

UNE APPROCHE INTÉGRÉE POUR RÉSOUDRE LE PROBLÈME DE FORMATION DES CELLULES DE PRODUCTION AVEC DES ROUTAGES ALTERNATIFS

E. Vin, P. De Lit, A. Delchambre
Département CAD/CAM
Université libre de Bruxelles (ULB)
50 av. F.D. Roosevelt, CP165/14
B-1050 Bruxelles, Belgique.
Mél : emmanuelle.vin@ulb.ac.be

RÉSUMÉ : Dans cet article, nous traitons le problème de formation de cellules de production en présence de routages alternatifs et de contraintes de capacité au niveau des machines. Connaissant la capacité des machines, le processus de fabrication des produits et les quantités à produire, le problème consiste à choisir, pour chaque produit, le routage préférentiel qui permettra d’optimiser la mise en cellules en terme de trafic intercellulaire. L’objectif principal est de minimiser ce trafic en respectant les contraintes de capacité. Nous proposons une approche basée sur l’intégration d’un algorithme génétique de groupement multi-objectifs pour la formation des cellules dans un premier algorithme génétique de groupement multi-objectifs pour le choix du routage préférentiel (en résolvant un problème associé de resource planning).

MOTS CLÉS : Cellule; routage alternatif; algorithme génétique de groupement; multi-objectifs.

1. INTRODUCTION

Les systèmes de production en cellules (cellular manufacturing) sont une des principales applications de la technologie de groupe (group technology) qui consiste à décomposer des systèmes en sous-systèmes en groupant ensemble les éléments similaires du point de vue opératoire ou fonctionnel. Un des principaux problèmes de l’implémentation de ce système cellulaire est celui de la formation de cellules de production (cell formation problem).

Ces dernières années, le problème de formation de cellules a été étudié dans de nombreux travaux. Un panel des différentes approches de ce problème a été proposé par Mungwattana (Mungwattana, 2000). Dans cet article, nous nous focalisons sur la formation de cellules en présence de contraintes de capacité au niveau des machines et en présence de routages alternatifs. Nous considérerons que plusieurs routages sont disponibles pour chaque produit dont la séquence opératoire (process) est déterminée.

De nombreux auteurs proposent différentes méthodes de résolution pour le problème de formation de cellules incluant les routages alternatifs (par exemple Diallo *et al.*, 1993 ; Askin *et al.*, 1997 ; Logendran *et al.*, 1994).

Gupta (Gupta, 1993) a proposé un algorithme en deux étapes. Pendant la première étape, un routage est déterminé définitivement pour chaque produit en respectant les contraintes de capacité. Ensuite, la mise en cellules est effectuée. Le désavantage de cette méthode est son approche séquentielle. La sélection des routages n’est améliorée qu’une fois au début du

traitement et la flexibilité donnée par les routages alternatifs n’est plus utilisée pour minimiser le trafic intercellulaire.

Nagi (Nagi *et al.*, 1990) a proposé une méthode itérative résolvant deux sous-problèmes distincts : la formation de cellules à l’aide d’une heuristique (Harhalakis *et al.*, 1990) et la sélection des routages par l’algorithme du Simplex. À chaque génération, un des deux problèmes est résolu en utilisant la solution de l’autre problème. Cette méthode est limitée par l’utilisation de la méthode du Simplex car le nombre de contraintes devient très important quand la taille des problèmes augmente. De plus, l’utilisation d’une heuristique ne garantit pas la convergence vers un optimum.

Caux (Caux *et al.*, 2000) a imaginé une approche basée sur le recuit simulé et l’algorithme de « branch-and-bound » pour améliorer simultanément la sélection des routages et la minimisation du flux intercellulaire. Dans ce cas, chaque opération est effectuée sur une machine définie et chaque produit possède différentes séquences opératoires.

Une bonne allocation des opérations aux machines disponibles n’implique pas forcément un bon groupement en cellules. Bien au contraire, ces deux buts sont en général contradictoires. Pour éviter les inconvénients de la résolution séquentielle des deux sous-problèmes, la méthode proposée dans cet article groupe simultanément les opérations des produits sur les machines et les machines en cellules. L’approche est basée sur deux algorithmes génétiques de

groupement multi-objectifs (MOGGA) intégrés. Le premier (RP-MOGGA, resource planning) permet de résoudre le problème de sélection des routages préférentiels et le second (CM-MOGGA, cellular manufacturing) intégré au RP-MOGGA traite simultanément le problème de formation de cellules en minimisant notamment le trafic intercellulaire.

Une description détaillée du problème est proposée au paragraphe 2. Le paragraphe 3 présente le MOGGA, l'outil principal utilisé dans l'algorithme. L'algorithme complet de résolution est expliqué dans le paragraphe 4 tandis que les deux MOGGAs sont détaillés respectivement dans les paragraphes 5 et 6. Les conclusions de cette recherche sont présentées au paragraphe 7.

2. DESCRIPTION DU PROBLÈME

Notre problème peut être décomposé en deux problèmes distincts : le groupement des opérations sur les machines et le groupement des machines en cellules de manière à minimiser le trafic intercellulaire.

Les données du problème et les hypothèses formulées sont présentées ci-dessous. Nous considérons un ensemble $M = \{m_1, m_2, \dots, m_m\}$ de m machines dans un système de production donné. Chaque machine n , considérée comme unique, est caractérisée par une disponibilité, d_n . Cette valeur doit tenir compte des pannes possibles et des variations de production. Nous définissons aussi un ensemble $\{p_1, p_2, \dots, p_p\}$ de p produits. Chaque produit k est défini par une quantité Q_k à produire, une et une seule séquence opératoire de NO_k opérations $\{o_{k1}, o_{k2}, \dots, o_{kNOk}\}$ et un facteur de transport FT_k compris entre 0 et 1. Ce facteur de transport doit tenir compte des caractéristiques suivantes : taille, forme, état du produit (liquide, solide, etc.), coût, quantité, risque de dommage du produit. L'ensemble de ces facteurs sont pondérés par un poids variable suivant les priorités que l'on accorde à ces critères.

La différence avec les études précédentes réside dans la définition des séquences opératoires. Chaque opération n'est pas limitée à une machine donnée mais est définie comme un type d'opération pouvant être réalisée par un type de machine donné. Chaque opération peut être réalisée par l'ensemble des machines appartenant à son type. Par exemple, nous pouvons considérer une opération d'alésage à effectuer sur le produit 1. Cette opération d'alésage sera du type "alésage" et pourra donc être effectuée par l'ensemble des machines appartenant au type de machines "aléseuses". Nous définissons donc un ensemble $T = \{tm_1, tm_2, \dots, tm_i\}$ de tm types de machines capables d'exécuter les tm types d'opérations. Chaque machine appartient à au moins un type de machines. Une machine multi-fonctionnelle appartiendra à plusieurs types de machines. Grâce à cette spécification, un produit possède différents

routages potentiels capables de réaliser la séquence opératoire donnée. Les temps opératoires t_{ikn} de chaque opération i des produits k peuvent être fixés de manière générale pour toutes les machines du type ou de manière spécifique pour une machine n .

Le choix du routage est effectué par l'algorithme simultanément à la formation des cellules. Ce concept est illustré à la Figure 1. La séquence opératoire d'un produit est définie par 4 opérations, et donc 4 types de machines. Comme mentionné précédemment, une machine donnée peut appartenir à plusieurs types (ici, m_1 appartient à tm_1 et tm_3). Le routage préférentiel pourrait être $\{m_1, m_3, m_1, m_7\}$.

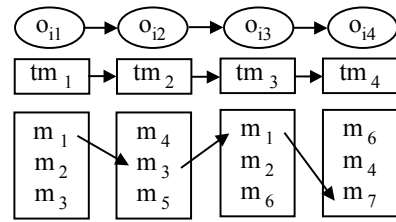


Figure 1. Une séquence opératoire correspond à différents routages.

Pour orienter les groupements, nous définissons un coefficient de similarité (SP_{kl}) entre tous les produits k et l , calculé suivant la méthode d'Irani (Irani et Huang, 2000). Ce coefficient se calcule à l'aide des formules ci-dessous :

$$mc(x, y) = \begin{cases} 1 - \frac{md(y, x) + \frac{id(y, x)}{N_x}}{N_y + 1} & \text{si } N_x > N_y \\ 1 - \frac{md(x, y) + \frac{id(x, y)}{N_y}}{N_x + 1} & \text{si } N_x < N_y \\ \max \left(\begin{array}{l} 1 - \frac{md(x, y) + \frac{id(x, y)}{N_y}}{N_x + 1} \\ 1 - \frac{md(y, x) + \frac{id(y, x)}{N_x}}{N_y + 1} \end{array} \right) & \text{si } N_x = N_y \end{cases}$$

où N_x et N_y représentent respectivement le nombre d'opérations appartenant à la séquence opératoire du produit x et du produit y . Le coefficient $md(x, y)$ (Merger Distance) ainsi que le coefficient $id(x, y)$ (Interruption Distance) sont développés par Irani (Irani et Huang, 2000). Ce coefficient de similarité (mc) est la seule mesure qui exprime l'exacte comparaison de séquence pour détecter les sous-séquences communes. Une comparaison de ce coefficient de similarité avec d'autres coefficients est donné dans (Irani, 2000).

Pour limiter la difficulté de récolte des données, les coûts sont pris en compte par le biais d'une limite inférieure d'utilisation (LL_n) pour chaque machine n . Cette limite s'exprime comme un pourcentage de la disponibilité (d_n) de la machine. La valeur sera proche

de 100 % si le coût de la machine est élevé et si elle doit être utilisée au maximum. À l'inverse, elle pourra être inférieure à 50 % si la machine est peu coûteuse, si elle ne doit pas être utilisée intensément pour être rentable et si elle peut être dédoublée.

De la même manière, on définit une limite supérieure d'utilisation (HL_n). Celle-ci permet d'imposer une certaine flexibilité au système. En cas de problème sur une machine, la production peut alors être réorientée vers une autre machine dont l'utilisation est inférieure à sa limite supérieure d'utilisation. Si l'utilisateur désire une grande flexibilité, il fixera HL_n à une valeur relativement basse (70 % par exemple). Comme le montre la Figure 2, pour différentes utilisations (U) de la machine, ces limites peuvent être dépassées. Elles le sont néanmoins au prix de pénalisations dans l'évaluation de la solution proposée.

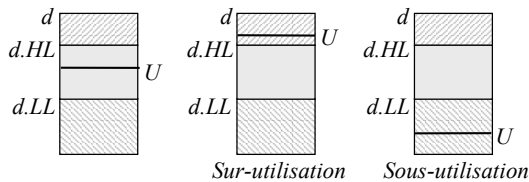


Figure 2. Trois cas d'utilisation de machine (U) en fonction des limites d'utilisation.

3. MO-GGA

Les algorithmes génétiques (GAs) sont une technique d'optimisation inspirée du processus d'évolution des organismes vivants. L'idée essentielle est de maintenir une population de chromosomes, chaque chromosome étant le codage (une description ou génotype) d'une solution (phénotype) du problème à résoudre. La valeur de chaque chromosome est mesurée par son "fitness", souvent simplement calculé par la valeur de la fonction objectif au point de l'espace de recherche défini par le chromosome considéré.

Falkenauer (Falkenauer, 1998) a montré la faiblesse des GAs standards quand ils sont appliqués à des problèmes de groupement. Il a introduit l'algorithme génétique de groupement (GGA), un GA fortement modifié pour supporter la structure des problèmes de groupement. Le but principal de ces problèmes est de grouper des membres d'un ensemble (c'est-à-dire trouver une bonne partition de cet ensemble). Le codage des chromosomes et les opérateurs du GGA (recombinaison, mutation et inversion) manipulent des groupes, plutôt que des éléments à grouper.

L'application des GAs à la résolution des problèmes multi-objectifs (MOP) doit prendre deux objectifs en considération : explorer un vaste espace de solutions et combiner plusieurs objectifs potentiellement conflictuels. L'approche proposée est basée sur l'encapsulation dans le GA de la méthode d'aide à la décision multicritère PROMETHEE II (Brans et

Mareschal, 1994). La description complète de cette méthode sort du cadre de cet article. Cependant, il est intéressant de savoir que le principe général de la méthode est le calcul d'un flux net ϕ entre les solutions. Ce flux net fournit un classement entre les différentes solutions de la population (appelé classement complet de PROMETHEE II). Ce flux net représente le "fitness" de la solution et va donc être utilisé par l'algorithme génétique pour ordonner l'ensemble des solutions.

Les avantages de PROMETHEE II sont expliqués ci-après. Tout d'abord, cette méthode laisse un maximum de liberté sur les critères à utiliser :

- le nombre de critères peut varier par simple ajout ou suppression ;
- les critères à maximiser et à minimiser interviennent simultanément ;
- l'utilisation de fonctions de préférences permet de quantifier les critères et de comparer les solutions. Ces fonctions de préférence permettent de comparer les solutions deux à deux ce qui donne le degré de préférence d'une solution 'a' sur une autre solution 'b' par rapport à un critère.

Ensuite, le poids de chaque critère qui définit l'importance relative des différents objectifs peut être modifié par l'utilisateur de manière transparente. Finalement, la rapidité de l'algorithme permet de tester différentes solutions en jouant sur les critères ainsi que sur leur poids.

L'intégration de l'algorithme PROMETHEE II dans l'algorithme génétique de groupement (MOGGA), utilisé dans notre cas, a été proposé par Rekiek (Rekiek *et al.*, 2002).

4. APPROCHE ITÉRATIVE DE RÉOLUTION

Pour résoudre le problème complet, nous utilisons deux MOGGAs intégrés comme l'illustre la Figure 3. Le premier module (RP-MOGGA) initialise la population de chromosomes. Chaque chromosome représente à ce moment une solution acceptable de groupement des opérations sur les machines spécifiques respectant les contraintes de capacité ainsi que les contraintes de faisabilité d'une opération sur une machine. Après avoir évalué chaque chromosome sur l'ensemble des critères, l'algorithme d'analyse multi-critère est appliqué. Les chromosomes sont ordonnés à l'aide de PROMETHEE II. Les meilleurs chromosomes sont sélectionnés pour être parents de la génération suivante du RP-MOGGA. Les opérateurs génétiques (crossover, mutation et inversion) sont appliqués à ces parents, offrant une nouvelle population qui doit être réévaluée.

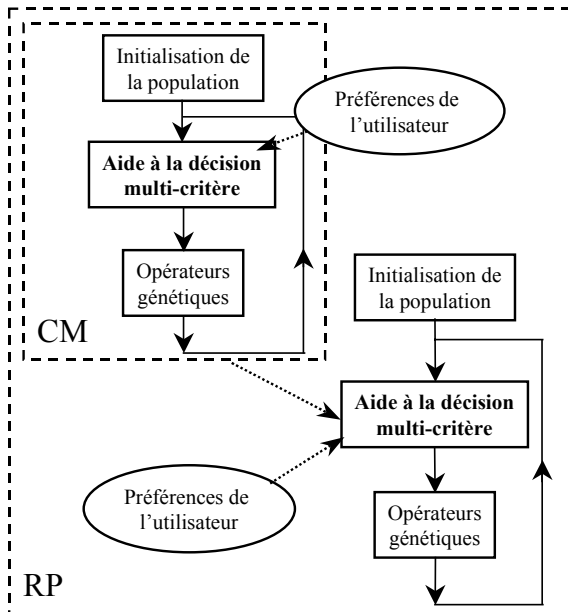


Figure 3. RP-MOGGA (resource planning) avec CM-MOGGA (cellular manufacturing) intégré.

Les critères utilisés dans le RP-MOGGA sont : un coefficient de formation de cellules, ainsi que des coefficients de similarité, d'utilisation des machines, de flexibilité et de coût. Ces coefficients seront détaillés au paragraphe 5.3. Il est important de commenter l'évaluation du critère principal (la formation de cellules). À ce niveau, le chromosome est constitué uniquement de la première partie de la solution, c'est-à-dire du groupement des opérations sur des machines spécifiques. Le groupement des machines en cellules n'a pas encore été réalisé. Pour évaluer ce coefficient de formation de cellules, il faut commencer par trouver la meilleure solution de mise en cellules pour un groupement donné d'opérations. Pour ce faire, un deuxième module de groupement, CM-MOGGA, est intégré au niveau de l'évaluation des solutions. À partir d'une solution de "resource planning", cet algorithme va permettre de trouver la solution de groupement en cellules résultant du meilleur compromis entre tous les critères du CM-MOGGA. Chaque chromosome de la population du RP-MOGGA sera donc complété par une solution de mise en cellules avant d'être évalué.

Ceci constitue l'une des originalités de l'approche : un des résultats d'un MOGGA (les cellules) est utilisé pour évaluer la qualité de la solution proposée par le RP-MOGGA. Ces deux MOGGAs intégrés permettent d'effectuer la recherche suivant les deux axes relatifs d'une part à l'allocation des opérations sur des machines et d'autre part au groupement des machines en cellules.

5. IMPLEMENTATION DE L'ALGORITHME – RP-MOGGA

5.1. Codage des chromosomes et opérateurs

Le codage des chromosomes et des opérateurs génétiques sont ceux classiquement utilisés dans les GGAs (Falkenauer, 1998). Les groupes représentent les machines et les éléments des groupes les opérations.

5.2. Contraintes strictes

Pour qu'un individu de la population soit valide, il doit remplir deux conditions : une contrainte de faisabilité ainsi qu'une contrainte de capacité. La contrainte de faisabilité est définie au niveau de l'opération. D'une part, chaque opération est caractérisée par un type de machine qui limite son allocation aux machines appartenant à ce type de machine. D'autre part, pour qu'une machine soit accessible à l'opération, il ne peut y avoir de restrictions particulières introduites au niveau de l'opération elle-même. Par exemple, si l'opération O_{12} est du type machine m_2 , comprenant la machine m_2 et m_3 , on peut mettre son temps opératoire sur la machine m_3 à zéro ($t_{123}=0$). Cette restriction particulière permet d'empêcher l'utilisation de la machine m_3 pour l'opération O_{12} .

La contrainte de capacité est définie au niveau de la machine. La disponibilité de chaque machine ne peut pas être dépassée. Cette limite pour chaque machine tient compte des pannes, des légères variations de la production et de l'entretien. Il est nécessaire de prévoir une marge pour supporter l'écart par rapport à la demande moyenne.

5.3. Fonction de coût

Comme mentionné au paragraphe 4, les chromosomes sont comparés entre eux à l'aide de la méthode PROMETHEE II (Brans et Mareschal, 1994). Cinq critères, présentés ci-dessous, sont pris en considération : la similarité (RS), l'utilisation des machines multi-fonctionnelles (RM), la flexibilité (RF), le coût (RC) et le groupement en cellules (RG). Chacun de ces critères sera illustré sur l'exemple suivant. Considérons 4 produits $P_1(O_{11}, O_{12}, O_{13})$; $P_2(O_{21}, O_{22})$; $P_3(O_{31}, O_{32}, O_{33}, O_{34}, O_{35})$; $P_4(O_{41}, O_{42}, O_{43})$, produits en quantité respective Q_1 , Q_2 , Q_3 et Q_4 . Supposons que l'assignation sur les 4 machines accessibles est celle représentée à la Figure 4. Les flux directionnels engendrés par cette assignation sont représentés par les flèches entre les machines. Si deux opérations successives sont assignées à la même machine, un flux interne à la machine est représenté.

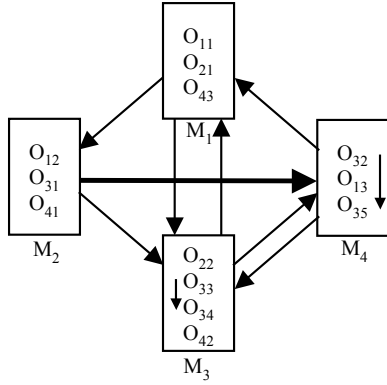


Figure 4. Représentation d'une solution du RP-MOGGA avec CM-MOGGA intégré

Les quatre machines sont caractérisées par une disponibilité de 710 heures et par les paramètres suivants exprimés en nombre d'heures :

Machines	U	$d_n.LL_n$	$d_n.HL_n$
M_1	560	350	650
M_2	430	450	550
M_3	620	260	600
M_4	450	600	700

Tableau 1. Utilisation, limites hautes et basses d'utilisation des machines de l'exemple de la figure 4

5.3.1 Maximiser la similarité

Le coefficient SP_{kl} présenté au paragraphe 2 (Irani et Huang, 2000) est utilisé pour calculer le coefficient S_{kl} filtré par deux paramètres p (seuil de préférence) et q (seuil d'indifférence) de la manière suivante :

Similarité forte : $S_{kl} = 1$ si $p \leq SP_{kl}$;
 Indifférence : $S_{kl} = SP_{kl}$ si $q \leq SP_{kl} < p$;
 Dissimilarité : $S_{kl} = 0$ si $SP_{kl} < q$.

Pour chaque machine, un coefficient RS_n , exprimant la similarité moyenne entre les produits assignés à cette machine, est calculé de la manière suivante :

$$RS_n = \frac{\sum_{l=1}^p \sum_{k=1}^p \sum_{i=1}^{NO_l} \sum_{j=1}^{NO_k} S_{kl} \cdot \delta_{kin} \cdot \delta_{ljn}}{\sum_{q=1}^{NbO_n-1} q}$$

où $\delta_{kin} = 1$ si l'opération i de la pièce k (O_{ki}) est assignée à la machine n , et NbO_n est le nombre d'opérations assignées à la machine n . Dans ce paramètre, on compare deux à deux les produits assignés à la machine et on somme les coefficients de similarité respectifs. Le résultat est divisé par le nombre de comparaisons effectuées pour donner le coefficient RS_n .

Le facteur de similarité RS utilisé pour évaluer la similarité est alors calculé comme la moyenne de ces coefficients RS_n , définis pour chaque machine n .

$$RS = \frac{1}{m} \sum_{n=1}^m RS_n$$

Nous pouvons illustrer ce principe avec la machine m_3 réalisant les opérations ($O_{22}, O_{33}, O_{34}, O_{42}$). Nous devons comparer les similarités entre les pièces P_2, P_3 et P_4 . Supposons que nous avons les coefficients de similarité suivants :

$$SP_{23} = 0,68 ; SP_{34} = 0,83 ; SP_{24} = 0,23 ;$$

les coefficients S_{kl} ($S_{23} = 0,68$; $S_{34} = 1$; $S_{24} = 0$) sont calculés en fonction de seuils d'indifférence ($q = 0,4$) et de préférence ($p = 0,8$) choisis par l'utilisateur. Le coefficient RS_3 de similarité pour la machine m_3 est la somme des six coefficients S_{kl} entre tous les produits concernés par les quatre opérations, comparés deux à deux ($P_2 - P_3$; $P_2 - P_3$; $P_2 - P_4$; $P_3 - P_3$; $P_3 - P_4$; $P_3 - P_4$). Nous avons donc un coefficient moyen de similarité élevé sur la machine 3 :

$$RS_3 = (0,68 + 0,68 + 0 + 1 + 1 + 1) / (3 + 2 + 1) = 0,726.$$

5.3.2 Minimiser le coefficient de "machine multi-fonctionnelle"

L'utilisation du coefficient de similarité présente l'inconvénient que l'algorithme privilégie le groupement d'opérations semblables sur les machines. Pour minimiser le trafic entre les machines, il est nécessaire d'allouer à des machines multi-fonctionnelles les opérations faisant partie de la même séquence de fabrication d'une pièce. Pour cette raison, nous introduisons le coefficient d'utilisation des machines multi-fonctionnelles. Le paramètre MM_k représente le rapport entre le nombre de machines différentes utilisées pour réaliser la pièce k et le nombre d'opérations de la séquence opératoire de la pièce k (NO_k) :

$$RM = \sum_{k=1}^{NO_k} MM_k / NO_k .$$

Ce paramètre ne tient pas compte de la séquence des opérations. Par exemple, si les opérations O_{11} et O_{13} sont placées sur une même machine, le coefficient MM_k sera le même que si ce sont les deux opérations successives O_{11} et O_{12} du produit P_1 qui sont assignées à cette machine k . Cependant, la première solution générera plus de flux que la seconde. Cette dernière sera donc préférée du point de vue du critère de mise en cellules comme nous le verrons au paragraphe 5.3.5 .

Le coefficient RM (à minimiser) de la solution proposée à la figure 4 sera

$$RM = \frac{\frac{3}{3} + \frac{2}{2} + \frac{3}{5} + \frac{3}{3}}{4} = 0,9 .$$

5.3.3 Minimiser le coefficient de flexibilité

Un des objectifs de l'approche proposée est de maximiser la flexibilité de l'environnement de production. Nous allons essayer d'obtenir une configuration du système de production qui permette de supporter les variations de la demande ainsi que les pannes. Lorsqu'une machine tombe en panne, il est utile de pouvoir réassigner sa charge sans pour autant produire un grand retard de production ou revoir toute l'allocation des opérations.

Pour ce faire, nous utilisons les limites hautes d'utilisation que nous avons introduites au paragraphe 2. Ces limites ne sont pas des contraintes strictes mais des contraintes douces qu'il serait préférable de respecter. Nous pénalisons donc tous les dépassements des limites supérieures d'utilisation ($d_n \cdot HL_n$) fixées pour chaque machine (voir paragraphe 2). Le coefficient RF représente la moyenne des pénalisations calculées sur toutes les machines utilisées :

$$RF = \frac{1}{m} \sum_{n=1}^m \max \left\{ 0, \frac{(U_n - HL_n \cdot d_n)}{(d_n - HL_n \cdot d_n)} \right\}.$$

Pour le calcul des pénalisations, nous utilisons une valeur relative car un dépassement de 2 heures de la limite haute d'utilisation n'a pas la même importance si la capacité que l'on veut laisser libre sur la machine est de 4 heures ou de 20 heures. Dans le premier cas, on dépasse de 50 % la limite tandis que dans le deuxième cas, le dépassement ne représente que 10%. Sur cet unique critère, l'algorithme préférera donc la deuxième solution à la première.

Pour l'exemple présenté à la figure 4, nous avons un seul dépassement de la limite haute d'utilisation. Le calcul du coefficient RF nous donne :

$$RF = \frac{(620 - 600)/(710 - 600)}{4} = 0,04545.$$

5.3.4 Minimiser le coefficient de coût

L'utilisation de la limite basse introduite au paragraphe 2 nous permet d'évaluer la rentabilité de la solution proposée. Toute sous-utilisation de machines introduit une pénalisation relative correspondant au manque de rentabilité. Les objectifs de coût sont exonsidérés par la pénalisation de toute utilisation de machines inférieure à sa limite inférieure d'utilisation ($d_n \cdot LL_n$) :

$$RC = \frac{1}{m} \sum_{n=1}^m \max \left\{ 0, \frac{(LL_n \cdot d_n - U_n)}{LL_n \cdot d_n} \right\}.$$

Pour l'exemple présenté à la figure 4, deux machines sont sous-utilisées. Ces machines m_2 et m_4 entraînent

donc des pénalités. Le calcul du coefficient RC nous donne :

$$RC = \frac{(450 - 430)/450 + (600 - 450)/600}{4} = 0,07361.$$

Le coefficient RF ainsi que le coefficient RC n'ont pas de réalité physique mais sont utilisées par PROMETHEE II pour comparer les solutions entre elles. Dans ces cas, nous ne pouvons pas parler de flexibilité ou de coût moyen.

5.3.5 Coefficient de groupement en cellules

Après l'application du second MOGGA qui groupe les machines en cellules, le coefficient de groupement est calculé sur base des flux intercellulaires résultant de l'allocation des opérations aux ressources de production. On calcule le flux intracellulaire (ϕ_{intra}) et on le divise par le flux total (ϕ_{tot}) pour obtenir le coefficient RG :

$$RG = \phi_{intra} / \phi_{tot}.$$

Dans le paragraphe suivant, nous précisons les différents critères qui interviendront lors du groupement des machines en cellules. La solution proposée sera optimisée sur l'ensemble des critères du CM-MOGGA. Cependant le coefficient RG exprimera seulement le pourcentage du flux intracellulaire de la meilleure solution.

6. IMPLEMENTATION DE L'ALGORITHME – CM-MOGGA

6.1. Initialisation

L'initialisation du MOGGA est basée sur l'heuristique d'Harhalakis (Harhalakis *et al.*, 1990) modifiée pour la rendre partiellement aléatoire, afin d'obtenir des individus différents à l'initialisation. On commence par créer pour chaque machine une cellule. Ensuite, on recherche le flux intercellulaire le plus élevé entre deux cellules. Pour peu que la taille maximale de cellule ne soit pas atteinte si on groupe ces deux cellules, on a 85% de chance de les grouper. Si ces deux cellules ne sont pas retenues, le flux suivant le plus élevé est recherché. Après chaque groupement de cellules, les flux intercellulaires correspondant sont convertis en flux intracellulaire.

Pour utiliser cette heuristique, il faut exprimer le flux existant entre chaque machine. Trois matrices de flux différentes sont utilisées dans le MOGGA.

6.1.1 Matrice de flux X

Dans cette matrice, les flux entre les machines sont exprimés en nombre d'heures de travail sur la machine de destination. Tout produit dont deux opérations successives sont assignées respectivement à la machine m_1 et la machine m_2 , engendre un flux entre ces deux

machines dont la valeur sera égale au nombre d'heures de travail à effectuer sur le produit par la machine n_2 . Le coefficient $x_{n_1-n_2}$ sera égal à la somme de tous les flux (partant de la machine n_1 et allant vers la machine n_2) engendré par l'ensemble des opérations de tous les produits entre les deux machines :

$$x_{n_1-n_2} = \sum_{k=1}^{NP} \sum_{i=1}^{NO_k-1} Q_k \cdot t_{k,(i+1),n_2} \cdot \delta_{k,i,n_1} \cdot \delta_{k,(i+1),n_2}$$

où $\delta_{k,i,n_2} \delta_{k,(i+1),n_2} = 1$ si l'opération i est réalisée sur la machine n_1 et $i+1$ sur n_2 .

Si nous calculons la matrice de flux X pour la solution proposée à la figure 4, nous obtenons le tableau 2.

X	M ₁	M ₂	M ₃	M ₄
M ₁	0	Q ₁ .t ₁₂₂	Q ₂ .t ₂₂₃	0
M ₂	0	0	Q ₄ .t ₄₂₃	Q ₁ .t ₁₃₄ ⁺ Q ₃ .t ₃₂₄
M ₃	Q ₄ .t ₄₃₁	0	Q ₃ .t ₃₄₃	Q ₃ .t ₃₅₄
M ₄	0	0	Q ₃ .t ₃₃₃	0

Tableau 2. Matrice de flux X pour l'exemple de la figure 4

6.1.2 Matrice de flux Y

Dans la matrice de flux Y , les flux directionnels sont exprimés en nombre de lots transférés entre deux machines pondérés par le facteur de transport des produits. Deux opérations successives d'un même produit impliquent un flux entre les deux machines n_1 et n_2 , dont la valeur est égale au nombre de lots transférés pondéré par le facteur de transport du produit.

$$y_{n_1-n_2} = \sum_{k=1}^{NP} \sum_{i=1}^{NO_k-1} FT_k \cdot Q_k / TL_k \cdot \delta_{k,i,n_1} \cdot \delta_{k,(i+1),n_2}$$

où TL_k représente la taille de lot du produit k exprimant la difficulté liée au transport du produit. La matrice de flux Y relative à l'exemple de la figure 4 est présentée dans le tableau 3.

Y	M ₁	M ₂	M ₃	M ₄
M ₁	0	N ₁ .Ft ₁	N ₂ .Ft ₂	0
M ₂	0	0	N ₄ .Ft ₄	N ₁ .Ft ₁ ⁺ N ₃ .Ft ₃
M ₃	N ₄ .Ft ₄	0	N ₃ .Ft ₃	N ₃ .Ft ₃
M ₄	0	0	N ₃ .Ft ₃	0

Tableau 3. Matrice de flux net Y pour l'exemple de la Figure 4

6.1.3 Matrice de flux Z

Cette matrice exprime le flux net unidirectionnel qui existe entre les machines. Les coefficients de cette matrice sont calculés à partir de la matrice Y et expriment la différence des flux entre deux machines n_1 et n_2 si elle est positive :

$$z_{n_1-n_2} = \max(0 ; y_{n_1-n_2} - y_{n_2-n_1})$$

La matrice Z relative à la solution de la figure 4 est calculée à partir de la matrice Y du tableau 3.

Z	M ₁	M ₂	M ₃	M ₄
M ₁	0	Y ₁₂	0	0
M ₂	0	0	Y ₂₃	Y ₂₄
M ₃	Y ₃₁ -Y ₁₃	0	0	Y ₃₄ -Y ₄₃
M ₄	0	0	0	0

Tableau 4. Matrice de flux unidirectionnel Z pour l'exemple de la Figure 4

6.2. Contraintes dures

Deux conditions doivent être satisfaites par les individus du MOGGA. D'une part, le nombre maximum de cellules ne peut être dépassé et d'autre part, la taille des cellules doit être respectée. Cette taille est limitée par le nombre de machines présentes, par la surface utilisée et/ou par le nombre d'utilisateurs nécessaires pour faire fonctionner la cellule.

6.3. Fonction de coût

De la même manière que pour le RP-MOGGA, les chromosomes sont comparés entre eux à l'aide de la méthode PROMETHEE II. Deux critères, présentés ci-dessous, interviennent : les flux intracellulaires (RF) et les flux intercellulaires directionnels (RFD).

6.3.1 Maximiser le flux intracellulaire (RF)

À partir de la matrice X des flux exprimée en nombre d'heures de travail de destination, nous pouvons calculer T_{cd} . Ce coefficient représente l'ensemble des flux partant d'une cellule c et aboutissant à une cellule d .

$$\Phi_{intracell} = \sum_{c=1}^{nbCell} T_{cc}$$

$$\text{où } T_{cd} = \sum_{i=1}^m \sum_{j=1}^m x_{n_i-n_j} \cdot \delta_{ic} \cdot \delta_{jd}$$

avec $\delta_{ic}=1$ si la machine n_i se trouve dans la cellule c .

6.3.2 Minimiser le flux intercellulaire directionnel (RFD)

Si les cellules ne sont pas parfaitement indépendantes (c'est-à-dire qu'il existe un flux intercellulaire), nous allons préférer qu'une cellule ne soit visitée qu'une seule fois, sans retour. Le principe est de privilégier les flux intercellulaires à faible facteur de transport si on a le choix. Dans le calcul du RFD, le paramètre φ_{cd} représente le flux net directionnel total de la cellule c à la cellule d ($\varphi_{dc}=0$).

$$\Phi_{\text{intercell-direct}} = \sum_{c=1}^{nbCell} \sum_{d=c+1}^{nbCell-1} \varphi_{cd}$$

$$\text{où } \varphi_{cd} = \max(0, \sum_{i \in C_c} \sum_{j \in C_d} z_{ij} - \sum_{i \in C_c} \sum_{j \in C_d} z_{ji})$$

7. CONCLUSIONS

Dans cet article, nous présentons une méthode innovante pour résoudre le problème de formation de cellules avec des routages alternatifs, basée sur deux algorithmes génétiques de groupement multi-objectifs, prenant en compte différents critères. Considérant les trois paramètres de production importants dans la formation de cellules (le volume de production, les séquences opératoires et les routages alternatifs), la méthode optimise la formation de cellules et choisit le routage préférentiel de chaque produit. À l'avenir, il sera intéressant de tester différents cas industriels avec un jeu de données assez important, pour étudier l'apport de l'intégration d'un CM-MOGGA de mise en cellules dans le RP-MOGGA par rapport à l'intégration d'une heuristique de groupement (Vin *et al.*, 2002). De plus, il est nécessaire de créer une heuristique spécifique pour l'allocation des opérations sur les machines qui tient compte des critères d'évaluation. Enfin, les routages alternatifs ont été envisagés pour optimiser la formation de cellules. Ils devraient être utilisés au niveau de l'évaluation pour tester la flexibilité de la solution en cas de panne. Le même travail devra être réalisé en tenant compte des séquences opératoires alternatives.

8. REMERCIEMENTS

Ce projet est réalisé avec la collaboration du Centre de Recherche belge pour l'Industrie métallurgique Francophone (CRIF) et financé partiellement par la Région wallonne et la Région flamande

9. REFERENCES

Askin R., Selim H. and Vakharia A., 1997. A methodology for designing flexible cellular manufacturing systems. *IIE transactions*, 29(7), pp. 599-610.
 Brans J.-P., Mareschal B., 1994. The PROMCALC & GAIA decision support system for multicriteria

decision aid. *Decision support systems*, 12, p. 297-310.
 Caux C., 2000. Cell formation with alternative process plans and machine capacity constraints: A new combined approach. *International Journal of Economics*, 64, p. 279-284.
 De Lit P., Falkenauer E., Delchambre A., 2000. Grouping genetic algorithms: an efficient method to solve the cell formation problem. *Mathematics and Computers in Simulation*, 51, p. 257-271.
 Diallo M., Pierreval H., Quilliot A., 1993. Manufacturing cells design with flexible routing capability in presence of unreliable machines. *International Journal of Production Research*, 74(1), p.175-182.
 Falkenauer E., 1998. Genetic algorithm and grouping problem. *Wiley & Sons*.
 Gupta T., 1993. Design of manufacturing cells for flexible environment considering alternative routing. *International Journal of Production Research*, 31(6), p. 1259-1273.
 Harhalakis G., Nagi R., Proth J.-M. 1990. 'An efficient heuristic in manufacturing cell formation for group technology applications'. *International Journal of Production Research* 28(1), p. 185-198.
 Holland J. H., 1975. Adaptation in Natural and Artificial Systems. *Ann Arbor:University of Michigan Press*.
 Irani A., Huang H., 2000. 'Custom design of facilities layouts for multi-product facilities using layout modules'. *IEEE Transaction Robotocs and Automation*, 16(3), p. 259-267.
 Logendran R., Ramakrishna P., Srikandarajah C., 1994. Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans. *International Journal of Production Research*, 32(2), p. 273-297.
 Mungwattana A., 2000, *Design of cellular manufacturing systems for dynamic and uncertain production requirements with presence of routing flexibility*. PhD thesis, Blacksburg, Virginia.
 Nagi R., Harhalakis G., Proth J.-M., 1990. Multiple routings and capacity considerations in group technology applications. *International Journal of Production Research*, 28(12), p. 2243-2257.
 Rekiek B., De Lit P. Delchambre A., 2002. Hybrid assembly line design and user's preference. *International Journal of Production Research*, 40(5), p. 1095-1111.
 Vivekanand P., Narendran T.T., 1998. Logical Cell Formation in FMS, Using Flexibility-Based Criteria. *International Journal of Flexible manufacturing System*, 10, p. 163-181.
 Vin E., Delit P., Delchambre A., 2002. A new combined approach to solve the Cell formation problem with alternative routings, *Proceeding of 9th International Multi-Conference Advanced Computer Systems (ACS'2002)*, Poland, Miedzydroje, 1, p. 43-50.