7e Conférence Francophone de MOdélisation et SIMulation - MOSIM'08 - du 31 mars au 2 avril 2008 - Paris - France
«Modélisation, Optimisation et Simulation des Systèmes : Communication, Coopération et Coordination»

Couplage Planification/Ordonnancement : une approche par décomposition et génération de coupes

O. Guyon[1] [2], P. Lemaire[2], É. Pinson[1] et D. Rivreau[1]

[1] Institut de Mathématiques Appliquées
44, rue Rabelais
B.P. 10808 - 49008 Angers Cedex 01
{olivier.guyon, eric.pinson, david.rivreau}@uco.fr

[2] École des Mines de Nantes
La Chantrerie - 4, rue Alfred Kastler
B.P. 20722 - 44307 Nantes Cedex 3
{pierre.lemaire}@emn.fr

RÉSUMÉ : *Cette étude porte sur un problème couplant les deux phases décisionnelles, dans un processus de production, que sont la planification d'agents et l'ordonnancement de production. D'un côté, nous devons gérer au moindre coût les emplois du temps d'un ensemble d'opérateurs (planification d'agents) et de l'autre, nous nous attelons à fournir un plan de production réalisable, compte tenu des ressources disponibles déduites de la phase de planification, et qui vise à satisfaire une certaine demande fournie en entrée (ordonnancement de production). Contrairement à la plupart des articles de la littérature, nous proposons ici des techniques de résolution résolvant simultanément ces deux problèmes. Deux approches exactes ont été étudiées : l'une d'elles est fondée sur une décomposition de Benders et l'autre est basée sur un processus spécifique de génération de coupes. L'intérêt de ces méthodes est discuté ici au travers de résultats expérimentaux.*

MOTS-CLÉS : *planification d'agents, ordonnancement, génération de coupes, Benders*

1. INTRODUCTION

Dans les systèmes de production, l'ordonnancement de production et la gestion de personnel sont deux problèmes généralement considérés comme trop complexes pour être traités simultanément. D'un côté, l'ordonnancement de production cherche à affecter des ressources (humaines ou matérielles) à différentes tâches (jobs) à réaliser. D'un autre côté, la gestion de personnel tend, la plupart du temps, à réduire les coûts de main d'œuvre. Nous nous référons à (Ernst, Jiang, Krishnamoorthy & Sier 2004, Soumis, Pesant & Rousseau 2005) pour un état de l'art sur le problème de gestion d'horaires et d'affectation du personnel. Bien qu'il semble admis que la meilleure approche pour résoudre ce problème global d'optimisation soit une prise en compte simultanée des deux sous-problèmes, en pratique, il est souvent résolu dans un processus de décision à deux niveaux. Ceci est essentiellement dû au fait qu'aucune technique de résolution n'a été jugée, pour le moment, suffisamment efficace pour résoudre le problème global concrètement. Néanmoins, nous pouvons mentionner un certain nombre de travaux, dont certains fournissent des résultats encourageants, proposant diverses méthodes pour résoudre le problème non décomposé et ce, dans le domaine de la production mais également dans celui du transport. (Artigues, Gendreau & Rousseau 2006) présente un état de l'art sur ces deux domaines. Dans cet article, nous intégrons l'idée proposée dans (Lasserre 1992) et développée dans (Dauzère-Pérès & Lasserre 1994) qui consiste à utiliser une approche multi-passe afin de résoudre alternativement un problème de production à deux niveaux différents (le premier fixant les tailles des lots à produire pour une séquence de jobs donnée sur une machine et le second établissant les séquences de jobs à réaliser suivant les tailles de lots à produire). Une telle décomposition a déjà été appliquée avec succès dans (Detienne, Péridy, Pinson & Rivreau 2007) pour un problème de planification d'agents avec une charge de travail prédéterminée.

Nous intégrons ici les deux sous-problèmes et proposons des méthodes exactes pour résoudre le problème résultant. La section suivante statue sur deux formalisations du problème. La section 4 décrit une méthode exacte fondée sur une décomposition de Benders. Nous introduisons ensuite une seconde méthode exacte basée, pour sa part, sur un processus spécifique de décomposition et de génération de coupes. Nous discutons enfin de l'intérêt de ces techniques de résolution en comparant leurs résultats expérimentaux.

2. FORMALISATION DU PROBLÈME

L'objet de cette étude est l'ordonnancement d'un ensemble J de n jobs (tâches) indépendants au moyen d'un ensemble O de m ressources (opérateurs) sur un horizon temporel H. Chaque job est caractérisé par une durée p_j, un domaine d'exécution $D_j = [r_j, d_j]$, et requiert pour son exécution un opérateur $o \in O$ maîtrisant une compétence $c_j \in C$. Notons que les jobs sont préemptifs et qu'ils peuvent, sous réserve de contraintes de compétence et de disponibilité, être exécutés par différents opérateurs. On ne peut cependant effectuer plus d'une unité d'un même job par instant. Il ne peut donc y avoir qu'au plus un opérateur affecté à un job sur un instant $t \in H$. Chaque opérateur possède un ensemble de compétences $C_o \subseteq C$ qu'il maîtrise ainsi qu'un ensemble de roulements $\Omega_o \subseteq \Omega$ qui peuvent lui être affectés. Un roulement $\omega \in \Omega$ est composé d'une suite ordonnée de Profils Horaires Hebdomadaires (PHH), définissant un canevas d'horaires de présence ou d'absence à l'échelle d'une semaine. Typiquement, un PHH peut représenter les horaires d'une semaine de travail de nuit, du matin ou du soir, qui peuvent être combinés en un roulement de trois-huit. Cette formalisation permet la prise en compte de nombreuses contraintes législatives, contractuelles ou autres (congés posés ...). Chaque couple roulement ω - opérateur o est caractérisé par un coût η^o_ω symbolisant le coût de main d'oeuvre engendré lorsque l'opérateur o se voit affecter le roulement ω, ce coût pouvant dépendre des horaires de travail mais aussi du statut de l'opérateur.

L'objectif du problème est alors de déterminer un ordonnancement de l'ensemble des n jobs en affectant à chaque opérateur un roulement de telle sorte que chacun des besoins en main d'oeuvre (effectif et compétence) soit rempli au moindre coût.

2.1 Formalisation indexée sur le temps

À partir de ces notations, nous proposons une première formalisation intuitive de la problématique en termes de Programme Linéaire en Nombres Entiers :

$$[P_1] : \quad \min \sum_{o \in O} \sum_{\omega \in \Omega_o} \eta^o_\omega \cdot y^o_\omega \tag{1}$$

$$\forall o \in O \ \sum_{\omega \in \Omega_o} y^o_\omega = 1 \tag{2}$$

$$\forall j \in J \ \sum_{t \in D_j} x_{jt} = p_j \tag{3}$$

$$\forall t \in H, \forall c \in C \ \sum_{\substack{j \in J / \\ c_j = c}} x_{jt} = \sum_{\substack{o \in O / \\ c \in C_o}} z_{oct} \tag{4}$$

$$\forall t \in H, \forall o \in O \ \sum_{c \in C_o} z_{oct} \leq \sum_{\omega \in \Omega_o} \sigma^t_\omega \cdot y^o_\omega \tag{5}$$

$$\forall o \in O, \forall \omega \in \Omega_o \ \ y^o_\omega \in \{0,1\} \tag{6}$$

$$\forall j \in J, \forall t \in H \ x_{jt} \in \{0,1\} \tag{7}$$

$$\forall o \in O, \forall c \in C_o, \forall t \in H \ z_{oct} \in \{0,1\} \tag{8}$$

Dans ce modèle, la variable de décision $y^o_\omega = 1$ si et seulement si le roulement ω est affecté à l'opérateur o, $x_{jt} = 1$ si et seulement si une unité du job j est exécutée à l'instant t et $z_{oct} = 1$ si et seulement si l'opérateur o utilise la compétence c à l'instant t.

Le vecteur caractéristique σ_ω permet de modéliser la couverture temporelle du roulement ω sur l'horizon H. Si ω couvre t, $\sigma^t_\omega = 1$, sinon $\sigma^t_\omega = 0$.

Les contraintes (2) astreignent l'affectation d'exactement un roulement à chaque opérateur. L'exécution complète des jobs durant leur fenêtre d'exécution est assurée par les contraintes (3). Pour leur part, les contraintes (4) indiquent que, sur un instant t, on effectue exactement autant d'unités de jobs requérant la compétence c qu'il y a d'opérateurs utilisant c sur t. Enfin, les contraintes (5) forcent les opérateurs à utiliser au plus une compétence sur chaque instant où ils sont présents (selon le roulement qui leur est attribué) et exactement 0 lorsqu'ils sont absents.

Notons ici qu'il y a deux niveaux de décision dans le processus de création d'emploi du temps. Tout d'abord, il est nécessaire d'affecter un roulement à chaque opérateur. Ensuite, sur chaque unité de temps où, en accord avec le roulement qui lui a été attribué, l'opérateur est considéré comme présent, nous devons décider quelle compétence il utilise concrètement.

2.2 Formalisation par intervalles de temps et profils de compétences

Afin de réduire le nombre de variables du modèle indexé sur le temps, nous proposons une seconde formalisation basée sur une représentation par intervalles de temps et sur une agrégation des opérateurs en profils de compétences.

Nous réduisons tout d'abord les instants de temps considérés dans le modèle en ne nous intéressant plus qu'à des intervalles de temps choisis. Pour cela, nous extrayons de H les instants de temps *significatifs*, à savoir, la date de disponibilité r_j et la date échue d_j de chaque job $j \in J$ ainsi que les bornes des intervalles de présence de chaque roulement $\omega \in \Omega$. Les instants ainsi *extraits* sont ensuite triés par ordre croissant puis appariés par paire successive de manière à former k_{max} intervalles de temps I_k ($k \in K = \{1, 2, \ldots, k_{max}\}$); par définition $H = \cup_{k \in K} I_k$ et $\forall k \neq k' : I_k \cap I_{k'} = \emptyset$. Une couverture complète de H est ainsi déterminée. Notons que chaque instant t non *extrait* est dit *inactif* car aucun job ne peut débuter ni ne doit se terminer, et aucun opérateur ne peut commencer ni ne doit arrêter de travailler en t. Ainsi, sur chaque intervalle I_k, les instants sont équivalents entre eux car travailler sur l'un est équivalent à tra-

vailler sur un autre en terme de coût et de disponibilité des ressources (opérateurs et compétences). Nous assurons qu'il est toujours possible d'extraire, via un problème de flot de coût maximum, une solution optimale globale (indexée sur le temps) à partir de la solution obtenue par cette formalisation.

Nous nous intéressons ensuite à agréger les variables dites de *ressources opérateurs-compétences* (variables z_{oct}) en profils de compétences. Nous exploitons pour cela le fait que ces variables apparaissent dans les contraintes mais pas directement dans la fonction coût du problème. Il est donc inutile de différencier les opérateurs possédant exactement les mêmes compétences entre eux lorsqu'on s'intéresse à l'attribution des compétences aux instants de temps. L'optimalité des solutions est alors laissée au juge-arbitre de l'affectation des roulements aux opérateurs (variables y^o_ω) qui jouent à la fois le rôle de facteur économique dans la fonction coût et de régulateur dans l'attribution des compétences aux opérateurs dans les contraintes couplantes avec les variables d'affectation de compétences aux opérateurs (variables z_{oct}). Une nouvelle notation $\Theta \subseteq \mathcal{P}(C)$ est alors définie pour désigner l'ensemble des profils de compétences $\theta \in \Theta$. Par suite, un unique profil de compétence θ_o est attribué à chaque opérateur $o \in O$. Deux opérateurs distincts se voient attribuer le même profil de compétences θ si et seulement s'ils maîtrisent tous les deux uniquement et exactement toutes les compétences $c \in \theta$. En d'autres termes, un profil de compétences peut être vu comme une généralisation des opérateurs au sens de leurs compétences. Grâce à cette formalisation, nous agrégeons (par les variables $z_{\theta ck}$ gérant l'attribution des compétences) les opérateurs possédant exactement les mêmes compétences.

Une seconde formalisation (dérivant de $[P_1]$) en termes de PLNE, sous forme d'intervalles de temps et de profils de compétences peut alors être proposée :

$$
\begin{aligned}
[P]: \quad & \min \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega \cdot y^o_\omega \\
\forall o \in O \quad & \sum_{\omega\in\Omega_o} y^o_\omega = 1 \\
\forall j \in J \quad & \sum_{\substack{k\in K/\\ I_k\subseteq D_j}} x_{jk} = p_j \\
\forall k \in K, \forall c \in C \quad & \sum_{\substack{j\in J/\\ c_j=c}} x_{jk} = \sum_{\substack{\theta\in\Theta/\\ c\in\theta}} z_{\theta ck} \\
\forall k \in K, \forall \theta \in \Theta \quad & \sum_{c\in\theta} z_{\theta ck} \leq \sum_{\substack{o\in O/\\ \theta_o=\theta}} \sum_{\substack{\omega\in\Omega_o/\\ I_k\subseteq\omega}} l_k \cdot y^o_\omega \qquad (9) \\
\forall o \in O, \forall \omega \in \Omega_o \quad & y^o_\omega \in \{0,1\} \\
\forall j \in J, \forall k \in K \quad & x_{jk} \in [0, \min(p_j, l_k)] \\
\forall \theta \in \Theta, \forall c \in \theta, \forall k \in K \quad & z_{\theta ck} \in [0, l_k.|\{o \in O/_{\theta_o=\theta}\}|]
\end{aligned}
$$

où l_k désigne la longueur de l'intervalle I_k, x_{jk} est le nombre d'unités du job j exécutées sur l'intervalle I_k et $z_{\theta ck}$ représente le nombre d'unités de compétence c utilisées par le profil de compétences θ durant I_k. Les autres notations sont identiques à celles utilisées pour la formalisation $[P_1]$ indexée sur le temps.

Les techniques de résolution proposées dans le reste de notre article sont basées sur cette seconde formalisation car elle s'avère, dans la pratique, nettement plus efficace en terme de temps de calcul.

Dans la suite de l'article, nous nous intéressons à la recherche d'une solution optimale de $[P]$. Nous présentons alors deux méthodes exactes, l'une d'elles repose sur une décomposition de Benders (Section 3) et l'autre (Section 4) a pour fondement une technique spécifique de décomposition et génération de coupes.

3. DÉCOMPOSITION DE BENDERS

En raison de sa structure en deux niveaux de décision, il semble assez naturel d'expérimenter une décomposition de Benders pour résoudre $[P]$. Afin de s'assurer, à chaque étape, une solution réalisable bornée, nous modifions légèrement le modèle $[P]$. Nous associons en effet un jeu de variables d'écart $s_{k\theta}$ positives aux contraintes (9). Une nouvelle modélisation est alors obtenue :

$$
\begin{aligned}
[P']: \quad & \min \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega \cdot y^o_\omega + M\left(\sum_{k\in K}\sum_{\theta\in\Theta} s_{k\theta}\right) \\
\forall o \in O \quad & \sum_{\omega\in\Omega_o} y^o_\omega = 1 \qquad (10) \\
\forall j \in J \quad & \sum_{\substack{k\in K/\\ I_k\subseteq D_j}} x_{jk} = p_j \\
\forall k \in K, \forall c \in C \quad & \sum_{\substack{j\in J/\\ c_j=c}} x_{jk} = \sum_{\substack{\theta\in\Theta/\\ c\in\theta}} z_{\theta ck} \\
\forall k \in K, \forall \theta \in \Theta \quad & \sum_{c\in\theta} z_{\theta ck} - s_{k\theta} \leq \sum_{\substack{o\in O/\\ \theta_o=\theta}} \sum_{\substack{\omega\in\Omega_o/\\ I_k\subseteq\omega}} l_k \cdot y^o_\omega \\
\forall o \in O, \forall \omega \in \Omega_o \quad & y^o_\omega \in \{0,1\} \qquad (11) \\
\forall j \in J, \forall k \in K \quad & x_{jk} \in [0, \min(p_j, l_k)] \\
\forall \theta \in \Theta, \forall c \in \theta, \forall k \in K \quad & z_{\theta ck} \in [0, l_k.|\{o \in O/_{\theta_o=\theta}\}|] \\
\forall k \in K, \forall \theta \in \Theta \quad & s_{k\theta} \geq 0
\end{aligned}
$$

Clairement, pour un choix approprié de M, $[P]$ et $[P']$ fournissent la même solution optimale.

Supposons une distribution $\bar{y}$ des roulements aux opérateurs fixée. Nous sommes alors en mesure d'introduire le sous-problème continu $[SP'(\bar{y})]$ résultant de $[P'(\bar{y})]$, défini en imposant la distribution $\bar{y}$ à $[P']$.

$$[SP'(\bar{y})] : M \cdot \min \sum_{k\in K}\sum_{\theta\in\Theta} s_{k\theta}$$

$$\forall j \in J \quad \sum_{k\in K/I_k\subseteq D_j} x_{jk} = p_j$$

$$\forall k \in K, \forall c \in C \quad \sum_{\substack{j\in J/\\ c_j=c}} x_{jk} = \sum_{\theta\in\Theta/c\in\theta} z_{\theta ck}$$

$$\forall k \in K, \forall \theta \in \Theta \sum_{c\in\theta} z_{\theta ck} - s_{k\theta} \leq \sum_{\substack{o\in O/\\ \theta_o=\theta}} \sum_{\substack{\omega\in\Omega_o/\\ I_k\subseteq\omega}} l_k.\bar{y}^o_\omega$$

$$\forall j \in J, \forall k \in K \quad x_{jk} \leq \min(p_j, l_k)$$

$$\forall \theta \in \Theta, \forall c \in C, \forall k \in K \quad z_{\theta ck} \leq l_k.|\{o \in O/_{\theta_o=\theta}\}|$$

$$\forall j \in J, \forall k \in K \quad x_{jk} \geq 0$$

$$\forall \theta \in \Theta, \forall c \in C, \forall k \in K \quad z_{\theta ck} \geq 0$$

$$\forall k \in K, \forall \theta \in \Theta \quad s_{k\theta} \geq 0$$

Il est trivial de vérifier qu'il existe nécessairement une solution optimale bornée (grâce aux $s_{k\theta}$). De plus, si la valeur de la solution optimale de $[SP'(\bar{y})]$ est zéro, alors $\bar{y}$ est une solution réalisable de $[P']$.

Le dual $[DSP'(\bar{y})]$ de $[SP'(\bar{y})]$ s'écrit :

$$[DSP'(\bar{y})] : \max g_{\bar{y}}(a,b,c,d,e) = \sum_{j\in J} a_j p_j + \sum_{k\in K}\sum_{\theta\in\Theta} c_{k\theta} \sum_{\substack{o\in O/\\ \theta_o=\theta}} \sum_{\substack{\omega\in\Omega_o/\\ I_k\subseteq\omega}} l_k \cdot \bar{y}^o_\omega + \sum_{j\in J}\sum_{k\in K} \min(p_j, l_k)\cdot d_{jk} + \sum_{\theta\in\Theta}\sum_{\substack{c\in C/\\ c\in\theta}}\sum_{k\in K} l_k \cdot |\{o \in O/_{\theta_o=\theta}\}| \cdot e_{\theta ck}$$

$$\forall j \in J, \forall k \in K \ \alpha^k_j a_j + b_{kc_j} + d_j \leq 0 \quad (12)$$

$$\forall \theta \in \Theta, \forall c \in \theta, \forall k \in K \ -b_{kc} + c_{k\theta} + e_{\theta ck} \leq 0 \quad (13)$$

$$\forall k \in K, \forall \theta \in \Theta \quad -c_{k\theta} \leq 1 \quad (14)$$

$$\forall j \in J \quad a_j \lesseqgtr 0 \quad (15)$$

$$\forall k \in K, \forall c \in C \quad b_{kc} \leq 0 \quad (16)$$

$$\forall k \in K, \forall \theta \in \Theta \quad c_{k\theta} \leq 0 \quad (17)$$

$$\forall j \in J, \forall k \in K \quad d_{jk} \leq 0 \quad (18)$$

$$\forall \theta \in \Theta, \forall c \in C, \forall k \in K \quad e_{\theta ck} \leq 0 \quad (19)$$

où la variable $\alpha^k_j = 1$ si $I_k \subseteq D_j$ et 0 sinon.

Soit $\zeta^*(\bar{y})$ la solution optimale de $[SP'(\bar{y})]$, $[P']$ peut ainsi être réécrit :

$$[P'] : \min \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega \cdot y^o_\omega + M \cdot \zeta^*(\bar{y})$$
$$\text{s.c.} \quad (10) \text{ et } (11)$$

Soit D le polyèdre défini par $D = \{(a,b,c,d,e) | (12) - (19)\}$. Par dualité, on obtient :

$$[P'] : \min \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega \cdot y^o_\omega + M \cdot \max_{(a,b,c,d,e)\in D} g_y(a,b,c,d,e)$$
$$\text{s.c. } (10) \text{ et } (11)$$

Dans la mesure où il existe toujours une solution réalisable bornée à $[SP'(y)]$, $[DSP'(y)]$ est également borné. Une solution optimale à $[DSP'(y)]$ correspond alors à un point extrêmal de D. De plus, D ne dépend pas de y. Par conséquent, si nous considérons l'ensemble Q de ses q points extrêmaux défini par $Q = \left\{(\tilde{a}_1,\tilde{b}_1,\tilde{c}_1,\tilde{d}_1,\tilde{e}_1),\ldots,(\tilde{a}_q,\tilde{b}_q,\tilde{c}_q,\tilde{d}_q,\tilde{e}_q)\right\}$, $[P']$ peut être réécrit :

$$[P'] : \min \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega \cdot y^o_\omega + M \cdot \max_{i\in[1..q]} g_y(\tilde{a}_i,\tilde{b}_i,\tilde{c}_i,\tilde{d}_i,\tilde{e}_i)$$
$$\text{s.c. } (10) \text{ et } (11)$$

$[P']$ est alors équivalent au problème maître suivant :

$$[MP'] : \min \ \psi$$
$$\text{s.c.} \quad (10) \text{ et } (11)$$
$$\sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega \cdot y^o_\omega + M \cdot g_y(\tilde{a}_i,\tilde{b}_i,\tilde{c}_i,\tilde{d}_i,\tilde{e}_i) \leq \psi \forall i \in [1..q]$$

Si on suppose qu'il existe au moins une solution réalisable au problème global, alors il existe une solution y_f telle que $optimum\,([DSP'(y_f)]) = 0$. Dans ce cas, $\psi = \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega (y_f)^o_\omega$ et nous pouvons poser :

$$[MP'] : \min \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega \cdot y^o_\omega$$
$$\text{s.c.} \quad (10) \text{ et } (11)$$
$$g_y(\tilde{a}_i,\tilde{b}_i,\tilde{c}_i,\tilde{d}_i,\tilde{e}_i) \leq 0 \quad \forall i \in [1..q]$$

Le problème maître peut être résolu par un processus de génération de coupes (Benders 1962). Soit $[RMP'^r]$ le problème maître relaxé avec seulement un petit sous-ensemble Q_r de contraintes :

$$[RMP'^r] \quad \min \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta^o_\omega \cdot y^o_\omega$$
$$\text{s.c.} \quad (10) \text{ et } (11)$$
$$g_y(\tilde{a}_i,\tilde{b}_i,\tilde{c}_i,\tilde{d}_i,\tilde{e}_i) \leq 0 \quad \forall i \in Q_r$$

En utilisant un solveur de PLNE, nous obtenons une solution optimale $\bar{y}_r$ à $[RMP'^r]$.

Afin de vérifier si $\bar{y}_r$ est une solution réalisable de $[MP']$, nous résolvons $[DSP'(\bar{y}_r)]$ et obtenons une solution $(\bar{a}_r,\bar{b}_r,\bar{c}_r,\bar{d}_r,\bar{e}_r)$ de coût $\bar{\nu}^*_r$.

Si $\bar{\nu}^*_r \leq 0$, alors : $\forall i \in Q \quad g_{\bar{y}_r}(\tilde{a}_i,\tilde{b}_i,\tilde{c}_i,\tilde{d}_i,\tilde{e}_i) \leq 0$. Dans ce cas, $\bar{y}_r$ est solution optimale de $[MP']$ et, par extension de $[P']$ et par conséquent de $[P]$.

Si $\bar{\nu}^*_r > 0$, la contrainte de $[MP']$ liée à $(\bar{a}_r,\bar{b}_r,\bar{c}_r,\bar{d}_r,\bar{e}_r)$ de D est violée par la solution $\bar{y}_r$. Nous devons alors ajouter la nouvelle contrainte $g_y(\bar{a}_r,\bar{b}_r,\bar{c}_r,\bar{d}_r,\bar{e}_r) \leq 0$ à $[RMP'^r]$ puis résoudre le problème résultant. Dans la mesure où il n'existe qu'un nombre fini de points extrêmes, le processus global s'arrête après un nombre fini de pas.

4. PROCESSUS SPÉCIFIQUE DE GÉNÉRATION DE COUPES

4.1 Processus global

Nous présentons ici une approche exacte alternative pour résoudre $[P]$. À l'instar de la décomposition de Benders, celle-ci exploite la décomposition de $[P]$ en deux sous-problèmes. Un problème maître $[MP]$ affecte tout d'abord un roulement à chaque opérateur. Utilisant ces informations comme données, le second sous-problème $[SP]$ vérifie ensuite la réalisabilité, aussi bien au niveau de l'exécution complète des jobs que de la disponibilité suffisante de compétence à chaque instant, de la solution précédemment générée. Si $[SP]$ ne possède aucune solution réalisable, une coupe invalidant l'affectation des roulements en cours est ajoutée à $[MP]$. Ainsi, le processus itère jusqu'à trouver une solution réalisable. Dans la mesure où, à chaque itération, $[MP]$ fournit l'affectation réalisable de roulements aux opérateurs de moindre coût, la première affectation qui conduira à une solution réalisable dans $[SP]$ sera optimale. $[MP]$ a pour but d'affecter, au moindre coût, un roulement à chaque opérateur, les contraintes liant les profils de compétences aux opérateurs ainsi que celles reliant les jobs aux opérateurs étant relaxées. $[MP]$ peut par conséquent s'écrire :

$$[MP]: \quad \min \sum_{o\in O}\sum_{\omega\in\Omega_o} \eta_\omega^o \cdot y_\omega^o$$

$$\forall o \in O \qquad \sum_{\omega\in\Omega_o} y_\omega^o \qquad = 1$$

$$Cut$$

$$\forall o \in O, \forall \omega \in \Omega_o \quad y_\omega^o \in \{0,1\}$$

où Cut désigne l'ensemble des coupes de réalisabilité ajoutées itérativement au modèle. Ces coupes permettent d'invalider des solutions de $[MP]$ ne conduisant pas à une solution réalisable au vu de l'ensemble des contraintes.

Considérons une solution $\bar{y}$ de $[MP]$ fixée. Nous devons par conséquent vérifier si $\bar{y}$ est réalisable au vu des autres contraintes de $[P]$, à savoir, les contraintes d'exécution complète des jobs et de disponibilité suffisante de compétence à chaque instant. Afin de vérifier ceci, le sous-problème $[SP(\bar{y})]$ est introduit :

$$[SP(\bar{y})]: \quad \max f_{\bar{y}} = \sum_{j\in J}\sum_{\substack{k\in K/\\ I_k\subseteq D_j}} x_{jk} \tag{20}$$

$$\forall j \in J \qquad \sum_{\substack{k\in K/\\ I_k\subseteq D_j}} x_{jk} \qquad \leq p_j$$

$$\forall k \in K, \forall c \in C \qquad \sum_{\substack{j\in J/\\ c_j=c}} x_{jk} \qquad = \sum_{\substack{\theta\in\Theta/\\ c\in\theta}} z_{\theta ck}$$

$$\forall k \in K, \forall \theta \in \Theta \qquad \sum_{c\in\theta} z_{\theta ck} \qquad \leq \sum_{\substack{o\in O/\\ \theta_o=\theta}}\sum_{\substack{\omega\in\Omega_o/\\ I_k\subseteq\omega}} l_k \cdot \bar{y}_\omega^o$$

$$\forall j \in J, \forall k \in K \qquad x_{jk} \in [0, \min(p_j, l_k)]$$

$$\forall \theta \in \Theta, \forall c \in \theta, \forall k \in K \; z_{\theta ck} \in [0, l_k.|\{o \in O/_{\theta_o=\theta}\}|]$$

Ce modèle est l'expression d'un problème de *flot maximum* sur un graphe orienté décomposable en niveaux $G_{\bar{y}} = (X, U)$ - cf. Figure 1 - où :

- Sommets : $X = \{s\} \cup J \cup KC \cup \Theta K \cup \{t\}$
  - s : source
  - J : ensemble des jobs J
  - KC : produit cartésien de K et C
  - ΘK : produit cartésien de Θ et K
  - t : puits
- Arcs : $\{\alpha, \beta\} \in U$ - (capacité $\gamma_{\alpha\beta}$) -
  - $\forall j \in J : (s, j) \in U$
    avec : $\gamma_{sj} = p_j$
  - $\forall j \in J, \forall a \in KC :$
    $(j, a) \in U \Leftrightarrow c_j = c_a \wedge I_{k_a} \subseteq D_j$
    avec : $\gamma_{ja} = \min(p_k, l_{k_a})$
  - $\forall a \in KC, \forall b \in \Theta K :$
    $(a, b) \in U \Leftrightarrow I_{k_a} = I_{k_b} \wedge c_a \in \theta_b$
    avec $\gamma_{ab} = l_{k_a} \cdot |\{o \in O/_{\theta_o=\theta_a}\}|$
  - $\forall b \in \Theta K : (b, t) \in U$
    avec $\gamma_{bt} = \sum_{o\in O/\theta_o=b_\theta} \sum_{\omega\in\Omega_o} \sigma_\omega^{k_b} \cdot \bar{y}_\omega^o \cdot l_{k_b}$
    où : $\sigma_\omega^k = 1$ si ω couvre I_k et 0 sinon.

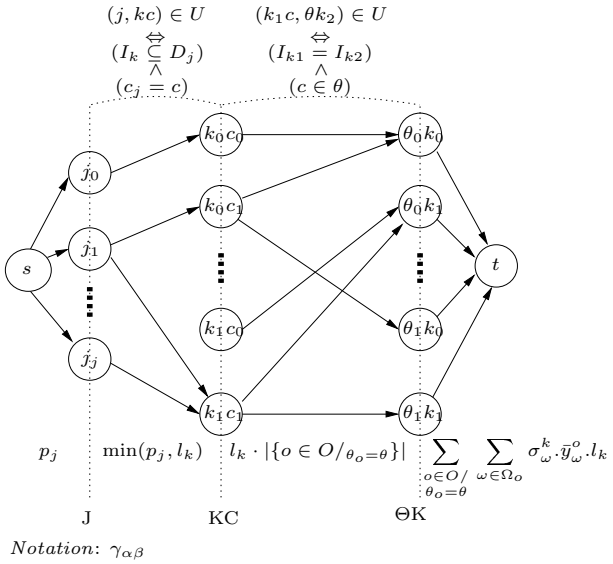

Figure 1: Structure de $G_{\bar{y}}$

La fonction objectif (20) de $[SP(\bar{y})]$ symbolise le nombre total d'unités de jobs qu'il est possible d'ordonnancer via $\bar{y}$. Rappelons ici que le but du problème global est d'ordonnancer l'ensemble des jobs entièrement. Par conséquent, seule une affectation $\bar{y}$ permettant d'aboutir à une valeur optimale de flot $f_{\bar{y}} = \sum_{j\in J} p_j$ est réalisable pour $[P]$.

Donc, si $f_{\bar{y}} = \sum_{j \in J} p_j$ pour une itération donnée, nous avons établi que $\bar{y}$ est nécessairement optimale pour $[P]$. Le processus peut alors être arrêté.

En revanche, si $f_{\bar{y}} < \sum_{j \in J} p_j$, il est impossible d'établir un plan d'ordonnancement global. Il convient alors d'éliminer $\bar{y}$ de l'ensemble des solutions de $[MP]$. Pour cela, nous exploitons la *coupe minimale du flot maximal de* $G_{\bar{y}}$.

Par la suite, nous utiliserons les notations suivantes :

- $\forall u \in U \quad (\phi_u, \gamma_u)$: (flot, capacité) de l'arc u
- $X = X^+ \cup X^-$ avec
  - $X^+ = \{s\} \cup J^+ \cup KC^+ \cup \Theta K^+$
  - $X^- = \overline{X^+} = J^- \cup KC^- \cup \Theta K^- \cup \{t\}$
  - $\forall u = (\alpha, \beta) \in U/_{\alpha \in X^+ \wedge \beta \in X^-} \quad \phi_u = \gamma_u$
  - $\forall u = (\alpha, \beta) \in U/_{\alpha \in X^- \wedge \beta \in X^+} \quad \phi_u = 0$

D'après (Ford & Fulkerson 1962), nous pouvons redéfinir quatre résultats importants:

1. L'ensemble des sommets X d'un graphe $G = (X, U)$ est toujours décomposable en X^+ et X^-, avec $X^+ \cap X^- = \emptyset$.
2. Il existe une *coupe minimale* Ξ, de valeur ξ :
   $$\Xi \subset U / u = (\alpha, \beta) \in \Xi \Leftrightarrow \alpha \in X^+ \wedge \beta \in X^-$$
3. La *coupe minimale* Ξ est, par définition, composée d'arcs *saturés*. Soit :
   $$\xi = \sum_{u \in \Xi} \phi_u = \sum_{u \in \Xi} \gamma_u$$
4. Theorème *Flot maximum - Coupe minimum* :
   ξ = valeur du flot maximal

La *coupe minimale* de $[SP(\bar{y})]$ peut être définie plus précisément (cf. figure 2) :

$$\Xi = \{(s, \beta) \in U/_{\beta \in J^-}\} \cup \{(\alpha, \beta) \in U/_{\alpha \in J^+ \wedge \beta \in KC^-}\} \cup \{(\alpha, \beta) \in U/_{\alpha \in KC^+ \wedge \beta \in \Theta K^-}\} \cup \{(\alpha, t) \in U/_{\alpha \in \Theta K^+}\}$$

avec :
$$\xi = \sum_{j \in J^-} \gamma_{sj} + \sum_{j \in J^+} \sum_{a \in KC^-} \gamma_{ja} + \sum_{a \in KC^+} \sum_{b \in \Theta K^-} \gamma_{ab} + \sum_{b \in \Theta K^+} \gamma_{bt}$$

Il a été établi que, si à une itération donnée i du processus, $f_{\bar{y}_i} < \sum_{j \in J} p_j$, nous devons invalider $\bar{y}_i$. Dans ce cas, la coupe devant être ajoutée à l'ensemble *Cut* est $f_{y_i} \geq \sum_{j \in J} p_j$. Soit, en appliquant le théorème *Flot maximal - Coupe minimale* :

$$\sum_{j \in J_i^-} \gamma_{sj} + \sum_{j \in J_i^+} \sum_{a \in KC_i^-} \gamma_{ja} + \sum_{a \in KC_i^+} \sum_{b \in \Theta K_i^-} \gamma_{ab} + \sum_{b \in \Theta K_i^+} \gamma_{bt} \geq \sum_{j \in J} p_j$$

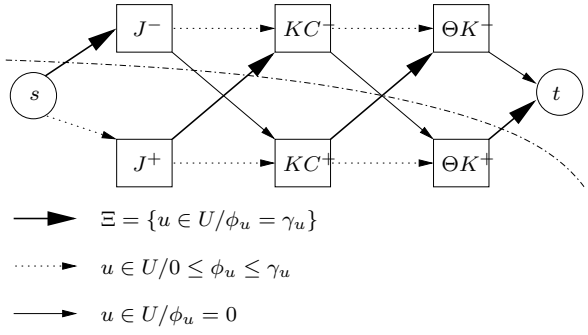

Figure 2: Coupe Minimale sur $G_{\bar{y}}$

En utilisant les capacités réelles γ des arcs de G, la coupe devient :

$$\sum_{b \in \Theta K_i^+} \sum_{\substack{o \in O/ \\ \theta_o = \theta_b}} \sum_{\omega \in \Omega_o} \sigma_\omega^{k_b} \cdot y_\omega^o \cdot l_{k_b} \geq \sum_{j \in J} p_j - \sum_{j \in J_i^-} p_j - \nu_i$$

avec :
$$\nu_i = \sum_{j \in J_i^+} \sum_{a \in KC_i^-} \min(p_j, l_{k_a}) + \sum_{a \in KC_i^+} \sum_{b \in \Theta K_i^-} l_{k_b} \cdot |\{o \in O/_{\theta_o = \theta_b}\}|$$

Dans la mesure où $J = J_i^+ \cup J_i^-$ avec $J_i^+ \cap J_i^- = \emptyset$, la coupe peut également s'écrire :

$$\sum_{b \in \Theta K_i^+} \sum_{\substack{o \in O/ \\ \theta_o = \theta_b}} \sum_{\omega \in \Omega_o} \sigma_\omega^{k_b} \cdot y_\omega^o \cdot l_{k_b} \geq \sum_{j \in J_i^+} p_j - \nu_i \quad (21)$$

Cette inégalité est par conséquent la coupe invalidant $\bar{y}_i$ qu'il convient d'ajouter à l'ensemble *Cut* de $[MP]$ pour les itérations suivantes du processus spécifique de génération de coupes.

De ce fait, si nous considérons un petit sous-ensemble Q de coupes, la formalisation de $[MP]$ devient :

$$[MP] : \quad \min \quad \sum_{o \in O} \sum_{\omega \in \Omega_o} \eta_\omega^o \cdot y_\omega^o$$

$$\forall o \in O \quad \sum_{\omega \in \Omega_o} y_\omega^o = 1$$

$$\forall i \in Q \quad \sum_{b \in \Theta K_i^+} \sum_{\substack{o \in O/ \\ \theta_o = \theta_b}} \sum_{\omega \in \Omega_o} \sigma_\omega^{k_b} . y_\omega^o . l_{k_b} \geq \sum_{j \in J_i^+} p_j - \nu_i$$

$$\forall o \in O, \forall \omega \in \Omega_o \quad y_\omega^o \in \{0, 1\}$$

Cette formalisation est une approximation d'un *Sac à Dos Multi-Choix Multi-dimensionnel* (NP-difficile). L'approximation tient au fait que les coupes (21) sont des contraintes de capacité de type $\geq$ alors que, classiquement, ces contraintes sont de type $\leq$.

Dans nos expérimentations, nous avons utilisé un solveur de programmation linéaire en nombres entiers pour résoudre $[MP]$ et un algorithme

d'*échelonnement des capacités* (Ahuja, Magnanti & Orlin 1993) pour résoudre $[SP(\bar{y})]$. Soulignons de nouveau que $[MP]$ est NP-difficile ($\approx$ *Sac à Dos Multi-Choix Multi-dimensionnel*). Par conséquent, l'utilisation d'un solveur de PLNE est d'une grande utilité quant à la recherche d'une solution optimale à $[MP]$. Toutefois, il est intéressant de noter ici qu'en raison de sa structure particulière et du grand nombre de travaux menés notamment par Akbar (Akbar, Manning, Shoja & Khan 2001) pour résoudre cette classe de problèmes de sacs à dos, on peut aisément chercher une solution heuristique de qualité à $[MP]$ et donc, approximer de manière intéressante, sans solveur, l'optimum de $[P]$. Dans cet exposé, nous cherchons une solution optimale du problème global, cette approximation heuristique pourrait donc être un prolongement intéressant à ce travail.

Soulignons également ici qu'une alternative à cette génération de coupes a été testée. Dans cette autre approche, l'optimum de $[MP]$ n'est pas nécessairement visé. La première affectation de roulements $\bar{y}$ réalisable pour $[MP]$ est fournie en entrée au sous-problème $[SP(\bar{y})]$. De cette manière, si $[SP(\bar{y})]$ est réalisable, nous ajoutons à $[MP]$ une *coupe de borne* invalidant toutes les affectations de coût strictement inférieur à celui de $\bar{y}$. En revanche, si $[SP(\bar{y})]$ n'est pas réalisable, la *coupe minimale* classique (décrite dans cet article) est ajoutée à Cut. Le processus itère jusqu'à ce qu'aucune solution réalisable de $[MP]$ ne soit trouvée. Cette technique permet d'hybrider des techniques de programmation linéaire et de programmation par contraintes pour résoudre $[MP]$. De telles tentatives ont déjà été expérimentées avec succès par Hooker (Hooker 2005, Hooker 2007) dans un problème de planification et d'ordonnancement de production.

Dans la suite de l'exposé, nous décrivons les coupes initiales ajoutées à Cut en pré-process.

4.2 Coupes initiales

La qualité des coupes générées lors du processus de génération de coupes est prépondérante. En effet, plus les coupes permettent d'invalider un grand nombre d'affectations de roulements aux opérateurs, meilleure est la vitesse de convergence du processus. Nous exposons ici les coupes ajoutées à l'ensemble Cut lors de l'initialisation du problème maître $[MP]$.

Le raisonnement utilisé pour définir ces coupes initiales est lié au concept d'*énergie*. Ce concept a été très largement utilisé avec succès, notamment par (Lopez, Erschler & Esquirol 1992, Baptiste, Pape & Nuijten 1999), dans des problèmes d'ordonnancement. Sous cette idée, les compétences sont considérées comme des ressources consommables, les opérateurs comme des fournisseurs de ressources et les jobs comme des activités consommatrices. Ce raisonnement prend tout son intérêt sur des périodes de temps où il est possible d'établir, pour un sous-ensemble de jobs, une *consommation obligatoire* en compétences strictement positive. Sur de telles périodes, la disponibilité en ressources concernées doit être suffisamment élevée pour permettre la consommation.

Considérons une période de temps $\delta = [\delta_b, \delta_e] \in \Delta$ avec $(\delta_b, \delta_e) \in H^2$, un sous-ensemble de compétences $\mathcal{C} \subseteq C$ et un sous-ensemble de jobs $\mathcal{J} \subseteq J$.

Dans la suite de cet exposé, nous utiliserons la notation $(a)^+ = \max(a, 0) \quad \forall a \in \Re$.

Consommation obligatoire d'un job j La consommation obligatoire de j sur δ correspond à la différence entre p_j et le nombre d'unités de j qu'il est possible d'exécuter en dehors de δ :

$$u_{j\delta} = \left(p_j - (\delta_b - r_j)^+ - (d_j - \delta_e)^+\right)^+$$

Disponibilité minimale en compétence c **requise** La disponibilité minimale en compétence c requise sur δ correspond à l'ensemble des *consommations obligatoires*, sur δ, des jobs requérant c :

$$U_{c\delta} = \sum_{\substack{j \in J / \\ c_j = c}} u_{j\delta}$$

Capacité des ressources disponibles Sur δ, chaque opérateur peut travailler autant d'instants qu'il est *présent* selon le roulement qui lui est affecté. Toutefois, il ne peut utiliser qu'une compétence à la fois. Par conséquent, dès lors qu'un opérateur o est *présent* et qu'il maîtrise une compétence de $\mathcal{C}$, une et une seule unité de compétence de o est disponible par instant.

$$\sum_{t \in \delta} \sum_{\substack{o \in O / \\ \{C_o \cap \mathcal{C} \neq \emptyset\}}} \sum_{\omega \in \Omega_o} \sigma_\omega^t y_\omega^o$$

Contrainte *énergétique* - Coupe initiale Une solution réalisable de $[P]$ doit par conséquent obligatoirement vérifier la coupe initiale suivante :

$$\sum_{t \in \delta} \sum_{\substack{o \in O / \\ \{C_o \cap \mathcal{C} \neq \emptyset\}}} \sum_{\omega \in \Omega_o} \sigma_\omega^t y_\omega^o \geq \sum_{c \in \mathcal{C}} \sum_{t \in \delta} U_{c\delta}$$

La figure 3 schématise un exemple où une coupe *initiale* peut être générée.
Il est important de souligner ici que l'utilisation de telles coupes est difficile à paramétrer. En effet, il existe un très grand nombre de périodes de temps δ (non nécessairement connexes) - et donc de coupes initiales possibles. Il convient alors de sélectionner les coupes initiales générées afin de ne pas confronter le processus de recherche à trop d'informations. Pour présenter la technique de sélection retenue dans nos tests, nous définissons tout d'abord $\tau_\delta^c = \frac{U_{c\delta}}{\delta_e - \delta_b}$ comme la

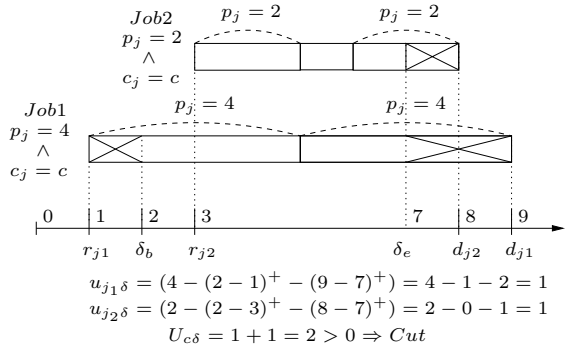

Figure 3: Coupe initiale pour une période de temps et une compétence

consommation proportionnelle en c obligatoire sur δ. Cet indicateur significatif nous permet de sélectionner des périodes de temps vraisemblablement plus contraignantes. En effet, si on considère deux périodes $\delta \in \Delta$ et $\delta' \in \Delta$ telles que $\tau^c_\delta > \tau^c_{\delta'}$, il semble logique de supposer que la coupe générée selon c sur δ soit plus significative que cela générée sur δ'.

L'idée sous-jacente de la sélection de périodes de temps retenue dans nos expérimentations repose sur l'intérêt tout particulier consacré à l'ensemble Υ des bornes υ des intervalles de présence de chaque roulement. En effet ces instants sont les seuls où le nombre d'opérateurs présents (et donc de compétences disponibles) peut varier. Par définition, ni $[\upsilon_i, \upsilon_{i+1}]$, ni $[\upsilon_{j-1}, \upsilon_j]$ (avec $\upsilon_i < \upsilon_j$) ne peuvent contenir d'instant marquant le début ou la fin d'un roulement. Notons RD l'ensemble (trié par ordre croissant) des dates de démarrage au plus tôt et des dates échues des n jobs. Dans nos expérimentations, nous considérons la période de temps $\delta' = [\delta'_b, \delta'_e] \in \Delta$ où $\delta'_b \in [\upsilon_i, \upsilon_{i+1}] \cap RD$ et $\delta'_e \in [\upsilon_{j-1}, \upsilon_j] \cap RD$ la plus *restrictive* pour l'ensemble des périodes générées pour tout couple $(i, j) \in |\Upsilon|^2$ avec $i < j$.

Dans nos tests, l'ensemble Θ des profils de compétences des opérateurs est l'ensemble de référence utilisé pour les coupes initiales énergétiques. Les singletons (une seule compétence) sont également employés.

Afin de s'affranchir d'un nombre trop important de coupes initiales, une technique complémentaire à l'approche énergétique consiste à intégrer au modèle l'ensemble des coupes générées puis, à résoudre sa relaxation continue. Il est alors intuitif de supposer que les coupes les plus significatives ont de faibles variables d'écart à l'optimum. Dans nos expérimentations, nous ne sélectionnons ainsi que 15% de l'ensemble des coupes énergétiques générées.

Expérimentalement, les coupes initiales s'avèrent d'une grande utilité car elles fournissent, à un moindre coût de temps de calcul, des informations autorisant l'invalidation d'un grand nombre d'affectations irréalisables. De plus, de par leur nature, elles mettent rapidement en exergue de fortes obligations d'affectation sur des périodes précises de l'horizon. Ce fonctionnement diffère légèrement des coupes générées via les coupes minimales de flot lors du processus global car ces dernières ont une vision plus générale du problème. Par conséquent, l'information retirée par l'usage des coupes *initiales* peut être longue à retrouver par le processus global. Ainsi, ces deux types de coupes sont complémentaires et leur utilisation commune est réellement bénéfique pour cette approche.

Les performances des différentes techniques de résolution présentées dans cet article sont discutées, à travers leurs résultats expérimentaux, dans la section suivante.

5. EXPÉRIMENTATIONS NUMÉRIQUES

Les méthodes décrites ci-dessus ont été implémentées en Java et testées sur un PC (Intel Pentium D 930, 3 GHz, 2 GB RAM) dont le système d'exploitation est MS Windows XP. Le solveur de programmation linéaire utilisé est ILOG CPLEX 9.1. D'après une étude préliminaire, il semble que les meilleurs paramètres du solveur sont ceux par défaut, à l'exception du guidage de la recherche de solution entière qui est, pour les 3 méthodes exactes, accenté sur la recherche de solution de meilleur coût plutôt que sur la preuve d'optimalité (paramètre *MipEmphasis* fixé à *MIPEmphasisFeasibility*). De plus, pour la résolution du problème maître $[MP]$ du processus spécifique de génération de coupes, l'*algorithme de barrière avec crossover* est utilisé pour la résolution de la première relaxation continue (paramètre *RootAlg* fixé à *Barrier*).

5.1 Jeu de données

Notre jeu de données est composé de données aléatoires. Les jobs sont générés en premier. Leur date de démarrage au plus tôt, leur durée et leur marge (taille du domaine de définition) sont respectivement distribuées selon une loi uniforme et deux lois binomiales. Une durée maximale ainsi qu'une marge maximale sont imposées. Les roulements sont ensuite générés. Afin de coller au mieux à la réalité des usines de production, les roulements de nos instances respectent les règles dites de *travail en trois-huit*. Le quart d'heure est l'unité de temps retenue pour la génération des roulements. Ces derniers sont générés sur une semaine (sauf week-end) puis dupli-

quer afin de couvrir l'horizon entier. Après cela, un ensemble de roulements éligibles et un ensemble de compétences maîtrisées sont aléatoirement attribués à chaque opérateur. Enfin, un coût est calculé pour chaque paire opérateur - roulement éligible. Ce coût dépend des horaires du roulement et des spécificités de l'opérateur (expérience, qualification, préférences). Le tableau 1 synthétise les paramètres utilisés pour générer nos 540 instances de test. Notons que 6 instances sont générées pour chaque jeu de paramètres.

Paramètre	min	max	pas
m	15	25	10
n	$4 \cdot m$	$6 \cdot m$	m
$marge_{max}$	30	90	30
$\|C\|$	1	5	1
p_{max}	30		
$\|H\|$	480		

Table 1: Paramètres de la génération des données

La limite de temps CPU imposée pour chaque technique est de 5 minutes.

5.2 Méthodes exactes

Le tableau 2 indique le pourcentage d'instances résolues par le solveur linéaire (MIP), la méthode exacte basée sur une décomposition de (Benders) et le processus spécifique de génération de coupes (Coupe).

Nous observons tout d'abord que la décomposition de Benders est peu efficace. Ses performances sont, quels que soient les paramètres, moins grandes que les deux autres techniques. De plus, lorsque elle parvient à trouver une solution, c'est au coût d'un très long temps de calcul. Sur les 98 instances où les trois méthodes réussissent à trouver l'optimum, $(Coupe)$ et (MIP) vont respectivement 32 et 2 fois plus vite que $(Benders)$. Nous constatons ensuite que $(Coupe)$ est le plus efficace sur presque tous les jeux de données. Seuls deux jeux de données ($m = 15$, $|C| \in \{3, 4, 5\}$ $marge_{max} \in \{60, 90\}$) donnent un léger avantage au solveur.

Le tableau 3 permet de juger des performances (en termes de temps de calcul) du solveur et du processus de coupes. Les comparaisons sont effectuées sur les instances où les deux méthodes atteignent l'optimum. Notons que le solveur résout 2 instances sur lesquelles le processus de coupes échoue et que, à l'inverse, le processus de coupes résout 85 instances où le solveur échoue. Les moyennes des nombres de coupes initiales (# coupes init.) et de flot (# coupes flot) nécessaires au processus de coupes ainsi que que le nombre d'instances comparées sont aussi indiquées.

			$marge_{max}$			
m	$\|C\|$	Méthode	30	60	90	Total
15	1-2	(MIP)	72.2%	91.7%	94.4%	86.1%
		(Coupe)	100%	100%	100%	100%
		(Benders)	41.7%	52.8%	75.0%	56.5%
	3-5	(MIP)	64.8%	83.3%	83.3%	77.2%
		(Coupe)	74.1%	72.2%	75.9%	74.1%
		(Benders)	0.00%	0.00%	1.90%	0.60%
(MIP) - $m = 15$			67.8%	86.7%	87.8%	80.7%
(Coupe) - $m = 15$			84.4%	83.3%	85.6%	84.4%
(Benders) - $m = 15$			16.7%	21.1%	31.1%	23.0%
25	1-2	(MIP)	25.0%	52.8%	66.7%	48.1%
		(Coupe)	58.3%	72.2%	97.2%	75.9%
		(Benders)	13.9%	50.0%	69.4%	44.4%
	3-5	(MIP)	18.5%	37.0%	38.9%	31.5%
		(Coupe)	35.2%	42.6%	63.0%	46.9%
		(Benders)	0.00%	0.00%	1.90%	0.60%
(MIP) - $m = 25$			21.1%	43.3%	50.0%	38.1%
(Coupe) - $m = 25$			44.4%	54.4%	76.7%	58.5%
(Benders) - $m = 25$			5.60%	20.0%	28.9%	18.1%
(MIP)			44.4%	65.0%	68.9%	59.4%
(Coupe)			64.4%	68.9%	81.1%	71.5%
(Benders)			11.1%	20.6%	30.0%	20.6%

Table 2: Pourcentage d'instances résolues par les méthodes exactes

Nous remarquons tout d'abord (cf. tableau 3) que le processus de coupes est plus rapide (de 1.2 à 40 fois plus rapide) sur chaque jeu de données. Ces résultats montrent donc l'intérêt tout particulier du processus de coupes pour ce problème. Nous observons aussi une différence notable entre les moyennes des nombres de coupes initiales et de flot nécessaires au processus de coupes. Un prolongement à cette étude serait donc de définir des règles de dominance et de sélection pour ne pas avoir à gérer autant de coupes initiales.

6. CONCLUSION

Nous avons défini deux méthodes exactes pour résoudre un problème couplant planification d'agents et ordonnancement de production. Notons notamment l'intérêt tout particulier d'une technique singulière de génération de coupes dont les résultats expérimentaux sont extrêmement compétitifs vis à vis de l'un des meilleurs solveurs actuels.

En perspective de ces travaux, de nombreuses pistes intéressantes demeurent à explorer. Les plus prometteuses semblent être l'hybridation de la programmation linéaire et de la programmation par contraintes ainsi que la recherche de coupes initiales de qualité pour résoudre le problème maître de la génération de coupes plus rapidement.

			$marge_{max}$			
m	$\|C\|$	Méthode	30	60	90	Total
15	1-2	(MIP)	71.6s	25.1s	12.2s	33.4s
		(Coupe)	11.1s	3.44s	0.90s	4.65s
		# coupes init.	65.9	60.4	55.2	60.0
		# coupes flot	3.88	4.27	2.71	3.59
		# instances	26	33	34	93
	3-5	(MIP)	61.2s	53.9s	37.6s	50.2s
		(Coupe)	9.10s	14.8s	29.2s	18.3s
		# coupes init.	220.0	209.5	197.5	208.3
		# coupes flot	17.6	21.3	31.9	24.0
		# instances	32	36	39	107
(MIP) - $m = 15$			65.9s	40.2s	25.8s	42.4s
(Coupe) - $m = 15$			9.99s	9.36s	16.0s	12.0s
# coupes init. - $m = 15$			150.9	138.2	131.2	139.3
# coupes flot - $m = 15$			11.5	13.1	18.3	14.5
# instances - $m = 15$			58	69	73	200
25	1-2	(MIP)	20.3s	34.5s	66.6s	46.9s
		(Coupe)	0.58s	2.76s	4.84s	3.34s
		# coupes init.	55.22	57.89	55.75	56.44
		# coupes flot	2.33	3.00	3.83	3.27
		# instances	9	19	24	52
	3-5	(MIP)	89.3s	69.4s	35.9s	59.1s
		(Coupe)	9.55s	17.6s	6.83s	11.3s
		# coupes init.	305.2	272.0	230.0	260.8
		# coupes flot	15.7	15.8	8.05	12.5
		# instances	10	18	21	49
(MIP) - $m = 25$			56.6s	51.5s	52.3s	52.8s
(Coupe) - $m = 25$			5.30s	9.97s	5.77s	7.22s
# coupes init. - $m = 25$			186.8	162.1	137.1	155.6
# coupes flot - $m = 25$			9.37	9.22	5.80	7.72
# instances - $m = 25$			19	37	45	101
(MIP)			63.6s	44.1s	35.9s	45.9s
(Coupe)			8.83s	9.58s	12.1s	10.3s
# coupes init.			159.8	146.6	133.5	144.8
# coupes flot			10.9	11.8	13.5	12.3
# instances			77	106	118	301

Table 3: Temps de résolution de (MIP) et ($Coupe$)

References

Ahuja, R. K., Magnanti, T. L. & Orlin, J. B. (1993). *Network Flows - Theory, Algorithms, and Applications*, Prentice Hall.

Akbar, M. M., Manning, E. G., Shoja, G. C. & Khan, S. (2001). Heuristic solutions for the multiple-choice multi-dimension knapsack problem, *ICCS '01: Proceedings of the International Conference on Computational Science-Part II*, Springer-Verlag, London, UK, pp. 659–668.

Artigues, C., Gendreau, M. & Rousseau, L.-M. (2006). A flexible model and a hybrid exact method for integrated employee timetabling and production scheduling, *CORS / Optimization Days 2006 Joint Conference* .

Baptiste, P., Pape, C. L. & Nuijten, W. (1999). Satisfiability tests and time-bound adjustments for cumulative scheduling problems, *Annals of Operations Research* **92**: 305–333.

Benders, J. (1962). Partioning procedures for solving mixed integer problem, *Numerische Mathematik* **4**: 238–252.

Dauzère-Pérès, S. & Lasserre, J. (1994). Integration of lotsizing and scheduling decisions in a job-shop, *European Journal of Operations Research* **75**: 413–426.

Detienne, B., Péridy, L., Pinson, E. & Rivreau, D. (2007). Cut generation for an employee timetabling problem, *European Journal of Operational Research* **To appear**.

Ernst, A., Jiang, H., Krishnamoorthy, M. & Sier, D. (2004). Staff scheduling and rostering : A review of applications, methods and models, *European Journal of Operational Research* **153**: 3 – 27.

Ford, L. & Fulkerson, D. (1962). *Flows in Networks*, Princeton University Press, Princeton.

Hooker (2005). A hybrid method for planning and scheduling, *Constraints* **10**: 385–401.

Hooker (2007). Planning and scheduling by logic-based benders decomposition, *Operations Research* **55**: 588–602.

Lasserre (1992). An integrated model for job-shop planning and scheduling, *Management Science* **38**: 1201–1211.

Lopez, P., Erschler, J. & Esquirol, P. (1992). Ordonnancement de tâches sous contraintes : une approche énergétique, *Automatique, Productique, Informatique Industrielle* **26**: 453–481.

Soumis, F., Pesant, G. & Rousseau, L.-M. (2005). *Gestion de Production et Ressources Humaines*, Presses Internationales Polytechnique, chapter 4, Gestion des horaires et affectation du personnel.