Approche par programmation linéaire pour la résolution du problème d'ordonnancement sur une machine avec dates d'arrivées des travaux

Chérif Sadfi **Bernard Penz** **Hacène Azzoug**
Laboratoire GILCO - INPG
46, avenue Félix Viallet
38031 Grenoble cedex 1, France
email: `{sadfi,penz}@gilco.inpg.fr`

1 Introduction

Le problème qu'on propose d'étudier dans ce papier est le problème d'ordonnancement sur une machine avec dates d'arrivées des travaux. Les travaux ne sont présents dans l'atelier qu'à partir d'une date notée r_i (date d'arrivée du travail J_i). L'objectif est la minimisation de la somme des dates d'achèvement des travaux ($\sum_{i=1}^{n} C_i$). Le problème est noté $1|r_i|\sum C_i$. Il est NP-difficile au sens fort [2]. Nous proposons de résoudre ce problème en utilisant une approche par programmation linéaire. Un programme linéaire en nombres entiers modélisant le problème est proposé. Ensuite, nous utilisons la relaxation linéaire pour en déduire une borne inférieure et développer quatre méthodes d'approximation.

Présentons d'abord le problème. On note par $r_{[k]}$ la date d'arrivée du travail ordonnancé en position k. Le temps opératoire du travail en position k est noté $p_{[k]}$ et sa date d'achèvement est notée $C_{[k]}$. Pour ce problème l'expression de $C_{[k]}$ (date de fin d'exécution du travail en position k) dépend de $r_{[k]}$ d'une part, et de la date de fin d'exécution du travail précédent ($C_{[k-1]}$) d'autre part. Si $r_{[k]} \geq C_{[k-1]}$ (la date d'arrivé du travail en position k est supérieure à la date de fin d'exécution du travail précédent) alors $C_{[k]} = r_{[k]} + p_{[k]}$; et si $r_{[k]} \leq C_{[k-1]}$ (la date d'arrivée du travail en position k est inférieure à la date de fin d'exécution du travail précédent) alors $C_{[k]} = C_{[k-1]} + p_{[k]}$. On déduit ainsi l'expression de $C_{[k]}$: $C_{[k]} = \max\{r_{[k]}, C_{[k-1]}\} + p_{[k]}$.

2 Approche par programmation linéaire

Dans un premier temps, nous proposons une modélisation du problème par programmation linéaire en nombres entiers. Les données du problèmes sont le nombre de travaux à ordonnancer (n), le temps opératoire de chaque travail J_i (p_i) et sa date d'arrivée (r_i). Le modèle proposé (qu'on note par M) utilise des variables de décision exprimant l'affectation des travaux aux positions dans la séquence. Les variables de décision sont : $C_{[k]}$ (date d'achèvement du travail $J_{[k]}$) et x_{ik} (variables entières exprimant l'affectation des travaux aux positions) avec : $x_{ik} = 1$ si le travail J_i est ordonnancé en position k et $x_{ik} = 0$ sinon. Pour la résolution du programme linéaire nous avons utilisé le logiciel Cplex [1]. La programmation linéaire en nombres entiers nous a permis de résoudre des instances allant jusqu'à 20 travaux en temps raisonnable.

Ensuite, nous proposons de résoudre le modèle M en relachant la contrainte entière des variables de décisions x_{ik}. Le nouveau modèle est noté $\overline{\mathrm{M}}$. On aura $x_{ik} \in [0, 1]$. La résolution du modèle relaxé permet d'obtenir une borne inférieure pour le problème qu'on note par LB. Une étude théorique a montré que le modèle $\overline{\mathrm{M}}$ donne la solution optimale pour deux cas "extrêmes" : lorsque les dates de disponibilité des travaux sont toutes nulles ($r_i = 0$ pour tout $i \in \{1, 2, ..., n\}$) et lorsqu'il n'y a pas de chevauchement entre les travaux ($r_{[i]} > r_{[i-1]} + p_{[i-1]}$ pour tout $i \in \{1, 2, ..., n\}$).

Enfin, nous proposons quatre méthodes d'approximation basées sur la résolution du programme linéaire $\overline{\mathrm{M}}$. A partir des valeurs des variables x_{ij} obtenues, on remplit une matrice $X = (x_{ij})$ où les lignes représentent les travaux et les colonnes représentent les positions. La matrice X étant de dimension $(n \times n)$ avec n le nombre de travaux à ordonnancer. En utilisant cette matrice, trois techniques de recherche d'une solution réalisable sont proposées (heuristiques H1, H2 et H3). La quatrième heuristique est une heuristique de reconstruction, elle utilise la résolution du programme linéaire relaxé de façon itérative pour la recherche d'une solution (heuristique H4).

Nous avons testé expérimentalement l'efficacité de la borne inférieure et des quatre heuristiques proposées en les comparant avec une borne inférieure de la littérature obtenue en autorisant la préemption (qu'on note par SRPT) et une heuristique de la littérature développée par Phillips *et al.* [3] (qu'on note par H5). Une étude préliminaire a été menée afin d'identifier les problèmes "difficiles" c'est-à-dire les problèmes pour lesquels le résultat de la résolution du programme linéaire relaxé $\overline{\mathrm{M}}$ s'éloigne le plus de la solution optimale (il s'agit de déterminer les distributions correspondantes des r_i). Les résultats expérimentaux ont montré que la borne inférieure SRPT donne de meilleurs résultats que notre borne LB. Pour les problèmes de petite taille, les quatre heuristiques proposées ont un meilleur comportement que l'heuristique de la littérature. Pour les problèmes de grande taille, les heuristiques présentent un comportement assez proche avec l'heuristique H1 qui donne les résultats les plus éloignés des bornes inférieures.

Références

[1] Cplex 6.0, ILOG, 1998. http://www.ilog.com/products/cplex

[2] J. K. Lenstra, A. H. G. Rinnooy Kan et P. Brucker, Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, vol. 1: 343–362, 1977.

[3] C. Phillips, C. Stein et J. Wein, Minimizing average completion time in the presence of release dates. *Mathematical Programming B*, vol. 82: 199–223, 1998.

[4] M. X. Goemans, M. Queyranne, A. S. Schulz, M. Skutella et Y. Wang, Single machine scheduling with release dates. *Mathematics of Operations Research*, 2001. Soumis.