

COMPARAISON DE DIFFERENTS OPERATEURS DE DEPLACEMENT POUR UNE METHODE TABOU : APPLICATION A UN PROBLEME INDUSTRIEL HAUTEMENT CONTRAINT

O. ROUX et V. DHAEVERS

Louvain School of Management et FUCaM
Chaussée de Binche, 151
7000 Mons - Belgique
roux@fucam.ac.be, dhaevers@fucam.ac.be

D. DUVIVIER

LIL, Université du Littoral Côte d'Opale
50, Rue Ferdinand Buisson BP 719
62228 Calais Cedex, - France
david.duvivier@lil.univ-littoral.fr

RESUME : Dans le cadre de l'aide à la décision, cet article présente les résultats d'une méthode de recherche tabou appliquée à l'ordonnancement de la production en industrie semi-continue. L'objectif est de fournir au décideur la stratégie d'ordonnancement la plus adaptée à ses préférences. Pour ce faire, une plate-forme a été développée pour ordonnancer la production, par l'exploitation de différentes méthodes d'optimisation et de la simulation. Les différents ordonnancements sont itérativement améliorés au moyen d'une méthode hybride constituée d'une méta-heuristique et d'un modèle de simulation. Cet article décrit plus particulièrement les résultats de plusieurs opérateurs de déplacement intégrés au sein d'une méthode de recherche Tabou. Cette étude compare et classe, sur base d'un ensemble d'indicateurs de performance, différentes stratégies d'ordonnancement au moyen d'un tri lexicographique.

MOTS-CLES : Simulation des systèmes de production, méta-heuristiques, méthodes hybrides, performance, aide à la décision, recherche Tabou.

1 INTRODUCTION

Dans le cadre des processus de production de type discret ou semi-continu organisés en job-shop généralisé, nous avons développé une plate-forme « générique » centrée sur l'optimisation et la simulation des problèmes de planification et d'ordonnancement (Carlier et Chrétienne, 1998 ; Chu et Proth, 1996 ; Esquirol et Lopez, 1999 ; Pinedo et Chao, 1999). Lors de la résolution de problèmes concrets, il apparaît clairement que les méthodes d'optimisation exactes sont limitées par leurs exigences en temps de calcul et en espace mémoire. La méthode de résolution mise en œuvre doit être capable de reproduire précisément le fonctionnement réel d'un atelier de production en un temps raisonnable.

Pour ce faire, notre plate-forme repose sur plusieurs modèles hybrides combinant la simulation avec différentes méthodes d'optimisation. Cette plate-forme, nommée *PlanOrdo*, est composée de modules réutilisables, organisés autour d'une base de données. Cette organisation extrêmement souple facilite le couplage à des outils complémentaires pour fournir aux décideurs une aide à la décision aussi complète, précise et fiable que possible.

Le module d'ordonnancement est constitué de méthodes sérielles classiques telles que SPT (*Shortest Processing Time*) auxquelles viennent s'ajouter plusieurs méthodes sérielles spécifiques. Afin de pallier les faiblesses caractéristiques de ces méthodes (Hansmann et Hoeck, 1997 ; Smith et Seidmann, 1983 ; Waikar *et al.* 1995),

les ordonnancements fournis par ce module sont améliorés par le module d'optimisation au moyen d'une (ou plusieurs) méta-heuristique(s) disponible(s) au sein de la plate-forme.

Le cœur de la plate-forme est constitué d'un modèle de simulation destiné à évaluer la performance des ordonnancements générés. Développé dans le langage de notre simulateur *RAO* (Artiba *et al.*, 1998), ce dernier est composé de *patterns* génériques décrivant le fonctionnement des ressources standards. Via notre modélisation graphique *ALIX* (Pichel *et al.*, 2003). Ce modèle peut éventuellement être complété, au moyen de *patterns* dédiés au problème étudié. Le modèle de simulation est donc construit pour tenir compte des moindres détails de fonctionnement du processus de production sous-jacent.

Dans le cadre d'une production sur commande, afin de prendre en compte les fortes variations de la demande, l'outil développé repose non pas sur une méthode d'ordonnancement unique mais sur un panel de modèles hybrides et de stratégies d'ordonnancement. Celui-ci offre une méthodologie robuste et adaptable à la prise de décision en temps limité.

Dans le cadre de cette recherche, notre plate-forme fournit un « banc d'essai » pour comparer différentes stratégies d'ordonnancement et déterminer dynamiquement la stratégie la plus adaptée au contexte. Chaque stratégie d'ordonnancement est composée de l'exécution en séquence d'une méthode sérielle, d'un

modèle de simulation et d'une optimisation par une (ou plusieurs) méta-heuristique(s).

Le recours à des méta-heuristiques dans nos méthodes hybrides améliore les résultats, sans garantir l'optimalité des solutions. Leur utilisation au sein de méthodes hybrides est toutefois induite par le besoin de réactivité des entreprises. En effet, la rapidité de la prise de décision peut s'avérer cruciale dans la course à la performance exigée par le contexte économique actuel. Une solution « acceptable » n'est pas nécessairement une solution optimale, mais souvent une solution satisfaisante obtenue en un temps raisonnable.

Le recours à un modèle de simulation au sein de nos méthodes hybrides est induit par la complexité des processus de production à évaluer. Pour de tels processus, un modèle analytique à la fois précis et évolutif est difficile voire impossible à concevoir. La simulation offre au contraire des techniques pour y parvenir facilement.

Il est à noter, que la méthodologie proposée dans cet article ne se justifie que lors de l'utilisation de la simulation comme fonction d'évaluation, ce qui n'est pas le cas dans les benchmarks de la littérature.

Notre étude synthétise les résultats de l'utilisation de la plate-forme dans une entreprise ayant un système de production semi-continu fortement contraint (Duvivier *et al.*, 2003). De nombreuses contraintes liées notamment à l'hétérogénéité des postes de travail, aux différents types de temps de préparation (notés *setups*) à prendre en compte et à la gestion des ressources auxiliaires, rendent ce problème particulièrement difficile à appréhender en termes de modélisation, d'optimisation et de simulation.

Dans cet article, nous nous concentrons sur l'utilisation d'un modèle hybride constitué d'une méthode sérielle, d'un modèle de simulation et d'une méthode Tabou. Nous évaluons plus particulièrement les performances de deux opérateurs de déplacement. Pour chacun d'eux, deux stratégies d'ordonnancement sont comparées et classées sur base d'un ensemble de mesures de performance classiques en ordonnancement.

Le plan suivi pour présenter notre étude est le suivant : nous décrivons le modèle hybride en section 2, nous exposons le problème industriel en section 3, nos résultats sont synthétisés en section 4, nous concluons ensuite l'exposé de notre étude en présentant quelques perspectives de développement de nos travaux.

2 DESCRIPTION DU MODELE HYBRIDE

Le fonctionnement global de notre modèle hybride est schématisé succinctement en Figure 1.

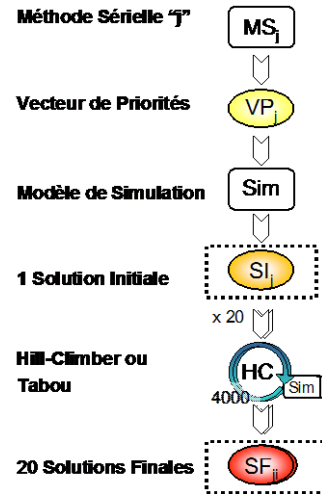


Figure 1. Plan d'expérimentation et fonctionnement général des méthodes hybrides

Sur base d'un carnet de commandes et d'informations complémentaires fournies par les décideurs (urgences de certaines commandes...), chaque méthode sérielle implémentée dans la plate-forme fournit au modèle de simulation un vecteur de priorités, associant une priorité à chaque commande ou ordre de fabrication. En retour, le modèle de simulation calcule l'ordonnancement et les mesures de performance associées (notées « critères » dans la suite de l'article). Ce vecteur de priorités des commandes et les informations relatives à la performance de l'ordonnancement obtenu constituent la solution initiale de plusieurs modèles hybrides basés sur une méthode itérative de recherche locale d'amélioration d'une solution unique (Hoos et Stützle, 2005). Parmi cette famille de méthodes itératives, cet article est focalisé sur une recherche Tabou (Glover F., 1986 ; Glover F., 1989a ; Glover F., 1989b). Il est à noter que notre modèle hybride est conçu pour être utilisé soit au sein de notre plate-forme, soit en tant que module autonome d'optimisation.

Dans sa version « de base », le principe de fonctionnement d'une méthode itérative de recherche locale dans le cadre de l'amélioration d'une solution unique est le suivant. A partir de la solution initiale, une solution candidate est engendrée via un « opérateur de déplacement ». Celle-ci est choisie parmi les solutions voisines, accessibles à partir de la solution initiale en une application de l'opérateur de déplacement. Si la solution candidate offre une meilleure évaluation que la solution initiale, alors la solution candidate devient la solution courante. L'opérateur est ensuite appliqué à la solution courante afin de donner naissance à une nouvelle solution candidate. Ce cycle, appelé itération, est répété jusqu'à la satisfaction de la condition d'arrêt prédéfinie (par exemple, un nombre d'itérations maximum). Il est à noter qu'une évaluation de la solution candidate est effectuée lors de chaque itération, si bien que le nombre d'itérations correspond au nombre d'évaluations dans cette version de base. Dans la suite de l'article, pour

faciliter la désignation d'une telle méthode, nous utiliserons le terme « hill-climber » (ou HC).

Le hill-climber est extrêmement simple à mettre en œuvre, mais ne comporte aucun dispositif lui permettant d'éviter d'être piégé par un optimum local. Il ne comprend pas de mécanisme de diversification, destiné à explorer l'espace de recherche. Il est par exemple facilement piégé par un *plateau* (ensemble de solutions voisines de même coût) sur lequel il risque de cycler indéfiniment sans parvenir à améliorer la solution courante. Pour éviter ces écueils, la méthode Tabou mémorise les derniers mouvements effectués dans une liste appelée liste Tabou. Cependant, comme le *Petit Poucet*, le nombre de cailloux est en nombre limité et par conséquent le nombre de mouvements mémorisés est borné par la *taille de la liste tabou*. Compte tenu de la taille de l'espace de recherche, il ne sera malheureusement pas possible de mémoriser l'ensemble du chemin conduisant de la solution initiale à toute solution rencontrée. Par contre, le fait de mémoriser les derniers mouvements effectués permettra d'interdire de les exécuter à nouveau et ainsi de cycler sur cette succession de mouvements récents. Afin d'être plus efficace encore, les mouvements inverses des mouvements effectués sont stockés dans la liste tabou. De ce fait, il est possible de détecter le début d'un cycle dès que la moitié du cycle (mouvements Mvt_1 à Mvt_5) a été parcourue (cf. Figure 2). Ce mécanisme permet de prévenir les cycles (courts).

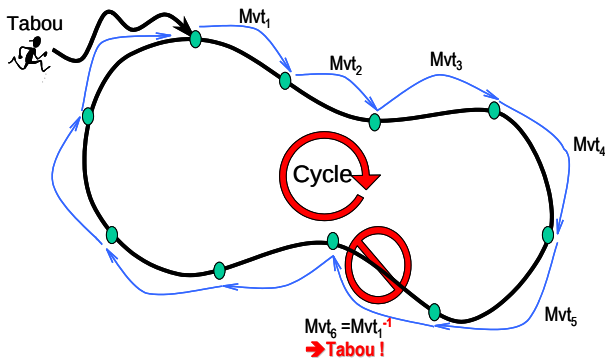


Figure 2. Détection des demi-cycles dans le Tabou

Cependant dans l'exemple de la Figure 2, le mouvement 6 (Mvt_6) qui n'est autre que l'inverse du mouvement 1 (Mvt_1^{-1}) est par conséquent interdit (Tabou). Or, il se pourrait que ce mouvement conduise à une solution meilleure que la meilleure solution trouvée jusqu'alors. En ce cas, la méthode tabou lève le caractère Tabou/interdit du mouvement en appliquant une règle particulière appelée « critère d'aspiration », visant à ne pas s'interdire la découverte de bonne solution au travers des mouvements Tabous. De plus, la prochaine solution visitée par le tabou est la « meilleure » solution voisine, même si cette dernière solution offre un coût moins intéressant que la solution courante. Ceci permet à la méthode tabou de dégrader le coût de la solution courante dans « l'espoir » de découvrir une meilleure

solution. Un « pas » pour la méthode Tabou consiste donc à évaluer un certain nombre de solutions voisines pour retenir la « meilleure » et se déplacer de la solution courante vers cette solution voisine. Nous utilisons par conséquent la notion de « pas » plutôt que la notion d'itération pour désigner un cycle de la méthode tabou. Afin de décrire précisément le principe de fonctionnement de la méthode tabou implantée, son pseudo-code est donné en Figure 3. La taille de liste Tabou $TabuListSize$ est fixée à 134, la fraction du voisinage explorée à chaque pas $SubNeighborhoodFrac$ est fixée à 1,5% (soit $SubNeighborhoodSize=134$ mouvements parmi 8911 dans le voisinage complet).

Les variables suivantes sont utilisées :

MaxEval :	Nombre maximum d'évaluations
Step :	Pas courant
SubNeighborhoodSize :	Taille du voisinage exploré à chaque pas
CurrentSol :	Solution courante
BestSol :	Meilleure solution trouvée jusqu'alors
NeighBestSol :	Meilleure solution voisine
CurrentCost :	Coût d'une solution (fonction agrégée)
BestCost :	Coût de la meilleure solution (actuelle)
NeighBestCost :	Coût de la meilleure solution voisine
u, v :	Positions pour le mouvement effectué
ShortMemory :	Mouvements déjà testés (matrice)
TabuList :	Liste tabou (matrice)

procédure EffectueUnPas : retourne le couple (u, v) sélectionné

```

(u,v) ← (ϕ, ϕ)
NeighBestCost ← ϕ
Effacer(ShortMemory)
pour n allant de 1 à SubNeighborhoodSize faire
  // Retourner (u, v) non déjà visité et mettre à jour ShortMemory
  SelectionneMouvement(u, v, ShortMemory) // Avec u≠v
  Deplacement(u, v, CurrentSol)
  CurrentCost ← Evaluer(CurrentSol)
  Evaluation ← Evaluation + 1
  si TabuList[u, v] ≤ Step ou CurrentCost < BestCost alors
    // Mouvement non tabou ou aspirant
    si NeighBestCost = ϕ ou NeighBestCost > CurrentCost alors
      (NeighBestSol, NeighBestCost) ← (CurrentSol, CurrentCost)
  Deplacement(v, u, CurrentSol) // Effectue le mouvement inverse
  (CurrentSol, CurrentCost) ← (NeighBestSol, NeighBestCost)

```

procédure Tabou : retourne le couple (BestSol, BestCost)

```

Evaluation ← 1
Step ← 0
Effacer(TabuList)
CalculerEtEvaluerSolutionInitiale(CurrentSol, CurrentCost)
(BestSol, BestCost) ← (CurrentSol, CurrentCost)
(NeighBestSol, NeighBestCost) ← (CurrentSol, CurrentCost)
Tant-Que Evaluation ≤ MaxEval faire
  Step ← Step + 1
  EffectueUnPas(u, v)
  MettreAJour(TabuList, u, v)
  si BestCost ≥ NeighBestCost alors
    (BestSol, BestCost) ← (NeighBestSol, NeighBestCost)
Resultat ← (BestSol, BestCost)

```

Figure 3. Fonctionnement de la méthode Tabou

Différents opérateurs de déplacement ont été testés. Ces opérateurs modifient de manière plus ou moins importante le vecteur de priorités associé à la solution courante pour engendrer un nouveau vecteur de priorités relatif à la solution candidate. Le premier opérateur est l'opérateur « swap ». Il consiste à échanger deux

éléments du vecteur courant, situés aux positions u et v sélectionnées aléatoirement, pour obtenir un nouveau vecteur (cf. Figure 4).

Le second opérateur est un « décalage circulaire ». Il consiste à sélectionner deux positions (u et v) aléatoirement dans le vecteur. Le premier élément sélectionné (en position u) est momentanément « éjecté » du vecteur et la sous-séquence située entre les deux positions préalablement sélectionnées est décalée (d'une position vers la droite ou d'une position vers la gauche) de manière à combler le vide laissé (en position u) par l'éjection de l'élément. L'élément éjecté est réintroduit dans le vecteur dans le vide laissé (en position v) par le décalage de la sous-séquence. Le principe de ces deux opérateurs est illustré en Figure 4. L'avantage principal de ces opérateurs est qu'ils ne nécessitent aucune information sur le processus en cours d'optimisation. De ce fait, ces opérateurs « génériques » sont applicables à une très large gamme de problèmes d'ordonnancement, ce qui nous permet d'évaluer aisément les performances des modules « génériques » assemblés au sein de notre plate-forme. Il est cependant possible d'améliorer les performances de ces opérateurs en tenant compte des spécificités du processus ordonné.

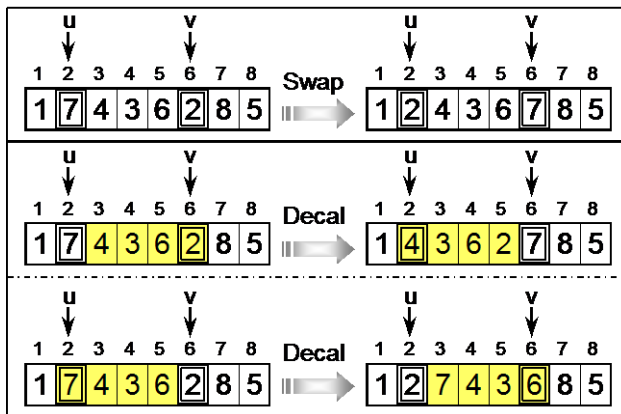


Figure 4. Fonctionnement des opérateurs

L'opérateur swap est particulièrement simple à implémenter et extrêmement rapide en terme d'exécution car il consiste simplement à échanger deux valeurs dans le vecteur. Il possède également une propriété intéressante pour la recherche Tabou en ce sens que le mouvement obtenu en échangeant u et v est le même qu'en échangeant v et u . C'est-à-dire que le mouvement (u,v) est identique au mouvement inverse (v,u) , ou que le fait d'appliquer deux fois le mouvement (u,v) ou (v,u) revient à l'identité (i.e. le vecteur reste inchangé). Du point de vue de l'ordonnancement associé au vecteur de priorités, ceci revient à échanger les priorités/rangs des commandes situées aux positions u et v . Dans l'exemple de la Figure 4, avant l'application de l'opérateur swap, la commande $u=2$ a la priorité 7 et la commande $v=6$ a la priorité 2. Après le swap, la commande $u=2$ a la priorité 2 et la commande $v=6$ a la priorité 7.

L'opérateur de décalage circulaire (decal) est légèrement plus complexe à implémenter, car il nécessite le décalage d'une tranche du vecteur vers la droite ou la gauche. Il ne possède pas la propriété de l'opérateur swap, c'est-à-dire qu'en appliquant deux fois un décalage nous n'obtenons pas l'identité. Du point de vue de l'ordonnancement, cet opérateur est beaucoup plus « agressif » que le swap, c'est-à-dire qu'il perturbe plus fortement l'ordonnancement étant donné qu'il modifie toutes les priorités des commandes situées entre les positions u et v . En terme de déplacement dans l'espace de recherche, ceci se traduit par une distance¹ plus élevée entre solutions voisines et donc une meilleure exploration du paysage. « L'agressivité » de l'opérateur se traduit par une variation plus forte du coût au cours du déplacement de la méthode de recherche.

Le plan d'expérimentation suivi dans le cadre de cette étude est illustré par la Figure 1. Parmi les différentes méthodes sérielles implémentées (Dhaevers *et al.*, 2004), deux d'entre-elles (MS_j ; $j = 1$ ou 3) ont été sélectionnées afin de réduire le volume des résultats présentés. Cette sélection a été effectuée de manière à ne retenir que deux méthodes sérielles représentatives de la qualité des solutions initiales obtenues pour le carnet de commandes utilisé, parmi toutes les méthodes sérielles disponibles au sein de notre plate-forme *PlanOrdo*.

Compte tenu du caractère stochastique de l'algorithme itératif de recherche locale, partant d'une solution initiale SI_i fournie par chaque méthode sérielle MS_j , nous avons effectué 20 essais ($i=1,2,...,20$) de 4000 évaluations, chaque évaluation nécessitant le lancement du modèle de simulation. Le nombre d'essais et d'évaluations a été établi sur la base de benchmarks préliminaires (Dhaevers *et al.*, 2005; Duvivier *et al.*, 2006). Ces travaux ont montré expérimentalement que 4000 évaluations étaient amplement suffisantes pour engendrer de bonnes solutions finales compte tenu des contraintes de temps préalablement établies. De ce fait l'ensemble des méthodes testées dispose d'un maximum de 4000 évaluations par essai pour engendrer une solution. A l'issue de chacun des 20 essais, nous obtenons 20 solutions finales SF_{ji} par stratégie j . Ce paramétrage vise à assurer un bon compromis entre temps de calcul et qualité des solutions finales.

Afin d'indiquer au décideur la stratégie d'ordonnancement à appliquer, les solutions optimisées SF_{ji} obtenues pour chaque stratégie, sont ensuite classées via un tri lexicographique. Nous sommes parfaitement conscients du fait que le recours à un tri lexicographique requiert un classement strict des critères à optimiser ce

¹ La distance entre deux permutations est généralement définie par le nombre minimum d'opérations élémentaires (swaps en général) nécessaires pour transformer la première permutation en la seconde. Avec cette définition deux permutations engendrées par l'opérateur swap sont situées à distance 1 (pourvu que u et v soient différents).

qui limite le champ d'application de notre méthode. Cependant la plate-forme PlanOrdo dispose également d'une méthode multicritère « Promethee II » (Brans *et al.*, 2002). Cette méthode a été désactivée dans le cadre de cet article afin de focaliser sur l'étude des opérateurs sans multiplier les paramètres.

De nombreux travaux antérieurs sur l'utilisation d'une méthode tabou ont déjà été publiés, mais rares sont les travaux comparant différents opérateurs de déplacement au sein d'une recherche tabou appliquée à un problème d'ordonnancement semi-continu fortement contraint. De plus, nos travaux se situent également dans le contexte de la prise de décision en temps limité. Or comme l'évaluation de chaque solution passe par l'exécution d'un modèle de simulation, consommatrice de temps processeur, nous nous devons d'obtenir une méthode de recherche robuste (pour limiter le nombre d'essais) et efficace (pour limiter le nombre de pas et d'évaluations nécessaires à la découverte de *bonnes* solutions). Les opérateurs présentés sont comparés en tenant compte de ces contraintes.

Après avoir détaillé le modèle hybride implémenté, la section suivante décrit le problème industriel de l'entreprise ArcelorMittal Fontaine.

3 PROBLEME INDUSTRIEL CONSIDERE

La validation de notre démarche est effectuée sur le problème d'ordonnancement du système de production de l'entreprise ArcelorMittal Fontaine, située à Fontaine-Lévêque, en Belgique. Sa production est composée de fils et de torons d'acier de haute qualité (câbles constitués de 2, 3 ou 7 fils), utilisés dans le monde entier, dans des constructions diverses telles que les aéroports, les stades, les ponts... Le système de production considéré est apparenté à un job-shop généralisé composé de dix-sept postes de travail (cf. Figure 5).

Le processus de production comporte sept groupes de machines, associés aux étapes du processus de fabrication. Les opérations continues et discrètes s'enchaînent via des opérations de soudure ou de découpe en vue du stockage des produits semi-finis sur des ressources auxiliaires, appelées *bobines*. Le simulateur RAO est utilisé pour modéliser la totalité du problème, y compris la partie continue du processus.

Chaque produit, obtenu à partir de fils machine (*Raw wire*) ou de fils galvanisés (*Galvanised wire*) (cf. Figure 5), est le résultat d'une séquence d'opérations ordonnancées en respectant les contraintes liées aux ressources et aux relations de précédence. Les contraintes liées aux ressources incluent des limitations au niveau des entrées/sorties, des *setups* et de la non-préemption. Les relations de précédence imposent qu'une opération ne peut commencer avant la fin de toutes les opérations précédentes identifiées dans la gamme opératoire du produit. L'hétérogénéité des

machines induit une dépendance entre le temps opératoire, le produit fabriqué et la machine utilisée pour réaliser l'opération. Enfin, la production de l'entreprise cible, de type « fabrication sur commande », nécessite la prise en compte de fortes variations des commandes d'un mois à l'autre.

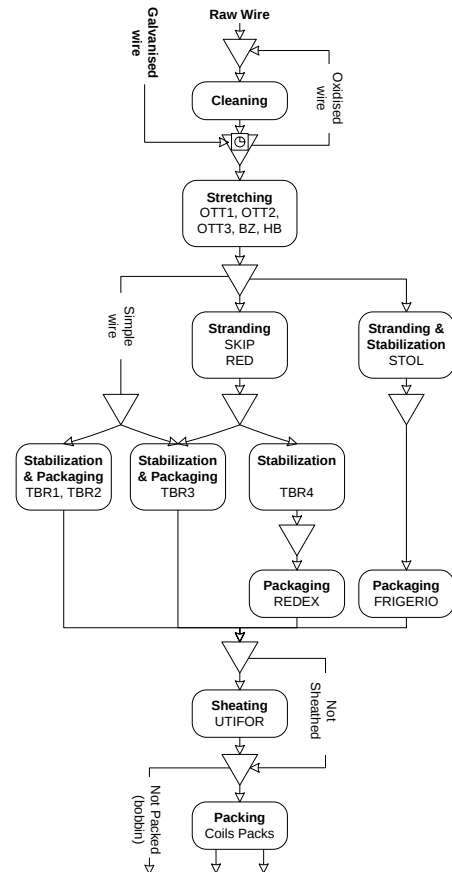


Figure 5. Système de production de l'entreprise ArcelorMittal Fontaine

Dans ce processus, différents types de *setups* doivent être considérés, dépendants et indépendants de la séquence de réalisation des produits sur les postes de travail. De nombreuses contraintes additionnelles rendent ce problème très difficile à appréhender en termes de modélisation, d'optimisation et de simulation. Parmi celles-ci, figurent l'utilisation de ressources auxiliaires en nombre limité (bobines), les contraintes sur les diamètres et les charges supportés en entrée/sortie des postes de travail. L'hétérogénéité de ces derniers implique que toutes les bobines ne sont pas compatibles avec toutes les machines. De même, toutes les machines ne sont pas capables de réaliser tous les produits et/ou tous les types de conditionnement. Mentionnons également qu'un sous-ensemble de commandes ne peut accuser de retard en raison notamment des moyens de transport utilisés lors de leur expédition. Le modèle réalisé doit également reproduire et anticiper, dans la mesure du possible, les dysfonctionnements du processus tels que des ruptures de fils. Les problèmes liés au stockage et au transport au sein de l'entreprise

étudiée ne sont pas pris en compte dans notre étude car ils n'ont pas d'impact significatif.

Le site travaille 24 heures sur 24, cinq jours sur sept avec une rotation journalière de trois équipes de 11 ouvriers.

L'évaluation des solutions courantes s'effectue sur la base des critères sélectionnés par le décideur. Ces critères servent également à la comparaison des solutions finales des stratégies d'ordonnancement pour les différents opérateurs testés. Dans cette recherche, quatre critères sont pris en compte. Le Tableau 1 présente leur description et la notation utilisée pour les identifier.

Notation	Description
n	Nombre de commandes
d_k	Date d'exigibilité de la commande k
C_k	Date de fin de production de la commande k
$E_k = \max(0, d_k - C_k)$	Durée de l'avance de la commande k
$T_k = \max(0, C_k - d_k)$	Durée du retard de la commande k

Critère	Description
(1) N_T	Nombre de commandes en retard
(2) $T = \frac{1}{n} \sum T_k$	Retard moyen
(3) $C_{\max} = \max C_k$	Durée de production totale (<i>makespan</i>)
(4) $E = \frac{1}{n} \sum E_k$	Avance moyenne

Tableau 1. Notations et descriptions des critères optimisés.

Il s'agit, dans cette étude, de minimiser les critères (1) à (3) et de maximiser le critère (4). Si la minimisation des trois premiers critères est couramment utilisée lors de la résolution d'un problème d'ordonnancement, la maximisation de l'avance est moins fréquente. Dans ce cas d'application, l'objectif est de se prémunir des dysfonctionnements apparaissant en cours de production afin de garantir, autant que possible, le respect des délais de livraison. Les impératifs de l'entreprise étudiée nous imposent de considérer ces quatre critères dans un ordre décroissant d'importance $N_T > T > C_{\max} > E$ (ici « > » désigne une relation de préférence). Les valeurs des critères (2) à (4) sont exprimées en heures.

Dans la section suivante, les solutions finales de deux stratégies d'ordonnancement sont présentées et comparées et ce, pour les différents opérateurs de déplacement testés.

4 RESULTATS EXPERIMENTAUX

Afin de respecter les clauses de confidentialité relatives à ce projet et aux données associées, les comparaisons directes entre la production réelle et la production simulée ne sont pas présentées. Notons que le comportement de notre modèle hybride a été validé sur différentes périodes.

De manière générale, l'objectif de nos modèles hybrides est d'ordonnancer avec précision un mois de production pour l'usine ArcelorMittal Fontaine. Notre simulation porte sur 134 commandes et prend une à trois secondes sur un Pentium IV 3GHz. Le temps exact dépend du nombre d'ordres de fabrication contenus dans la période analysée et des statistiques calculées pendant la simulation. Afin de prendre en compte les fortes variations de la demande subie par l'entreprise, l'outil développé repose, non pas sur une méthode d'ordonnancement unique, mais sur un panel de modèles hybrides et de stratégies d'ordonnancement.

Les paragraphes suivants montrent les résultats obtenus par deux de ces stratégies basées sur un modèle hybride contenant une méthode Tabou.

Nous utilisons les notations suivantes pour désigner les méta-heuristiques et opérateurs comparés :

- HC+swap : Hill-climber utilisant l'opérateur swap
- Tabou+swap : Tabou utilisant l'opérateur swap
- Tabou+decal : Tabou utilisant l'opérateur Decal

Précédemment, nous avons utilisé un hill-climber dans le cadre de l'optimisation d'ordonnancement en industrie semi-continue. Cette méthode a donné de très bons résultats, mais souffre cependant d'un problème de robustesse. Nous avons donc recherché une autre méta-heuristique simple (en raison de la contrainte liée à la durée de l'évaluation par simulation en contexte d'aide à la décision en temps limité). Notre choix s'est porté sur une méthode de recherche Tabou, car cette dernière a été conçue de manière à la rendre plus robuste en évitant les pièges que sont les optima locaux et les cycles courts. Comme expliqué ci-dessus, nous avons expérimenté la recherche Tabou avec deux opérateurs : swap et decal. Les résultats ont été mesurés sur les solutions fournies par deux méthodes sérielles pour un carnet de commandes donné. Les deux ordonnancements obtenus par ces deux méthodesérielles sont caractéristiques des deux cas de figure se présentant dans la phase initiale, c'est-à-dire un premier ordonnancement (stratégie 1) de bonne qualité et un second (stratégie 3) de mauvaise qualité. La qualité d'un ordonnancement est mesurée par les critères de performance précédemment définis.

Pour la stratégie 1, la Figure 6 montre que les différentes méta-heuristiques (hill-climber et tabou avec decal et swap) donnent des résultats proches. Cependant, les résultats détaillés présentés en Tableau 2 indiquent que la moyenne et la médiane des résultats sont meilleures pour le Tabou, quel que soit l'opérateur utilisé. Il s'avère toutefois que l'opérateur fournissant les meilleures solutions est le swap. Cette différence de performances semble être due à l'aspect plus agressif du décalage face au swap.

La solution initiale de la stratégie 1 est déjà de bonne qualité. Le hill-climber semble donc ne pas devoir cheminer longtemps pour améliorer cette solution de proche

en proche avec l'opérateur swap, peu agressif. Le tabou donne de meilleurs résultats et c'est normal car il exploite mieux le voisinage. L'opérateur de décalage apparaît comme trop agressif, ne permettant que difficilement l'amélioration de la solution en la faisant vagabonder

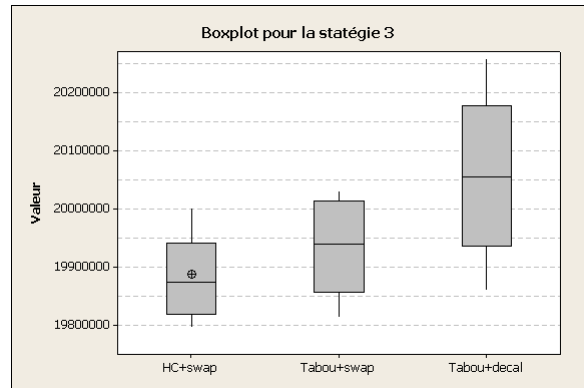
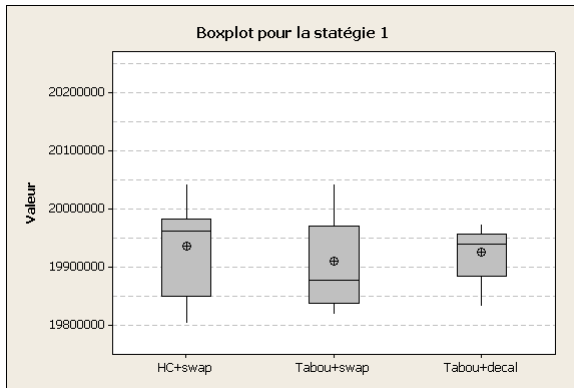


Figure 6. Comparaison des méta-heuristiques utilisées
(croix-cerclée : moyenne, séparation interne du *boxplot* : médiane)

Stratégie 1					Stratégie 3				
	N_T	T	C_{max}	E	N_T	T	C_{max}	E	
		(H)	(H)	(H)		(H)	(H)	(H)	
HC swap	moyenne	0	0	742.88	280.56	0	0	741.07	279.84
	médiane	0	0	743.87	280.66	0	0	740.59	280.23
	écart-type	0	0	2.79	1.37	0	0	2.51	2.05
Tabou swap	moyenne	0	0	741.95	281.28	0.15	1.01	743.08	279.65
	médiane	0	0	740.72	281.30	0	0	743.04	278.74
	écart-type	0	0	2.76	0.95	0.37	3.14	3.39	3.28
Tabou decal	moyenne	0	0	742.48	273.07	0.15	2.17	745.65	275.90
	médiane	0	0	743.00	273.06	0	0	745.41	275.16
	écart-type	0	0	1.61	1.90	0.37	6.43	3.93	3.07

Tableau 2. Résultats détaillés

Pour la stratégie 3, la solution initiale n'est pas correctement structurée et le hill-climber donne de meilleures solutions car il a la possibilité de cheminer plus longtemps à l'aide d'un opérateur moins agressif (le swap). La recherche Tabou donne de moins bonnes solutions au final car elle semble perdre son temps en évaluant le sous-voisinage de chaque solution visitée. De plus, en raison de la limite de temps imposée, la recherche Tabou n'effectue que 30 pas dans l'espace de recherche. Ce qui explique les moins bonnes performances du Tabou+swap. Le Tabou+decal se comporte comme pour la stratégie 1 : trop agressif, l'opérateur decal semblerait faire vagabonder la recherche Tabou autour de solutions de bonne qualité.

Pour synthétiser ces résultats, nous pouvons conclure que la recherche Tabou+swap donne de bons résultats en intensification au vu du faible nombre d'évaluations imposé par la contrainte de temps. Par contre, ce nombre d'évaluations est insuffisant pour que la méthode Tabou puisse être efficace sur des solutions initiales de mauvaise qualité. Par conséquent l'utilisation de la recherche Tabou actuellement préconisée au sein de la plate-forme est la suivante :

- 1) Un panel de méthodes sérielles sont appliquées au carnet de commandes à ordonnancer ;

« autour » des zones contenant des bonnes solutions. Le fait d'échantillonner des solutions plus diversifiées à chaque pas pour ne retenir que la meilleure donne au final des solutions plus proches les unes des autres.

- 2) Les solutions obtenues par simulation sont évaluées et comparées par un tri lexicographique ;
- 3) Le meilleur ordonnancement est optimisé par une méthode Tabou.

5 CONCLUSIONS ET PERSPECTIVES

Dans cet article, une approche hybride d'optimisation et simulation pour un problème industriel réel d'ordonnancement est présentée. Nous avons décrit plus particulièrement les résultats de plusieurs opérateurs de déplacement intégrés au sein d'une méthode de recherche Tabou. Dans le cadre de l'aide à la décision en temps limité, nous avons démontré expérimentalement l'efficacité du modèle hybride développé. Il est à noter que l'approche proposée n'est pas limitée au problème de l'entreprise ArcelorMittal Fontaine.

Au-delà de l'hybridation de plusieurs méthodes issues de différents domaines de recherche (modélisation, optimisation, simulation, performance) et de leur application dans un contexte d'aide à la décision en temps limité, l'un des points forts de nos travaux est la prise compte des ressources auxiliaires au sein, notamment, du modèle de simulation. Une autre originalité de ces travaux réside dans la précision de ce modèle de simulation ainsi que dans la diversité des indicateurs de performance et des stratégies d'ordonnancement disponibles au sein de notre plateforme.

Le contexte particulier de la prise de décision en temps limité implique de borner le temps d'exécution des méthodes hybrides appliquées en limitant le nombre d'évaluations. De ce fait, il n'est pas possible d'utiliser un mécanisme de diversification au sein de notre recherche Tabou. Cette dernière n'aurait effectivement plus suffisamment de temps pour converger vers de

bonnes solutions. C'est pour cette raison qu'aucun mécanisme de diversification n'a été intégré dans la méthode Tabou utilisée. Pour pallier cette absence de diversification, nos modèles hybrides sont basés sur un ensemble de méthodes sérielles, offrant un panel de solutions initiales dispersées dans l'espace de recherche. Le modèle hybride présenté complète cette hybridation par l'utilisation d'une méthode Tabou comme intensificateur de la recherche qui explore le voisinage de la solution initiale.

Nous travaillons actuellement à la mise au point d'une version itérée de l'opérateur de décalage circulaire dont l'objectif est d'apporter à notre hybride un mécanisme élaboré d'exploration en contrôlant « l'agressivité » de l'opérateur de manière à obtenir une bonne exploration de l'espace de recherche pour *intelligemment* focaliser sur une zone au fil de chaque essai à la manière d'un recuit simulé. L'objectif principal est de rendre la méthode Tabou moins sensible à la solution initiale pré-calculée au moyen d'une méthode sérielle et de la doter d'un mécanisme d'exploration élaboré ; ceci, sans allonger la durée des essais.

Comme cela a été précédemment évoqué, nous avons désactivé la méthode Promethee II, disponible au sein de notre plate-forme, afin de focaliser sur l'étude des opérateurs sans multiplier les paramètres. Il est clair que dans une prochaine étape cette méthode sera réactivée afin de tenir compte de nuances dans le jugement du décideur et de fournir une comparaison « plus réaliste » des solutions engendrées. Celle-ci apporte en effet des nuances dans le classement des stratégies d'ordonnancement et absorbe les incertitudes liées à l'acquisition, la synthèse des données ou encore aux aléas du processus de production (pannes, dysfonctionnement...). Cette caractéristique évite les changements intempestifs de stratégie d'ordonnancement en intégrant les seuils de préférence définis par le décideur et conduit, au niveau de l'atelier, à une stabilité accrue de l'ordonnancement.

A ce jour, nous avons terminé la collecte et la synthèse des informations destinées à enrichir l'ensemble des critères d'évaluation actuels par la prise en compte d'aspects financiers et relatifs à la maintenance (Jamali *et al.* 2004). Il s'agit d'offrir au décideur une vue aussi complète que possible, non limitée aux critères directement liés à la production, afin de sélectionner la stratégie d'ordonnancement la plus adaptée.

6

7 REMERCIEMENTS

Cette recherche a été réalisée grâce au financement de la Région Wallonne et du Fond Social Européen. Les auteurs souhaitent remercier le personnel de l'entreprise ArcelorMittal Fontaine pour leur contribution à ce travail.

REFERENCES

- Artiba, A., Emelyanov, V. V. et Iassinovski, S. I., 1998. *Introduction to Intelligent Simulation: The RAO Language*. Boston, Kluwer Academic Publishers.
- J.P. Brans et B. Mareschal, 2002. PROMÉTHÉE-GAIA: Une méthodologie d'aide à la décision en présence de critères multiples. Éditions Ellipses, ISBN 2729812539.
- Carlier, J. et Chrétienne, P., 1988. *Problèmes d'ordonnancement, modélisation, complexité, algorithmes*. France, Masson.
- Chu C. et Proth J-M., 1996. *L'ordonnancement et ses applications*. Masson.
- Dhaevers V., Roux O., Lazraq A., Duvivier D., Meskens N. et Artiba A., 2005. Analyse statistique et classement de stratégies d'ordonnancement: cas d'une industrie se-mi-continue, Revue électronique E-STA Sciences et Technologies de l'Automatique, Année 2005, quatrième trimestre, Volume 2, n°4, <http://www.see.asso.fr/esta>.
- Duvivier, D., Dhaevers, V., Bachelet, B. et Artiba, A., 2003. Integrating simulation and optimization of Manufacturing systems. IEEE - Systems, Man, and Cybernetics, part C: Applications and reviews 33 (2), 186-192.
- Duvivier D., Artiba A. et Dhaevers V., 2005. Simulation based performance measurement and analysis: an industrial application. International Journal of Computer Integrated Manufacturing, 18(5), p. 402-407.
- Duvivier D., Roux O., Dhaevers V. et Lazraq A., 2006. Comparison of scheduling strategies via a statistical analysis in an industrial context, In proc. of International Conference Service Systems and Service Management IEEE SSSM'06, Troyes.
- Esquirol P. et Lopez P., 1999. *L'ordonnancement*. Economica.
- Glover F., 1986. Future paths for integer programming and link to A.I. Computers and Operations Research, p. 533-549.
- Glover F., 1989a. Tabu search - part 1. ORSA Journal on Computing, 1((3)), 190-206.
- Glover F., 1989b. Tabu search - part 2. ORSA Journal on Computing, 2((1)), 4-31.
- Hansmann, K.-W. et Hoeck, M., 1997. Production control of a flexible manufacturing system in a job shop environment. International Transactions in Operational Research 4 (5/6), 341-351.
- Hoos H. H. et Stützle Th., 2005. *Stochastic local search – Foundations and applications*. Morgan-Kaufmann publishers.
- Jamali M.A., Roux O., Ait-Kadi D. et Artiba. A., 2004. An integrated approach using optimization methods and simulation in maintenance strategies analysis. 14th International Institution for Production Engineering Research, CIRP Design Seminar, Cairo, Egypt.
- Pichel D., Dontaine A., Iassinovski S., Artiba A. et Fagnart C., 2003. ALIX : une méthode de modélisation

- des flux d'un système de production. Conf. CPI 2003, Maroc.
- Pinedo, M. et Chao, X., 1999. *Operations Scheduling with applications in manufacturing and services*. Irwin/McGraw-Hill.
- Smith, M. L. et Seidmann, A., 1983. Due date selection procedures for job-shop simulation. *Computers and Industrial Engineering Journal* 7 (3), 199-207.
- Waikar, A. M., Sarker, B. R. et Lal, A. M., 1995. A comparative study of some priority dispatching rules under different shop loads. *Production Planning Control* 6 (4), 301-310.