique floue pour inciter l'AG à accentuer l'exploration des régions les plus prometteuses.

Dans le domaine vaste de l'optimisation, donner un coup de main à l'AG, en utilisant la logique floue, est une idée qui ne date pas d'hier. Cet apport flou se voulait un moyen pour accélérer la convergence de l'AG et/ou de l'aider à atteindre des solutions de meilleure qualité [Richter2003]. Le principe général de ces travaux est d'adapter de façon dynamique les valeurs des opérateurs génétiques en utilisant des règles d'inférence floues dans le but de déterminer les valeurs qui améliorent les performances de l'AG. Parmi les travaux publiés, nous pouvons trouver l'adaptation de la taille de la population [Lee1993, Lee1995], le taux de croisement [Lee1995, Wang1996] et le taux de mutation [Lee1993, Lee1995, Arnone1994]. Ces efforts ont permis de construire des AGs surclassant de façon significative les AGs conventionnels. Cependant, bien qu'ils soient intéressants, ces travaux ne considèrent pas les causes qui "dopent" l'AG dans sa course à la quête de bonnes solutions mais leurs effets. En effet, les modèles flous

¹ Une solution est *active* si elle est constituée d'une séquence de compositions différentes. Elle est dite *passive* si ses compositions sont toutes identiques.

² Nous entendons par *pseudo-passive* une solution dans la majorité des transitions est réalisée entre des compositions identiques.

cités utilisent comme input soit la vitesse de convergence, le degré de diversité de la population ou la valeur moyenne de la fitness. Ces paramètres sont des indicateurs qui permettent de contrôler la performance de l'AG qui doit assurer une bonne diversité du code génétique pour éviter une convergence prématuée. Ainsi, si la convergence est trop rapide, on procède, par exemple, à l'augmentation de la taille de la population ou du taux de mutation. Nous estimons qu'il serait judicieux de prendre en compte les causes qui poussent l'AG à réaliser les bonnes performances en définissant des opérateurs spécifiques et de ne pas se contenter de remèdes superficiels. Cependant, rien n'empêche de combiner les deux approches pour construire un AG plus performant.

Avant de se lancer dans la mise en œuvre de l'idée proposée, il serait préférable de résumer l'essentiel des notions théoriques liés à la logique floue inférentielle. Pour de plus amples détails, on peut consulter [Kaufmann1975, Buhler1993, Ying2000] pour lesquels le paragraphe suivant présente une synthèse.

10.3.1 Recueils de la théorie des sous-ensembles flous

La théorie des sous-ensembles flous a été introduite en 1965 par le mathématicien Zadeh L., professeur à l'université de Californie à Berkeley, dans son article intitulé « Fuzzy Sets » [Zadeh1965]. Les premières applications industrielles datent des années soixante-dix. Mais, l'essor applicatif et médiatique n'est apparu que vers les années quatre-vingt grâce aux travaux de Mamdani [Mamdani1977].

La vocation de la théorie des sous-ensembles flous est de représenter et de manipuler des données imprécises et incertaines. Les approches traditionnelles ont, en effet, du mal à traiter des concepts mal définis. Par exemple, un système de régulation de la température a besoin qu'on lui indique si 32° représente du « chaud » et jusqu'où s'étend cette notion.

La manière d'exprimer en mathématique classique la notion de plage de température consiste à considérer l'ensemble des températures, à le partitionner, puis à décider de l'appartenance d'une valeur donnée à l'une des classes. L'intérêt de l'approche « Floue » vient du fait qu'un élément peut appartenir simultanément à plusieurs classes à des degrés divers.

Les éléments de base de la logique floue peuvent être résumés en trois étapes [Zadeh1975] :

1. La fuzzification.
2. Les inférences.
3. La défuzzification.

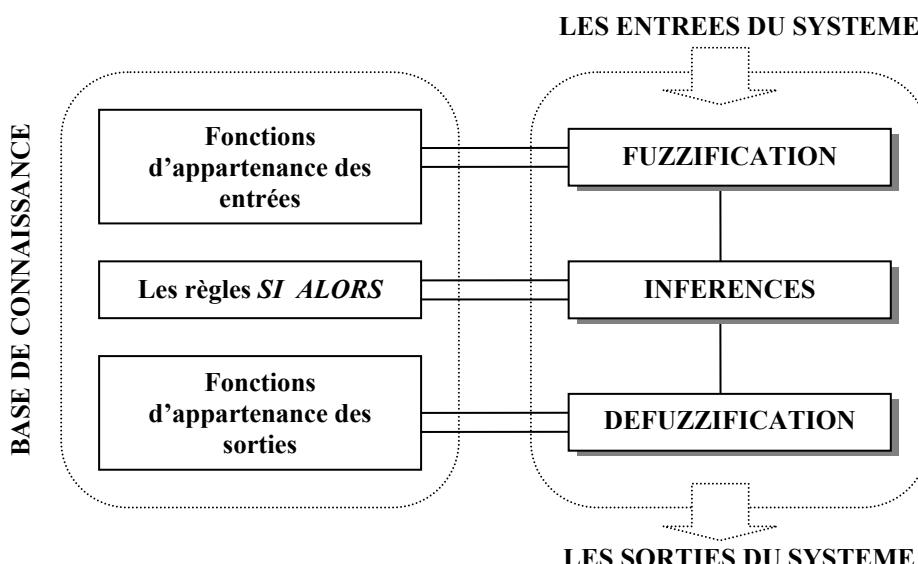


Figure 10.4

Les fondements de la logique floue.

10.3.1.1 La fuzzification

Cette étape consiste en la transformation des entrées du problème en nuances qualitatives. Elle permet de passer des variables réelles à des variables floues qu'on appelle *variables linguistiques*.

a) Variables linguistiques

La description d'une certaine situation ou d'un procédé contient en général des expressions floues comme : grand, petit, lent, rapide, proche, loin, etc. Les expressions de ce genre forment les valeurs des variables linguistiques. Une variable linguistique est donc une variable dont les valeurs ne sont pas des nombres mais des mots du langage naturel ou artificiel. A titre d'exemple, la variable linguistique *Température* peut, par exemple, prendre les valeurs {*Froid*, *Très chaud*} ou bien {*F*, *T*}. Pour chaque valeur d'une variable linguistique, on associe une restriction floue qui représente les valeurs les plus ou moins compatibles avec celle-ci. Cette restriction est définie par la notion de *fonction d'appartenance*.

b) Fonction d'appartenance

Dans la théorie des ensembles, la fonction caractéristique d'un ensemble est booléenne : un élément appartient ou n'appartient pas à un ensemble donné. Par conséquent, le fait d'appartenir à une classe exclut le fait d'appartenir à une classe différente. En logique floue, cette notion est étendue en une fonction pouvant prendre toutes les valeurs comprises entre 0 et 1, appelée *fonction d'appartenance*. Cette fonction notée $\mu(\cdot)$ indique le degré d'appartenance d'un élément à un ensemble. Ainsi, pour la variable linguistique *Température*, chaque valeur représente un sous-ensemble flou. A titre d'exemple (voir fig. 10.5), une température 16° appartient avec un facteur d'appartenance $\mu = 0.7$ à l'ensemble « *Froid* » et avec $\mu = 0.3$ à l'ensemble « *Très chaud* ».

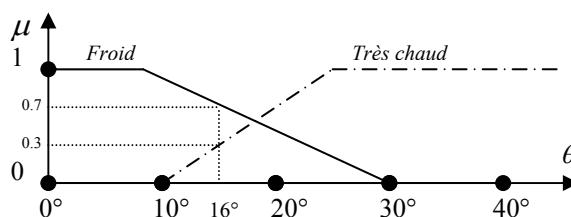


Figure 10.5 Fonctions d'appartenance.

Un des problèmes que posent les personnes qui s'initient à la théorie des ensembles flous est la manière de déterminer ces fonctions d'appartenance. Malheureusement, celle-ci ne fournit aucune méthode systématique pour déterminer ces fonctions. Cependant, l'expérience montre que la forme la plus utilisée est la forme trapézoïdale ou triangulaire (voir fig. 10.6).

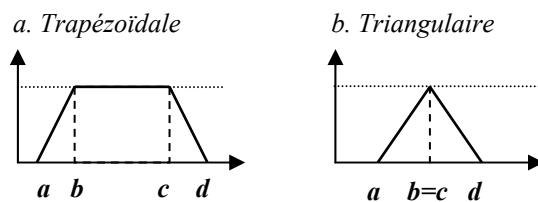


Figure 10.6 Quelques formes de fonctions d'appartenance.

10.3.1.2 Les inférences

En général, les variables linguistiques sont liées entre elles par des règles. Ce qui nous permet de tirer, à partir de ces règles, des conclusions. On parle, alors, de déductions floues ou d'inférences. Ces règles peuvent être exprimées sous la forme générale :

$$\begin{aligned} \text{opération} \leftarrow & \underline{\text{SI condition1, ALORS opération1, OU}} \\ & \underline{\text{SI condition2, ALORS opération2, OU}} \\ & \dots \\ & \underline{\text{SI condition m, ALORS opération m)}} \end{aligned}$$

Les *conditions* dépendent d'une ou de plusieurs variables définies dans la partie fuzzification. Ces variables sont liées entre elles par des opérateurs ET, OU ou NON. Ces opérateurs, dont les règles sont bien connues en logique classique, nécessitent une toute autre spécification dans le langage flou. Il existe une panoplie de méthodes réalisant ces opérateurs. Le plus souvent les opérateurs ET et OU sont réalisés respectivement par les règles « min » et « max » due à Mamdani [Mamdani1977].

Les *opérations* sont des décisions simples ou floues. Dans ce dernier cas, ces opérations sont associées à des variables linguistiques auxquelles on associe des fonctions d'appartenance. La procédure qui permet d'obtenir une décision simple (non floue) à partir des décisions floues est appelée *défuzzification*.

10.3.1.3 La défuzzification

Cette dernière étape consiste à prendre une décision : un résultat bien déterminé à partir de l'information floue déduite des inférences de la deuxième étape. Dans ce cas aussi, il existe plusieurs façons d'appréhender le problème. Les deux façons les plus courantes sont : le centre de gravité et la moyenne des maximums [Ying2000].

Faire appel au raisonnement flou, pour la mise en œuvre d'un AG piloté en tenant compte des causes de la bonne performance et non pas de ces effets, requiert une discussion concernant les inputs influant sur le bon déroulement de l'AG.

10.3.2 Détermination des informations influentes

Le fait qu'une transition soit passive ou active (c'est à dire, qu'on maintient ou non la même composition de cellules) dépend essentiellement des données définissant les deux scénarios de demandes associés aux deux périodes de ladite transition. En effet, chaque scénario engendre un trafic de produits entre machines, représenté par le graphe de flux. En observant de près les deux graphes de flux associés à chaque transition, on peut constater que si la concentration des flux est « presque » la même dans les deux graphes, les compositions de cellules seraient probablement similaires. Par conséquent, si on représente chaque graphe par une carte géographique indiquant le relief, on peut prévoir une transition passive pour deux cartes « proches » l'une de l'autre.

Il y a, en outre, une autre information qui n'est moins influente. En effet, si le coût de reconfiguration est « très prohibitif », recourir à une transition passive est très probable même si les similarités des graphes de flux ne sont pas aussi importantes. Par conséquent, l'utilisation de ces deux informations dans le pilotage de la recherche peut procurer une aide considérable pour le contrôle de n'importe quelle heuristique de résolution.

10.3.2.1 Extraction d'informations depuis les scénarios de l'horizon de planification

Chaque scénario engendre un graphe de flux dont les informations sont décrites fidèlement par la matrice de flux, qui est une matrice d'ordre $m \times m$ définissant pour chaque couple de machines la quantité du trafic de produits

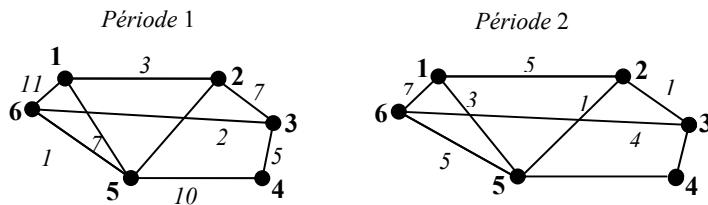
qui les relie. Cependant, puisque la matrice de flux est symétrique, nous pouvons la compacter en représentant seulement les composantes qui sont au dessus de la première diagonale. Ceci conduit à la définition d'une matrice compacte de dimension $(m/2) \times (m-1)$ ou $m \times ((m-1)/2)$, selon la parité du nombre de machines m .

Arrivé à ce stade, nous devons constater que l'information extraite des graphes de flux doit refléter les zones de concentration de flux et non pas leur grandeur. En effet, si, pour un graphe de flux donné, nous ajoutons une même valeur à chaque poids d'arête, une solution optimale pour le graphe d'origine serait nécessairement optimale pour le second. Par conséquent, il serait judicieux de normaliser les valeurs de la matrice compacte. A cette fin, nous proposons de soustraire de toutes ces valeurs leur minimum, ensuite de les diviser par leur maximum. Cette opération consiste à changer la valeur de toute arête comme suit :

$$t_{ijh} \leftarrow \frac{t_{ijh} - \min_{(i,j)} t_{ijh}}{\max_{(i,j)} t_{ijh} - \min_{(i,j)} t_{ijh}}$$

où t_{ijh} est le poids de l'arête reliant les machines M_i et M_j dans le graphe de flux associé à la période h .

a) Graphes de flux



b) Matrices de flux brutes

	M_2	M_3	M_4	M_5	M_6		M_2	M_3	M_4	M_5	M_6
M_1	3	0	0	7	11		5	0	0	3	7
M_2		7	0	0	0		1	0	10	0	0
M_3			5	0	2		0	0	4		
M_4				10	0		0	0			
M_5					1					5	

c) Matrices de flux réduites

$$\begin{vmatrix} 3/11 & 0 & 0 & 7/11 & 1 \\ 7/11 & 0 & 0 & 0 & 5/11 \\ 0 & 2/11 & 10/11 & 0 & 1/11 \end{vmatrix} \quad \begin{vmatrix} 0.5 & 0 & 0 & 0.3 & 0.7 \\ 0.1 & 0 & 1 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0.5 \end{vmatrix}$$

d) Images de flux 3D

Figure 10.7

Des graphes de flux aux images de flux.

Dans ce qui suit, nous désignons la matrice, résultant de ces traitements, la matrice de flux (réduite). Si nous associons alors à chaque composante de cette matrice un niveau de gris approprié, elle peut être confondue à une

carte relief que nous appelons image de flux. La figure 10.7 montre les étapes qui permettent de générer des images de flux à partir de graphes de flux.

Pour chaque couple d'images consécutives, on définit le degré de proximité (ou similarité) par le calcul de la distance entre leurs matrices de flux. Si la distance les séparant est petite, les images de flux sont considérées « proches », sinon elles sont considérées « éloignées ». Dans le but d'exploiter cette information en utilisant la logique floue, on définit la variable linguistique *Distance* dont les valeurs sont {*Proche*, *Eloigné*}.

10.3.2.2 Extraction de l'information de l'importance du coût de reconfiguration

La deuxième information influente, à savoir, l'importance du coût de reconfiguration, peut être extraite depuis l'un des poids associés aux critères et définis par le décideur ([voir paragraphe 10.2](#)). On convient d'utiliser le poids w_2 associé au critère coût de reconfiguration. Si ce poids est proche de un, ceci indique que le coût de reconfiguration est très important en comparaison avec le critère coût du trafic intercellulaire. Par contre, s'il est proche de zéro, ce critère est considéré de faible importance. Cette information est exploitée par l'introduction de la variable linguistique *Reconfiguration* dont les valeurs sont dans l'ensemble {*Petite*, *Grande*} pour designer une « petite » et une « grande » importance du coût de reconfiguration respectivement.

Notons enfin la non consistance d'extraction de cette information du coût de déplacement de machines (moyennant leur moyenne ou leur somme) car le procédé de normalisation que nous adoptons dans la définition des fonctions objectif rend cette initiative infructueuse.

Remarque 10.1

Une autre information peut être aussi influente que les précédentes. Il s'agit des durées des périodes. En effet, il est évident qu'une longue période (disant qu'elle occupe la moitié de l'horizon) peut avoir beaucoup plus d'influence sur la séquence de compositions à adopter que si elle était courte (disons le un dixième). Cependant, le souci de simplification des règles d'inférence nous a poussé à inclure la prise en charge de cette information d'une façon qui sera décrite ultérieurement ([voir paragraphe 10.3.4.1](#)).

10.3.3 Les règles d'inférence

Dans le but d'exploiter les connaissances acquises dans les entrées floues, nous utilisons des règles d'inférence de la forme SI A ET B ALORS C. Ces règles sont appliquées pour chaque transition afin de tirer une réponse appropriée qui sera utilisée ensuite dans le pilotage de l'AG. Les règles sont définies comme suit :

SI « *Distance est Proche* » ET « *Reconfiguration est Petite* » ALORS « *Similarité est Moyenne* » OU,
SI « *Distance est Proche* » ET « *Reconfiguration est Grande* » ALORS « *Similarité est Grande* » OU,
SI « *Distance est Eloignée* » ET « *Reconfiguration est Petite* » ALORS « *Similarité est Faible* » OU,
SI « *Distance est Eloignée* » ET « *Reconfiguration est Grande* » ALORS « *Similarité est Moyenne* ».

Le décision à prendre est définie par la variable linguistique *Similarité*, qui peut prendre l'une des valeurs de l'ensemble {*Faible*, *Moyenne*, *Grande*}, pour designer un « *faible* », « *moyen* » ou « *grand* » degré de similarité devant lier les deux compositions de la transition considérée.

Dans le but d'avoir la valeur de sortie liée à la variable *Similarité*, étant données les valeurs d'entrée des variables *Distance* et *Reconfiguration*, on peut utiliser l'une des méthodes de défuzzification telle que la méthode max-min due à Mamdani [Mamdani1977].

10.3.4 L'AG flou

Incorporer le raisonnement flou à notre approche de résolution évolutionnaire, pour orienter la recherche vers les zones prometteuses, consiste à redéfinir les points suivants de l'AG décrits dans le [paragraphe 10.2.2](#) :

10.3.4.1 Le générateur d'individus

Le générateur d'individus est une procédure sollicitée pour avoir un nouvel individu soit pour la génération de tous les individus de la population initiale ([voir paragraphe 10.2.2.2](#)), ou dans l'étape de remplacement qui permet de mettre des individus nouveaux à la place de ceux non sélectionnés ([voir paragraphe 10.2.2.3](#)).

Chaque nouvel individu est généré séquentiellement, en commençant depuis une partie du code relative à une période choisie aléatoirement. La probabilité P_{cp} de choix d'une période dépend de sa durée et de la quantité relative de produits qui y sont fabriqués. Elle peut être définie par la quantité normalisée du flux total donnée par la formule suivante :

$$P_{cp} = \frac{d_h \cdot \sum_{e_{kl} \in E} W_h(e_{kl})}{\sum_h d_h \cdot \sum_{e_{kl} \in E} W_h(e_{kl})}.$$

Ceci permet de favoriser les périodes ayant relativement une longue durée et un volume de production considérable du moment que sur cette période de démarrage que le restant des compositions va être construit. En effet, la composition C_h associée à la période de démarrage, que nous appelons composition de base, est générée aléatoirement. Ensuite, en appliquant une perturbation sur le code génétique de C_h , nous obtenons la composition C_{h+1} . Cette perturbation est proportionnelle au résultat des règles d'inférence associées à la transition $(h, h+1)$. Par la suite, nous déterminons C_{h+2} de C_{h+1} en utilisant le résultat des inférences de la transition $(h+1, h+2)$, et ainsi de suite jusqu'à avoir la composition C_H . Le même procédé est répété dans le sens opposé pour avoir les compositions C_{h-1} , C_{h-2} , et ainsi de suite jusqu'à obtenir C_1 .

10.3.4.2 Le croisement

Utiliser l'opérateur de croisement pour diriger la recherche vers la zone des solutions passives ou active peut être réalisé en jouant sur le nombre de points de coupures et leurs emplacements. En effet, un croisement simple, avec un unique point de coupure, encourage le maintien des solutions passives ou pseudo passives puisqu'un nombre considérable de transitions des parents reste intact. D'autre part, accroître le nombre de points de coupure encourage les solutions actives du moment que la probabilité d'avoir une progéniture avec des compositions assez différentes augmente. En outre, l'implantation de ces points de coupure doit être encouragée dans les parties du code correspondant à des compositions qui ne sont pas similaires à leurs voisines, le résultat des inférences faisant foi.

A travers ces consignes, nous pouvons modifier l'algorithme de croisement de l'AG classique ([voir paragraphe 10.2.2.4](#)) par la redéfinition des étapes 1.2 et 1.3 qui déterminent le nombre de points de coupure et leur emplacement dans le code génétique de la façon suivante :

Soient r_1, r_2, \dots, r_{H-1} les résultats des inférences relatifs aux transitions $(1,2), (2,3), \dots, (H-1,H)$ respectivement. Pour chaque partie du code associé à une période h , $h=1..H$, est affecté un point de coupure avec la probabilité p_h définie comme suit :

$$p_h = \begin{cases} 1 - r_1 & \text{si } h = 1, \\ 1 - r_{H-1} & \text{si } h = H, \\ 1 - ((r_{h-1} + r_h)/2) & \text{autrement.} \end{cases}$$

En supposant que les allèles sont indexés de 1 à $H \times |E|$, l'emplacement de chaque éventuel point de coupure est, pour chaque période h , $h=1..H$, un élément uniformément choisi parmi l'ensemble des indexées $\{(h-1) \times |E| + 1, \dots, h \times |E|\}$.

Cette manière de faire permet d'éviter de positionner des points de coupures dans les parties de code des compositions qui ont une grande similarité avec les voisines, et incite à les mettre dans celles qui en ont une faible similarité.

10.3.4.3 La mutation

La mutation exploite les résultats des règles d'inférence r_i , $i=1..H-1$, en les utilisant pour encourager la permutation des codes des compositions consécutives qui sont suffisamment similaires. Ceci peut être accompli par le parcours du code génétique, de l'individu choisi pour subir une mutation, transition par transition. Pendant ce parcours, on permute progressivement les parties du code, relatives aux compositions incidentes, avec une probabilité égale au résultat des règles d'inférence de la transition associée.

10.4 RESULTATS ET PERFORMANCES

Dans cette section, des exemples sont présentés dans le but de montrer l'efficience de l'apport flou aux performances de l'AG. Une comparaison avec les méthodes décrites dans le [paragraphe 10.2](#) y est présentée. Une évaluation comparative avec des méthodes publiées dans le domaine de la composition cellulaire dynamique a été impossible pour différentes raisons : ces méthodes utilisent soit des fonctions objectif différentes ou requièrent des données additionnelles non justifiées pratiquement (tel que le nombre de cellules prédéterminé), ou ne tiennent pas compte de certaines contraintes, telles que la cohabitation et la non cohabitation entre machines. Cependant, dans le but de montrer l'efficacité de l'AG flou, nous le comparons avec la méthode du plus court chemin et avec l'AG binaire. Notons que la méthode du plus court chemin a été implémentée par génération aléatoire d'un même nombre de compositions réalisables dans chaque période.

Nous considérons deux situations opposées : La première est lorsque le décideur estime que le coût de reconfiguration possède une faible importance comparé au coût du trafic intercellulaire. Par conséquent, il propose les poids 0.3 et 0.7 pour ces deux critères respectivement. Dans le deuxième, on considère le cas inverse (c'est à dire, 0.7 pour le poids du coût de reconfiguration et 0.3 pour celui de trafic). Les trois applications ont été codées en utilisant un compilateur C++ sur la même machine qui a été utilisée dans le précédent tests (processeur Cyrix MII 300 à 233 MHz et 32 Mo de RAM). Dans les paragraphes suivants, les trois méthodes sont désignées par **PCC** pour la méthode du plus court chemin, **AG** pour l'AG binaire sans raisonnement flou et **AGF** pour l'AG flou.

10.4.1 Les données des exemples

Trois exemples sont présentés dans ce qui suit. Nous détaillons l'exposé du premier exemple afin d'illustrer l'incorporation du raisonnement flou à l'AG.

La taille du premier exemple est $16 \times 6 \times 4$. La taille est supposée égale au produit $p \times m \times H$ (nombre de produits \times nombre de machines \times nombre de périodes).

La table 10.1 montre la matrice d'incidence machine-produit et les ratios de production associés à chacune des quatre périodes. Ces deux types d'information définissent les quatre scénarios de l'horizon de planification. Le nombre maximum de machines dans chaque cellule est égal à 3 pour toutes les périodes.

Produits	Machines						Ratios			
	M1	M2	M3	M4	M5	M6	r₁	r₂	r₃	r₄
P ₁	1	2			4	3	1	0	0	0
P ₂	3	1	2	4			3	0	0	0
P ₃				2	1,3		5	0	0	0
P ₄	1	2,4	3				2	0	0	0
P ₅	1		3	2,4			0	2	2	0
P ₆	1	3,6	5	2,4			0	3	0	0
P ₇	1	2,4	5	3,6			0	1	0	0
P ₈	1	3		2,4			0	3	0	0
P ₉	1	4		3	2		0	2	0	0
P ₁₀	1		3	4	2,5		0	0	1	0
P ₁₁	4	3	1,6	5,7	2		0	0	1	0
P ₁₂			4	1,3	2		0	0	2	0
P ₁₃		2			1	3	0	0	0	1
P ₁₄	3		2		1		0	0	0	3
P ₁₅		2		3		1	0	0	0	2
P ₁₆	1	2		4	3,5	6	0	0	0	1
Durées des périodes							2	4	2	4

En outre, les machines M₁ et M₃ doivent être placées dans des cellules différentes.

Pour le deuxième exemple, la taille s'élève à $20 \times 15 \times 5$. Les scénarios des périodes peuvent être déduits de la table 10.2.

Table 10.2 Les scénarios de l'horizon de planification de l'exemple 2

Produits	Machines															Ratios				
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	r ₁	r ₂	r ₃	r ₄	r ₅
P ₁	.	.	6	5	4	2	3,7	.	1	1	0	0	0	0
P ₂	6	1	.	3	4	7	.	.	2,5	8	.	3	0	0	0	0
P ₃	.	1	2	3	1	0	3	0	0
P ₄	1	2,4	.	3	7	.	5	6	.	.	1	0	0	0	0
P ₅	2	.	1	3	0	3	0	0	0	0
P ₆	.	.	.	2	.	6	3	5	7	.	.	8	.	1,4	0	2	0	3	0	0
P ₇	.	.	.	1	.	3	2	0	2	0	0	0	0
P ₈	.	1	2	.	.	.	3	.	0	2	0	0	0	0
P ₉	.	2	1	.	.	.	3	.	0	0	3	0	0	0
P ₁₀	7	.	8	1,3	.	2	5	.	4	.	.	6	.	.	0	3	1	0	0	0
P ₁₁	3	.	.	5	.	.	.	2	4,6	.	1	.	.	.	0	0	3	0	2	0
P ₁₂	2	6	3,5	.	4	1	2	0	1	0	0	0
P ₁₃	3	.	8	.	1,5	2	4	.	7	.	.	.	6	3	0	0	1	0	0	0
P ₁₄	.	2	1	.	3	0	0	0	2	0	0
P ₁₅	.	.	.	1	4	.	3,6	2,5	.	0	0	0	1	0	0	0
P ₁₆	.	2	3	4	.	.	1	0	0	0	2	0	0
P ₁₇	1,4	.	.	2	.	3	0	0	1	0	3	0
P ₁₈	.	1	.	.	6	3	4	.	2	.	7	.	.	5	0	0	0	0	1	0
P ₁₉	.	4	6	.	1,3	5	.	2	.	0	3	0	0	1	0
P ₂₀	3	.	2	4	1	.	.	.	0	0	0	0	3	0
Durées des périodes															1	2	4	3	3	

Le nombre minimum de machines par cellule est porté à 5 et une contrainte de non cohabitation est considérée entre les machines M₈ et M₁₁.

Le dernier exemple, qui est le plus large, est de taille 24×20×6. Les informations des six scénarios sont illustrées dans la table 10.3.

Table 10.3 Les scénarios de l'horizon de planification de l'exemple 3

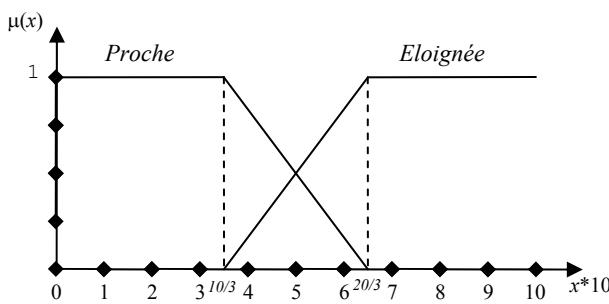
Produits	Machines																			Ratios								
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	r ₁	r ₂	r ₃	r ₄	r ₅	r ₆		
P1	2	.	.	3	4	.	1	.	.	7	.	.	6	4	.	3,7	.	1,5	.	3	0	0	0	0	0	0		
P2	6	3	.	.	5	.	.	.	2	.	8	2	0	0	0	0	0	0		
P3	.	2	1	3	4	3	0	0	0	0	0	0		
P4	1	5	2,4	.	3	7	.	6	.	1	0	0	0	0	0	0		
P5	.	.	.	3	2	.	1	.	.	0	1	0	0	0	0	0		
P6	8	.	.	4	6	.	5	2,7	1,3	0	1	0	0	0	0	0		
P7	.	3	4	2	.	.	.	1	0	1	0	0	0	0	0	0		
P8	.	6	.	2,7	.	.	.	7	3	.	.	5	.	5	3	.	4	.	1	0	1	0	0	0	0	0		
P9	.	.	6,8	.	.	.	7	3	.	.	5	.	2	.	4	.	4	1	.	0	0	2	0	0	0	0		
P10	1	4	2,5	.	6	3	0	0	1	0	0	0	0	0		
P11	.	.	2	1,3	5	4	0	0	2	0	0	0	0	0		
P12	1	.	8	.	7	5	.	.	.	2	4	.	3	6	0	0	3	0	0	0	0	0		
P13	2	1	.	3	.	0	0	0	3	0	0	0	0		
P14	4	.	.	2,5	.	.	6	3	1	0	0	0	3	0	0	0	0		
P15	4	.	2	5	7	6	1	.	3	0	0	0	1	0	0	0	0		
P16	.	4	.	.	7	.	.	.	1,3	.	.	2,6	.	5	.	8	.	3	0	0	0	2	0	0	0	0		
P17	7	1	4	2,6	.	.	.	5	.	8	.	3	.	0	0	0	0	2	0	0	0	0		
P18	3	.	8	.	7	.	5	1,4	.	.	2	6	0	0	0	0	3	0	0	0	0		
P19	6	.	3	.	.	1	5	.	.	.	2	4	.	3	0	0	0	0	2	0	0	0	0	
P20	6	1	.	4	2	5	2	4	.	3	0	0	0	0	2	0	0	0	0	
P21	1	3	.	5	.	.	1	.	.	.	2	4	.	3	0	0	0	0	0	0	0	0	0	
P22	.	.	.	3	.	4	1	3	.	2	.	2	.	0	0	0	0	0	0	0	0	0	0	
P23	1	.	5	2	.	1	7	.	8	.	3	.	0	0	0	0	0	0	0	0	0	0		
P24	.	.	4	.	6	.	5	2	.	1	7	.	8	.	3	.	0	0	0	0	0	0	2	0	0	0		
Durées des périodes																			2	4	3	2	3	2				

En ce qui concerne les contraintes, nous considérons qu'on ne peut avoir plus de 5 machines par cellule et que les machines M₄ et M₁₆ ne peuvent cohabiter.

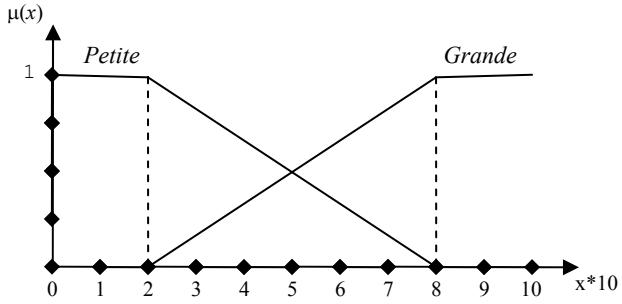
10.4.2 Embarquement de connaissance dans l'AG

Comme décrit dans le [paragraphe 10.3.2](#), deux inputs influent sur la l'identité de la meilleure solution

a) Distance



b) Reconfiguration



c) Similarité

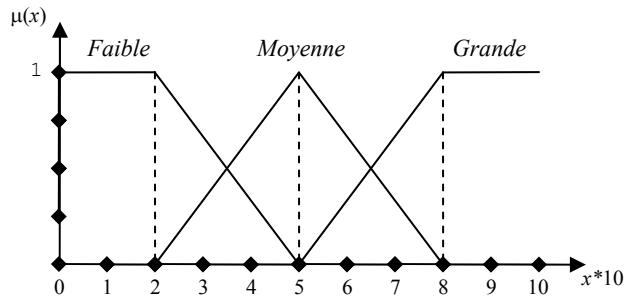


Figure 10.6 Fonctions d'appartenance des variables linguistiques.

recherchée : les matrices de flux et l'importance du coût de reconfiguration. L'information du premier paramètre est extraite par la variable linguistique *Distance* dont les fonctions d'appartenance sont représentées dans la figure 6.a.

La table 10.4 montre les facteurs d'appartenance aux valeurs de la variable *Distance* correspondant à l'exemple illustratif.

Table 10.4 Facteurs d'appartenance associés à la variable *Distance* de l'exemple 1

Transition	Distance de Hamming entre images de flux	Proche	Eloignée
(1,2)	0.323958	1	0
(2,3)	0.228588	1	0
(3,4)	0.509259	0.456790	0.543210

Pour le deuxième paramètre influant, nous utilisons la variable linguistique *Reconfiguration* dont les fonctions d'appartenance sont représentées dans la figure 6.b. La table 10.5 en montre les facteurs d'appartenance.

Table 10.5 Facteurs d'appartenance liés à la variable *Reconfiguration*

Poids du critère : coût de Reconfiguration	petite	grande
0.3	0.833333	0.166667
0.7	0.166667	0.833333

Ces deux variables linguistiques sont liées par les règles d'inférences floues qui donnent, en appliquant la méthode max-min de Mamdani, puis la méthode de la moyenne des maximums lors de la défuzzification [Ying2000], les valeurs r_1, r_2, r_3 représentées dans la table 10.6. Le choix de la méthode de la moyenne des maximums est justifié par son efficacité face à la technique du centre de gravité. Il nous a paru, à priori, que cette dernière pourrait donner des résultats meilleurs. Cependant, après l'avoir implémenté, nous n'avons pas décelé une amélioration significative qui justifie le surcoût de calcul en comparaison avec la technique de la moyenne des maximums. Par conséquent, nous avons opté pour cette dernière.

Table 10.6 Résultats de la défuzzification de la variable floue *Similarité*

Poids du critère de reconfiguration	Transition	Similarité
0.3	(1,2)	0.50
	(2,3)	0.50
	(3,4)	0.33
0.7	(1,2)	0.74
	(2,3)	0.74
	(3,4)	0.50

Les valeurs de la défuzzification de la variable *Similarité* sont utilisées par AGF pour prévoir la forme de la solution recherchée, en considérant que chaque r_i , $i \in \{1,2,3\}$, représente le degré de similarité devant lier les compositions consécutives, associées aux périodes i et $i+1$, dans toute bonne solution.

10.4.3 Analyse des résultats

Comme pour les précédents tests, les trois méthodes ont été lancées plusieurs fois dans le but de déterminer les valeurs des paramètres qui rendent les résultats stables. Nous rappelons que la stabilité est supposée atteinte si la meilleure solution est toujours obtenue au moins deux fois parmi cinq exécutions. La meilleure solution pour chaque méthode a été ensuite portée avec son propre temps d'exécution.

Pour le premier exemple, les résultats obtenus concernant la première situation ($w_1=0.7$ et $w_2=0.3$) sont portés dans la table 10.7.

Table 10.7 Performances pour une reconfiguration de faible importance liées à l'exemple 1

	Méthodes		
	PCC	AG	AGF
Trafic	0.3072	0.3072	0.3072
Reconfiguration	0.3611	0.3056	0.3056
Agrégation	0.3234	0.3067	0.3067
CPU (sec.)	11.53	49.30	30.02

Les deux méthodes génétiques, AG et AGF, ont atteint la même solution qui est meilleure que celle trouvée par la méthode PCC (voir la Table 10.8 pour les détails des solutions trouvées par les trois méthodes). Cependant, AGF a pu atteindre sa stabilité avec un temps de calcul moindre.

Table 10.8 Les compositions pour une reconfiguration de faible importance de l'exemple 1

Méthodes	Compositions de cellules			
	Période 1	Période 2	Période 3	Période 4
PCC	{1,4,5} {2,3} {6}	{1,5,6} {2,3,4}	{1,2} {3,4,5} {6}	{1,2,6} {3,4,5}
AG	{1,4,5} {2,3} {6}	{1,5} {2,3,4} {6}	{1,2,6} {3,4,5}	{1,2,6} {3,4,5}
AGF	{1,4,5} {2,3} {6}	{1,5} {2,3,4} {6}	{1,2,6} {3,4,5}	{1,2,6} {3,4,5}

En effet, la table 10.9 montre que AGF requiert une population de taille plus petite et un nombre de générations moins important pour atteindre la bonne solution. Un tel gain montre l'apport du raisonnement flou qui permet de diriger la recherche, par anticipation, vers les zones prometteuses.

Table 10.9 Valeurs des paramètres pour une reconfiguration de faible importance de l'exemple 1

Paramètres	Méthodes		
	PCC	AG	AGF
Taille de population (s)	-	200	150
Nombre de générations (i_{max})	-	2000	1000
Probabilité pour un allèle non nul (P_1)	-	0.80	0.80
Taux de sélection (P_2)	-	0.70	0.70
Taux de croisement (P_3)	-	0.90	0.90
Taux de mutation (P_4)	-	0.01	0.01
Nombre de solutions par période	40	-	-

Dans le deuxième exemple, les résultats vont aussi en faveur de AGF (voir table 10.10) qui a pu trouver une solution statique meilleure que la solution dynamique déterminée par AG. La solution trouvée par PCC est aussi statique mais elle est moins bonne comparée à celle des deux méthodes génétiques.

Table 10.10 Performance pour une reconfiguration de faible importance de l'exemple 2

	Méthodes		
	PCC	AG	AGF
Trafic	0.6563	0.5641	0.4786
Reconfiguration	0	0.0545	0
Agrégation	0.4594	0.4113	0.3350
CPU (sec.)	28.47	152.82	53.17

Les détails des solutions trouvées sont présentés dans la table 10.11.

Table 10.11 Compositions de cellules pour une reconfiguration de faible importance de l'exemple 2

Meth.	Les compositions				
	Période 1	Période 2	Période 3	Période 4	Période 5
PCC	{1,3,4,7,13} {2,5,9,11,14} } {6,8,10,12,15}	{1,3,4,7,13} {2,5,9,11,14} } {6,8,10,12,15}	{1,3,4,7,13} {2,5,9,11,14} } {6,8,10,12,15}	{1,3,4,7,13} {2,5,9,11,14} } {6,8,10,12,15}	{1,3,4,7,13} {2,5,9,11,14} } {6,8,10,12,15}
AG	{1,5,7,14,15} {2,3,8,10,13} } {4,6,9,12} {11}	{1,5,7,14,15} {2,3,8,10,13} } {4,6,9,12} {11}	{1,4,5,7,11} {2,3,8,10,13} } {6,9,12,14,15}	{1,4,5,7,11} {2,3,8,10,13} } {6,9,12,14,15}	{1,4,5,7,11} {2,3,8,10,13} } {6,9,12,14,15}
AGF	{1,7,9,10,13} {2,3,8,12,15} } {4,5,6,11,14}	{1,7,9,10,13} {2,3,8,12,15} } {4,5,6,11,14}	{1,7,9,10,13} {2,3,8,12,15} } {4,5,6,11,14}	{1,7,9,10,13} {2,3,8,12,15} } {4,5,6,11,14}	{1,7,9,10,13} {2,3,8,12,15} } {4,5,6,11,14}

Dans le troisième exemple, les trois méthodes ont toutes trouvées des solutions à composition active. Les solutions de PCC et AG réalisent une meilleure performance en ce qui concerne le coût de reconfiguration (voir table 10.12). Cependant, la performance de AGF pour le critère le plus important qu'est le trafic intercellulaire a mis la balance en faveur de cette dernière dans fonction d'agrégation.

Table 10.12 Performance pour une reconfiguration de faible importance de l'exemple 3

	Méthodes		
	PCC	AG	AGF
Trafic	0.6878	0.5854	0.5187
Reconfiguration	0.0719	0.0854	0.1236
Agrégation	0.5030	0.4354	0.4002
CPU (sec.)	116.47	1123.76	817.64

Les détails des résultats de chaque méthode sont présentés dans la table 10.13.

Table 10.13 Compositions de cellules pour une reconfiguration de faible importance de l'exemple 3

Période	Les compositions des méthodes		
	SP	GA	FGA
1	{1,5,12,14,16} {2,6,8,13,18} {3,4,11,20} {7,9,10,15,19} {17}	{1,10,15,17,19} {2,4,8,20} {3} {5,11,13,16,18} {6,14} {7} {9} {12}	{1,10,15,17,19} {2,4,5,7,9} {3} {6,8,13,20} {11,12,14,16,18}
2	{1,5,12,14,16} {2,6,8,13,18} {3,4,11,20} {7,9,10,15,19} {17}	{1,10,15,17,19} {2,4,8,20} {3} {5,11,13,16,18} {6,14} {7} {9} {12}	{1,10,15,17,19} {2,4,5,7,9} {3} {6,8,13,20} {11,12,14,16,18}
3	{1,5,12,14,16} {2,6,8,13,18} {3,4,11,20} {7,9,10,15,19} {17}	{1,9,12,13,16} {2,4,8} {3,5,6,19} {7,1,0,15} {11,18,20} {14} {17}	{1,10,15,17,19} {2,5,7,9} {3,11,14,18} {4} {6,8,13,20} {12,16}
4	{1,5,12,14,16} {2,6,8,13,18} {3,4,11,20} {7,9,10,15,19} {17}	{1,9,12,13} {2,4,8} {3,5,6,19} {7,10,1,5} {11,18,20} {14} {16} {17}	{1,9,13,18} {2} {3,11,12,15,16} {4,6,8,14,20} {5,7,10,19} {17}
5	{1,5,12,14,16} {2,6,8,13,18} {3,4,11,20} {7,9,10,15,19} {17}	{1,2,4,12,13} {3,5,6,14,19} {7,8,10,1,5,17} {9} {11,16,18,20}	{1,8,12,13,18} {2,4,10,20} {3,5,6,14,19} {7} {9,11,15,17} {16}
6	{1,2,7,9,10} {3,5,6,12,14} {4,11,19} {8,13,15,18,20} {16} {17}	{1,2,4,12,13} {3,5,6,14,19} {7,8,10,1,5,17} {9} {11,16,18,20}	{1,8,12,13,18} {2,4,10,20} {3,5,6,14,19} {7} {9,11,15,17} {16}

Pour la deuxième situation ($w_1=0.3$ et $w_2=0.7$), les résultats obtenus sont portés dans les tables 10.14, 10.17 et 10.19 pour les trois exemples respectivement. Les détails des solutions associées sont portés dans les tables 10.15, 10.18 et 10.20.

Table 10.14 Performances pour une reconfiguration de grande importance liées à l'exemple 1

	Méthodes		
	PCC	AG	AGF
Trafic	0.6627	0.5361	0.4940
Reconfiguration	0.0000	0.0000	0.0000
Agrégation	0.1988	0.1608	0.1482
CPU (sec.)	11.47	81.27	2.01

Table 10.15 Les compositions pour une reconfiguration de grande importance de l'exemple 1

Méthodes	Les compositions			
	Période 1	Période 2	Période 3	Période 4
PCC	{1}{2}{3,4,5}{6}	{1}{2}{3,4,5}{6}	{1}{2}{3,4,5}{6}	{1}{2}{3,4,5}{6}
AG	{1,2,4}{3,5,6}	{1,2,4}{3,5,6}	{1,2,4}{3,5,6}	{1,2,4}{3,5,6}
AGF	{1,5,6}{2,3,4}	{1,5,6}{2,3,4}	{1,5,6}{2,3,4}	{1,5,6}{2,3,4}

Table 10.16 Valeurs des paramètres pour une reconfiguration de grande importance de l'exemple 1

Paramètres	Méthodes		
	PCC	AG	AGF
Taille de population (s)	-	200	50
Nombre de générations (i_{max})	-	2500	300
Probabilité pour un allèle non nul (P_1)	-	0.80	0.80
Taux de sélection (P_2)	-	0.70	0.70
Taux de croisement (P_3)	-	0.90	0.90
Taux de mutation (P_4)	-	0.02	0.02
Nombre de solutions par période (T_h)	40	-	-

Table 10.17 Performance pour une reconfiguration de grande importance de l'exemple 2

	Méthodes		
	PCC	AG	AGF
Trafic	0.7549	0.5641	0.5066
Reconfiguration	0	0	0
Agrégation	0.2265	0.1692	0.1520
CPU (sec.)	28.82	170.46	78.06

Table 10.18 Compositions de cellules pour une reconfiguration de faible importance de l'exemple 2

Meth.	Les compositions				
	Période 1	Période 2	Période 3	Période 4	Période 5
PCC	{1,4,8,10,14}{2,5,11}{3,7,12,13,15}{6,9}	{1,4,8,10,14}{2,5,11}{3,7,12,13,15}{6,9}	{1,4,8,10,14}{2,5,11}{3,7,12,13,15}{6,9}	{1,4,8,10,14}{2,5,11}{3,7,12,13,15}{6,9}	{1,4,8,10,14}{2,5,11}{3,7,12,13,15}{6,9}
AG	{1,5,11,13,14}{2,7,8,10,12}{3,4,6,9,15}	{1,5,11,13,14}{2,7,8,10,12}{3,4,6,9,15}	{1,5,11,13,14}{2,7,8,10,12}{3,4,6,9,15}	{1,5,11,13,14}{2,7,8,10,12}{3,4,6,9,15}	{1,5,11,13,14}{2,7,8,10,12}{3,4,6,9,15}
AGF	{1,7,9,10,13}{2,3,5,11,14}{4,6,8,12,15}	{1,7,9,10,13}{2,3,5,11,14}{4,6,8,12,15}	{1,7,9,10,13}{2,3,5,11,14}{4,6,8,12,15}	{1,7,9,10,13}{2,3,5,11,14}{4,6,8,12,15}	{1,7,9,10,13}{2,3,5,11,14}{4,6,8,12,15}

Table 10.19 Performance pour une reconfiguration de grande importance de l'exemple 3

	Méthodes		
	PCC	AG	AGF
Trafic	0.7154	0.7463	0.6325
Reconfiguration	0.0472	0.0067	0
Agrégation	0.2477	0.2286	0.1898
CPU (sec.)	60.76	595.27	219.38

Table 10.20 Compositions de cellules pour une reconfiguration de grande importance de l'exemple 3

Période	Les compositions des méthodes		
	SP	GA	FGA
1	{1,3,5,9,12}{2,6,13,14,16}{4,7,15,17,19}{8,10,11,18,20}	{1,12,13,14,19}{2,4,7,8,18}{3,6,11,15}{9,16}{17}	{1,5,12,13,16}{2,6,7,14,18}{3,4,9,11,20}{8,10,15,17,19}
2	{1,3,5,9,12}{2,6,13,14,16}{4,7,15,17,19}{8,10,11,18,20}	{1,12,13,14,19}{2,4,7,8,18}{3,6,11,15}{9,16}{17}	{1,5,12,13,16}{2,6,7,14,18}{3,4,9,11,20}{8,10,15,17,19}
3	{1,3,5,9,12}{2,6,13,14,16}{4,7,15,17,19}{8,10,11,18,20}	{1,12,13,14,19}{2,4,7,8,18}{3,6,11,15}{9,16}{17}	{1,5,12,13,16}{2,6,7,14,18}{3,4,9,11,20}{8,10,15,17,19}
4	{1,3,5,9,12}{2,6,13,14,16}{4,7,15,17,19}{8,10,11,18,20}	{1}{2,4,7,8,18}{3,6,11,15}{5,10,20}{9}{12,13,14,19}{16}{17}	{1,5,12,13,16}{2,6,7,14,18}{3,4,9,11,20}{8,10,15,17,19}
5	{1,3,5,8,12}{2,10,14,17}{4,6,7,15,19}{9}{11,13,18,20}{16}	{1}{2,4,7,8,18}{3,6,11,15}{5,10,20}{9}{12,13,14,19}{16}{17}	{1,5,12,13,16}{2,6,7,14,18}{3,4,9,11,20}{8,10,15,17,19}
6	{1,3,5,8,12}{2,10,14,17}{4,6,7,15,19}{9}{11,13,18,20}{16}	{1}{2,4,7,8,18}{3,6,11,15}{5,10,20}{9}{12,13,14,19}{16}{17}	{1,5,12,13,16}{2,6,7,14,18}{3,4,9,11,20}{8,10,15,17,19}

En adoptant une vue globale, on peut constater que AG trouve des solutions meilleures, en terme de qualité, en comparaison avec PCC. Cependant, c'est la deuxième qui donne le temps d'exécution le plus petit. D'autre part, AGF permet, en général, d'améliorer les performances de AG. Lorsqu'on attribue au coût de reconfiguration une faible importance, les solutions trouvées par les méthodes sont, en général, dynamiques. Cette constatation est déduite de par la valeur non nulle du coût de reconfiguration de la plupart des méthodes, dans les tables 10.7, 10.10 et 10.12. Cependant, dans le deuxième exemple, AGF a pu déterminer une meilleure solution statique en exploitant les similitudes entre les images de flux de cet exemple (voir figure 10.7).

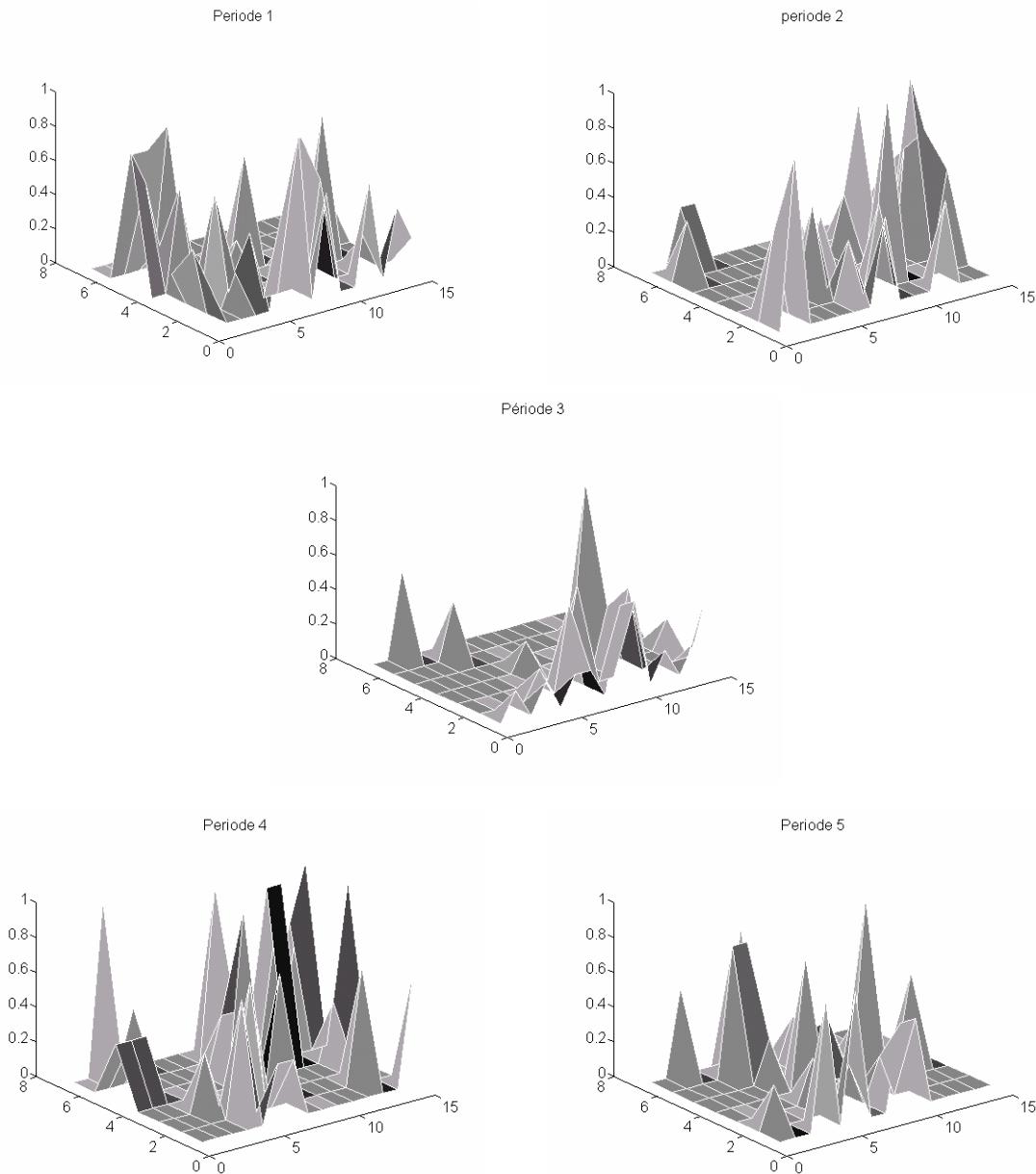


Figure 10.7 Les images de flux de l'exemple 2

Par contre, lorsque le coût de reconfiguration est relativement important et au fur et à mesure que la taille des problèmes augmente, AG est un petit peu perdu dans l'espace de recherche inondé par les solutions à composition active. En effet, les résultats de la table 10.19 montre que AG a du mal à déceler une bonne solution statique pour l'instance la plus large que représente l'exemple 3.

En résumé, nous pouvons affirmer que, AGF fait preuve d'une bonne performance en terme de qualité et d'une performance acceptable en terme d'efficacité, que le système donne plus ou moins d'importance au coût de reconfiguration.

Les exemples utilisés dans cette comparaison sont relativement de taille moyenne. Dans ceux-ci, le raisonnement flou a pu améliorer les performances de l'algorithme génétique binaire. En se basant sur ces performances, il est plausible de prévoir qu'il maintiendra ses bonnes performances pour des instances de grande taille.

CONCLUSION GENERALE

1. EN RESUME

Dans cette thèse, nous avons proposé une approche multicritère au problème de composition de cellules flexibles dans les systèmes cellulaires de production.

Après avoir décrit le domaine d'étude, dans une première partie, nous avons réalisé, dans une deuxième, un bref survol sur les principales méthodes de modélisation et de résolution multicritères.

La troisième partie inaugure notre approche au problème de composition. Pour ce faire, nous abordons le problème sous son aspect statique. Après avoir positionné le problème et présenté son état de l'art, nous proposons une modélisation basée sur la théorie des graphes tenant compte de deux parmi les critères les plus utilisés dans la littérature. Une version monocritère du problème nous permet de situer notre problème dans la classe des problèmes *NP-Complets*. Nous nous sommes orientés, par conséquent, vers les méthodes évolutionnaires qui ont montré leur efficacité dans la résolution des problèmes difficiles d'optimisation multicritères.

Nous mettons le ton sur le codage utilisé par la plupart des approches évolutionnaires déjà publiés. Après avoir mis en relief les insuffisances associées à cette codification, nous proposons une nouvelle méthode de codage qui permet une meilleure représentation des caractéristiques des individus en étalant les informations qu'ils contiennent sur une chaîne, certes, plus longue mais avec un alphabet moindre. Une méthode astucieuse pour le calcul de l'adaptation par transformation des fonctions objectif est proposée, permettant de ne pas pénaliser sévèrement les mauvais individus de la population (solutions non réalisables) mais de les garder afin de tirer profits des bonnes caractéristiques qu'ils peuvent éventuellement cacher. Afin de renforcer les performances de l'AG binaire, nous avons proposé une nouvelle méthode par séparation et évaluation qui a permis de donner à l'AG binaire un apport considérable consistant en un préraffinage de l'espace des solutions. Ensuite, une extension du modèle pour la prise en compte des routages multiples est abordée.

Dans une quatrième partie, nous traitons l'aspect dynamique du problème. Contrairement à l'aspect statique, cet aspect est peu traité dans la littérature. Après avoir décrit la nécessité de la prise en compte de l'aspect dynamique, nous préconisons que la résolution de celui-ci peut se faire selon deux stratégies différentes, à savoir, la stratégie passive et celle active. Une attention particulière est prêtée à un nouveau critère associé à l'aspect dynamique. Il s'agit du coût de reconfiguration qui est la clef de la décision permettant de choisir entre ces deux stratégies. Nous signalons la maladresse de la technique utilisée dans la littérature pour sa définition et proposons une nouvelle définition plus élégante.

Nous démontrons, pour la stratégie passive, que la résolution revient à résoudre un problème sous l'aspect statique. Par conséquent, toutes les méthodes proposées dans la troisième partie peuvent être appliquées. Pour la deuxième stratégie, nous proposons une première méthode basée sur un algorithme de recherche d'un plus court chemin sur un graphe que nous définissons de façon originale. Aussi, une méthode basée sur l'algorithme génétique binaire a été suggérée.

Lorsque le décideur est amené à choisir entre l'une ou l'autre stratégie, nous proposons de procurer à l'algorithme génétique une aide basée sur la théorie des sous-ensembles flous. Cette dernière offre à la communauté des chercheurs une application d'un procédé de résolution pouvant être copié pour aborder de façon similaire les problèmes dynamiques multipériodiques.

2. CONTRIBUTIONS

Les contributions proposées dans cette thèse peuvent être classifiées en deux niveaux : modélisation et résolution. Dans chaque niveau, ces contributions peuvent être énumérées comme suit :

2.1 Modélisation

- Codage binaire : Parmi les apports que nous avons apporté à la communauté des chercheurs dans le domaine de la composition de cellules l'utilisation du codage à base d'arêtes du graphe de flux.
- Approche à la composition par les co-cycles : Dans la littérature, les approches au problème de composition basées sur la théorie des graphes utilisent deux modèles : le modèle de partitionnement ou celui de l'arbre recouvrant. Le modèle utilisant les co-cycles pour définir une composition est original.
- Modélisation par la théorie des graphes avec routages alternatifs : La prise en compte de l'aspect multiroutage est réalisé dans la littérature en utilisant des modèles de programmation mathématique. L'extension du modèle graphique à ce cas par la considération d'un graphe paramétré est un travail nouveau.
- Modélisation du coût de reconfiguration : La transformation de la détermination du coût de reconfiguration en un problème de couplage dans un graphe biparti est un apport original.
- Considération des stratégies passive et active pour le problème dynamique : Dans la littérature, aucun travail ne fait allusion aux stratégies statique et dynamique dans le problème de composition. Nous nous sommes inspirés des travaux réalisés dans le module d'agencement pour proposer l'approche présentée dans cette thèse. Cependant, la façon avec laquelle nous abordons le problème de décision entre ces deux stratégies est un effort original.
- Le graphe H-partite de la méthode du plus court chemin : La modélisation du problème dynamique en utilisant un graphe H-partite est inspirée des travaux réalisés dans le domaine d'agencement. Cependant, la façon par laquelle on définit les poids du graphe est originale.

2.2 Résolution

- Méthode par séparation et évaluation co-cyclique : La méthode de résolution par séparation et évaluation progressive est originale.
- Méthode epsilon-contrainte : L'amélioration de l'évaluation par la méthode epsilon contrainte est un apport original.
- Pilotage flou off-line de l'AG bi-critère : Le pilotage de l'AG par un raisonnement flou en off-line utilisant les informations emmagasinées dans les inputs pour guider la recherche est un effort qui nous est propre.

Cependant, comme tout travail humain, on estime que la synthèse présentée est loin d'atteindre la perfection car la recherche est une discipline qui n'a pas de fin. En outre, malgré les efforts déployés, on aura toujours un

sentiment de travail inachevé dans l'esprit. A cet effet, afin de combler les limites du travail présenté, le paragraphe suivant présente plusieurs axes de recherche pour son amélioration.

3. AXES POUR DES RECHERCHES FUTURES

Comme on peut le constater au fil des chapitres de ce manuscrit, nous posons dans certains problèmes des questions à développer. Celles-ci peuvent constituer des voies d'amélioration intéressantes. Mais pour parler des grands axes, nous estimons que les contributions présentées peuvent être étendue en œuvrant dans les voies suivantes :

3.1 L'aspect modélisation

Le travail que nous avons présenté traite le problème de composition des cellules dans les systèmes cellulaires de production sous l'aspect déterministe. Ce qui suppose que les scénarios de production sont parfaitement connus. Ceci oblige le concepteur à prévoir l'évolution du système dans le temps, en tenant compte de l'historique du comportement du système et des prévisions faites en fonction des tendances du marché.

L'une des extensions possibles à notre travail est d'essayer d'aborder le problème en considérant des données aléatoires (probabilistes ou non probabilistes). Des travaux ont été menés dans ce sens pour les problèmes d'agencement intra et intercellulaires. Dans le domaine de la composition statique, les apports de la littérature restent très pauvres. Dans le domaine de la composition dynamique, aucun travail n'a été réalisé dans ce sens en notre connaissance.

3.2 L'aspect résolution

Les méthodes de résolution proposées dans ce manuscrit peuvent être améliorées suivant plusieurs axes. Par exemple, pour la méthode SEPAG, on peut étudier l'impact de l'insertion du critère d'équilibrage de charge intercellulaire de façon directe de sorte à avoir un procédé d'évaluation bi-critère. Pour ce faire, il faut trouver une méthode qui permet de borner la fonction objectif associée au critère de charge chaque fois qu'il y a une scission par un co-cycle. Pour la méthode AG-Tabou, notre investigation a concerné un procédé d'hybridation séquentiel. Il serait intéressant, de mener une étude comparative avec un procédé de coopération parallèle. Enfin, pour la méthode floue, il serait intéressant de mener une étude sur l'impact de l'utilisation d'autres définitions des opérateurs flous tels que la disjonction (ou), la conjonction (et) et l'implication sur les performances de la méthode.

3.3 L'aspect implémentation

Beaucoup reste à faire du côté de l'implémentation. A titre d'exemple, la parallélisation des méthodes proposées est une voie très importante vu la réduction du temps d'exécution qu'elle peut procurer. Celle-ci devient plus qu'une nécessité lors de l'application des méthodes pour des instances réelles du problème.

Le fait que les algorithmes génétiques sont des méthodes « hautement » parallélisables est un paramètre encourageant pour une telle démarche. En effet, l'exploration par nuage de points que ces méthodes réalisent autorise le fait que chaque individu de la population puisse être évalué indépendamment de ses frères. De même, chaque opération de croisement et de remplacement après sélection.

BIBLIOGRAPHIE

• Ouvrages et articles de recherche

- [Adil1996] **ADIL GK.**, RAJAMANI D. and STRONG D., Cell formation considering alternate routeings. *International Journal of Production Research*, 34 (5): 1361-80. 1996.
- [Aguirre2003] **AGUIRRE A. H.**, S. B. RIONDA, C. A. C. COELLO, G. L. LIZARRAGA, E. M. MONTES, Handling constraints using multiobjective optimization concepts, International journal for numerical methods in engineering, John Wiley & Sons, Ltd, 2003.
- [Aitzai1998] **AITZAI A.**, Agencement statique et dynamique des systèmes flexibles de production. Thèse de Magister, USTHB. 1998.
- [Akturk1996] **AKTURK MS.** and BALKOSE, H. O., Part-machine grouping using a multiobjective cluster analysis. *International Journal of Production Research*, 34, 2299-315. 1996.
- [Al-Sultan1997] **AL-SULTAN KS**, FEDJKI CA. A genetic algorithm for the part family formation problem. *Production Planning & Control*. 8(8):788-96. 1997.
- [Albadawi2005] **ALBADAWI Z.**, BASHIR H. A., CHEN M., A mathematical approach for the formation of manufacturing cells, *Computers & Industrial Engineering* 48:3–21. 2005.
- [Anderberg1973] **ANDERBERG, MR.**, 1973. Cluster analysis for applications (New York: Academic Press).
- [Anderson1999] **ANDERSON J.**, A survey of multiobjective optimization in engineering design, Technical report: LiTH-IKP-R-1097. Linkoping University, Sweden.
- [Askin1990] **ASKIN RG.** and CHIU K.S., A graph partitioning procedure for machine assignment and cell formation in GT, *IJPR* 28 (2): 1555-72. 1990.
- [Askin1991] **ASKIN RG.**, CRESSWELL S.H., GOLDBERG J.B. and VAKHARIA A.J., A Hamiltonian Path Approach to Reordering the Part-Machine Matrix for Cellular Manufacturing, *International Journal of Production Research*, 29, 1081-1100. 1991.
- [Atmani1995] **ATMANI A.**, LASHKARI R.S. and CARON R.J., A mathematical programming approach to joint manufacturing cell formation and operation allocation in cellular manufacturing. *Int. J. Prod. Res.* 33: 1-15 1995.
- [Barichard2003] **BARICHARD V.**, Approches hybrides pour les problèmes multiobjectifs. Thèse de Doctorat. Université d'Angers. 2003.
- [Batta1987] **BATTA R.**, Comments on "The Dynamics of Plant Layout", *Management Science*. 33(8):1065. 1987.
- [Benders2000] Benders, J., R. Badham. A history of cell-based manufacturing in Work Teams: Past, Present and Future. M. Beyerlein (ed.), Kluwer Academic Publishers, Boston, Massachusetts, pp.45–57. 2000. in [Johnson2004].
- [Bently1997] **BENTLY P.**, The revolution of evolution for real-world applications. 1997. From Internet.
- [Béranger1987] **BERANGER P.**, Les nouvelles règles de production vers l'excellence industrielle. Dunod entreprise, 1987.
- [Berro2001] **BERRO A.**, Optimisation multiobjectif et stratégies d'évolution en environnement dynamique, Thèse de Doctorat en Informatique, Université des sciences sociales Toulouse I, France.
- [Boctor1991] **BOCTOR FF.**, A Linear Formation of the Machine-Part Cell Formation Problem. *International Journal of Production Research*, 29(2):343-56, 1991.
- [Boctor1996] **BOCTOR FF.**, The minimum cost, machine-part cell formation problem. *Ibid.*, 34: 1045-63. 1996. In [Mansouri2000].
- [Boulif2000] **BOULIF M.**, Composition de cellules dans les systèmes cellulaires de production statiques et dynamique. Thèse de Magister, USTHB. 2000.
- [Boulif2006a] **BOULIF M.** and ATIF K., A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. *Computers & Operations Research* 33:2219-2245. 2006.
- [Boulif2006b] **BOULIF M.** and BENAISSE N., On the definition of the reconfiguration cost for the multi-periodic manufacturing cell formation problem. In preparation. Fundamental and applied computer science laboratory. Boumerdes University, Algeria. 2006.
- [Boulif2006c] **BOULIF M.** and ATIF K., An exact multiobjective epsilon constraint approach for the manufacturing cell formation problem. International conference on service systems and service management, (ICSSSM'06) IEEE. University of Troyes, France. 2006.
- [Brans1985] **BRANS JP.** and VINCKE P., A preference ranking organization method, *Management science*. 31(6): 647-56. 1985. In [Vin2003].
- [Buhler1994] **BUHLER H.**, Réglage par logique floue. Collection Electricité, Presses Polytechnique et Universitaires Romandes. 1994.

- [Burbidge1963] **BURBIDGE** JL. Production flow analysis. *Production Engineer*, 42:742-52. 1963. In [Burbidge1975].
- [Burbidge1975] **BURBIDGE** JL., The introduction of Group Technology. John Wiley. 1975.
- [Cantamessa1997] **CANTAMESSA** M. and TURRONI A., A pragmatic approach to machine and part grouping in cellular manufacturing system design, IJPR, 35(4): 1031-50. 1997.
- [Charnes1961] **CHARNES** A., Cooper W. W., Management models and industrial applications of linear programming. New York, John Wiley & Sons, 1961. In [Anderson1999]
- [Chatelet1996] **CHATELET** JM., Méthodes productique et qualité, Ellipses, 1996.
- [Chen1995a] **CHEN** C., COTRUVO NA., BAEK W., A simulated annealing solution for the cell formation problem. International Journal of Production Research. 33:2601–14. 1995.
- [Chen1995b] **CHEN** SJ. and CHENG C., A Neural Network-Based Cell Formation Algorithm in Cellular Manufacturing. International Journal of Production Research, 33(2):293- 318, 1995.
- [Chen1998] **CHEN** M., A Mathematical Programming Model for System Reconfiguration in a Dynamic Cellular Manufacturing Environment". Annals of Operations Research, 77(1):109-28, 1998.
- [Chen1999] **CHEN** SJ. and HEARAGU S., Stepwise decomposition approaches for large scale cell formation problems, EJOR 113: 64-79. 1999.
- [Cheng1996] **CHENG** CH., GOH, C. H. and LEE, A., Solving the generalized machine assignment problem in group technology. *Journal of the Operational Research Society*, 47, 794-802. 1996.
- [Chetouane1998] **CHETOUANE** F. and Binder Z., Robustesse et flexibilité des les systèmes de production manufacturiers. CNP'1998.
- [Choi1992] **CHOI** M., Manufacturing Cell Design. Production and Inventory Management Journal, Vol. 33 N°2 p. 66-69, 1992. In [Mungwattana2000].
- [Cochet] in [Everaere1999].
- [Collette2002] **COLLETTE** Y. et SIARRY P., Optimisation multiobjectif. Eyrolles, 2002.
- [Coriat1983] in [Everaere1999].
- [Coriat1990] **CORIAT** B., L'atelier et le robot. 1990. in [Everaere1999].
- [Coello2005] **COELLO** C. A. C., PULIDO G. T., MONTES E. M., Current and future trends in evolutionary multiobjective optimization. From Internet.
- [Dahel1993] **DAHEL**, N.-E. and SMITH, S. B., Designing flexibility into cellular manufacturing systems. *International Journal of Production Research*, 31, 933-45. 1993. in [Mansouri2000].
- [Deb2000a] **DEB** K., AGRAWAL S., PRATAP A., MEYARIVAN T., A fast elitist Non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, KanGAL Report N°200001, Indian Institute of technology Kanpur, 2000.
- [Deb2000b] **DEB** K., PRATAP A. and MEYARIVAN T., Constrained test problem for multiobjective evolutionary optimization, KanGal Report N° 200002. Indian Institute of technology Kanpur, 2000.
- [Deb2001] **DEB** K., Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Chichester, UK. 2001, in [Purshouse2002].
- [DeToni1998] **DE TONI** A. and TONICHIA S., Manufacturing flexibility: a literature review. IJPR, 36(6): 1587-617. 1998.
- [Edgeworth1881] **EDGEWORTH** F. Y., Mathematical physics. P. Keagan, London, England, 1881. In [Aguirre2003].
- [Ehrgott2000] **EHRGOTT** M. and GANDIBLEUX X., A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum, 22:425-460, 2000.
- [Everaere1999] **EVERAERE** C. Perrier P., La flexibilité dans les organisations industrielles, Techniques de l'ingénieur, 1999.
- [Fogel1963] **FOGEL** L. J., Biotechnology: Concepts and Applications. Englewood Cliffs, NJ: Prentice Hall. 1963. In [Bently1997]
- [Fonseca1998] **FONSECA** C. M. and Fleming P. J., Multiobjective optimization and multiple constraint handling with evolutionary algorithms — Part II: An application example, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28(1): 38–47. 1998. In [Purshouse2002].
- [Frazier1991] **FRAZIER** GV., and N.GAITHER, 1991, Seed Selection Procedures for Cell Formation Heuristics, *International Journal of Production Research*, 29:2229-2237.
- [Fry1987] **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. In [Mungwattana2000].
- [Garcia1986] **GARCIA** H. and PROTH J.M., 1986, A New Cross-Decomposition Algorithm: The GPM, *Control And Cybernetics*, 15(2), P.155-165.

- [Garey1979] **GAREY M.** and **JOHNSON, D.** Computers and Intractability: A Guide to Theory of NP-completeness. Freeman, San Francisco, CA, 1979.
- [Goldberg1987] **GOLDBERG DE.** and **RICHARDSON J.**, Genetic algorithms with sharing for multimodal function optimization. (Grefenstette J., Editor). Proceedings of the second international conference on Genetic Algorithms. Hillsdale, 1987, pp. 41-49. In [Horn1997].
- [Goldberg1989] **GOLDBERG DE.**, Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley. 1989.
- [Goldberg1993] **GOLDBERG DE.** et al., Rapid Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms, Proceedings of the Fifth International Conference on Genetic Algorithms, edited by Stephanie Forrest, 56-64, San Mateo, CA: Morgan Kaufmann Publishers, July 1993.
- [Gondran1994] **GONDTRAN M.** et **MINOUX M.**, Graphes et Algorithmes, Eyrolles. 1994.
- [Gonçalves2002] **GONÇALVES JF.** and **RESENDE MGC.**, A hybrid genetic algorithm for manufacturing cell formation. AT&TLabs Research Technical Report TD-5FE6RN; 2002.
- [Gupta1990] **GUPTA T.**, and **SEIFODDINI H.** Production data based similarity coefficient for machine-component grouping decision in the design of a cellular manufacturing system., 28:1247-69. 1990.
- [Gupta1995] **GUPTA Y.**, **GUPTA M.**, **KUMAR, A.** and **SUNDARAM 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.
- [Gupta1996] **GUPTA Y.**, **GUPTA M.**, **KUMAR A.** and **SUNDARAM C.**, 1996, A genetic algorithm-based approach to cell composition and layout design problems. *International Journal of Production Research*, 34, 447-82. in [Mansouri2000].
- [Hansen1997] **HANSEN MP.**, Tabu Search in Multiobjective Optimization : MOTS. In Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM'97), Cape Town, South Africa, 1997.
- [Harhalakis1990] **HARHALAKIS G.**, **PROTH J-M**, **XIE X.**, Manufacturing cell design using simulated annealing. International Journal of Production Research.1:185-91. 1990.
- [Harhalakis1994] **HARHALAKIS G.**, **IOANNOU I.**, **MINIS I.** and **NAGI R.**, Manufacturing Cell Formation under Random Product Demand. International Journal of Production Research, 32(1):47-64, 1994.
- [Harik1997] **HARIK G.**, Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty Using Genetic Algorithms, PhD dissertation, University of Michigan, 1997.
- [Harik1999] **HARIK G.**, Linkage Learning via Probabilistic Modeling in the ECGA, IlliGAL Technical Report 99010, January 1999.
- [Heragu1997] **HERAGU SS.** and **CHEN J.**, Optimal solution of cellular manufacturing system design, Benders' decomposition approach. *Eur. J. Oper. Res.*, 1997.
- [Ho1996] **HO YC.** and **MOODIE C.L.**, Solving cell formation problems in a manufacturing environment with flexible processing and routing capabilities. *International Journal of Production Research*, 34, 2901-23. 1996. in [Mansouri2000].
- [Holland1975] **HOLLAND JH.**, *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor. 1975. In [Goldberg1989].
- [Hopcroft1973] **HOPCROFT JE.** and **KARP RM.**, An $O(n^{5/2})$ algorithm for maximum matchings in bipartite graphs, SIAM Journal on Computing, 2:225-31,. 1973. In [Gondran1994].
- [Hormozzi1996] **HORMOZI A.** and **KHUMAWALA B.**, An improved algorithm for solving a multi-period facility location problem, IIE Trans., 28:105-114. 1996.
- [Horn1994] **HORN J.**, **NAFPLIOTIS N.**, **GOLDBERG D. E.**, A Niched Pareto Genetic Algorithm for multiobjective optimization, Proceedings of the first IEEE conference on Evolutionary Computation (ICEC'94), 1994. In [Horn1997].
- [Horn1997] **HORN J.**, The Nature of Niching: Genetic Algorithms and the Evolution of Optimal Cooperative Populations. PhD dissertation, University of Illinois. 1997.
- [Hyer1984] **HYER N.**, The Potential of Group Technology for U.S. Manufacturing. *Journal of Operations Management*, 4(3): 183-202. 1984. In [Mungwattana2000].
- [Irani1991] **IRANI SA.**, **COHEN P. H.** and **CAVALIER T. M.**, Design of Cellular Manufacturing Systems, PED-Vol 53, Design, Analysis, and Control of Manufacturing Cells, ASME. 1991.
- [Irani1993] **IRANI SA.**, **CAVALIER T.M.**, **COHEN P.H.**, Virtual Manufacturing Cells: Exploiting Layout Design And Intercell Flows For The Machine Sharing Problem, *International Journal of Production Research*, (31)4:791-810. 1993.
- [Jayakrishnan1998] **JAYAKRISHNAN NAIR G.** and **NARENDRAN T.T.**, A Clustering Algorithm for Cell Formation with Sequence Data, *International Journal of Production Research*, (36)1:157-79. 1998.

- [Jeon2006] JEON G. and LEEP H.R., Forming part families by using genetic algorithm and designing machine cells under demand changes. *Computers & Operations Research*, 33: 263–83. 2006.
- [Johnson2004] JOHNSON DJ. and Wemmerlöv U., Why does cell Implementation stop? Factors influencing cell penetration in manufacturing plants. *Production and Operations Management*, 13(3):272–289. 2004.
- [Joines1996] JOINES J.A., CULBRETH C.T. and KING R.E., Manufacturing cell design: an integer programming model employing genetic algorithms. *IIE Transactions*, 28: 69-85. 1996.
- [Joines1997] JOINES JA, KAY MG, KING RE., A hybrid genetic algorithm for manufacturing cell design. Technical Report, Department of Industrial Engineering, North Carolina State University; 1997.
- [Jubin1994] JUBIN M., Ateliers flexible d'usinage, Techniques de l'ingénieur, 1994.
- [Kaparthi1992] KAPARTHI S., SURESH NC., Machine-component cell formation in group technology: a neural network approach. *International Journal of Production Research*, 30(6): 1353-67. 1992.
- [Kaufmann1975] KAUFMANN A., Introduction à la théorie des sous-ensembles flous : Tome 3. Application à la classification et à la reconnaissance des formes. *Masson*. 1975.
- [King1980] KING J., Machine-Component Grouping in Production Flow Analysis: An Approach Using a Rank Order Clustering Algorithm. *International Journal of Production Research*, 18(2): 213-32, 1980.
- [Klincewics1994] KLINCEWICS JG. and RAJAN A., Using GRASP to solve the component grouping problem. *Naval Research Logistics*, 41: 893-912. 1994.
- [Knowles2002] KNOWLES JD., Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization. PhD thesis. The University of Reading, January 2002.
- [Knowles2005] KNOWLES JD., Mimetic algorithms for multiobjective optimization: issues, methods and prospects, From Internet. 2005.
- [Kumar1990] KUMAR KR, CHANDRASEKHARAN MP., Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in GT. *IJPR*, 28(2): 233-43. 1990.
- [Kusiak1986] KUSIAK A., VANNELLI A., KUMAR K.R., Clustering analysis: models and algorithms. *Control and Cybernetics*, 15:139-54. 1986.
- [Kusiak1987a] KUSIAK A., The Generalized Group Technology Concept, *International Journal of Production Research*, 25, 561-569. 1987.
- [Kusiak1987b] KUSIAK A., and CHOW W.S., Efficient Solving of The Group Technology Problem. *Journal of Manufacturing Systems*, 6, 117-124. 1987.
- [Kusiak1993] 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.
- [Lamouri1999] LAMOURI S. et THOMAS A., Juste à temps et qualité total : concepts et outils. Techniques de l'ingénieur, AG 5 190. 1999.
- [Lee1997] LEE SD. and CHEN Y.L., A weighted approach for cellular manufacturing design: minimizing intercell movement and balancing workload among duplicated machines. *International Journal of Production Research*, 35, 1125-46. 1997. in [Mansouri2000].
- [Levasseur1995] LEVASSEUR G., HELMS M. and ZINK A., A Conversion from a Functional to a Cellular Manufacturing Layout at Steward, Inc. *Production and Inventory Management Journal*, 36(3):37-42, 1995.
- [Liang1995] LIANG M. and TABOUN S.M., Converting functional manufacturing systems into focused machine cells, a bicriterion approach. *International Journal of Production Research*, 33: 2147-61. 1995.
- [Logendran1991] LOGENDRAN R., Impact of sequence of operations and layout of cells in cellular manufacturing, *IJPR*, 29(2): 375-90.
- [Logendran1993] LOGENDRAN R., A binary integer programming approach for simultaneous machine-part grouping in cellular manufacturing systems. *Computers & Industrial Engineering*, 24:329-36. 1993. in [Mansouri2000].
- [Logendran1994] LOGENDRAN R., RAMAKRISHNA P. and SRISKAANDARAJAH C., Tabu search based heuristics for cellular manufacturing systems in the presence of alternate process plans. *IJPR*, 32(2): 273-97. 1994.
- [Logendran1995] LOGENDRAN R. and RAMAKRISHNA P., Manufacturing cell formation in the presence of lot splitting and multiple units of the same machine. *International Journal of Production Research*, 33, 675-693. 1995.
- [Logendran1997] LOGENDRAN R. and RAMAKRISHNA P., A methodology for simultaneously dealing with machine duplication and part subcontracting in cellular manufacturing systems. *Computers & Operations Research*, 24(2): 97-116. 1997.
- [Mahadevan2003] MAHADEVAN B. and VENKATARAMAIAH S., Re-aligning research objectives in cellular manufacturing system design: A user's perspective. *Asian Journal of Operation Management..* 2003. From Internet.

- [Mamdani1977] **MAMDANI**, EH., Application of fuzzy logic to approximate reasoning using linguistic systems. *Fuzzy Sets and Systems*, 26:1182–91. 1977.
- [Mansouri2000] **MANSOURI** S.A., MOATTAR-HUSSEINI S.M. and NEWMAN S.T., A review of the modern approaches to multi-criteria cell design, *Int. J. Prod. Res.*, 38(5):1201-18. 2000.
- [Mansouri2002] **MANSOURI** S.A., MOATTAR-HUSSEINI S.M. and ZEGORDI SH., Multi-criterion tackling bottleneck machines and exceptional parts in cell formation using genetic algorithms. 2002. In [laniaCo].
- [Maystre1994] **MAYSTRE** L. Y. PICTET J. ET SIMOS J., Méthodes multicritères ELECTRE : Description, conseils pratiques et cas d'application à la gestion environnementale. Presses polytechniques et universitaires romandes, Lausanne, 1994. In [Berro2001].
- [McAuley1972] **McAULEY** J., Machine Grouping For Efficient Production, *Production Engineer*, Vol. 52, February, p.53-57, 1972. In [Souilah1994].
- [Min1993] **MIN** H. and **SHIN** D., Simultaneous formation of machine and human cells in group technology: a multiple objective approach”, *Int. J. Prod. Res.*, 31, 2307-18. 1993.
- [Mitrofanov1959] **MITROFANOV**, S.P., *The scientific principles of group technology*, Lenizdat (National Lending Library, England, 1966-1959. In [Souilah1994, Riezbos2001]
- [Moon1999] **MOON** C and **GEN** M., A genetic algorithm based approach for design of independent manufacturing cells. *International Journal of Production Economics*, 60-61 pp. 421-426. 1999.
- [Mukhopadhyay1995] **MUKHOPADHYAY** S.K. and **GOPALAKRISHNAN** A., A vector analytic (VECAN) method for solving the machine-part grouping problem in GT. *IJPR*, 33(3): 795-818. 1995.
- [Mungwattana2000] **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.
- [Murthy1995] **MURTHY** CVR. and **SRINIVASAN** G., Fractional Cell Formation in Group Technology, *International Journal of Production Research*, 33(5):1323-37. 1995.
- [Nagi1990] **NAGI** R., **HARHALAKIS** G., **PROTH** J. M., Multiple routings and capacity considerations in Group Technology applications, *International Journal of Production Research*, 28:2243-57. 1990.
- [Opitz1970] **OPITZ** H., A classification system to describe workpieces, Pergamon Press Ltd., Oxford, 1970. in [Burbidge1975].
- [Papadimitriou1982] **PAPADIMITRIOU** CH. and **STEIGLITZ** K., Combinatorial optimization: Algorithm and complexity. Prentice Hall, 1982.
- [Pareto1896] **PARETO** V., Cours d'économie politique, vol I et II. F. Rouge, Lausanne 1896. In [Aguirre2003].
- [Pierreval1998] **PIERREVAL** H. and **PLAQUIN** M., An evolutionary approach of multicriteria manufacturing cell formation. *Int. Trans. Opl. Res.* 5(1):13-25. 1998.
- [Purcheck1975] **PURCHECK** GFK., A linear programming method for the combinatorial grouping of an incomplete power set. *Journal of Cybernetics*, 5:51–76. 1975. In [AlBadawi2005].
- [Purshouse2003] **PURSHOUSE** RC., On the evolutionary optimization of many objectives. PhD thesis, University of Sheffield. 2003.
- [Rajagopalan1975] **RAJAGOPALAN** R. and **BARTA** J., Design of manufacturing cellular production systems: a graph-theoretic approach". *International Journal of Production Research*, 13(6):567-79. 1975.
- [Rajamani1996] **RAJAMANI** D., **SINGH** N. and **ANEJA** Y. P., Design of cellular manufacturing systems. *International Journal of Production Research*, 34, 1917-28. 1996.
- [Riezbos2001] **RIEZBOS** J., Design of a period batch control planning system for Cellular Manufacturing. PhD Thesis, ISBN 90-367-1341-2, University of Groningen, The Netherlands. 2001.
- [Rosenblatt1986] **ROSENBLATT** MJ., The dynamics of plant layout. *Management Science*, 32(1): 76-86. 1986.
- [Sankaran1990] **SANKARAN** S., 1990, Multiple objective decision making approach to cell formation: a goal programming model. *Mathematical and Computer Modeling*, 13, 71-81.
- [Sankaran1993] **SANKARAN** S. and **KASILINGAM** R.G., 1993, On cell size and machine requirements planning in group technology. *European Journal of Operational Research*, 69, 373-83.
- [Schaffer1984] **SCHAFFER** JD., Multiple objective optimization with Vector Evaluated Genetic Algorithm, PhD thesis, Vanderbilt University, 1984. in [Berro2001].
- [Schott1995] **SCHOTT** JR., Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, MIT. 1995. In [Knowles2002].
- [Seifoddini1989] **SEIFODDINI** H., Single Linkage v/s Average Linkage Clustering In Machine Cells Formation Applications, *Computers & Industrial Engineering*, 16, 419-426. 1989.
- [Shafer1991] **SHAFER**, SM. and **ROGERS** DF., A goal programming approach to the cell formation problem. *Journal of Operations Management*, 10, 28-43. 1991. In [Mansouri2000].

- [Shafer1992] **SHAFER SM.**, KERN G.M. and WEI J.C., A mathematical programming approach for dealing with exceptional elements in cellular manufacturing. *International Journal of Production Research*, 30, 1029-36. 1992. In [Mansouri2000].
- [Sofianopoulou1997] **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, 501-11. 1997.
- [Souilah1994] **SOUILAH A.**, Les systèmes cellulaires de production : Agencement intracellulaire. Thèse de Doctorat; Université de Metz. 1994.
- [Souilah1994] **SOUILAH A.**, Simulated annealing for manufacturing systems layout design. *European Journal of operational research*, 82:592-614. 1995.
- [Srinivas1993] **SRINIVAS N.** and DEB K., Multiobjective optimization using non-dominated sorting in Genetic Algorithm, Technical report, Indian Institute of technology. 1993.
- [Srinivas1994] **SRINIVAS N.** and DEB K., Multiobjective optimization using non-dominated sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221-48. 1994.
- [Steudel1987] **STEUDEL H.J.** and BALLAKUR A., A dynamic programming based heuristic for machine grouping in manufacturing cell formation, *Computers & Industrial Engineering*, 12:215-22. 1987.
- [Stone2000] **STONE S.**, Implementation of cellular manufacturing. From Internet. 2000.
- [Su1998] **SU CT.**, HSU CM., Multi-objective machine-part cell formation through parallel simulated annealing. *International Journal of Production Research*. 36(8):2185-207. 1998.
- [Sureh1995] **SURESH NC.**, SLOMP J. and KAPARTHI S., The capacitated cell formation problem: a new hierarchical methodology. *International Journal of Production Research*, 33, 1761-84. 1995.
- [Sweeney1976] **SWEENEY D.S.** and TATHAM R.L., An improved long-run model for the problem of multiple warehouse location, *Management Science*. 22(7): 748-58. 1976.
- [Taboun1991] **TABOUN SM.** and SHARMA A., A weighted index for the design of cellular manufacturing systems. *Computers & industrial Engineering*, 21: 273-77, 1991.
- [Taboun1998] **TABOUN SM.**, MERCHANTI NS. and ULGER T., Part Family And Machine Cell Formation In Multi-Period Planning Horizons of Cellular Manufacturing Systems. *Production Planning & Control*. 9(6): 561-71. 1998.
- [Talbi1999] **TALBI E.**, Métaheuristiques pour l'optimisation combinatoire multiobjectif : Etat de l'art, Rapport interne, Université de sciences et technologie de Lille, 1999.
- [Upton1995] **UPTON D.M.**, What really makes factories flexible? Harvard Business Review, July/August 1995. (in [DeToni1998]).
- [Vakharia1990] **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.
- [Vanelli1986] **A.VANELLI, R.KUMAR**, A Method For Finding Minimal Bottleneck Cells For Grouping Part-Machine Families, *International Journal of Production Research*, 24:387-400. 1986.
- [Veldhuizen1999] **VAN VELDHUIZEN D.A.**, Multiobjective Evolutionary Algorithms: Classifications Analyses and New Innovations, PhD Thesis AFIT/DS/ENG/99-01. Air Force Institute of Technology, Wright-Patterson AFB, June 1999.
- [Venugopal1992a] **VENUGOPAL V.**, A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers & Industrial Engineering*, 22(4):469-80. 1992.
- [Venugopal1992b] **VENUGOPAL V.**, NARENDRAN T.T., Cell formation in manufacturing systems through simulated annealing: an experimental evaluation. *European Journal of Operational Research*, 63:409-22 1992. In [Mansouri2000].
- [Verma1995] **VERMA P.** and DING F.Y., A sequence based materials flow procedure for designing manufacturing cells, 33(12):3267-81. 1995.
- [Vin2003] **VIN E.**, DELIT P. et DELACHAMBRE A., Une approche intégrée pour résoudre le problème de formation des cellules de production avec routage alternatifs, 4^{ème} Conférence Francophone de Modélisation et Simulation, MOSIM'03, France. 2003.
- [Wang2003] **WANG J.**, Formation of machine cells and part families in cellular manufacturing systems using a linear assignment algorithm. *Automatica*, 39:1607-15. 2003.
- [Wei1990] **WEI J. C.** and GAITHER N., A capacity constrained multiobjective cell formation method. *Journal of Manufacturing Systems*, 9, 222-32. 1990.
- [Wemmelöv1989] **WEMMERLOV U.** and HYER N., Cellular Manufacturing in the US Industry: A Survey of Users. *International Journal of Production Research*, Vol. 27 N°9 p. 1511-1530, 1989. In [Mungwattana2000].

- [Wemmelöv1997] **WEMMERLOV** U. and JOHN D., Cellular Manufacturing at 46 User Plants: Implementation experiences and Performance Improvements. International Journal of Production Research, Vol. 35 N°1 p. 29-49, 1997. In [Mungwattana2000].
- [Wicks1995] **WICKS** EM. Designing Cellular Manufacturing Systems with Time Varying Product Mix and Resource Availability. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, 1995.
- [Wicks1999] **WICKS** EM. and REASOR RJ., Designing cellular manufacturing systems with dynamic part populations. IIE Transactions, 31: 11-20. 1999.
- [Wolpert1997] **WOLPERT** D. and MACREADY W., No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1): 67-82. 1997.
- [Yin2001] **YIN** Y. and YASUADA K., Similarity coefficient methods applied to the cell formation problem: a taxonomy and review. From Internet. 2001.
- [Ying2000] **YING** H., Fuzzy control and modeling, analytical foundations and applications, IEEE Press. 2000.
- [Zadeh1965] **ZADEH** L.A., Fuzzy Sets. Information and Control, 8:338-53. 1965.
- [Zadeh1975] **ZADEH** L.A., Calculus of fuzzy restrictions. **ZADEH** L.A., K.S.FU, K.TANAKA, M.SHIMURA, (Eds) Fuzzy Sets and their Applications to Cognitive and Decision Processes. *Proceedings of the U.S-Japan Seminar on Fuzzy Sets and their Applications, University of California, Berkeley, California*. Academic Press. 1-39. 1975.
- [Zhao2000] **ZHAO** C. and WU Z., A genetic algorithm approach for manufacturing cell formation with multiple routes and multiple objectives, IJPR, 38(2):385-95. 2000.
- [Zitzler1998] **ZITZLER** E., THIELE L., An evolutionary algorithm for multiobjective optimization: the Strength Pareto approach, TIK-Report 43, Swiss Federal Institute of Technology (Zurich), 1998.
- [Zitzler1999a] **ZITZLER** E., DEB K., THIELE L., Comparison of multiobjective evolutionary algorithms: empirical results. Technical report No. 70. Computer engineering and network laboratory (TIK), Swiss Federal Institute of Technology. 1999.
- [Zitzler1999b] **ZITZLER** E., Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, Swiss Federal Institute of Technology, Zurich, Switzerland. November 1999.
- [Zitzler2001] **ZITZLER** E., LAUMANNS M., and THIELE L., SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology, Zurich, Switzerland, 2001.
- [Zitzler2002] **ZITZLER** E., LAUMANNS M., THIELE L., FONSECA CM., and GRUNERT DA FONSECA V., Why Quality Assessment of Multiobjective Optimizers Is Difficult. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002), pp. 666-673. Morgan Kaufmann Publishers, July 2002. in [Zydallis2003].
- [Zolfaghari1997] **ZOLFAGHARI** S., LIANG M., An objective-guided ortho-synapse Hopfield network approach to machine grouping problems. International Journal of Production Research, 35(10):2773-92. 1997.
- [Zydallis2003] **ZYDALLYS** J. B., Explicit building bloc multiobjective genetic algorithms: Theory, analysis and development. PhD dissertation, Air Force University of Technology, USA, 2003.

• **Sources électroniques (Sites Web, CD)**

- [laniaCo] <http://www.lania.mx/~ccoello/EMOO/EMO> : site extraordinaire regroupant des thèses et des articles traitant de l'optimisation multicritère.
- [DEMAp1999] Dictionnaire Encyclopédique Micro Application 2000. Data Becker, 1999.
- [Geronim] <http://geronim.free.fr> : site exposant un ensemble de concepts et définitions liés au domaine de la production.
- [SMEorga] <http://www.sme.org/index.html> : site de l'organisation "Society of the Manufacturing Engineers".
- [Strateg] <http://www.strategosinc.com> : site supervisé par le chercheur Quaterman Lee.

ANNEXE 1.

LEXIQUE

A

Allèle : *n.m.* Dans le langage génétique se dit d'un caractère héréditaire situé sur un chromosome. Dans celui des algorithmes génétiques, il s'agit d'un élément du vecteur représentatif d'une solution donnée.

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 (*Ang. Cell*).

Centre d'usinage : Machine outils à commande numérique multifonction.

Composition de cellules virtuelles : Approche au problème de composition dans laquelle les cellules n'ont pas le sens d'un groupement physique de machines sur le sol de l'atelier, mais sont vues en tant que tel uniquement par le logiciel de contrôle du système de production [Irani1993].

D

Distance : Soit E un ensemble. On appelle *distance* la fonction numérique $d(\cdot)$ définie sur $E \times E$, telle que :

$$1. \quad d(x,y) \geq 0 \quad 2. \quad (x=y) \Rightarrow (d(x,y)=0) \quad 3. \quad d(x,y) = d(y,x) \quad 4. \quad d(x,z) \leq d(x,y)*d(y,z)$$

* est un opérateur associé à la fonction de distance.

E

Ecole 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 (*Ang. Optimal and pragmatic schools*) [Cantamessa1997].

Economie d'échelle : Baisse du coût unitaire de fabrication d'un produit à mesure qu'augmente le volume de production. On peut expliquer ce phénomène par le fait que l'augmentation de la production ne nécessite pas

de nouveaux investissements à long terme. Les coûts fixes sont donc répartis sur un volume de production plus grand, ce qui permet de réduire le coût moyen de fabrication d'un produit [DEMAP1999].

En-cours : Les produits en cours de fabrication (*Ang. Work-in-progress, WIP*).

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.

Flexibilité : la capacité d'adaptation de la production à tout phénomène nouveau qui peut intervenir sur la fonction production [Everaere1999].

G

Gamme de fabrication : Suite ordonnée d'opérations subies par un produit donné durant sa fabrication avec les temps opératoires correspondants (*Syn. séquence opératoire. Ang. Process Plan*).

H

Horizon de planification : Etendue de temps sur laquelle le système cellulaire de production sera implanté (*Ang. Planning horizon*).

L

LINDO : Acronyme de *Linear INteractive Discrete Optimizer*, logiciel commercial de résolution de programmes linéaires.

M

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

Matrice d'Incidence Produits/Machines : 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.

MIPM : voir **Matrice d'Incidence Produits/Machines**.

O

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

Organisation scientifique de travail : voir *Taylorisme*.

OST : Acronyme de *Organisation scientifique de travail*.

P

Productivité : Rapport mesurable entre la production réalisée (biens, etc.) et les moyens (machines, personnels, etc.) mis en oeuvre pour y parvenir. 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 [DictMApp2000].

R

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

Schème : (Dans le cas d'un codage binaire). On appelle schème (ou schéma) tout chromosome particulier construit sur l'alphabet à trois éléments suivants {0, 1, *} où le symbole (*) indique que la valeur 0 ou 1 n'est pas spécifiée. Par exemple, le schème (*0101) décrit les chaînes suivantes : {(00101), (10101)} [Goldberg1989].

Scénario : Ensemble de couples (P_i, r_i) désignant les produits P_i destinés à être fabriqués avec un ratio de demande r_i durant une période de temps donnée.

SCP : Acronyme de Système(s) Cellulaire(s) de production

Séquence opératoire : Voir *gamme de fabrication*.

Sous-traitance : Exécution d'une partie d'un travail pour le compte d'un autre à qui le travail a été commandé en totalité [DEMAp1999]. Il y a deux types de sous-traitance : la sous-traitance de capacité et la sous-traitance de technicité. La première offre un équipement de production comparable à celui des donneurs d'ordre. La deuxième offre un parc machine hautement spécialisé [Chatelet1996].

Statistical Process Control : Technique basée sur l'étude statistique des produits par échantillonnage et qui cherche à connaître, surveiller, améliorer tous les paramètres affectant un procédé de fabrication (Syn. Maîtrise Statistique des Procédés [Chatelet1996, Lamouri1999]).

Système : Ensemble d'objets interactifs. Un système est défini par trois entités : 1. Les entrées. 2. Les sorties. 3. L'état des objets qui le constituent.

Système dynamique : Système dont l'interaction entre objets provoque des changements éventuels dans les états des objets qui le constituent.

T

Taylorisme : (Syn. *Organisation scientifique de travail*, OST) Méthode proposée par Taylor visant essentiellement l'amélioration de la productivité. Elle préconise que cette amélioration n'est possible que si l'activité réelle des

ouvriers est rigoureusement contrôlée. Le Taylorisme s'appuie sur l'analyse scientifique des gestes, des temps et des pauses. La suppression des gestes inutiles, la décomposition des tâches, l'analyse des outils employés sont nécessaires pour que chaque ouvrier atteignent un niveau de production optimal [DEMAP1999].

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

TG : abréviation de *Technologie de Groupe* (Ang. Group Technology ou GT)

DETAILS DES RESULTATS

du chapitre 8

```
*****  
***** Exemple 1 *****  
***** Les donnees *****  
*****
```

Le nombre de machines :5

Le nombre de produits :4

- Les gammes de produits

Produit 1:2-5-3/4-5

Produit 2:1-3/4-1-5

Produit 3:3/4-1-3/4-5

Produit 4:3/4-2-3/4

- Temps opératoires :

Produit 1:2.00-1.00-3.00-1.00 Ratio:2.00

Produit 2:1.00-3.00-2.00-5.00 Ratio:3.00

Produit 3:4.00-1.00-3.00-2.00 Ratio:1.00

Produit 4:1.00-3.00-1.00 Ratio:2.00

```
*****  
***** Les Contraintes *****  
*****
```

Le Nombre maximum de machines par cell: 3

Le Nombre maximum de cell: 5

Le Nombre minimum de cell: 1

le nbr de contraintes de cohabitation entre les machines:0

le nbr de contraintes de non cohabitation entre les machines:1

les machines de la contraintes de non cohabitation 1:M3 M4

- Les contraintes de charge

Charge Max de M1:10.00 Charge Max de M2:10.00

Charge Max de M3:20.00 Charge Max de M4:20.00

Charge Max de M5:25.00

```
***** AG simple *****
```

Taille de la population:120

Taux de croisement:0.80

Taux de Selection:0.80

Nombre de générations:600

Temps d'exécution:94.37

HyperArea: 0.151

Nbr pareto solutions: 6

HDist:0.151299

```
***** Details des individus du front *****
```

** Composition0

C1:1,3,5, C2:2,4,

** Affectation des operations

Op(1,3):M3 Op(2,2):M3 Op(3,1):M4 Op(3,3):M3 Op(4,1):M4 Op(4,3):M4

** Les charges des machines :

M1:10.00 M2:10.00 M3:18.00 M4:8.00 M5:21.00

** Les charges des cellules :

Ch(C1):49.00 Ch(C2):18.00

** Criteres: TRAFIC: 0.136364 CHARGE: 0.214079

** Composition1

C1:1,4,5, C2:2,3,

** Affectation des operations

Op(1,3):M4 Op(2,2):M4 Op(3,1):M3 Op(3,3):M4 Op(4,1):M3 Op(4,3):M3

** Les charges des machines :

```

M1:10.00      M2:10.00      M3:8.00       M4:18.00      M5:21.00
** Les charges des cellules :
    Ch(C1):49.00      Ch(C2):18.00
** Criteres: TRAFIC: 0.136364 CHARGE: 0.214079
** Composition2
C1:1,3, C2:2,4,5,
** Affectation des operations
Op(1,3):M4 Op(2,2):M3 Op(3,1):M3 Op(3,3):M3 Op(4,1):M4 Op(4,3):M4
** Les charges des machines :
    M1:10.00      M2:10.00      M3:16.00       M4:10.00      M5:21.00
** Les charges des cellules :
    Ch(C1):26.00      Ch(C2):41.00
** Criteres: TRAFIC: 0.181818 CHARGE: 0.050123
** Composition3
C1:1,4, C2:2,3,5,
** Affectation des operations
Op(1,3):M3 Op(2,2):M4 Op(3,1):M4 Op(3,3):M4 Op(4,1):M3 Op(4,3):M3
** Les charges des machines :
    M1:10.00      M2:10.00      M3:10.00       M4:16.00      M5:21.00
** Les charges des cellules :
    Ch(C1):26.00      Ch(C2):41.00
** Criteres: TRAFIC: 0.181818 CHARGE: 0.050123
** Composition4
C1:1,2,3, C2:4,5,
** Affectation des operations
Op(1,3):M4 Op(2,2):M3 Op(3,1):M3 Op(3,3):M4 Op(4,1):M3 Op(4,3):M3
** Les charges des machines :
    M1:10.00      M2:10.00      M3:17.00       M4:9.00       M5:21.00
** Les charges des cellules :
    Ch(C1):37.00      Ch(C2):30.00
** Criteres: TRAFIC: 0.272727 CHARGE: 0.010916
** Composition5
C1:1,2,4, C2:3,5,
** Affectation des operations
Op(1,3):M3 Op(2,2):M4 Op(3,1):M3 Op(3,3):M3 Op(4,1):M4 Op(4,3):M4
** Les charges des machines :
    M1:10.00      M2:10.00      M3:13.00       M4:13.00      M5:21.00
** Les charges des cellules :
    Ch(C1):33.00      Ch(C2):34.00
** Criteres: TRAFIC: 0.318182 CHARGE: 0.000223

```

***** AG-TABU *****

Taille de la population Genetique:40
 Nombre de générations AG:100
 Taux de croisement:0.80
 Taux de Selection:0.80
 Nbr de solutions explorees dans le voisinage:20
 Nombre d'iterations Tabu:50
 Temps d'exécution:6.45
 HyperArea: 0.151
 Nbr pareto solutions: 7
 HDist:0.151299

***** Details des individus du front *****
 ** Composition0
 C1:1,4,5, C2:2,3,
 ** Affectation des operations
 Op(1,3):M4 Op(2,2):M4 Op(3,1):M3 Op(3,3):M4 Op(4,1):M3 Op(4,3):M3
 ** Les charges des machines :
 M1:10.00 M2:10.00 M3:8.00 M4:18.00 M5:21.00
 ** Les charges des cellules :
 Ch(C1):49.00 Ch(C2):18.00
 ** Criteres: TRAFIC: 0.136364 CHARGE: 0.214079
 ** Composition1

```

C1:1,3, C2:2,4,5,
** Affectation des operations
Op(1,3):M4 Op(2,2):M3 Op(3,1):M3 Op(3,3):M3 Op(4,1):M4 Op(4,3):M4
** Les charges des machines :
    M1:10.00    M2:10.00    M3:16.00    M4:10.00    M5:21.00
** Les charges des cellules :
    Ch(C1):26.00    Ch(C2):41.00
** Criteres: TRAFIC: 0.181818 CHARGE: 0.050123
** Composition2
C1:1,4, C2:2,3,5,
** Affectation des operations
Op(1,3):M3 Op(2,2):M4 Op(3,1):M4 Op(3,3):M4 Op(4,1):M3 Op(4,3):M3
** Les charges des machines :
    M1:10.00    M2:10.00    M3:10.00    M4:16.00    M5:21.00
** Les charges des cellules :
    Ch(C1):26.00    Ch(C2):41.00
** Criteres: TRAFIC: 0.181818 CHARGE: 0.050123
** Composition3
C1:1,2,4, C2:3,5,
** Affectation des operations
Op(1,3):M3 Op(2,2):M4 Op(3,1):M4 Op(3,3):M3 Op(4,1):M4 Op(4,3):M4
** Les charges des machines :
    M1:10.00    M2:10.00    M3:9.00    M4:17.00    M5:21.00
** Les charges des cellules :
    Ch(C1):37.00    Ch(C2):30.00
** Criteres: TRAFIC: 0.272727 CHARGE: 0.010916
** Composition4
C1:1,2,3, C2:4,5,
** Affectation des operations
Op(1,3):M4 Op(2,2):M3 Op(3,1):M3 Op(3,3):M4 Op(4,1):M3 Op(4,3):M3
** Les charges des machines :
    M1:10.00    M2:10.00    M3:17.00    M4:9.00    M5:21.00
** Les charges des cellules :
    Ch(C1):37.00    Ch(C2):30.00
** Criteres: TRAFIC: 0.272727 CHARGE: 0.010916
** Composition5
C1:1,2,4, C2:3,5,
** Affectation des operations
Op(1,3):M3 Op(2,2):M4 Op(3,1):M3 Op(3,3):M3 Op(4,1):M4 Op(4,3):M4
** Les charges des machines :
    M1:10.00    M2:10.00    M3:13.00    M4:13.00    M5:21.00
** Les charges des cellules :
    Ch(C1):33.00    Ch(C2):34.00
** Criteres: TRAFIC: 0.318182 CHARGE: 0.000223
** Composition6
C1:1,2,3, C2:4,5,
** Affectation des operations
Op(1,3):M4 Op(2,2):M3 Op(3,1):M4 Op(3,3):M4 Op(4,1):M3 Op(4,3):M3
** Les charges des machines :
    M1:10.00    M2:10.00    M3:13.00    M4:13.00    M5:21.00
** Les charges des cellules :
    Ch(C1):33.00    Ch(C2):34.00
** Criteres: TRAFIC: 0.318182 CHARGE: 0.000223

```

***** SPEA *****

Taille de la population:120
 Taille de l'archive:20
 Taux de croisement:0.80
 Taux de Selection:0.80
 Nombre de générations:100
 Temps d'exécution:178.70
 HyperArea: 0.209
 Nbr pareto solutions: 3
 HDist:0.209370

```
***** Details des individus du front *****
** Composition0
C1:1,4, C2:2,3,5,
** Affectation des operations
Op(1,3):M3 Op(2,2):M4 Op(3,1):M4 Op(3,3):M3 Op(4,1):M3 Op(4,3):M3
** Les charges des machines :
    M1:10.00    M2:10.00    M3:13.00    M4:13.00    M5:21.00
** Les charges des cellules :
    Ch(C1):23.00    Ch(C2):44.00
** Criteres: TRAFIC: 0.181818 CHARGE: 0.098240
** Composition1
C1:1,4, C2:2,3, C3:5,
** Affectation des operations
Op(1,3):M3 Op(2,2):M4 Op(3,1):M4 Op(3,3):M4 Op(4,1):M3 Op(4,3):M3
** Les charges des machines :
    M1:10.00    M2:10.00    M3:10.00    M4:16.00    M5:21.00
** Les charges des cellules :
    Ch(C1):26.00    Ch(C2):20.00    Ch(C3):21.00
** Criteres: TRAFIC: 0.454545 CHARGE: 0.006906
** Composition2
C1:1,4, C2:2,3, C3:5,
** Affectation des operations
Op(1,3):M3 Op(2,2):M4 Op(3,1):M4 Op(3,3):M3 Op(4,1):M3 Op(4,3):M3
** Les charges des machines :
    M1:10.00    M2:10.00    M3:13.00    M4:13.00    M5:21.00
** Les charges des cellules :
    Ch(C1):23.00    Ch(C2):23.00    Ch(C3):21.00
** Criteres: TRAFIC: 0.500000 CHARGE: 0.000891
```

```
*****
***** Exemple 2 *****
*****
***** Les donnees *****
*****
Le nombre de machines :9
```

Le nombre de produits :6

Les gammes de produits

Produit 1:5/6-5/9-3/7-4/6
Produit 2:7/9-3/4
Produit 3:2/7-2/7
Produit 4:4/9-1/9-6/7
Produit 5:1/3-1/4-2/6-5/8
Produit 6:2/3-3/8-4/6

Temps opératoires :

Produit 1:1.23-4.13-4.69-3.65	Ratio:0.94
Produit 2:1.23-3.55	Ratio:0.90
Produit 3:3.85-2.90	Ratio:1.07
Produit 4:3.35-3.07-2.68	Ratio:0.80
Produit 5:4.78-14.10-2.14-3.11	Ratio:0.91
Produit 6:1.91-3.81-1.95	Ratio:0.84

```
*****
***** Les Contraintes *****
*****
Le Nombre maximum de machines par cell: 3
```

Le Nombre maximum de cell: 9

Le Nombre minimum de cell: 1

Le Nombre de contraintes de cohabitation entre les machines:0
Le Nombre de contraintes de non cohabitation entre les machines:1

Les machines de la contrainte de non cohabitation 1: M2 M7
 Les contraintes de charge
 Charge Max de M1:15.00 Charge Max de M2:7.50
 Charge Max de M3:15.00 Charge Max de M4:20.50
 Charge Max de M5:7.50 Charge Max de M6:7.50
 Charge Max de M7:10.50 Charge Max de M8:6.50
 Charge Max de M9:7.50

***** AG simple *****
 Taille de la population:120
 Taux de croisement:0.80
 Taux de Selection:0.80
 Nombre de générations:600
 Temps d'exécution:122.76
 Nbr pareto solutions: 9
 HDist:0.249859

***** Details des individus du front AGS *****
 ** Composition0
 C1:1,2,8, C2:3,4,9, C3:5,6,7,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M4 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:12.83 M2:1.95 M3:13.53 M4:10.94 M5:5.02 M6:0.00 M7:9.39
 M8:2.83 M9:3.57

** Les charges des cellules :
 Ch(C1):17.61 Ch(C2):28.05 Ch(C3):14.41
 ** Criteres: TRAFIC: 0.245344 CHARGE: 0.042290

** Composition1
 C1:1,2,8, C2:3,4,9, C3:5,6,7,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:12.83 M2:1.95 M3:13.53 M4:8.26 M5:5.02 M6:0.00 M7:9.39
 M8:2.83 M9:6.26

** Les charges des cellules :
 Ch(C1):17.61 Ch(C2):28.05 Ch(C3):14.41
 ** Criteres: TRAFIC: 0.245344 CHARGE: 0.042290

** Composition2
 C1:1,2,8, C2:3,4,9, C3:5,6,7,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M2 Op(6,2):M8 Op(6,3):M6
 ** Les charges des machines :
 M1:12.83 M2:3.55 M3:8.74 M4:6.63 M5:5.02 M6:1.63 M7:9.39
 M8:6.02 M9:6.26

** Les charges des cellules :
 Ch(C1):22.40 Ch(C2):21.63 Ch(C3):16.04
 ** Criteres: TRAFIC: 0.322894 CHARGE: 0.010002

** Composition3
 C1:1,2,8, C2:3,4,9, C3:5,6,7,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M2 Op(6,2):M8 Op(6,3):M6
 ** Les charges des machines :
 M1:12.83 M2:3.55 M3:8.74 M4:3.21 M5:5.02 M6:5.05 M7:9.39
 M8:6.02 M9:6.26

** Les charges des cellules :
 Ch(C1):22.40 Ch(C2):18.21 Ch(C3):19.46

** Criteres: TRAFIC: 0.409617 CHARGE: 0.003842
 ** Composition4
 C1:1,2,6, C2:3,5,8, C3:4,7,9,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M2 Op(6,2):M8 Op(6,3):M6
 ** Les charges des machines :
 M1:12.83 M2:3.55 M3:8.74 M4:6.63 M5:5.02 M6:1.63 M7:9.39
 M8:6.02 M9:6.26
 ** Les charges des cellules :
 Ch(C1):18.01 Ch(C2):19.78 Ch(C3):22.28
 ** Criteres: TRAFIC: 0.410451 CHARGE: 0.003827
 ** Composition5
 C1:1,2,9, C2:3,5,6, C3:4,7,8,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M8 Op(6,3):M6
 ** Les charges des machines :
 M1:12.83 M2:1.95 M3:10.34 M4:3.21 M5:5.02 M6:5.05 M7:9.39
 M8:6.02 M9:6.26
 ** Les charges des cellules :
 Ch(C1):21.04 Ch(C2):20.40 Ch(C3):18.62
 ** Criteres: TRAFIC: 0.481794 CHARGE: 0.001308
 ** Composition6
 C1:1,2,8, C2:3,4,9, C3:5,6,7,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M8 Op(6,3):M6
 ** Les charges des machines :
 M1:12.83 M2:1.95 M3:10.34 M4:3.21 M5:5.02 M6:5.05 M7:9.39
 M8:6.02 M9:6.26
 ** Les charges des cellules :
 Ch(C1):20.80 Ch(C2):19.81 Ch(C3):19.46
 ** Criteres: TRAFIC: 0.487168 CHARGE: 0.000401
 ** Composition7
 C1:1,2,5, C2:3,4,8, C3:6,7,9,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M8 Op(6,3):M6
 ** Les charges des machines :
 M1:12.83 M2:1.95 M3:10.34 M4:3.21 M5:5.02 M6:5.05 M7:9.39
 M8:6.02 M9:6.26
 ** Les charges des cellules :
 Ch(C1):19.80 Ch(C2):19.57 Ch(C3):20.70
 ** Criteres: TRAFIC: 0.503382 CHARGE: 0.000300
 ** Composition8
 C1:1,2,5, C2:3,6,9, C3:4,7,8,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:12.83 M2:1.95 M3:13.53 M4:8.26 M5:5.02 M6:0.00 M7:9.39
 M8:2.83 M9:6.26
 ** Les charges des cellules :
 Ch(C1):19.80 Ch(C2):19.79 Ch(C3):20.48
 ** Criteres: TRAFIC: 0.577689 CHARGE: 0.000131

***** AG-TABU *****

Taille de la population Genetique:40

Nombre de générations AG:1100
 Taux de croisement:0.80
 Taux de Selection:0.80
 Nbr de solutions explorees dans le voisinage:20
 Nombre d'iterations Tabu:50
 Temps d'exécution:29.94
 Nbr pareto solutions: 14
 HDist:0.000463
 ***** Details des individus du front AGT *****
 ** Composition0
 C1:1,2,8, C2:3,5,6, C3:4,7,9,
 ** Affectation des operations
 Op(1,1):M6 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M7 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6
 ** Les charges des machines :
 M1:17.18 M2:1.95 M3:9.18 M4:3.21 M5:3.87 M6:6.20
 M7:10.50 M8:2.83 M9:5.15
 ** Les charges des cellules :
 Ch(C1):21.96 Ch(C2):19.24 Ch(C3):18.86
 ** Criteres: TRAFIC: 0.000000 CHARGE: 0.002371
 ** Composition1
 C1:1,2,8, C2:3,5,6, C3:4,7,9,
 ** Affectation des operations
 Op(1,1):M6 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M7 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M4 Op(4,2):M9 Op(4,3):M7 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6
 ** Les charges des machines :
 M1:17.18 M2:1.95 M3:9.18 M4:5.90 M5:3.87 M6:6.20
 M7:10.50 M8:2.83 M9:2.46
 ** Les charges des cellules :
 Ch(C1):21.96 Ch(C2):19.24 Ch(C3):18.86
 ** Criteres: TRAFIC: 0.000000 CHARGE: 0.002371
 ** Composition2
 C1:1,2,8, C2:3,5,6, C3:4,7,9,
 ** Affectation des operations
 Op(1,1):M6 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M4 Op(4,2):M9 Op(4,3):M7 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M5 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:17.18 M2:1.95 M3:9.18 M4:7.53 M5:6.70 M6:4.57 M7:9.39
 M8:0.00 M9:3.57
 ** Les charges des cellules :
 Ch(C1):19.13 Ch(C2):20.44 Ch(C3):20.49
 ** Criteres: TRAFIC: 0.161864 CHARGE: 0.000498
 ** Composition3
 C1:1,2,8, C2:3,5,6, C3:4,7,9,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M7 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M4 Op(4,2):M9 Op(4,3):M7 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M5 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:17.18 M2:1.95 M3:9.18 M4:7.53 M5:7.85 M6:3.42
 M7:10.50 M8:0.00 M9:2.46
 ** Les charges des cellules :
 Ch(C1):19.13 Ch(C2):20.44 Ch(C3):20.49
 ** Criteres: TRAFIC: 0.161864 CHARGE: 0.000498
 ** Composition4
 C1:1,2,8, C2:3,5,6, C3:4,7,9,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M7 Op(2,2):M4 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M5 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :

M1:17.18 M2:1.95 M3:9.18 M4:4.84 M5:7.85 M6:3.42
M7:10.50 M8:0.00 M9:5.15

** Les charges des cellules :
Ch(C1):19.13 Ch(C2):20.44 Ch(C3):20.49

** Criteres: TRAFIC: 0.161864 CHARGE: 0.000498

** Composition5
C1:1,2,8, C2:3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M7 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
Op(5,4):M5 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4

** Les charges des machines :
M1:17.18 M2:1.95 M3:9.18 M4:4.84 M5:7.85 M6:3.42 M7:9.39
M8:0.00 M9:6.26

** Les charges des cellules :
Ch(C1):19.13 Ch(C2):20.44 Ch(C3):20.49

** Criteres: TRAFIC: 0.161864 CHARGE: 0.000498

** Composition6
C1:1,2,8, C2:3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M6 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M4 Op(4,2):M9 Op(4,3):M7 Op(5,1):M1 Op(5,2):M1 Op(5,3):M6
Op(5,4):M5 Op(6,1):M3 Op(6,2):M8 Op(6,3):M4

** Les charges des machines :
M1:17.18 M2:0.00 M3:5.99 M4:7.53 M5:7.85 M6:5.36 M7:9.39
M8:3.19 M9:3.57

** Les charges des cellules :
Ch(C1):20.37 Ch(C2):19.20 Ch(C3):20.49

** Criteres: TRAFIC: 0.239414 CHARGE: 0.000424

** Composition7
C1:1,2,8, C2:3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M6 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M7 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M9 Op(4,2):M1 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6

** Les charges des machines :
M1:15.29 M2:1.95 M3:13.53 M4:6.63 M5:3.87 M6:2.78
M7:10.50 M8:2.83 M9:2.69

** Les charges des cellules :
Ch(C1):20.07 Ch(C2):20.18 Ch(C3):19.82

** Criteres: TRAFIC: 0.319652 CHARGE: 0.000028

** Composition8
C1:1,2,8, C2:3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M6 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M4 Op(4,2):M1 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6

** Les charges des machines :
M1:15.29 M2:1.95 M3:13.53 M4:9.31 M5:3.87 M6:2.78 M7:9.39
M8:2.83 M9:1.11

** Les charges des cellules :
Ch(C1):20.07 Ch(C2):20.18 Ch(C3):19.82

** Criteres: TRAFIC: 0.319652 CHARGE: 0.000028

** Composition9
C1:1,2,8, C2:3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M4 Op(4,2):M1 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6

** Les charges des machines :
M1:15.29 M2:1.95 M3:13.53 M4:9.31 M5:5.02 M6:1.63 M7:9.39
M8:2.83 M9:1.11

** Les charges des cellules :
Ch(C1):20.07 Ch(C2):20.18 Ch(C3):19.82

```

** Criteres: TRAFIC: 0.319652 CHARGE: 0.000028
** Composition10
C1:1,2,8, C2:3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M6 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M9 Op(4,2):M1 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6
** Les charges des machines :
    M1:15.29      M2:1.95      M3:13.53      M4:6.63      M5:3.87      M6:2.78      M7:9.39
    M8:2.83      M9:3.80
** Les charges des cellules :
    Ch(C1):20.07      Ch(C2):20.18      Ch(C3):19.82
** Criteres: TRAFIC: 0.319652 CHARGE: 0.000028
** Composition11
C1:1,2,8, C2:3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M5 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M7 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M4 Op(4,2):M1 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6
** Les charges des machines :
    M1:15.29      M2:1.95      M3:13.53      M4:9.31      M5:5.02      M6:1.63
    M7:10.50      M8:2.83      M9:0.00
** Les charges des cellules :
    Ch(C1):20.07      Ch(C2):20.18      Ch(C3):19.82
** Criteres: TRAFIC: 0.319652 CHARGE: 0.000028
** Composition12
C1:1,2,8, C2:3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M6 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M7 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M4 Op(4,2):M1 Op(4,3):M7 Op(5,1):M3 Op(5,2):M1 Op(5,3):M2
Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6
** Les charges des machines :
    M1:15.29      M2:1.95      M3:13.53      M4:9.31      M5:3.87      M6:2.78
    M7:10.50      M8:2.83      M9:0.00
** Les charges des cellules :
    Ch(C1):20.07      Ch(C2):20.18      Ch(C3):19.82
** Criteres: TRAFIC: 0.319652 CHARGE: 0.000028
** Composition13
C1:1,8, C2:2,3,5,6, C3:4,7,9,
** Affectation des operations
Op(1,1):M6 Op(1,2):M5 Op(1,3):M3 Op(1,4):M4 Op(2,1):M7 Op(2,2):M4 Op(3,1):M7
Op(3,2):M7 Op(4,1):M4 Op(4,2):M9 Op(4,3):M6 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M6
** Les charges des machines :
    M1:17.18      M2:1.95      M3:9.18      M4:9.31      M5:3.87      M6:4.93      M7:8.35
    M8:2.83      M9:2.46
** Les charges des cellules :
    Ch(C1):20.01      Ch(C2):19.92      Ch(C3):20.13
** Criteres: TRAFIC: 0.329658 CHARGE: 0.000009

***** SPEA *****
Taille de la population:120
Taille de l'archive:20
Taux de croisement:0.80
Taux de Selection:0.80
Nombre de générations:100
Temps d'exécution:124.99
HyperArea: 0.252
Nbr pareto solutions: 4
HDist:0.251654
***** Details des individus du front SPEA *****
** Composition0
C1:1, C2:2,3,8, C3:4,7, C4:5,6,9,
** Affectation des operations

```

Op(1,1):M5 Op(1,2):M5 Op(1,3):M7 Op(1,4):M4 Op(2,1):M7 Op(2,2):M4 Op(3,1):M2
 Op(3,2):M2 Op(4,1):M9 Op(4,2):M9 Op(4,3):M6 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:17.18 M2:9.19 M3:4.79 M4:8.26 M5:5.02 M6:2.15 M7:5.50
 M8:2.83 M9:5.15
 ** Les charges des cellules :
 Ch(C1):17.18 Ch(C2):16.81 Ch(C3):13.76 Ch(C4):12.32
 ** Criteres: TRAFIC: 0.248587 CHARGE: 0.006198
 ** Composition1
 C1:1, C2:2,3,8, C3:4,7, C4:5,6,9,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M9 Op(1,3):M7 Op(1,4):M4 Op(2,1):M7 Op(2,2):M4 Op(3,1):M2
 Op(3,2):M2 Op(4,1):M9 Op(4,2):M9 Op(4,3):M6 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:17.18 M2:9.19 M3:4.79 M4:8.26 M5:1.15 M6:2.15 M7:5.50
 M8:2.83 M9:9.01
 ** Les charges des cellules :
 Ch(C1):17.18 Ch(C2):16.81 Ch(C3):13.76 Ch(C4):12.32
 ** Criteres: TRAFIC: 0.248587 CHARGE: 0.006198
 ** Composition2
 C1:1, C2:2,3,8, C3:4,7, C4:5,6,9,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M5 Op(1,3):M7 Op(1,4):M4 Op(2,1):M9 Op(2,2):M3 Op(3,1):M7
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M6 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:17.18 M2:1.95 M3:8.00 M4:5.05 M5:5.02 M6:2.15
 M7:11.63 M8:2.83 M9:6.26
 ** Les charges des cellules :
 Ch(C1):17.18 Ch(C2):12.77 Ch(C3):16.68 Ch(C4):13.43
 ** Criteres: TRAFIC: 0.332345 CHARGE: 0.005547
 ** Composition3
 C1:1, C2:2,3,8, C3:4,7, C4:5,6,9,
 ** Affectation des operations
 Op(1,1):M5 Op(1,2):M9 Op(1,3):M7 Op(1,4):M4 Op(2,1):M9 Op(2,2):M4 Op(3,1):M2
 Op(3,2):M7 Op(4,1):M9 Op(4,2):M9 Op(4,3):M6 Op(5,1):M1 Op(5,2):M1 Op(5,3):M2
 Op(5,4):M8 Op(6,1):M3 Op(6,2):M3 Op(6,3):M4
 ** Les charges des machines :
 M1:17.18 M2:6.08 M3:4.79 M4:8.26 M5:1.15 M6:2.15 M7:7.50
 M8:2.83 M9:10.13
 ** Les charges des cellules :
 Ch(C1):17.18 Ch(C2):13.70 Ch(C3):15.76 Ch(C4):13.43
 ** Criteres: TRAFIC: 0.431761 CHARGE: 0.003513

Exemple 3

***** Les données *****

Le nombre de machines :10

Le nombre de produits :12

Les gammes de produits

Produit 1:2/3-9-7/8
 Produit 2:1/9-7-6/7-5/9-10
 Produit 3:10-3/8-1/2
 Produit 4:7-3/7-9/10-1/5
 Produit 5:10-1-3-7
 Produit 6:1-5/7-4-5/8-8
 Produit 7:7-6/10-9-4/10-2
 Produit 8:5/8-3-9/10

Produit 9:5/9-5-1
 Produit 10:2-6-1/5-9/10
 Produit 11:7-8-4
 Produit 12:9-3/10-4/7
 Temps opératoires :
 Produit 1:3.52-5.40-5.50 Ratio:0.88
 Produit 2:3.53-5.44-3.03-4.85-4.18 Ratio:0.89
 Produit 3:4.18-4.64-4.63 Ratio:0.81
 Produit 4:3.68-2.26-3.17-3.66 Ratio:0.92
 Produit 5:5.82-2.35-4.48-5.17 Ratio:0.87
 Produit 6:2.25-4.13-3.20-5.39-5.12 Ratio:0.98
 Produit 7:5.78-5.71-2.22-2.89-5.91 Ratio:0.93
 Produit 8:3.58-2.67-3.61 Ratio:0.91
 Produit 9:5.92-5.57-3.93 Ratio:0.94
 Produit 10:5.04-5.67-2.45-3.13 Ratio:0.80
 Produit 11:2.51-3.68-5.94 Ratio:0.93
 Produit 12:2.42-4.52-4.33 Ratio:0.82

***** Les Contraintes *****

Le Nombre maximum de machines par cell: 4

Le Nombre maximum de cell: 10

Le Nombre minimum de cell: 1

Le Nombre de contraintes de cohabitation entre les machines:1
 Les machines de la contraintes de cohabitation 1: M1 M10
 Le Nombre de contraintes de non cohabitation entre les machines:1
 Les machines de la contraintes de non cohabitation 1: M5 M7

Les contraintes de charge

Charge Max de M1:83.10 Charge Max de M2:88.80
 Charge Max de M3:95.80 Charge Max de M4:73.90
 Charge Max de M5:94.10 Charge Max de M6:78.80
 Charge Max de M7:91.90 Charge Max de M8:84.90
 Charge Max de M9:100.00 Charge Max de M10:71.50

***** AG simple *****

Taille de la population:120
 Taux de croisement:0.80
 Taux de Selection:0.80
 Nombre de générations:600
 Temps d'exécution:302.75
 Nbr pareto solutions: 18
 HDist:0.344757

***** Details des individus du front AGS *****

** Composition0
 C1:1,3,7,10, C2:2,5,6,9, C3:4,8,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
 Op(12,3):M7
 ** Les charges des machines :

M1:18.20	M2:12.62	M3:10.09	M4:11.35	M5:12.75	M6:9.84
M7:32.77	M8:21.82	M9:15.63	M10:22.07		

** Les charges des cellules :
 Ch(C1):83.13 Ch(C2):50.84 Ch(C3):33.16

** Criteres: TRAFIC: 0.339574 CHARGE: 0.068945

** Composition1

C1:1,3,7,10, C2:2,4,8, C3:5,6,9,

** Affectation des operations
Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
Op(12,3):M7

** Les charges des machines :
M1:18.20 M2:12.62 M3:10.09 M4:11.35 M5:12.75 M6:9.84
M7:32.77 M8:21.82 M9:15.63 M10:22.07

** Les charges des cellules :
Ch(C1):83.13 Ch(C2):45.78 Ch(C3):38.22

** Criteres: TRAFIC: 0.365068 CHARGE: 0.062091

** Composition2
C1:1,3,7,10, C2:2,4,8, C3:5,6,9,

** Affectation des operations
Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M9 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
Op(12,3):M7

** Les charges des machines :
M1:18.20 M2:12.62 M3:10.09 M4:11.35 M5:7.19 M6:9.84
M7:32.77 M8:21.82 M9:21.19 M10:22.07

** Les charges des cellules :
Ch(C1):83.13 Ch(C2):45.78 Ch(C3):38.22

** Criteres: TRAFIC: 0.365068 CHARGE: 0.062091

** Composition3
C1:1,3,7,10, C2:2,5,6,9, C3:4,8,

** Affectation des operations
Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
Op(12,3):M4

** Les charges des machines :
M1:18.20 M2:12.62 M3:10.09 M4:14.89 M5:12.75 M6:9.84
M7:29.22 M8:21.82 M9:15.63 M10:22.07

** Les charges des cellules :
Ch(C1):79.59 Ch(C2):50.84 Ch(C3):36.71

** Criteres: TRAFIC: 0.367268 CHARGE: 0.051286

** Composition4
C1:1,3,7,10, C2:2,5,6,9, C3:4,8,

** Affectation des operations
Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M9 Op(10,3):M5 Op(10,4):M9 Op(12,2):M3
Op(12,3):M4

** Les charges des machines :
M1:18.20 M2:12.62 M3:13.79 M4:14.89 M5:7.19 M6:9.84
M7:29.22 M8:21.82 M9:21.19 M10:18.38

** Les charges des cellules :
Ch(C1):79.59 Ch(C2):50.84 Ch(C3):36.71

** Criteres: TRAFIC: 0.367268 CHARGE: 0.051286

** Composition5
C1:1,3,7,10, C2:2,5,6,9, C3:4,8,

** Affectation des operations
Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
Op(3,3):M1 Op(4,2):M3 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M3
Op(12,3):M4

** Les charges des machines :
M1:18.20 M2:12.62 M3:15.86 M4:14.89 M5:17.08 M6:9.84
M7:27.15 M8:21.82 M9:11.30 M10:18.38

** Les charges des cellules :
Ch(C1):79.59 Ch(C2):50.84 Ch(C3):36.71

** Criteres: TRAFIC: 0.367268 CHARGE: 0.051286

** Composition6

C1:1,3,7,10, C2:2,5,6,9, C3:4,8,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M3
 Op(12,3):M4
 ** Les charges des machines :
 M1:18.20 M2:12.62 M3:13.79 M4:14.89 M5:12.75 M6:9.84
 M7:29.22 M8:21.82 M9:15.63 M10:18.38
 ** Les charges des cellules :
 Ch(C1):79.59 Ch(C2):50.84 Ch(C3):36.71
 ** Criteres: TRAFIC: 0.367268 CHARGE: 0.051286
 ** Composition7
 C1:1,3,7,10, C2:2,4,8, C3:5,6,9,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M9 Op(10,3):M5 Op(10,4):M9 Op(12,2):M3
 Op(12,3):M4
 ** Les charges des machines :
 M1:18.20 M2:12.62 M3:13.79 M4:14.89 M5:7.19 M6:9.84
 M7:29.22 M8:21.82 M9:21.19 M10:18.38
 ** Les charges des cellules :
 Ch(C1):79.59 Ch(C2):49.33 Ch(C3):38.22
 ** Criteres: TRAFIC: 0.392762 CHARGE: 0.049232
 ** Composition8
 C1:1,3,7,10, C2:2,4,8, C3:5,6,9,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M8 Op(2,1):M1 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M3
 Op(12,3):M4
 ** Les charges des machines :
 M1:18.20 M2:12.62 M3:13.79 M4:14.89 M5:12.75 M6:9.84
 M7:29.22 M8:21.82 M9:15.63 M10:18.38
 ** Les charges des cellules :
 Ch(C1):79.59 Ch(C2):49.33 Ch(C3):38.22
 ** Criteres: TRAFIC: 0.392762 CHARGE: 0.049232
 ** Composition9
 C1:1,3,5,10, C2:2,4,6,9, C3:7,8,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
 Op(12,3):M7
 ** Les charges des machines :
 M1:15.05 M2:12.62 M3:10.09 M4:11.35 M5:21.12 M6:9.84
 M7:33.57 M8:16.98 M9:14.45 M10:22.07
 ** Les charges des cellules :
 Ch(C1):68.34 Ch(C2):48.26 Ch(C3):50.54
 ** Criteres: TRAFIC: 0.420320 CHARGE: 0.012974
 ** Composition10
 C1:1,3,5,10, C2:2,4,6,9, C3:7,8,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
 Op(12,3):M7
 ** Les charges des machines :
 M1:15.05 M2:12.62 M3:10.09 M4:11.35 M5:21.12 M6:9.84
 M7:33.57 M8:16.98 M9:14.45 M10:22.07
 ** Les charges des cellules :
 Ch(C1):68.34 Ch(C2):48.26 Ch(C3):50.54
 ** Criteres: TRAFIC: 0.420320 CHARGE: 0.012974

```

** Composition11
C1:1,3,5,10, C2:2,4,6,9, C3:7,8,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M3
Op(12,3):M4
** Les charges des machines :
    M1:11.28      M2:16.39      M3:13.79      M4:14.89      M5:21.12      M6:9.84
    M7:30.03      M8:16.98      M9:14.45      M10:18.38
** Les charges des cellules :
    Ch(C1):64.57      Ch(C2):55.57      Ch(C3):47.00
** Criteres: TRAFIC: 0.447879 CHARGE: 0.008285
** Composition12
C1:1,3,5,10, C2:2,4,6,9, C3:7,8,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
Op(12,3):M7
** Les charges des machines :
    M1:11.28      M2:16.39      M3:10.09      M4:11.35      M5:21.12      M6:9.84
    M7:33.57      M8:16.98      M9:14.45      M10:22.07
** Les charges des cellules :
    Ch(C1):64.57      Ch(C2):52.03      Ch(C3):50.54
** Criteres: TRAFIC: 0.447879 CHARGE: 0.006373
** Composition13
C1:1,3,5,10, C2:2,4,6,9, C3:7,8,
** Affectation des operations
Op(1,1):M2 Op(1,3):M8 Op(2,1):M9 Op(2,3):M7 Op(2,4):M9 Op(3,1):M10 Op(3,2):M3
Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
Op(12,3):M7
** Les charges des machines :
    M1:15.05      M2:12.62      M3:10.09      M4:11.35      M5:16.79      M6:9.84
    M7:28.73      M8:21.82      M9:18.78      M10:22.07
** Les charges des cellules :
    Ch(C1):64.00      Ch(C2):52.59      Ch(C3):50.54
** Criteres: TRAFIC: 0.450554 CHARGE: 0.005650
** Composition14
C1:1,5,10, C2:2,4,6,9, C3:3,7,8,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M1 Op(10,4):M9 Op(12,2):M10
Op(12,3):M4
** Les charges des machines :
    M1:13.24      M2:16.39      M3:10.09      M4:14.89      M5:19.16      M6:9.84
    M7:30.03      M8:16.98      M9:14.45      M10:22.07
** Les charges des cellules :
    Ch(C1):54.47      Ch(C2):55.57      Ch(C3):57.10
** Criteres: TRAFIC: 0.475438 CHARGE: 0.000186
** Composition15
C1:1,5,10, C2:2,4,6,9, C3:3,7,8,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
Op(7,4):M4 Op(8,1):M5 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
Op(12,3):M4
** Les charges des machines :
    M1:15.05      M2:12.62      M3:13.79      M4:14.89      M5:24.39      M6:9.84
    M7:30.03      M8:13.71      M9:15.24      M10:17.59
** Les charges des cellules :
    Ch(C1):57.02      Ch(C2):52.59      Ch(C3):57.53

```

** Criteres: TRAFIC: 0.479229 CHARGE: 0.000793
 ** Composition16
 C1:1,3,5,10, C2:2,4,6,9, C3:7,8,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M5 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
 Op(12,3):M7
 ** Les charges des machines :
 M1:7.93 M2:16.39 M3:10.09 M4:11.35 M5:20.44 M6:9.84
 M7:37.61 M8:16.98 M9:17.75 M10:18.78
 ** Les charges des cellules :
 Ch(C1):57.23 Ch(C2):55.32 Ch(C3):54.58
 ** Criteres: TRAFIC: 0.511900 CHARGE: 0.000200
 ** Composition17
 C1:1,3,6,10, C2:2,4,5,9, C3:7,8,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M1 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M7 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M9 Op(12,2):M10
 Op(12,3):M7
 ** Les charges des machines :
 M1:15.05 M2:12.62 M3:10.09 M4:11.35 M5:17.08 M6:9.84
 M7:37.61 M8:16.98 M9:14.45 M10:22.07
 ** Les charges des cellules :
 Ch(C1):57.05 Ch(C2):55.50 Ch(C3):54.58
 ** Criteres: TRAFIC: 0.546873 CHARGE: 0.000168

***** AG-TABU *****

Taille de la population Genetique:40
 Nombre de générations AG:600
 Taux de croisement:0.80
 Taux de Selection:0.80
 Nbr de solutions explorees dans le voisinage:20
 Nombre d'iterations Tabu:50
 Temps d'exécution:36.45
 Nbr pareto solutions: 13
 HDist:0.335223

***** Details des individus du front AGT *****

** Composition0
 C1:1,5,10, C2:2,4,6,8, C3:3,7,9,
 ** Affectation des operations
 Op(1,1):M3 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
 Op(3,3):M2 Op(4,2):M3 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M1 Op(10,4):M10 Op(12,2):M3
 Op(12,3):M7
 ** Les charges des machines :
 M1:9.89 M2:13.29 M3:15.18 M4:11.35 M5:22.51 M6:4.54
 M7:31.50 M8:20.75 M9:15.24 M10:22.89
 ** Les charges des cellules :
 Ch(C1):55.29 Ch(C2):49.93 Ch(C3):61.92
 ** Criteres: TRAFIC: 0.335071 CHARGE: 0.003872
 ** Composition1
 C1:1,5,10, C2:2,4,6,8, C3:3,7,9,
 ** Affectation des operations
 Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
 Op(3,3):M2 Op(4,2):M3 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
 Op(12,3):M7
 ** Les charges des machines :
 M1:7.93 M2:16.39 M3:12.08 M4:11.35 M5:24.47 M6:4.54
 M7:31.50 M8:20.75 M9:15.24 M10:22.89
 ** Les charges des cellules :

```

Ch(C1):55.29      Ch(C2):53.03      Ch(C3):58.82
** Criteres: TRAFIC: 0.364864 CHARGE: 0.000915
** Composition2
C1:1,5,10, C2:2,4,6,8, C3:3,7,9,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
Op(3,3):M2 Op(4,2):M3 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M1 Op(10,4):M10 Op(12,2):M3
Op(12,3):M7
** Les charges des machines :
    M1:9.89      M2:16.39      M3:12.08      M4:11.35      M5:22.51      M6:4.54
    M7:31.50      M8:20.75      M9:15.24      M10:22.89
** Les charges des cellules :
    Ch(C1):55.29      Ch(C2):53.03      Ch(C3):58.82
** Criteres: TRAFIC: 0.364864 CHARGE: 0.000915
** Composition3
C1:1,5,10, C2:2,4,6,8, C3:3,7,9,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
Op(12,3):M7
** Les charges des machines :
    M1:7.93      M2:16.39      M3:10.01      M4:11.35      M5:24.47      M6:4.54
    M7:33.57      M8:20.75      M9:15.24      M10:22.89
** Les charges des cellules :
    Ch(C1):55.29      Ch(C2):53.03      Ch(C3):58.82
** Criteres: TRAFIC: 0.364864 CHARGE: 0.000915
** Composition4
C1:1,5,10, C2:2,4,6,8, C3:3,7,9,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
Op(3,3):M2 Op(4,2):M3 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M1 Op(10,4):M10 Op(12,2):M3
Op(12,3):M7
** Les charges des machines :
    M1:9.89      M2:16.39      M3:12.08      M4:11.35      M5:22.51      M6:4.54
    M7:31.50      M8:20.75      M9:15.24      M10:22.89
** Les charges des cellules :
    Ch(C1):55.29      Ch(C2):53.03      Ch(C3):58.82
** Criteres: TRAFIC: 0.364864 CHARGE: 0.000915
** Composition5
C1:1,5,10, C2:2,4,6,8, C3:3,7,9,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
Op(12,3):M7
** Les charges des machines :
    M1:7.93      M2:16.39      M3:10.01      M4:11.35      M5:24.47      M6:4.54
    M7:33.57      M8:20.75      M9:15.24      M10:22.89
** Les charges des cellules :
    Ch(C1):55.29      Ch(C2):53.03      Ch(C3):58.82
** Criteres: TRAFIC: 0.364864 CHARGE: 0.000915
** Composition6
C1:1,5,10, C2:2,4,6,8, C3:3,7,9,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
Op(3,3):M2 Op(4,2):M3 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
Op(12,3):M4
** Les charges des machines :
    M1:7.93      M2:16.39      M3:12.08      M4:14.89      M5:24.47      M6:4.54
    M7:27.95      M8:20.75      M9:15.24      M10:22.89

```

** Les charges des cellules :
 Ch(C1):55.29 Ch(C2):56.57 Ch(C3):55.28

** Criteres: TRAFIC: 0.392558 CHARGE: 0.000060

** Composition7
 C1:1,5,10, C2:2,4,6,8, C3:3,7,9,

** Affectation des operations
 Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
 Op(3,3):M2 Op(4,2):M3 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
 Op(12,3):M4

** Les charges des machines :
 M1:7.93 M2:16.39 M3:12.08 M4:14.89 M5:24.47 M6:4.54
 M7:27.95 M8:20.75 M9:15.24 M10:22.89

** Les charges des cellules :
 Ch(C1):55.29 Ch(C2):56.57 Ch(C3):55.28

** Criteres: TRAFIC: 0.392558 CHARGE: 0.000060

** Composition8
 C1:1,5,10, C2:2,4,6,8, C3:3,7,9,

** Affectation des operations
 Op(1,1):M3 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
 Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M10 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M1 Op(10,4):M10 Op(12,2):M3
 Op(12,3):M4

** Les charges des machines :
 M1:13.24 M2:13.29 M3:13.11 M4:12.21 M5:19.16 M6:9.84
 M7:30.03 M8:20.75 M9:11.94 M10:23.56

** Les charges des cellules :
 Ch(C1):55.96 Ch(C2):56.09 Ch(C3):55.08

** Criteres: TRAFIC: 0.425094 CHARGE: 0.000032

** Composition9
 C1:1,5,10, C2:2,4,6,8, C3:3,7,9,

** Affectation des operations
 Op(1,1):M3 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
 Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M10 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M1 Op(10,4):M10 Op(12,2):M3
 Op(12,3):M4

** Les charges des machines :
 M1:9.89 M2:13.29 M3:13.11 M4:12.21 M5:22.51 M6:9.84
 M7:30.03 M8:20.75 M9:11.94 M10:23.56

** Les charges des cellules :
 Ch(C1):55.96 Ch(C2):56.09 Ch(C3):55.08

** Criteres: TRAFIC: 0.425094 CHARGE: 0.000032

** Composition10
 C1:1,5,10, C2:2,4,6,8, C3:3,7,9,

** Affectation des operations
 Op(1,1):M3 Op(1,3):M7 Op(2,1):M9 Op(2,3):M7 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
 Op(3,3):M2 Op(4,2):M7 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M8 Op(7,2):M6
 Op(7,4):M10 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
 Op(12,3):M4

** Les charges des machines :
 M1:11.28 M2:13.29 M3:13.11 M4:12.21 M5:21.12 M6:9.84
 M7:30.03 M8:20.75 M9:11.94 M10:23.56

** Les charges des cellules :
 Ch(C1):55.96 Ch(C2):56.09 Ch(C3):55.08

** Criteres: TRAFIC: 0.425094 CHARGE: 0.000032

** Composition11
 C1:1,5,10, C2:2,4,6,8, C3:3,7,9,

** Affectation des operations
 Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M6 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
 Op(3,3):M2 Op(4,2):M3 Op(4,3):M10 Op(4,4):M1 Op(6,2):M5 Op(6,4):M5 Op(7,2):M6
 Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M1 Op(10,4):M9 Op(12,2):M3
 Op(12,3):M4

** Les charges des machines :
 M1:13.24 M2:16.39 M3:15.86 M4:14.89 M5:24.43 M6:12.55

```

M7:25.25      M8:11.71      M9:14.45      M10:18.38
** Les charges des cellules :
    Ch(C1):56.05      Ch(C2):55.53      Ch(C3):55.56
** Criteres: TRAFIC: 0.574601  CHARGE: 0.000009
** Composition12
C1:1,5,10, C2:2,4,6,8, C3:3,7,9,
** Affectation des operations
Op(1,1):M2 Op(1,3):M7 Op(2,1):M9 Op(2,3):M6 Op(2,4):M5 Op(3,1):M10 Op(3,2):M3
Op(3,3):M2 Op(4,2):M3 Op(4,3):M10 Op(4,4):M5 Op(6,2):M5 Op(6,4):M5 Op(7,2):M6
Op(7,4):M4 Op(8,1):M8 Op(8,3):M10 Op(9,1):M5 Op(10,3):M1 Op(10,4):M9 Op(12,2):M3
Op(12,3):M4
** Les charges des machines :
    M1:9.89      M2:16.39      M3:15.86      M4:14.89      M5:27.78      M6:12.55
    M7:25.25      M8:11.71      M9:14.45      M10:18.38
** Les charges des cellules :
    Ch(C1):56.05      Ch(C2):55.53      Ch(C3):55.56
** Criteres: TRAFIC: 0.574601  CHARGE: 0.000009

***** SPEA *****
Taille de la population:120
Taille de l'archive:20
Taux de croisement:0.80
Taux de Selection:0.80
Nombre de générations:1100
Temps d'exécution:889.96
Nbr pareto solutions: 2
HDist:0.397586

***** Details des individus du front SPEA *****
** Composition0
C1:1,5,6, C2:2,4,8, C3:3,7,9,10,
** Affectation des operations
Op(1,1):M3 Op(1,3):M8 Op(2,1):M9 Op(2,3):M6 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
Op(3,3):M2 Op(4,2):M3 Op(4,3):M9 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
Op(7,4):M10 Op(8,1):M5 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
Op(12,3):M7
** Les charges des machines :
    M1:7.93      M2:13.29      M3:15.18      M4:8.67      M5:27.74      M6:7.25
    M7:23.95      M8:22.33      M9:18.14      M10:22.66
** Les charges des cellules :
    Ch(C1):42.92      Ch(C2):44.28      Ch(C3):79.94
** Criteres: TRAFIC: 0.388699  CHARGE: 0.047328
** Composition1
C1:1,5,6, C2:2,4,8, C3:3,7, C4:9,10,
** Affectation des operations
Op(1,1):M3 Op(1,3):M8 Op(2,1):M9 Op(2,3):M6 Op(2,4):M5 Op(3,1):M10 Op(3,2):M8
Op(3,3):M2 Op(4,2):M3 Op(4,3):M9 Op(4,4):M5 Op(6,2):M5 Op(6,4):M8 Op(7,2):M10
Op(7,4):M10 Op(8,1):M5 Op(8,3):M9 Op(9,1):M5 Op(10,3):M5 Op(10,4):M10 Op(12,2):M3
Op(12,3):M7
** Les charges des machines :
    M1:7.93      M2:13.29      M3:15.18      M4:8.67      M5:27.74      M6:7.25
    M7:23.95      M8:22.33      M9:18.14      M10:22.66
** Les charges des cellules :
    Ch(C1):42.92      Ch(C2):44.28      Ch(C3):39.13      Ch(C4):40.81
** Criteres: TRAFIC: 0.569760  CHARGE: 0.000740

```