

Optimisation par colonie de fourmis pour un problème d'ordonnancement industriel avec temps de réglages dépendants de la séquence

Caroline GAGNÉ, Marc GRAVEL

Département d'informatique et de mathématique
Université du Québec à Chicoutimi
555, Boul. de l'Université
Chicoutimi, Québec, Canada
G7H 2B1
caroline_gagne@uqac.quebec.ca
marc_gravel@uqac.quebec.ca

Wilson L. PRICE

Faculté des Sciences de l'administration,
Université Laval
Ste-Foy, Québec, Canada
G1K 7P4

wilson.price@fsa.ulaval.ca

RÉSUMÉ : Ce document présente l'adaptation d'un OCF pour traiter un problème d'ordonnancement industriel multi-objectifs dans un centre de coulée d'aluminium. Une série d'éléments nouveaux ont été ajoutés à l'OCF de base pour en améliorer sa performance. La métaheuristique est utilisée pour la construction des séquences de commandes et un modèle de simulation sert à l'évaluation de chacune d'elles sur trois objectifs. Les résultats obtenus confirment la performance de l'algorithme développé et ces méthodes seront incorporées au logiciel déjà utilisé par l'entreprise.

MOTS-CLÉS: ordonnancement, métaheuristique, colonie-de-fourmis, multi-objectifs, réglages dépendants de la séquence.

1. INTRODUCTION

La complexité des problèmes industriels fait en sorte qu'il est souvent difficile d'utiliser des méthodes exactes de solution compte tenu des temps de calculs exigés et du manque de flexibilité des méthodes pour intégrer diverses contraintes spécifiques. Les métaheuristices constituent alors une stratégie de résolution de plus en plus privilégiée. L'objectif principal de cet article est de montrer l'efficacité de nouveaux éléments introduits à la métaheuristique « optimisation par colonies de fourmis » pour solutionner un problème réel d'ordonnancement. Les auteurs ont déjà démontré la performance de cette métaheuristique pour solutionner ce problème industriel. Le lecteur peut consulter (Gravel et al., 2000a), (Gravel et al., 2000b) pour plus amples détails concernant la transposition de la version de base OCF pour le problème du voyageur de commerce (Dorigo, 1992) à ce problème d'ordonnancement industriel. Cette méthode de solution a même été intégrée à un logiciel utilisé par le planificateur de l'entreprise concernée.

La situation industrielle est celle d'un centre de coulée d'une aluminerie où un carnet de n commandes de lingots de différents alliages doit être ordonné sur une machine à couler horizontale (voir figure 1). Il faut préciser que la fabrication successive de deux commandes d'alliages différents peut engendrer certaines opérations de drainage et de nettoyage des fours de préparation qui correspondent, en fait, à un temps de réglage d'une machine dans la littérature classique. Également, la fabrication successive de commandes d'alliages de dimensions différentes oblige le changement du moule et représente une deuxième

forme de réglage. Notons également la notion d'optimisation multi-objectifs où la méthode de résolution doit prendre en compte à la fois un objectif de minimisation de perte de capacité de l'usine, de minimisation des retards des commandes et d'utilisation optimale des modes de livraison des produits dans la recherche de solutions. Certaines contraintes technologiques relatives à la disponibilité d'équipements sont également présentes et complexifient la situation. La métaheuristique est utilisée pour la construction des séquences de commandes et un modèle de simulation sert à l'évaluation de chacune d'elles sur les trois objectifs. Le planificateur doit toutefois préciser l'ordre d'importance des objectifs et la recherche est alors dirigée selon un ordre lexicographique. Ce n'est, qu'en cas d'égalité sur l'objectif principal, que l'objectif suivant devient opérationnel dans la recherche de solutions et ainsi de suite.

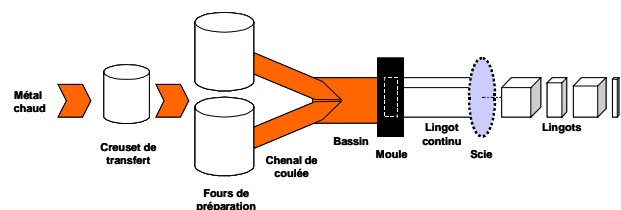


Figure 1. Le processus de coulée horizontale

De nouveaux éléments ont été intégrés à la métaheuristique et validés sur une série de problèmes classiques d'ordonnancement de n commandes sur une seule machine avec objectif unique et temps de réglages

dépendants de la séquence. Ceux-ci ont démontré leur efficacité et le lecteur peut consulter Gagné et al. (2000) pour plus de détails. Nous allons présenter dans cet article comment ces éléments doivent être adaptés au contexte industriel multi-objectifs et nous montrerons la performance supplémentaire obtenue en comparaison à la version présentement implantée dans l'usine.

2. REVUE DE LA DOCUMENTATION

Le problème d'ordonnement d'une machine unique sans la caractéristique des réglages dépendants de la séquence a retenu l'attention de nombreux chercheurs. À cet égard, Koulamas (1994) présente un travail de revue sur le problème de minimisation du retard total et répertorie un grand nombre d'approches de résolution. En ce qui concerne la complexité de ce problème, Du et Leung (1990) ont apporté la preuve que celui-ci est NP-difficile. L'ajout de la caractéristique des réglages dépendants de la séquence procure une dimension supplémentaire au problème qui a pour effet d'augmenter le niveau de complexité.

L'importance de traiter explicitement les réglages dans l'ordonnement a été soulignée à plusieurs reprises dans la documentation. Notamment, Wilbrecht et Prescott (1969) déclarent que la considération des réglages dépendants de la séquence est significative et ce, surtout dans le cas où la capacité de production est utilisée à son maximum ou près de celui-ci. De plus, Wortman (1992) souligne que la gestion efficace de la capacité de production exige la considération explicite des réglages.

D'un point de vue industriel, plusieurs contextes justifient la considération des opérations de réglages et le développement d'outils adaptés. De nombreux cas réels provenant de différents environnements de production exposent cette situation. Pinedo (1995) décrit la situation d'une usine de fabrication de sacs en papier où des réglages sont exigés entre la production de différents types de sacs; Das et al. (1995) et Franca et al. (1996) ont observé une situation semblable dans l'industrie du plastique; Conway et al. (1967) décrivent une situation dans l'imprimerie où les réglages consistent au nettoyage des machines lors de changements de couleurs d'impression. Les industries du textile, pharmaceutique, chimique, métallurgique représentent d'autres exemples de contextes réels où les réglages dépendants de la séquence sont fréquents.

Malgré la vaste documentation en ordonnancement, ce n'est qu'en 1999 qu'apparu deux synthèses sur les travaux traitant des réglages dépendants de la séquence (Allahverdi et al., 1999), (Yang et Liao, 1999). La première synthèse souligne, entre autres, le peu de travaux d'optimisation multi-objectifs et, la seconde, le peu d'applications à des situations réelles. En ce qui concerne l'utilisation de métaheuristiques pour solutionner de tels problèmes, il se limite aux travaux de

Ragatz (1993), Rubin et Ragatz (1995), Tan et Narasimhan (1997) et Tan et al. (2000).

3. OPTIMISATION PAR COLONIE DE FOURMIS

Cette métaheuristique (Colomi et al., 1991) (Dorigo, 1992) a été inspirée par les études sur le comportement des fourmis réelles (Deneubourg et al., 1983) (Deneubourg et Goss, 1989), (Goss et al., 1990). À l'origine, l'optimisation par colonie de fourmis a été conçue pour résoudre le problème du voyageur de commerce (VC). L'objectif de ce problème est de trouver la tournée la plus courte pour un ensemble de villes données. Une matrice fournissant les distances, d_{ij} , entre toutes les paires de villes est utilisée pour estimer la longueur d'une tournée.

Dans l'optique d'un problème de VC, chaque fourmi représente un agent (Dorigo et al., 1991) et lorsqu'il se déplace de la ville i à la ville j , il laisse une trace sur le chemin (ij). De plus, un agent choisit la prochaine ville à visiter à l'aide d'une probabilité $p_{ij}^k(t)$ basée sur un compromis entre l'intensité de la trace $\tau_{ij}(t)$ et la visibilité ($1/d_{ij}$) qui prend en considération la distance entre les villes. Les coefficients α et β sont des paramètres qui permettent de contrôler l'importance relative des deux éléments. Finalement, une fourmi k possède une forme de mémoire, tabou_k , lui rappelant la liste ordonnée des villes déjà visitées afin d'obliger celle-ci à former une solution admissible. Si le nombre total de fourmis est m et le nombre de villes à visiter est n , un cycle est réalisé lorsque chacune des m fourmis complètent une tournée des n villes.

Dorigo et Gambardella (1997) ont proposé certaines améliorations à la version de base (Dorigo, 1992) concernant la règle de transition, la mise à jour de la trace qui se fait globalement mais aussi localement, l'intégration de méthodes d'amélioration et l'utilisation d'une liste de candidats. Plusieurs autres travaux (Dorigo et al., 1991), (Dorigo et al., 1996), (Dorigo et Di Caro, 1999) procurent des explications supplémentaires sur le fonctionnement de l'algorithme, sur la notation et sur le choix des divers paramètres.

Dans Gagné et al. (2000), nous avons proposé et validé certaines modifications à l'algorithme de base d'OCF pour la résolution d'un problème d'ordonnement avec machine unique et temps de réglages dépendants de la séquence dans un objectif de minimisation du retard total. Ces modifications sont présentées à la figure 2. Entre autres, la matrice de distance (d_{ij}) est remplacée par plusieurs matrices qui tiennent compte individuellement de certains aspects importants du contexte d'ordonnement en regard de l'objectif à optimiser. À cet effet, on retrouve dans l'équation (1), \bar{s}_{ij} , qui représente le temps de réglages normalisés et, \bar{m}_{ij} , qui représente la marge normalisée de chacune des commandes. Ces deux éléments ont une incidence

directe sur le retard total. De plus, l'utilisation d'une liste de candidats dans la construction d'une solution limite le choix de la prochaine commande à celles qui sont les plus intéressantes en fonction de l'objectif recherché. Dans le contexte du problème d'ordonnement concerné, les cl commandes candidates sont sélectionnées en ordre croissant de leur marge. L'intégration d'une fonction de regard vers l'avant est une innovation importante proposée à l'OCF. Ce nouvel élément apparaît dans la règle de transition aux équations (1) et (2) et consiste à estimer la qualité de

la solution qui pourrait être obtenue suite à la sélection d'une commande donnée. Le terme \bar{B}_{ij} représente ainsi l'estimation de la qualité de solution normalisée. Finalement, la méthode d'échanges d'arcs successifs 3-Opt restreint (Johnson et McGeoch, 1997) a été choisie comme méthode d'amélioration pour le problème d'ordonnement avec réglages dépendants de la séquence. Cette méthode présente l'avantage de ne pas inverser l'ordre des commandes dans le branchement des arcs comme le fait la version courante du 3-Opt.

Étape 1: [Initialisation]

$t := 0; NC := 0;$ {t = compteur de temps; NC = Nombre de cycles}

Pour chaque arc (i,j) , initialiser l'intensité de la trace $\tau_{ij}(0) := \tau_0$

Étape 2: [Commande de départ]

Pour chaque fourmi k :

└ Initialiser la commande de départ et conserver cette information dans $Tabou_k$

Étape 3: [Construction d'une séquence pour chaque fourmi]

Pour i allant de 1 à n :

└ Pour k allant de 1 à m :

└ Choisir la prochaine commande $j, j \notin Tabou_k$, parmi les cl commandes candidates selon:

$$j = \begin{cases} \arg \max_{j \in Tabou_k} \{ [\tau_{it}(t)]^\alpha \cdot [\frac{1}{s_{it}}]^\beta \cdot [\frac{1}{m_{ij}}]^\delta \cdot [\frac{1}{\bar{B}_{ij}}]^\phi \} & \text{si } q \leq q_0 \\ J & \text{si } q > q_0 \end{cases} \quad (1)$$

où J est choisi selon la probabilité:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\frac{1}{s_{ij}}]^\beta \cdot [\frac{1}{m_{ij}}]^\delta \cdot [\frac{1}{\bar{B}_{ij}}]^\phi}{\sum_{\ell \notin Tabou_k} [\tau_{i\ell}(t)]^\alpha \cdot [\frac{1}{s_{i\ell}}]^\beta \cdot [\frac{1}{m_{i\ell}}]^\delta \cdot [\frac{1}{\bar{B}_{i\ell}}]^\phi} \quad (2)$$

└ Conserver la commande choisie dans $Tabou_k$

└ Mise à jour locale de l'intensité de la trace pour le chemin (i,j) sélectionné:

$$\tau_{ij}(t) := \rho \cdot \tau_{ij}(t) + (1-\rho) \cdot \Delta \tau_{ij} \quad \text{où } \Delta \tau_{ij} = \tau_0 \quad (3)$$

Étape 4: [Mise à jour globale de l'intensité de la trace]

└ Calculer le retard total obtenu, L_k , pour chaque fourmi k

└ Appliquer une méthode d'amélioration pour chaque fourmi k et recalculer L_k

└ Pour tous les arcs $(i,j) \in Cycle^*$, Mise à jour globale de la trace selon:

$$\tau_{ij}(t+1) := \rho_g \cdot \tau_{ij}(t) + (1-\rho_g) \cdot \Delta \tau_{ij} \quad \text{où } \Delta \tau_{ij} = \frac{1}{L_{Cycle^*}} \quad (4)$$

$t := t + 1; NC := NC + 1$

{Cycle* = meilleure solution du cycle}

Étape 5: [Enchaînement]

└ si $(NC < NC_{MAX})$ et (autres critères d'arrêt non respectés)

└ alors Vider les $Tabou_k$ et recommencer à l'étape #2

└ sinon Arrêt

Figure 2. Algorithme d'optimisation par colonie de fourmis pour un problème d'ordonnement avec machine unique

4. ÉLÉMENTS DE L'APPLICATION INDUSTRIELLE

L'aspect principal de l'application industrielle est la notion d'ordonnancement multi-objectifs. Ceux-ci sont la minimisation de la perte de capacité du centre de coulée, la minimisation du retard total de l'ensemble du carnet de commandes et la minimisation de la perte de capacité de transport. Pour le premier objectif, les réglages liés au changement d'alliages et ceux liés au changement des moules influencent directement celui-ci. Étant donné que la durée d'un réglage relatif à un changement d'alliage dépend de la quantité à drainer et de la vitesse de coulée de l'alliage, nous avons choisi de traiter cet aspect par l'ajout d'une pénalité dans la matrice s_{ij} lorsqu'un drainage est nécessaire. Pour sa part, le temps de changement de moule est constant et nous avons également choisi d'ajouter une pénalité dans la matrice s_{ij} lorsqu'un changement de moule est nécessaire entre deux commandes. La procédure de construction des matrices est présentée au tableau 1.

En ce qui a trait au deuxième objectif, la marge d'une commande a un impact sur le retard total et les éléments de réglages influencent également cet objectif car ils retardent la production. Ce dernier aspect sera tenu en compte par la valeur accordée à α .

Pour le troisième objectif, on doit chercher à regrouper ensemble les commandes d'une même destination en fonction d'un tonnage cible à respecter. Nous considérons alors la compatibilité des modes de transport dans la matrice c_{ij} .

Les équations (1) et (2) de la figure 2 deviennent (1') et (2') pour tenir compte de tous ces éléments.

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot \left[\frac{1}{s_{ij}}\right]^\beta \cdot \left[\frac{1}{m_{ij}}\right]^\delta \cdot \left[\frac{1}{c_{it}}\right]^\lambda \cdot \left[\frac{1}{B_{ij}}\right]^\phi}{\sum_{k \in \text{Tabouk}} [\tau_{ik}(t)]^\alpha \cdot \left[\frac{1}{s_{ik}}\right]^\beta \cdot \left[\frac{1}{m_{ik}}\right]^\delta \cdot \left[\frac{1}{c_{it}}\right]^\lambda \cdot \left[\frac{1}{B_{ik}}\right]^\phi} \quad (1')$$

$$j = \begin{cases} \arg \max_{k \in \text{Tabouk}} \left\{ [\tau_{ik}(t)]^\alpha \cdot \left[\frac{1}{s_{ik}}\right]^\beta \cdot \left[\frac{1}{m_{ik}}\right]^\delta \cdot \left[\frac{1}{c_{it}}\right]^\lambda \cdot \left[\frac{1}{B_{ik}}\right]^\phi \right\} & \text{si } q \leq q_0 \\ J & \text{si } q > q_0 \end{cases} \quad (2')$$

Nous devons également expliquer la fonction d'anticipation B_{ij} . Tout d'abord, celle-ci est calculée sur l'objectif principal identifié. Rappelons que c'est le gestionnaire qui définit l'ordre de priorité des objectifs et que la recherche de solutions est dirigée selon un ordre lexicographique de ceux-ci. La fonction

d'anticipation est calculée selon l'équation (5) par une évaluation aussi précise que possible de la partie complétée de la séquence (Q_h), incluant la commande j à la suite de la dernière commande i sélectionnée, et par une estimation de la partie non complétée de la séquence (Q'_h). La partie complétée n'est pas toujours évaluée de manière exacte par le modèle de simulation afin de ne pas trop pénaliser le temps de calcul étant donné la nécessité d'obtenir rapidement des solutions de qualité pour le gestionnaire. La partie non complétée de la solution est estimée par le calcul d'une borne inférieure qui utilise le principe décrit par Ragatz (1993) dans sa méthode de séparation et d'évaluation progressive. La borne inférieure a été définie dans l'équation (5) pour l'objectif de minimisation du retard total et une adaptation est nécessaire pour les objectifs de minimisation de la perte de capacité de l'usine et de transport.

$$B_{ij} = T_{Q_h} + T_{Q'_h} \quad (5)$$

Lorsque l'objectif principal est la minimisation de la perte de capacité, l'évaluation de Q_h représente uniquement le temps total de réglage lié au changement de dimension. Cette évaluation est simplifiée puisqu'elle ne tient pas compte du temps de réglage lié au changement d'alliages afin de réduire le temps de calcul exigé. En ce qui concerne l'estimation de la perte de capacité de la partie Q'_h , les commandes qui ne sont pas encore sélectionnées dans la solution servent à estimer, de façon optimiste, les réglages de changement de moules qui risquent d'être encourus.

Lorsque l'objectif principal est la minimisation du temps total de retard, la date de fin des commandes pour la partie complétée de la solution est calculée afin de déterminer le retard de chacune d'elles. La date de fin de production des commandes est calculée en fonction du temps de fabrication et du temps de réglage pour les changements de dimensions. Encore cette fois, les temps de réglages pour les changements d'alliages sont négligés. La partie non complétée de la solution est estimée selon la borne inférieure de Ragatz (1993).

Finalement, lorsque l'objectif principal est la minimisation de la perte de capacité en transport, la partie complétée de la solution est évaluée de manière exacte. En ce qui concerne les commandes qui ne sont pas encore sélectionnées, elles sont regroupées selon leur destination pour permettre une estimation optimiste de la perte de capacité en transport pour la partie de la solution non complétée.

Matrice s_{ij}

- (1) Initialisation de $s_{ij} = 1$ pour tous les ij .
- (2) Lorsqu'un drainage est requis entre l'alliage de la commande i et celui de la commande j , une pénalité de +2 est ajoutée à s_{ij} .
- (3) Lorsqu'un changement de moule est requis entre la commande i et j parce que les dimensions sont différentes, une pénalité de +2 est ajoutée à s_{ij} .

Matrice m_{ij}

- (1) Initialisation de $m_{ij} = 1$ pour tous les ij .
- (2) Pour encourager le respect des dates dues et ainsi minimiser le retard total, une pénalité correspond à deux fois le ratio $(\max(0, \text{marge de la commande}) / \text{maximum marge de toutes les commandes})$ est calculée et ajoutée à m_{ij} , où:

$$\text{marge de la commande} = (\text{date due}) - (\text{date de début de production}) - (\text{temps de production de la commande})$$

Matrice c_{ij}

- (1) Initialisation de $c_{ij} = 1$ pour tous les ij .
- (2) Si les modes de transport diffèrent entre les commandes i et j , une pénalité de +2 est ajoutée à c_{ij} .

Note : Une pénalité de +500 est ajoutée dans toutes ces matrices lorsque des contraintes technologiques empêchent la commande j de suivre la commande i .

Tableau 1. Coefficients des matrices

5. EXPÉRIENCES NUMÉRIQUES

Pour juger de la performance de l'algorithme présenté, huit carnets de commandes réelles de tailles différentes ont été testés. Dans Gravel et al. (2000b), ces mêmes problèmes ont été résolus à partir d'une version de base de l'optimisation par colonie de fourmis dans laquelle une seule matrice d_{ij} incorporait tous les éléments spécifiques du contexte industriel (s_{ij} , m_{ij} et c_{ij}). Dans cette version, il est possible de contrôler l'importance relative de chacun de ces éléments par les valeurs accordées aux paramètres β , δ , λ dans la construction des solutions. Ces paramètres prendront donc des valeurs différentes l'objectif principal optimisé. Les données numériques des problèmes sont disponibles des auteurs sur demande. Notez que les objectifs de minimisation de la perte de capacité et de retard total sont mesurés en jours tandis que la perte de capacité de transport est mesurée en tonnes métriques.

Le tableau 2 (a, b, c) présente les résultats obtenus lorsque les objectifs sont, tour à tour, désignés comme objectif principal. Au tableau 2 (a), en considérant la perte de capacité comme objectif principal, on retrouve, dans la première partie du tableau, les résultats obtenus par l'algorithme de base (Gravel et al., 2000b) et, dans la seconde partie, les résultats obtenus avec l'algorithme décrit dans cet article. Chaque problème a été résolu à dix reprises par les deux versions de l'algorithme et nous présentons le résultat moyen et l'écart-type obtenu. On peut constater que la nouvelle version de l'algorithme est

supérieure ou égale pour tous les problèmes testés en ce qui a trait à l'objectif principal. Comme les objectifs sont contradictoires, l'amélioration de la solution sur le premier objectif entraîne quelquefois une baisse de performance sur les autres objectifs. On note toutefois que, pour cet objectif, les gains de performance sont minimes mais, dans l'ensemble, les résultats ont beaucoup moins de variabilité. Les paramètres α , β , δ , λ et ϕ , qui accordent une importance aux différents éléments dans la construction des solutions, sont fixés à (1,4,2,1,2) et ont été choisis après une série d'essais numériques. On note qu'une plus grande importance est accordée à la matrice s_{ij} en accordant la valeur 4 à β car les différentes formes de réglage ont une grande incidence sur l'objectif principal.

Au tableau 2 (b), en considérant le retard total comme objectif principal, on observe une meilleure performance du nouvel algorithme pour tous les problèmes. Dans ce cas, les améliorations sont plus substantielles sur le critère principal et atteignent près de 75% pour les problèmes de 50 et 60 commandes. On observe généralement une perte de performance sur le deuxième objectif, la perte de capacité, avec l'amélioration de l'objectif principal. Les paramètres α , β , δ , λ et ϕ ont fixés à (1,2,3,1,2) pour les problèmes de 40 commandes et moins et à (1,5,20,1,2) pour les autres problèmes. Notons encore une fois que la plus grande importance ($\delta = 3$) est accordée à la matrice en relation directe avec l'objectif principal. On doit également souligner que les réglages peuvent avoir une influence sur l'objectif principal, ce qui explique l'importance accordée au paramètre β . Dans le cas de

problèmes de grande taille, nous devons augmenter l'importance accordée aux matrices s_{ij} et m_{ij} pour diriger davantage la méthode de recherche vers

l'objectif principal et ainsi obtenir une bonne performance.

Taille du problème	OCF implanté						OCF nouvelle version					
	Capacité		Retard		Transport		Capacité		Retard		Transport	
10	1.64	(0.00)	6.97	(0.00)	12.71	(0.00)	1.64	(0.00)	6.97	(0.00)	12.71	(0.00)
20	3.69	(0.00)	53.49	(5.30)	0	(0.00)	3.69	(0.00)	47.46	(1.72)	1.73	(3.46)
30	3.69	(0.00)	97.26	(11.11)	0.43	(0.00)	3.69	(0.00)	83.22	(4.30)	0.43	(0.00)
40	3.80	(0.00)	100.52	(11.99)	14.83	(7.64)	3.69	(0.00)	133.80	(4.68)	17.75	(0.00)
50	4.85	(0.00)	183.12	(12.07)	7.31	(8.80)	4.75	(0.00)	204.75	(9.02)	54.20	(15.00)
60	4.90	(0.08)	279.52	(53.20)	12.53	(6.67)	4.75	(0.00)	291.91	(5.18)	43.54	(24.93)
70	4.93	(0.08)	441.76	(42.29)	13.64	(13.01)	4.75	(0.00)	426.72	(15.96)	30.44	(16.93)
80	4.88	(0.06)	674.37	(131.13)	5.34	(8.72)	4.75	(0.00)	643.20	(25.36)	15.45	(11.25)

Tableau 2 (a). Résultats comparatifs des deux versions de OCF lorsque la "perte de capacité" est l'objectif principal. Le premier résultat de chacune des colonnes correspond à la moyenne de dix essais et le second résultat est l'écart-type

Taille du problème	OCF implanté						OCF nouvelle version					
	Capacité		Retard		Transport		Capacité		Retard		Transport	
10	1.93	(0.36)	6.88	(0.11)	11.55	(1.78)	2.20	(0.38)	6.81	(0.10)	10.70	(2.01)
20	3.77	(0.00)	17.86	(0.64)	0.00	(0.00)	3.77	(0.00)	16.70	(0.19)	0.00	(0.00)
30	4.40	(0.27)	38.62	(2.95)	7.84	(7.58)	4.51	(0.11)	29.42	(0.91)	7.18	(6.89)
40	4.48	(0.45)	75.15	(4.93)	33.65	(7.47)	5.11	(0.44)	61.89	(3.25)	27.59	(16.67)
50	7.13	(0.43)	88.68	(8.96)	11.28	(15.15)	7.86	(0.22)	21.33	(0.77)	56.65	(5.73)
60	8.54	(0.65)	142.86	(7.68)	17.91	(12.99)	8.82	(0.29)	36.34	(1.70)	62.80	(11.52)
70	8.37	(0.89)	277.29	(17.35)	11.99	(13.78)	9.05	(0.21)	165.12	(3.18)	29.20	(11.92)
80	8.81	(0.50)	384.39	(17.63)	8.41	(8.75)	9.22	(0.33)	258.75	(8.42)	25.97	(14.46)

Tableau 2 (b). Résultats comparatifs des deux versions de OCF lorsque le "retard" est l'objectif principal. Le premier résultat de chacune des colonnes correspond à la moyenne de dix essais et le second résultat est l'écart-type

Taille du problème	OCF implanté				OCF nouvelle version							
	Capacité		Retard		Transport		Capacité		Retard		Transport	
10	1.64	(0.00)	15.03	(0.00)	0.00	(0.00)	1.64	(0.00)	15.03	(0.00)	0.00	(0.00)
20	3.69	(0.00)	54.79	(9.70)	0.00	(0.00)	3.69	(0.00)	50.84	(4.38)	0.00	(0.00)
30	3.77	(0.00)	67.23	(9.00)	0.00	(0.00)	3.77	(0.00)	51.75	(3.48)	0.00	(0.00)
40	3.88	(0.12)	113.71	(23.71)	0.00	(0.00)	3.80	(0.00)	91.49	(14.08)	0.00	(0.00)
50	4.94	(0.11)	203.23	(42.19)	0.00	(0.00)	4.75	(0.00)	220.04	(25.06)	0.00	(0.00)
60	4.97	(0.12)	306.26	(39.58)	0.00	(0.00)	4.75	(0.00)	331.76	(27.72)	0.00	(0.00)
70	4.98	(0.11)	529.69	(82.17)	0.00	(0.00)	4.75	(0.00)	452.70	(20.70)	0.00	(0.00)
80	4.89	(0.08)	759.12	(111.21)	0.00	(0.00)	4.75	(0.00)	686.41	(49.01)	0.00	(0.00)

Tableau 2 (c). Résultats comparatifs des deux versions de OCF lorsque la "perte de capacité de transport" est l'objectif principal. Le premier résultat de chacune des colonnes correspond à la moyenne de dix essais et le second résultat est l'écart-type

Au tableau 2 (c), on peut constater que l'objectif de minimisation de la perte de capacité de transport est facile à optimiser pour les deux versions de l'algorithme. Alors, en accordant une plus grande importance au deuxième objectif, la perte de capacité, dans le choix des paramètres, il est possible d'améliorer celui-ci lorsque la taille des problèmes dépasse 30 commandes. On peut conclure que la nouvelle version de l'algorithme couvre mieux l'espace de solutions pour les gros problèmes. Les paramètres α , β , δ , λ et ϕ sont fixés à (1,4,2,2,2).

6. CONCLUSIONS

Dans cet article, nous avons montré comment transposer la métaheuristique d'optimisation par colonie de fourmis à un problème industriel multi-objectifs. Les résultats obtenus montrent que les éléments intégrés à l'OCF tels la liste de commandes candidates, la fonction d'anticipation, la méthode d'amélioration et l'utilisation de plusieurs matrices de distance contribuent grandement à l'amélioration des solutions.

Dans la version initiale (Gravel et al., 2000b), l'utilisation d'une matrice unique occasionnait une perte d'information sur la provenance des différentes pénalités additionnées les unes aux autres. En séparant maintenant l'information dans plusieurs matrices distinctes, il est alors plus facile de contrôler l'importance relative de chaque élément dans la recherche de solution pour en améliorer la performance.

L'ajout de tous ces éléments à l'algorithme augmente toutefois le temps de calculs. Étant donné la nécessité pour un planificateur d'obtenir rapidement des solutions, nous devons évaluer précisément la contribution de chacun d'eux à la qualité des résultats. Nos recherches futures s'orientent également vers l'introduction, dans la métaheuristique, de méthodes pour l'élaboration de solutions de compromis lorsque le décideur ne veut pas diriger exclusivement sa recherche dans une seule direction. Les notions d'ensemble de solutions non-dominées représentent une direction intéressante à cet effet. Les éléments proposés dans cet article seront intégrés à l'application utilisée par le planificateur de production de l'entreprise.

RÉFÉRENCES

- Allahverdi A., Gupta J.N.D. and T. Aldowaisan, 1999. A review of scheduling research involving setup considerations, *Omega*, 27, p. 219-239.
- Conway R.W., Maxwell W.L. and L.W. Miller 1967. *Theory of scheduling*, Addison Wesley, MA.
- Das S.R., Gupta J.N.D. and Khumawala B.M, 1995. A saving index heuristic algorithm for flowshop scheduling with sequence dependent set-up times, *The Journal of the Operational Research Society*, 46, p. 1365-1373.
- Deneubourg J.L., Pasteels J.M. and J.C. Verhaeghe, 1983. Probabilistic behaviour in ants: A strategy of errors?, *Journal of Theoretical Biology*, 105, p. 259-271.
- Deneubourg J.L. and S. Goss, 1989. Collective patterns and decision-making, *Ethology & Evolution*, 1, p. 295-311.
- Dorigo M., 1992. *Optimization, learning and natural algorithms*, Ph.D. Thesis, Politecnico di Milano, Italy.
- Dorigo M. and . Di Caro, 1999. The Ant Colony Optimization Meta-Heuristic, In: D. Corne, M. Dorigo and F. Glover Editors, *New Ideas in Optimization*, McGraw-Hill.

- Dorigo M. and L.M. Gambardella, 1997. Ant colonies for the traveling salesman problem, *BioSystems*, 43, p. 73-81.
- Dorigo M., Maniezzo V. and A. Colorni, 1991. *Positive feedback as a search strategy*, Technical Report No 91-016, Politecnico di Milano, Italy, 20 pages.
- Dorigo M., Maniezzo V. and A. Colorni, 1996. Ant system: optimization by a colony of cooperating agents, *IEEE transactions on System, Man & Cybernetic*, 26 (1), p. 29-41.
- Du J. and J.Y. Leung, 1990. Minimizing total tardiness on one machine is NP-hard, *Mathematics of Operations Research*, 15, p. 483-494.
- Franca P.M., Gendreau M., Laporte G. and F.M. Muller, 1996. A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times, *International Journal of Production Economics*, 43, p. 79-89.
- Gagné C., Price W.L. and M. Gravel, 2000., Scheduling a single machine with sequence dependent setup times using Ant Colony Optimization, soumis pour publication.
- Goss S., Beckers R., Deneubourg J.L., Aron S. and J.M. Pasteels, 1990. How trail laying and trail following can solve foraging problems for ant colonies, In: *Behavioural Mechanisms of Food Selection*, R.N. Hughes ed., NATO-ASI Series, vol. G20, Berlin: Springer-Verlag.
- Gravel M., Price W. and C. Gagné, 2000a. Scheduling jobs in a Alcan aluminium factory using a genetic algorithm, *International Journal of Production Research*, 38 (13), p. 3031-3041.
- Gravel M., Price W. and C. Gagné, 2000b. Scheduling continuous casting of aluminum using a multiple-objective ant-colony heuristic, *European Journal of Operational Research*, (to appear)
- Johnson D.S. and L.A. McGeoch, 1997. *The traveling salesman problem: a case study in local optimization*, Local Search in Combinatorial Optimization, E.H.L. Aarts & J.K. Lenstra editors, John Wiley and Sons Ltd., pp. 215-310.
- Koulamas C., 1994. The total tardiness problem: review and extensions, *Operations research*, 42 (6), p. 1025-1041.
- Pinedo M., 1995. *Scheduling theory, algorithms and systems*, Prentice-Hall, NJ.
- Ragatz G.L., 1993. A branch-and-bound method for minimum tardiness sequencing on a single processor with sequence dependent setup times, *Proceedings: Twenty-fourth Annual Meeting of the Decision Sciences Institute*, p. 1375-1377.
- Rubin P.A. and G.L. Ragatz, 1995. Scheduling in a sequence dependent setup environment with genetic search, *Computers and Operations Research*, 22 (2), p. 85-99.
- Tan K.C. and R. Narasimhan, 1997. Minimizing tardiness on a single processor with sequence-dependent setup times: a simulated annealing approach, *Omega*, 25 (6), p. 619-634.
- Tan K.C., Narasimhan R., Rubin P.A. and G.L. Ragatz, 2000., A comparison of four methods for minimizing total tardiness on a single processor with sequence dependent setup times, *Omega*, 28, p. 313-326.
- Wilbrecht J.K. and W.B. Prescott, 1969. The influence of setup time on job performance, *Management Science*, 16, p. 274-280.
- Wortman D.B., 1992. Managing capacity: getting the most from your firms assets, *Industrial Engineering*, 24, p. 47-49.
- Yang W.-H., Liao C.-J., (1999), Survey of scheduling research involving setup times, *International Journal of Systems Science*, 30 (2), p. 143-155.