

INTEGRATION DE L'OPTIMISATION ET DE LA SIMULATION EN INDUSTRIE SEMI-CONTINUE

V. DHAEVERS¹, D.DUVIVIER¹, O. ROUX¹ et A. ARTIBA¹

¹ Centre de Recherches et d'Etudes en Gestion Industrielle (CREGI)
Facultés Universitaires Catholiques de Mons
151, Chaussée de Binche
B-7000 Mons – Belgique
[f{dhaevers, duvivier, roux, artiba}@fucam.ac.be](mailto:{dhaevers, duvivier, roux, artiba}@fucam.ac.be)

RÉSUMÉ : *Cet article porte sur l'ordonnement de la production en industrie semi-continue. Le souci de réutilisabilité et de modularité de l'outil mis au point, nous a conduits à développer une plate-forme « générique » centrée sur la planification, l'ordonnement et la simulation des processus de production. La plate-forme est implantée depuis trois ans au sein du processus de décision de l'entreprise Fontainunion. L'étude, présentée dans cet article, synthétise les résultats de l'application de cette plate-forme à un système de production semi-continu fortement contraint, de type jobshop généralisé non-préemptif. Les résultats fournis par la plate-forme conduisent systématiquement à une amélioration des résultats obtenus antérieurement à sa mise en place.*

MOTS-CLÉS : *Ordonnement, optimisation, simulation, modèles hybrides, aide à la décision, performance.*

1. INTRODUCTION

Les problèmes d'ordonnement se rencontrent dans de nombreux domaines d'applications [5, 6, 8, 11, 12, 14, 16, 23, 24, 28]. Dans le cadre des processus de production de type discrets ou semi-continus, nous avons développé une plate-forme « générique » centrée sur l'optimisation et la simulation. Lors de l'étude de problèmes concrets, il apparaît clairement que les méthodes exactes sont limitées par leurs exigences en temps de calcul et en espace mémoire. Plus spécifiquement, un argument à l'encontre de ces méthodes s'appuyant sur un modèle mathématique, est la difficulté de traduire le fonctionnement réel d'un atelier de production. Il est également très ardu de se baser directement sur ce type de formalisme pour expliquer la logique de fonctionnement des modèles aux décideurs. Afin de prendre en compte ces aspects, notre simulateur RAO [2] comporte un puissant langage dédié à la simulation des systèmes discrets. Il est basé sur une représentation du processus sous forme de règles de production modifiées. Ce langage nous fournit une méthode rapide pour décrire la logique de fonctionnement de tout processus de production. Dans le cas d'un processus semi-continu deux approches ont été envisagées au sein de notre plate-forme [14] : si la « partie continue » du processus simulé est modélisable à l'aide d'équations mathématiques simples, ces dernières sont intégrées dans notre modèle de simulation. Par contre, si la partie continue du processus simulé s'avère

plus complexe à modéliser ou à résoudre, la solution retenue consiste à coupler notre simulateur à un solveur.

Concernant l'optimisation du processus, nous avons étudié l'efficacité de divers outils appliqués à des problèmes industriels [9, 10, 17, 19, 26]. Dans cet article, nous nous focalisons sur l'utilisation de méthodes sérielles couplées à notre modèle de simulation, ainsi que sur l'amélioration des résultats via un algorithme itératif de recherche locale. En effet, dans de nombreuses applications, il a été montré que la coopération (hybridation) de méthodes (d'optimisation, de simulation...) conduit à une amélioration des performances obtenues par ces mêmes méthodes prises séparément. Loin de constituer la panacée, nous sommes parfaitement conscients de leurs limitations en terme de résultats théoriques (preuve d'optimalité...). Cependant, nous sommes intimement convaincus que ces méthodes hybrides sont capables de fournir de précieux outils, adaptés aux problèmes industriels. Un autre argument en leur faveur est induit 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 [21]. Une solution « acceptable » n'est pas nécessairement une solution optimale, mais souvent une solution satisfaisante obtenue en un temps raisonnable.

Intégré dans notre plate-forme, notre modèle hybride est utilisé afin d'optimiser l'ordonnement d'un problème industriel concret, la planification étant assurée par un

ERP (*Enterprise Requirement Planning*). Le problème traité est issu de la modélisation de l'outil de production de l'entreprise Fontainunion (située à Fontaine-Lévêque, en Belgique). La production est composée de fils et de torons d'acier de haute qualité, utilisés dans le monde entier, dans des constructions diverses (aéroports, stades, hôtels, ponts...). Le processus est constitué d'opérations discrètes et d'opérations continues. Les résultats présentés concernent un modèle de simulation simplifié où les temps opératoires sont fixés pour toute la durée de simulation. Cette approche de simulation « déterministe » peut sembler restrictive au premier abord mais nous avons opté pour un modèle simplifié afin de faciliter la comparaison des différents hybrides. Toutefois, selon la précision requise, notre simulateur peut également intégrer les aspects non-déterministes du processus sous-jacent, pour tenir compte par exemple des variations des temps opératoires, des fréquences et des durées de pannes ou encore des stratégies de maintenance. Ces variations peuvent être introduites sous forme de distributions statistiques ou sur base d'historiques issus du processus considéré.

Le plan suivi pour présenter notre étude est le suivant : nous décrivons notre plate-forme en section 2, nous exposons le « problème Fontainunion » en section 3, nos résultats sont synthétisés en section 4, nous concluons ensuite l'exposé de notre étude en présentant quelques perspectives de nos travaux (section 5).

2. PLATE-FORME PLANORDO

Notre objectif est de réaliser une plate-forme « générique » centrée sur la simulation et l'optimisation des problèmes de planification et d'ordonnancement de types discrets ou semi-continus. Cependant, il serait illusoire de chercher à concevoir un modèle « générique » sans autre considération [1, 29, 32]. En réalité, nous devons définir clairement le spectre d'application de nos modèles (de simulation, de planification, d'ordonnancement...). Dans le contexte de l'étude présentée, la généralisation de nos résultats s'applique en fait aux problèmes d'ordonnancement de type jobshop généralisé tels qu'ils sont définis en section 3. Guidés par un certain nombre de concepts issus du génie logiciel [3, 15, 20, 30, 33], nous avons conçu notre plate-forme sous forme de modules hautement réutilisables, organisés autour d'une base de données. Cette organisation extrêmement souple permet d'intégrer un grand nombre d'outils complémentaires mis à la disposition des décideurs avec pour but commun de fournir une aide à la décision aussi complète, précise et robuste/fiable que possible (cf. figure 1).

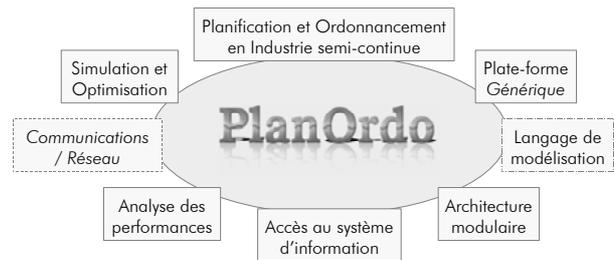


Figure 1. Plate-forme *PlanOrdo*

Dans cet article, l'objectif de notre plate-forme, nommée *PlanOrdo*, est de fournir un « banc d'essai » pour comparer différentes stratégies d'ordonnancement et déterminer dynamiquement la stratégie la plus adaptée au contexte (taille du carnet de commandes, charge de l'atelier...).

L'interface graphique de *PlanOrdo* pilote un certain nombre de modules décrits ci-dessous.

Module d'ordonnancement : A ce jour, ce module comporte un ensemble de méthodes sérielles classiques (cf. section 4). Ces méthodes élémentaires ne sont pas uniquement faciles et rapides à implémenter, à contrôler et à exécuter ; elles sont également extrêmement simples à expliquer aux opérateurs comme aux décideurs. Du point de vue de la « généralité » de *PlanOrdo*, un autre avantage à ne pas négliger est l'étendue du spectre d'application de ces méthodes sérielles. Elles présentent toutefois un inconvénient majeur lié à leur faible robustesse. En effet, ces règles ne sont pas toujours très performantes lors de la variation de certains paramètres [5, 13, 27, 31]. Afin de pallier cette faiblesse, les ordonnancements fournis par ce module peuvent être optimisés par le module décrit ci-dessous.

Module d'optimisation : Celui-ci peut prendre plusieurs formes. Il est réduit à un algorithme de tri (selon les critères optimisés) dans le cas où seules les méthodes sérielles sont utilisées. Il est remplacé par une méthode d'optimisation (méta-heuristique ou méthode exacte) dans le cas général.

Module de performance : Ce module est capable de générer de nombreuses statistiques sous forme de rapports, courbes d'évolution, diagrammes de Gantt... Il peut établir un lien vers des outils externes (tableurs...).

Traducteur de bases de données : Ce traducteur assure les échanges d'informations entre le système d'informations de l'entreprise et la base de données de *PlanOrdo*.

Générateur de code RAO : Ce générateur traduit divers paramètres du modèle de simulation en langage RAO.

Interfaces : Plusieurs modules gèrent l'interfaçage entre les composants de la plate-forme et les outils externes.

Accessible au travers du module d'ordonnancement, le modèle de simulation de *PlanOrdo* est composé d'un ensemble de *patterns* génériques, éventuellement complété d'un petit nombre de *patterns* dédiés au problème étudié. Ces derniers peuvent être facilement générés à partir de notre modélisation graphique ALIX [17, 22].

Après avoir défini l'objectif et la constitution de notre plate-forme, la section suivante décrit le problème industriel relatif à l'entreprise Fontainunion.

3. PROBLEME INDUSTRIEL CONSIDERE

La validation de notre démarche est effectuée sur le problème d'ordonnancement du système de production de l'entreprise Fontainunion. Ce dernier est apparenté à un jobshop généralisé composé de dix-sept postes de travail divisés en sept groupes (cf. figure 2).

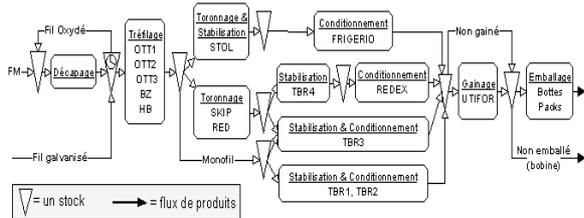


Figure 2. Processus de fabrication « Fontainunion »

Dans une organisation de type jobshop, chaque produit passe sur les postes selon un chemin prédéfini qui lui est propre. Dans un jobshop simple [5, 18], à chaque produit correspond exactement un chemin et chaque étape du processus ne peut être effectuée que sur un seul poste. Un jobshop généralisé est un jobshop dans lequel au moins une étape du processus peut être réalisée sur plus d'un poste et/ou au moins un produit peut emprunter plus d'un chemin. Le processus de production comporte sept groupes de machines, associés aux étapes du processus de fabrication : décapage, tréfilage, toronnage, stabilisation, conditionnement, gainage, emballage. Dans le processus étudié, les opérations continues et discrètes s'enchaînent via (respectivement) des opérations de soudure ou de découpe en vue du stockage sur des bobines. Ainsi le tréfilage, le toronnage et la stabilisation correspondent à des sous-systèmes continus. Toutefois, la partie continue du problème est facilement modélisable à l'aide d'équations mathématiques simples. De ce fait, le simulateur RAO [2] est utilisé pour modéliser la totalité du problème.

La production est divisée en quatre grandes familles : Monofils, petits, moyens et gros torons (câbles d'acier constitués de 2, 3 ou 7 fils). Ces produits sont obtenus à partir de fils machine (FM) ou de fils galvanisés (cf. figure 2). Chaque produit est le résultat d'une séquence d'opérations ordonnancées en respectant les contraintes liées aux ressources et aux relations de précedence. Les contraintes liées aux ressources incluent des limitations au niveau des entrées/sorties, des temps de préparation (notés setups), et de préemption (une opération ne peut être stoppée pour en démarrer une autre). Les relations de précedence imposent qu'une opération ne peut commencer avant la fin de toutes les opérations précédentes, identifiées dans la gamme opératoire de chaque produit. Pour le processus industriel considéré, l'hétérogénéité des machines implique que le temps opératoire dépend à la fois du produit et de la machine

utilisée pour réaliser chaque opération. Enfin, la production de l'entreprise cible est de type « fabrication sur commande ». Il faut tenir compte des fortes variations des commandes d'un mois à l'autre. Dans ce processus, différents types de setups doivent être pris en considération (dépendants et indépendants de la séquence de réalisation des produits sur les postes, proportionnels au tonnage réalisé). 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 utilisées pour supporter les produits semi-finis), les contraintes sur les diamètres maximum/minimum et les charges maximales 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 conditionnements. Mentionnons également qu'un sous-ensemble de commandes ne peut pas être en retard (notamment à cause des moyens de transport utilisés pour certaines d'entre-elles). Le modèle réalisé doit également reproduire et/ou anticiper, dans la mesure du possible, les dysfonctionnements du processus telles que les ruptures de fils. Les problèmes liés au stockage et au transport au sein de l'entreprise ne sont pas pris en compte dans notre étude car ils n'ont aucun impact significatif sur le processus de fabrication étudié.

Outre son aspect multidisciplinaire (modélisation, optimisation, simulation, performance), l'une des originalités de nos travaux réside dans la prise en compte des ressources auxiliaires. Dans le cas de l'entreprise Fontainunion, ces ressources sont exprimées en terme de bobines. De ce fait, lorsque la flexibilité des lignes est mise en exergue, la stratégie doit garder des bobines vides en réserve afin d'autoriser l'insertion de commandes imprévues/urgentes.

Les « meilleurs » ordonnancements doivent satisfaire un certain nombre de contraintes [9, 10] parmi lesquelles :

- Maximiser le tonnage produit (tonnes par heure) ;
- Considérer les niveaux de priorités de commandes ;
- Respecter les délais (dates d'exigibilité) ;
- Minimiser les temps d'attente, le *makespan* ;
- Equilibrer la charge des lignes ;
- Minimiser les temps de préparation (setups) ;
- Maximiser la flexibilité (garder des bobines vides afin de parer aux commandes imprévues) ;
- Minimiser l'utilisation d'intérimaires (et les heures supplémentaires les week-ends) ;
- Minimiser les chutes.

La prise en compte simultanée de l'ensemble de ces contraintes nous a confortés dans notre choix de la simulation pour modéliser le comportement du processus de fabrication, au détriment de l'utilisation d'un modèle mathématique résolu par une méthode exacte. Il semble, de plus, difficile de résoudre ce problème sans passer par

une approche hybride combinant la simulation à une ou plusieurs méthode(s) de résolution (exacte ou approchée).

La version actuelle du modèle de simulation prend directement en compte les quatre premières contraintes. Les autres sont considérées de manière indirecte via une sélection, du meilleur compromis entre les critères optimisés, parmi les stratégies d'ordonnancement engendrées par la plate-forme. Le tableau 1 introduit les notations utilisées.

Notation	Description
n	Nombre de commandes
d_j	Exigibilité de la commande j
C_j	Date de fin de la commande j
$E_j = \max(0, d_j - C_j)$	Avance de la commande j
$T_j = \max(0, C_j - d_j)$	Retard de la commande j
N_T	Nombre de commandes en retard
$\bar{E} = \frac{1}{n} \sum_j E_j$	Avance moyenne
$\bar{T} = \frac{1}{n} \sum_j T_j$	Retard moyen
$C_{\max} = \max_j C_j$	Date de fin de l'ensemble des commandes (<i>makespan</i>)

Tableau 1. Notations associées aux critères optimisés

La qualité d'une solution est mesurée par :

1. le tonnage produit (en tonnes par heure) ;
2. le nombre de commandes en retard N_T ;
3. le retard moyen \bar{T} ;
4. le *makespan* C_{\max} ;
5. l'avance moyenne \bar{E} .

L'objectif global consiste à minimiser les critères 2 à 4, et à maximiser les critères 1 et 5. Or, le modèle de simulation réalise toujours l'ensemble des commandes sélectionnées. De ce fait, le critère 1 est lié au *makespan* : réduire le *makespan* augmente le tonnage (*i.e.* le poids réalisé divisé par le nombre total de périodes de temps). Nous maximisons l'avance moyenne des commandes afin de pallier les retards imputables aux dysfonctionnements inhérents au procédé de fabrication. Ces critères ont été arbitrairement choisis en guise d'exemple, mais PlanOrdo n'est pas limité à ces derniers. Dans sa version actuelle, PlanOrdo effectue un classement des stratégies d'ordonnancement à l'aide d'un tri étagé. Son fonctionnement est le suivant :

- un ordre est défini entre les critères (du plus important au moins important) ;
- les stratégies sont triées en fonction du premier critère (noté C1) ;
- les stratégies de valeurs identiques pour C1 sont triées en fonction du second critère (C2) ;
- les stratégies de même valeur pour C1 et C2 sont triées en fonction du troisième critère (C3) ; etc.

La section suivante présente les résultats expérimentaux relatifs à l'application de la plate-forme PlanOrdo au système de production de l'entreprise Fontainunion.

4. RESULTATS EXPERIMENTAUX

Afin de respecter les clauses de confidentialité relatives à ce projet, les comparaisons directes entre la production réelle et la production simulée ne sont pas présentées. Des données factices peuvent être utilisées en lieu et place des données réelles en s'assurant toutefois que les ordres de grandeur restent cohérents.

L'objectif de notre modèle hybride est d'ordonner de façon précise un mois de production. Sur cette période, la durée d'exécution de notre modèle de simulation varie de trois à six secondes (sur un Pentium IV 1.5GHz) en fonction du nombre de commandes et des statistiques calculées en cours de simulation.

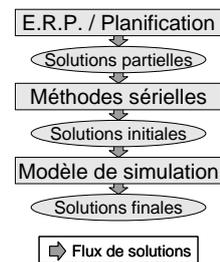


Figure 3. Modèle hybride 1

En combinant des versions « étendues » de méthodes sérielles élémentaires telles que SPT (Shortest Processing Time) et LPT (Largest Processing Time), nous obtenons un ensemble de méthodes applicables au problème traité. Chaque méthode sérielle fournit une solution initiale sous forme d'un vecteur de priorités, associant une priorité à chaque commande. Le modèle de simulation utilisé accepte en entrée un vecteur de priorités et fournit en retour une solution « finale » (*cf.* figure 3). Cette solution est constituée d'une liste de commandes ordonnancées sur les postes de travail et d'un ensemble de mesures de performance. Afin de réduire le volume des résultats présentés, nous avons retenu cinq méthodes sérielles, représentatives de l'ensemble des résultats obtenus : D+F+HP, D+F+SPT, D+F+LPT, SPT+F+D, LPT+F+D. Dans les notations utilisées ci-dessus « D » représente la date de lancement des commandes et « F » la date d'exigibilité. La notation C1+C2+C3 signifie que les commandes sont d'abord triées en fonction du critère C1, puis du critère C2, puis du critère C3, en utilisant un tri étagé. La règle D+F+HP est particulière car elle utilise les informations données par la personne responsable de l'ordonnancement (Human Planner).

Dans la suite de cet article, les méthodes sérielles, éventuellement suivies d'une optimisation, sont appelées stratégies d'ordonnancement. Leurs résultats ont été comparés aux choix des décideurs. Dans chaque cas, PlanOrdo génère des résultats similaires ou meilleurs que les résultats obtenus par la personne chargée d'effectuer l'ordonnancement (sur les critères préalablement définis). De plus, les calculs effectués par PlanOrdo facilitent grandement le travail des décideurs

et fournissent très rapidement un panel de stratégies. Il est également possible de déterminer à tout instant si une nouvelle commande peut être introduite dans le système, sans trop perturber la réalisation des autres commandes.

Les paragraphes suivants présentent un exemple d'utilisation de PlanOrdo où 134 commandes sont simulées sur une période d'un mois. Le tableau 2 présente une comparaison des stratégies d'ordonnancement sur la période sélectionnée. Celles-ci sont classées selon les quatre critères présentés en section 3, via un tri étagé (du critère le plus important au critère le moins important) :

1. Minimiser le nombre de commandes en retard N_T ;
2. Minimiser le retard moyen \bar{T} ;
3. Minimiser le *makespan* C_{max} ;
4. Maximiser l'avance moyenne \bar{E} .

N°	Méthode sérielle	N_T	\bar{T} (H)	C_{max} (H)	\bar{E} (H)
0	D+F+HP	0	0	753,9	271,5
1	D+F+SPT	0	0	757,71	277,4
2	D+F+LPT	0	0	760,17	256,84
3	SPT+F+D	2	17,65	749,77	287,42
4	LPT+F+D	6	10,55	756,68	266,57

Tableau 2. Comparaison de quelques méthodes sérielles

Sur la période sélectionnée, pour les critères retenus, la meilleure stratégie d'ordonnancement est D+F+HP. Ce résultat montre que la règle s'inspirant de la technique utilisée par la personne chargée de l'ordonnancement surpasse les autres règles élémentaires.

Afin d'améliorer ces résultats, nous avons implanté une seconde méthode hybride incluant une méta-heuristique simple.

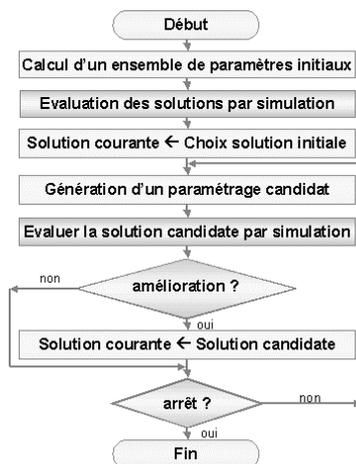


Figure 4. Modèle hybride 2

Au sein de la seconde méthode hybride (notée hybride 2), les stratégies d'ordonnancement fournissent les paramètres initiaux sous forme de vecteurs de priorités. Le modèle de simulation se base sur ce vecteur de

priorités pour fournir en retour un ensemble de mesures de performance agrégées au sein d'une fonction coût. Le principe de fonctionnement de la méta-heuristique utilisée est présenté en figure 4, il repose sur une méthode itérative d'amélioration d'une solution unique. A partir d'une solution courante (*i.e.* un vecteur de priorités utilisé pour « paramétrer » le modèle de simulation), une solution candidate est engendrée via un « opérateur de déplacement ». Si la solution candidate possède un meilleur coût, par rapport à la solution courante, alors la solution candidate devient la solution courante. Le cycle décrit ci-dessus est appelé « itération ». Il est répété jusqu'à une condition d'arrêt prédéfinie (en général 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. Nous avons testé différents opérateurs de déplacement applicables aux solutions encodées sous forme de vecteurs de priorités. Ces opérateurs modifient de manière plus ou moins importante le vecteur de priorités relatif à la solution courante pour engendrer un nouveau vecteur de priorités relatif à la solution candidate. En guise d'exemple, l'opérateur « swap » utilisé dans le cadre des résultats présentés figure parmi les opérateurs testés : il consiste à échanger deux éléments du vecteur initial pour obtenir un nouveau vecteur. L'avantage principal de cet opérateur est qu'il ne nécessite aucune information sur le processus en cours d'optimisation. De ce fait, cet opérateur « générique » est applicable à une très large gamme de problèmes d'ordonnancement. Il correspond aux objectifs de l'étude présentée où nous évaluons les performances de modules « génériques » assemblés au sein de notre plate-forme. Il est cependant possible d'améliorer les performances de cet opérateur en tenant compte des spécificités du processus ordonnancé.

Notre méta-heuristique fait partie de la classe des algorithmes itératifs de recherche locale. Elle améliore le jeu de paramètres (*i.e.* le vecteur de priorités) du modèle de simulation du système de production par évaluations successives. Elle se base sur une fonction coût constituée d'une agrégation des mesures de performance du système. Une « solution » au sens de la méta-heuristique est caractérisée par le vecteur de priorités et la valeur de la fonction coût associée à ce vecteur. Dans le cas particulier de l'hybridation entre optimisation et simulation mise en œuvre, l'évaluation d'une solution nécessite un appel systématique au modèle de simulation pour obtenir la valeur des mesures de performance agrégées au sein de la fonction coût. En effet, en l'absence d'un module multicritère dans la version actuelle de PlanOrdo, nous utilisons une fonction coût étagée [7], basée sur une somme pondérée des critères optimisés pour guider notre seconde méthode hybride dans l'espace de recherche. Les résultats obtenus après 4000 itérations sont présentés dans le tableau 3.

Hybride 2	Stratégie/solution finale				Gain après 4000 itérations				
	N _T	\bar{T} (H)	C _{max} (H)	\bar{E} (H)	N _T	\bar{T} (H)	C _{max} (H)	\bar{E} (H)	
3	moyenne	0	0	741,1	279,84	2	17,65	8,71	-7,58
	ecart-type	0	0	2,509	2,054				
1	moyenne	0	0	742,9	280,56	0	0	14,84	3,16
	ecart-type	0	0	2,791	1,374				
2	moyenne	0	0	744,1	277,06	0	0	16,09	20,23
	ecart-type	0	0	1,359	2,379				
0	moyenne	0	0	745	278,57	0	0	8,99	7,03
	ecart-type	0	0	1,344	1,338				
4	moyenne	0	0	744,6	275,47	6	10,55	12,04	8,89
	ecart-type	0	0	2,294	2,574				

Tableau 3. Résultats de l'hybride 2 après 4000 itérations

Etant donné le caractère stochastique de l'algorithme itératif de recherche locale utilisé, nous avons lancé 20 fois l'hybride à partir de chacune des solutions initiales (obtenues via les méthodes sérielles présentées ci-dessus). De ce fait, nous obtenons une valeur moyenne représentative du comportement de la méthode hybride et nous indiquons la valeur de l'écart-type calculée sur chaque groupe de vingt exécutions (cf. Tableau 3). Pour chaque critère, nous calculons également le gain moyen entre la solution initiale et la moyenne des solutions finales obtenues après 4000 itérations (i.e. 4000 évaluations). Le tableau 3 met en évidence deux propriétés importantes de la méthode hybride : d'une part, à l'issue de n'importe quelle exécutions de 4000 itérations, il n'existe plus aucune commande en retard (les colonnes N_T et \bar{T} sont à 0) ; d'autre part, quelle que soit la stratégie initiale utilisée, nous parvenons en moyenne à réduire le makespan d'au moins huit heures, ce qui conduit à « libérer » huit heures de travail pour une équipe de 11 ouvriers sur le mois simulé. Ce type de résultat a été fréquemment obtenu lors de l'utilisation de la plate-forme, bien qu'il existe des périodes extrêmement chargées (en terme de nombre de commandes lancées par unité de temps) pour lesquelles il est difficile – à la fois pour le responsable de production et pour l'hybride – d'optimiser les différents critères considérés.

Toutefois, une analyse approfondie des résultats de ce second hybride requiert la mise en vis-à-vis des gains réalisés et de la durée d'exécution nécessaire à leur obtention. En effet, si nous considérons une durée de simulation de trois secondes par itération de l'hybride, il faut trois heures et vingt minutes par exécution, soit deux jours, dix-huit heures et quarante minutes pour vingt exécutions. En réalité, l'étude de l'évolution des critères optimisés en fonction du nombre d'itérations montre que l'accroissement du gain devient de plus en plus faible à mesure que le nombre d'itérations augmente (cf. tableau 4). Ceci traduit la difficulté croissante d'améliorer la solution courante. Il ne faut cependant pas oublier que dans le cas général, seules les méthodes sérielles sont utilisées. L'hybride 2 n'est utilisé que dans le cas où le meilleur résultat fourni par les méthodes sérielles est jugé insuffisant par le responsable de la production. De ce fait, la durée d'exécution de l'hybride 2 devient moins significative, car il n'est invoqué que pour les « cas difficiles » et non de manière systématique. De plus, il

fournit une solution à chaque itération, ce qui fait qu'il peut être interrompu à tout instant si le gain est jugé suffisant ou si l'amélioration du gain devient inférieur à un seuil préalablement fixé. De même, il n'est pas nécessaire de lancer systématiquement vingt exécutions pour chaque stratégie. En général quelques unes suffisent pour obtenir une solution de bonne qualité. Le fait de lancer un ensemble d'exécutions ne se justifie que lorsque nous nous intéressons aux résultats moyens de l'hybride afin d'estimer sa robustesse par exemple. Enfin, mentionnons que l'évolution des performances des ordinateurs tend à réduire considérablement le temps d'exécution des programmes. Ainsi, nous avons dernièrement effectué des tests sur un Pentium IV 3.2GHz. Sans surprise, notre hybride s'en trouve accéléré par un facteur supérieur à deux. Il est également possible d'accélérer l'obtention des résultats issus de l'hybride en lançant en parallèle plusieurs exécutions sur un ordinateur multiprocesseur par exemple.

Hybride2	Ecart-type				Gain moyen				
	N _T	\bar{T} (H)	C _{max} (H)	\bar{E} (H)	N _T	\bar{T} (H)	C _{max} (H)	\bar{E} (H)	
Itération									
méthode 0	20	0	0	1,955	1,50	0	0	1,58	0,40
	40	0	0	2,358	1,36	0	0	1,93	0,80
	60	0	0	2,791	1,80	0	0	2,90	1,03
	80	0	0	2,558	1,89	0	0	3,23	1,15
	100	0	0	2,327	2,29	0	0	3,68	1,20
	200	0	0	2,015	2,28	0	0	4,62	2,09
	500	0	0	1,85	2,21	0	0	5,60	3,43
	1000	0	0	1,843	2,05	0	0	6,52	4,76
	2000	0	0	1,891	1,87	0	0	7,17	6,03
	4000	0	0	1,344	1,34	0	0	8,99	7,03
méthode 1	20	0	0	0,524	0,67	0	0	0,12	0,59
	40	0	0	1,348	0,77	0	0	0,62	0,79
	60	0	0	1,456	0,85	0	0	0,96	0,88
	80	0	0	3,681	0,88	0	0	2,17	1,47
	100	0	0	4,974	0,98	0	0	3,48	1,69
	200	0	0	5,872	1,78	0	0	6,42	1,89
	500	0	0	4,705	1,47	0	0	11,04	2,06
	1000	0	0	3,128	1,52	0	0	12,93	2,16
	2000	0	0	3,00	1,56	0	0	14,19	2,41
	4000	0	0	2,791	1,37	0	0	14,84	3,16
méthode 2	20	0	0	3,677	0,91	0	0	5,15	1,08
	40	0	0	3,657	1,30	0	0	7,03	1,69
	60	0	0	3,792	1,41	0	0	8,37	2,13
	80	0	0	3,63	1,63	0	0	9,27	2,79
	100	0	0	3,907	2,62	0	0	9,52	3,67
	200	0	0	2,335	2,78	0	0	12,09	5,84
	500	0	0	1,842	2,98	0	0	13,81	12,36
	1000	0	0	1,615	2,33	0	0	14,75	16,45
	2000	0	0	1,529	2,54	0	0	15,46	18,83
	4000	0	0	1,359	2,38	0	0	16,09	20,23
méthode 3	20	0,41	8,51	5,51	2,66	0,20	-3,46	0,42	-0,93
	40	0,47	7,39	3,55	2,48	0,30	-2,59	1,76	-1,13
	60	0,47	7,10	3,51	2,85	0,30	-1,90	1,95	-1,22
	80	0,49	5,72	1,96	3,13	0,35	-1,66	2,60	-1,54
	100	0,51	5,28	1,77	3,37	0,45	-1,62	2,90	-2,18
	200	0,57	7,06	3,77	3,52	0,70	1,27	1,85	-3,54
	500	0,39	12,73	5,44	3,16	1,05	6,69	1,52	-5,30
	1000	0,47	8,21	4,22	2,83	1,30	11,78	4,05	-6,43
	2000	0,44	2,50	3,22	2,48	1,75	16,45	6,69	-7,56
	4000	0,00	0,00	2,51	2,05	2,00	17,65	8,71	-7,58
méthode 4	20	1,00	5,35	1,76	1,76	2,50	0,13	7,96	-3,24
	40	1,08	8,49	2,42	1,70	3,30	-2,57	6,92	-3,63
	60	0,94	8,35	3,94	2,38	3,60	-1,31	6,17	-3,59
	80	0,89	7,77	4,22	2,12	3,80	-0,30	5,63	-4,06
	100	0,93	9,34	4,40	2,59	4,35	-0,91	5,22	-4,56
	200	0,76	11,96	2,79	3,19	5,05	-2,32	5,79	-3,54
	500	0,50	7,56	4,26	4,54	5,60	5,85	6,41	0,14
	1000	0,22	4,03	2,95	4,50	5,95	9,65	8,90	2,74
	2000	0,22	0,27	2,72	3,50	5,95	10,49	10,73	6,34
	4000	0,00	0,00	2,29	2,57	6,00	10,55	12,04	8,89

Tableau 4. Résultats détaillés de l'hybride 2 (issus des méthodes sérielles)

Ces résultats nous conduisent à envisager plusieurs pistes afin d'améliorer la robustesse de la plate-forme :

- L'accélération du modèle de simulation et l'ajout d'optimisations locales, afin de réduire la durée d'une itération et de rendre l'hybride plus performant en terme d'optimisation de l'outil de production.
- L'ajout de nouvelles méthodes sérielles, pour diversifier le panel des solutions fournies ;
- L'utilisation d'une méta-heuristique plus robuste (recherche Tabou, colonies de fourmis...) et d'un opérateur de déplacement « plus agressif » afin d'augmenter le gain par itération.

Avant de conclure, il est utile de rappeler que les classements présentés, relatifs à la période sélectionnée, ne sont pas généralisables directement à d'autres périodes, à cause de l'influence de divers paramètres sur les performances des méthodes sérielles (cf. section 2). Cependant, les résultats obtenus lors de l'étude de nos hybrides sur différentes périodes demeurent cohérents par rapport aux résultats engendrés par la personne chargée de l'ordonnancement de la production.

5. CONCLUSIONS ET PERSPECTIVES

Dans cet article, nous avons présenté une approche de simulation et d'optimisation d'un problème industriel concret afin d'en optimiser l'ordonnancement. Pour ce faire, le processus de fabrication a été décrit en insistant sur la prise en considération simultanée de plusieurs critères opérationnels et en mettant en évidence le caractère semi-continu du problème.

Les résultats obtenus sont cohérents par rapport aux résultats engendrés par la personne chargée de l'ordonnancement de la production. Les résultats sont consistants et montrent la robustesse de la plate-forme et des modèles sous-jacents. Résolument génériques, l'approche proposée et la plate-forme qui en résulte ne sont pas restreintes au « problème Fontainunion ». Les résultats présentés concernent l'utilisation de méthodes sérielles et d'une méta-heuristique élémentaire (sous la forme d'un algorithme itératif de recherche locale) pour optimiser les stratégies d'ordonnancement. Cependant, PlanOrdo n'est pas limitée à ces méthodes. La plate-forme est en effet prévue pour accueillir d'autres outils d'optimisation telles que des méta-heuristiques plus évoluées (recherche tabou, algorithmes génétiques, colonies de fourmis...) ou des méthodes exactes. Au-delà du calcul rapide d'un ensemble de solutions de bonne qualité (en comparaison des solutions engendrées par la personne chargée de l'ordonnancement), l'un des points forts de la plate-forme est de fournir simultanément un large panel de critères de comparaison. Il est envisagé à court terme d'y intégrer d'autres aspects tels que le caractère stochastique du processus simulé, la maintenance avec les notions de fiabilité et de risque (rupture de fil, pannes...), la gestion des coûts de production... Ainsi que nous le précisons dans [9], classer les stratégies selon plusieurs critères

n'est pas chose facile. De plus, nous avons fait l'hypothèse selon laquelle il était possible de classer les critères selon un ordre d'importance (afin d'effectuer le tri étagé). Dans la réalité, il est fréquent d'accorder la même importance à plusieurs critères. Dans ce cas, la mise en œuvre de méthodes multicritères telles que *Prométhée* [4] est nécessaire. Pour ce faire, le développement d'un module multicritère basé sur la méthode *Prométhée II* est en cours. Il permettra de transformer la plate-forme en un véritable outil d'aide à la décision multicritère. Il sera secondé d'un module chargé de la synthèse des indicateurs de performance sous forme de tableaux de bord.

REMERCIEMENTS

Ces travaux ont été effectués dans le cadre de projets financés par la Région Wallonne. Les auteurs remercient le personnel de l'Entreprise Fontainunion et les membres du CREGI.

REFERENCES

- [1] M. Aguillar, T. Rautert et A.J.G. Pater. Business Process Simulation : a fundamental step supporting process centered management in Proceedings of the 1999 Winter Simulation Conference, pp.1386. 1999.
- [2] A. Artiba, V. Emelyanov, and S. Iassinovski. Introduction to Intelligent Simulation: The RAO Language. *Kluwer Academic Publishers*. ISBN 0792381769, 1998.
- [3] G. Booch, J. Rumbaugh, and I. Jacobson. The Unified Modeling Language - User Guide. *New York: Addison Wesley*. ISBN 0201571684, 1998.
- [4] J.P. Brans et B. Mareschal, PROMÉTHÉE-GAIA: Une méthodologie d'aide à la décision en présence de critères multiples. *Éditions Ellipses*. ISBN: 2-729-81253-9, 2002.
- [5] J. Carlier and P. Chrétienne. Problèmes d'ordonnancement, modélisation, complexité, algorithmes. *Masson*. ISBN 2225812756, 1988.
- [6] C. Chu and J-M. Proth. L'ordonnancement et ses applications. *Masson*. ISBN: 2-225-85193-X, 1996.
- [7] D. Duvivier, P. Preux, E-G. Talbi, C. Fonlupt, et D. Robilliard. The fitness function and its impact on local search methods. Conf. IEEEESMC, 2478-2483, San Diego, *IEEE Press*. 1998.
- [8] D. Duvivier. Étude de l'hybridation des méta-heuristiques, Application à un problème d'ordonnancement de type jobshop. Thèse de Doctorat, Université du Littoral Côte d'Opale, Laboratoire d'Informatique du Littoral, Calais, France, 2000.
- [9] D. Duvivier, A. Artiba et V. Dhaevers. Simulation based performance measurement and analysis : an industrial application. Conf. CARS&FOF'2002, Porto, 2002. Accepté pour publication dans la revue IJCM.

- [10] D. Duvivier, V. Dhaevers, B. Bachelet, A. Artiba. Integreting simulation and optimization of Manufacturing systems. IEEE/SMC, part C: Applications and reviews, 33(2):186-192. ISSN 10946977, 2003.
- [11] P. Esquirol and P. Lopez. L'ordonnancement. *Economica*. ISBN: 2-7178-3798-1, 1999.
- [12] Groupe d'Ordonnancement Théorique et Appliqué (GOThA). Les problèmes d'ordonnancement. *Operations Research*, 27(1):77-150, 1993.
- [13] K.-W. Hansmann and M. Hoeck. Production control of a flexible manufacturing system in a job shop environment. *International Transactions in Operational Research*, 4(5/6):341-351. ISSN 09696016, 1997.
- [14] J.C. Hennet. Concepts et outils pour les systèmes de production. *Cépaduès-Éditions*. ISBN: 2-854-28437-2, 1997.
- [15] R. C. Hill David, Object-Oriented Analysis and Simulation. *Addison-Wesley Publishing Company*, New-York, 1996.
- [16] A. A. Hssain. Optimisation des Flux de Production - Méthodes et simulation. *Dunod*. 2000. ISBN: 2-100-05181-4.
- [17] S. Iassinovski, A. Artiba, V. Bachelet and F. Riane. Integration of simulation and optimization for solving complex decision making problems, *International Journal of Production Economics*, 85(1):3-10, 2003.
- [18] A. Jain and S. Meeran. Deterministic job-shop scheduling: past, present and future. *European Journal of Operational Research*, 113:390-434, 1999.
- [19] M.A. Jamali, O. Roux, D. Ait-Kadi, A. Artiba. An integrated approach using optimization methods and simulation in maintenance strategies analysis. 14th CIRP Design Seminar, Egypte, 2004.
- [20] L. Józefowska, M. Mika, R. Rózycki, G. Waligóra, and J. Weglarz. Local search metaheuristics for discrete-continuous scheduling problems. *European Journal of Operational Research*, 107:354-370, 1998.
- [21] B. Legros, V. Dhaevers, F. Riane et A. Artiba. Méthodologie d'évaluation de la performance opérationnelle d'un système de production au départ d'un diagnostic financier. Conf. CEMASI, 2000.
- [22] D. Pichel, A. Dontaine, S. Iassinovski, A. Artiba, C. Fagnart. ALIX : une méthode de modélisation des flux d'un système de production. Conf. CPI 2003, Maroc, 2003.
- [23] M. Pinedo. Scheduling - Theory, Algorithms, and Systems. *Prentice Hall*. ISBN 0137067577, 1995.
- [24] M. Pinedo and X. Chao. Operations Scheduling with applications in manufacturing and services. *Irwin/McGraw-Hill*. ISBN 0072897791, 1999.
- [25] C. Rajendran, and O. Holthaus. A Comparative Study of Dispatching Rules in Dynamic Flow Shops and Job Shops. *European Journal of Operational Research*, 116:156-170, 1999.
- [26] F. Riane, A. Artiba, S. Iassinovski. An integrated production planning and scheduling system for hybrid flowshop organizations. *Int. J. Production Economics*, 74 (2001) 33-48.
- [27] M.L. Smith and A. Seidmann. Due date selection procedures for job-shop simulation. *Computers and Industrial Engineering Journal*, 7(3):199-207, 1983.
- [28] F. Vernadat. Techniques de Modélisation en Entreprise: Applications aux Processus Opérationnels. *Economica*. ISBN 2717838538, 1999.
- [29] F. Vernadat and B. Zeigler, "New simulation requirements for model-based Enterprise Engineering" in IFAC/IEEE/INRIA 2nd Int. Conf. on Manufacturing Control and Production Logistics (MCPL'2000), Grenoble, 2000, CD-ROM.
- [30] J. Vlissides. Pattern Hatching - Design Patterns Applied. *New York: Addison-Wesley*. ISBN 0201432935, 1998.
- [31] A. M. Waikar, B. R. Sarker and A. M. Lal. A comparative study of some priority dispatching rules under different shop loads. *Production Planning Control*, 6(4):301-310. ISSN 09537287, 1995.
- [32] B. Zeigler, H. Praehofer and Tag Gon Kim, Theory of Modeling and Simulation. *Academic Press*, 2nd Edition, 2000.
- [33] G.W. Zobrist, and J.V. Leonard. Object-Oriented Simulation - Reusability, Adaptability, Maintainability. *IEEE Press*. ISBN 0780310616, 1997.