Le meilleur des cas dans un ordonnancement de groupes

Guillaume Pinot et Nasser Mebarki

IRCCyN — UMR CNRS 6597, 1 rue de la Noé, BP 92101, 44321 Nantes Cedex 3
prenom.nom@irccyn.ec-nantes.fr

1 Introduction

L'ordonnancement de groupes permet d'introduire une flexibilité séquentielle importante tout en garantissant une certaine qualité dans le pire des cas. Une évaluation du meilleur des cas d'un ordonnancement de groupes pourrait également être utile. Elle permettrait une description plus complète de l'ordonnancement de groupes dans sa globalité. Son utilisation dans un outil d'aide à la décision en temps réel basée sur l'ordonnancement de groupes pourrait également être bénéfique en apportant plus d'information au décideur.

Cet article présente donc différents algorithmes pour l'évaluation dans le meilleur des cas d'un ordonnancement de groupes : bornes inférieures, heuristiques et méthodes exactes. [1,2] détaillent respectivement les bornes inférieures et les heuristiques.

2 Ordonnancement de groupes

L'ordonnancement de groupes fut créé au LAAS il y a plus de 30 ans ([3,4,5]). Il est également connu sous le nom ORABAID (ORdonnancement d'Atelier Basé sur l'AIde à la Décision) et est implémenté dans le progiciel ORDO [1]. Cette méthode est décrite dans [6].

Un groupe d'opérations permutables est un ensemble d'opérations qui seront exécutées successivement sur une même machine, dans un ordre qui n'est pas fixé à l'avance. Un ordonnancement de groupes est défini par une séquence de groupes sur chaque machine. Il est dit réalisable si toute permutation des opérations au sein de chaque groupe conduit à un ordonnancement qui satisfait les contraintes du problème. Un ordonnancement de groupes définit ainsi plusieurs ordonnancements réalisables de manière implicite.

Cette représentation implicite d'un ensemble d'ordonnancements possède une propriété très intéressante : la qualité de l'ordonnancement dans le pire des cas est calculable en temps polynomial pour les objectifs de type *minmax* (comme $C_{\max}$, $L_{\max}$ et $T_{\max}$). Ainsi, cette méthode est utilisable pour des problèmes de taille conséquente.

3 Bornes inférieures

Avoir une borne inférieure sur la date de fin des opérations permettrait d'obtenir une borne inférieure pour tout critère régulier. Nous proposons pour cela un algorithme de programmation dynamique basé sur la relaxation sur les ressources et sur une borne inférieure de la date de fin d'un groupe. Cet algorithme permet de calculer une borne inférieure de la date de début d'une opération dans le meilleur des cas (θ_i), une borne inférieure de la date de fin d'une opération dans le meilleur des cas (χ_i), et une borne inférieure de la date de fin d'un groupe dans le meilleur des cas (γ_i). Ainsi, une borne inférieure de $L_{\max}$ peut être calculée comme $\max_{\forall O_i} L_i(\chi_i)$ et une borne inférieure de $C_{\max}$ peut être calculée comme $\max_{\forall G_k} \gamma_k$.

Une relaxation très utilisée pour le problème du *job shop* classique est la relaxation en *one-machine problem* [7]. Pour réaliser une telle relaxation sur l'ordonnancement de groupes, il nous faut des *heads* et *tails*[2] pour chaque opération. θ_i est une *head* valide pour l'opération O_i. Par symétrie des *heads* et *tails*, une *tail* valide peut être obtenue de façon similaire. La relaxation en

[1] Le progiciel ORDO est la solution d'ordonnancement et de planification du groupe Schneider Electric. Voir http://www.ordosoftware.com/.

[2] Une *head* d'une opération est une borne inférieure de sa date de début. Une *tail* d'une opération est une borne inférieure de la différence entre sa date de fin et la date de fin de l'ordonnancement.

one-machine problem se fait alors simplement pour l'ordonnancement de groupes, en effectuant la relaxation au niveau des groupes plutôt qu'au niveau des machines. Grâce à cette relaxation, nous obtenons une borne inférieure efficace du $C_{\max}$ pour l'ordonnancement de groupes.

4 Heuristiques

Comme l'ordonnancement de groupes correspond à l'ordonnancement classique avec des contraintes supplémentaires, il est possible d'utiliser les heuristiques appelées « règles de priorité ». Nous utilisons directement des règles classiques et proposons une nouvelle règle basée sur la *tail* d'une opération (*cf.* section précédente). Pour améliorer ces règles de priorité, nous les combinons avec la borne inférieure du $C_{\max}$ de la section précédente.

Grâce à la relaxation en *one-machine problem* décrite à la section précédente, nous avons adapté l'heuristique *shifting bottleneck* [8] au problème de l'ordonnancement de groupes.

Ces heuristiques aux performances interessantes permettent différents compromis entre temps de calcul et qualité de la solution.

5 Méthode exacte

Pour completer notre étude sur le meilleur des cas, nous avons développé un algorithme de résolution exacte adapté à tout objectif régulier. Cet algorithme se base sur l'énumération des ordonnancements actifs. Le parcours de l'arbre de recherche se fait en largeur jusqu'à un certain nombre de nœuds stockés, puis en profondeur. Pour en améliorer les performances, nous avons trouvé une condition suffisante permettant d'ordonnancer un groupe complet sans avoir à énumerer l'ensemble des ordonancements actifs.

Nous avons expérimenté cet algorithme sur le $C_{\max}$. Notre condition suffisante pour ordonnancer un groupe nous permet de diviser le temps de calcul par quatre en moyenne. Sur les 40 instances de *job shop* testées, 15 sont résolues en moins d'une seconde, 32 en moins d'une minute, 34 en moins d'une heure et 36 en moins d'une journée. 4 instances restent non résolues au bout d'une semaine de calcul.

6 Conclusion

Dans cet article, nous proposons de résoudre le meilleur des cas dans un ordonnancement de groupes. Pour parvenir à cela, des bornes inférieures, des heuristiques et des méthodes exactes sont proposées. Les résultats obtenus par ces méthodes sont satisfaisants.

La méthode exacte pourrait être améliorée, notamment en utilisant notre condition suffisante dans la méthode de parcours de l'arbre de recherche.

Références

1. Pinot, G., Mebarki, N. : Best-case lower bounds in a group sequence for the job shop problem. In : Proceedings of the 17th IFAC World Congress. (2008) To be reviewed.
2. Pinot, G., Mebarki, N. : Heuristiques pour le meilleur des cas dans un ordonnancement de groupes. In : Actes de la 7ème Conférence Francophone de Modélisation et Simulation (MOSIM'08). (2008) To be reviewed.
3. Erschler, J. : Analyse sous contraintes et aide à la décision pour certains problèmes d'ordonnancement. Thèse de doctorat, Université Paul Sabatier, Toulouse (1976)
4. Demmou, R. : Étude de familles remarquables d'ordonnancements en vue d'une aide à la décision. Thèse de doctorat, Université Paul Sabatier, Toulouse (1977)
5. Thomas, V. : Aide à la décision pour l'ordonnancement d'atelier en temps réel. Thèse de doctorat, Université Paul Sabatier, Toulouse (1980)
6. Artigues, C., Billaut, J.C., Esswein, C. : Maximization of solution flexibility for robust shop scheduling. European Journal of Operational Research **165** (2005) 314–328
7. Carlier, J. : The one-machine sequencing problem. European Journal of Operational Research **11** (1982) 42–47
8. Adams, J., Balas, E., Zawack, D. : The shifting bottleneck procedure for job shop scheduling. Management Science **34** (1988) 391–401