Minimisation du délai moyen : une contrainte globale

Camille Bonnin[1,2], Margaux Nattaf[1], Arnaud Malapert[2], Marie-Laure Espinouse[1]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP[†], G-SCOP, 38000 Grenoble, France
{camille.bonnin, margaux.nattaf, marie-laure.espinouse}@grenoble-inp.fr
[2] Université Côte d'Azur, CNRS, I3S, France
arnaud.malapert@univ-cotedazur.fr

Mots-clés : *Programmation Par Contraintes, Ordonnancement, Filtrage, Délai Moyen.*

Introduction De nos jours, les problèmes d'ordonnancement sont omniprésents dans la société. Une méthode prometteuse pour les résoudre est la Programmation Par Contraintes (PPC), pour laquelle de nombreux travaux ont été menés [1]. Néanmoins, ces derniers ont longtemps été principalement tournés vers l'optimisation du délai total ($C_{\max}$), négligeant les autres fonctions objectif.

L'objectif de notre travail est d'améliorer la prise en compte de l'optimisation du délai moyen ($\sum C_j$) en PPC pour le problème $1|r_j; d_j|\sum C_j$. Nous avons choisi ce problème pour sa généricité. Notre objectif est de définir une contrainte globale qui permet de minimiser le délai moyen pour les problèmes d'ordonnancement à une machine. $1|r_j; d_j|\sum C_j$ étant $\mathcal{NP}$-difficile [4], nous allons utiliser des bornes inférieures obtenues grâce à la relaxation $1|r_j; pmtn|\sum C_j$.

Contrainte globale A notre connaissance, aucune contrainte globale n'a encore été proposée pour le délai moyen. Toutefois, Kovács et Beck ont proposé une contrainte globale pour le problème plus général de minimisation du délai moyen pondéré [2]. Leurs travaux utilisent des bornes inférieures obtenues grâce à la relaxation polynomiale qu'est le problème préemptif à une machine avec des dates de disponibilité dont l'objectif est la minimisation du temps d'occupation moyen pondéré ($1|r_j; pmtn|\sum w_j M_j$). Comme notre travail porte sur le cas particulier du délai moyen sans pondération, la relaxation utilisée ici est différente de celle de Kovács et Beck. En effet, contrairement à sa version pondérée, le problème préemptif à une machine avec des dates de disponibilité dont l'objectif est la minimisation du délai moyen ($1|r_j; pmtn|\sum C_j$) est polynomial. La relaxation utilisée ici est donc $1|r_j; pmtn|\sum C_j$.

La contrainte possède deux phases, une phase de vérification et une phase de filtrage des domaines. La phase de vérification permet, en utilisant la relaxation, de mettre à jour la borne inférieure de l'ordonnancement avec les données du nœud courant de l'arbre de recherche. La phase de réduction des domaines permet de retirer des domaines des variables les valeurs pour lesquelles il n'existe aucune solution optimale les contenant.

A chaque nœud, la contrainte commence par la phase de vérification. En tenant compte des choix faits et des données présentes au nœud courant de l'arbre, elle résout la relaxation $1|r_j; pmtn|\sum C_j$ grâce à la règle de Smith modifiée [3] et met à jour la valeur de la borne inférieure du solveur avec la valeur de la relaxation. Cette borne inférieure est ensuite comparée à la borne supérieure calculée par le solveur, ce qui permet de détecter une infaisabilité.

Ensuite, la contrainte entre dans une phase de filtrage des bornes des domaines. Pour ce faire, nous proposons une première version naïve qui consiste à tester une à une toutes les dates de début possibles pour chacune des tâches.

Supposons que la contrainte examine la tâche j dont le domaine des dates de début possibles est $\mathcal{S}_j$. La contrainte fixe la date de début de j à sa plus petite valeur s_j. Une fois ceci fait, la contrainte résout la relaxation $1|r_j; pmtn|\sum C_j$ avec la règle de Smith modifiée. Cela permet d'obtenir une borne inférieure du nœud courant. Si cette borne inférieure dépasse la borne

[†]Institute of Engineering Univ. Grenoble Alpes

supérieure fournie par le solveur, alors la tâche j ne peut pas commencer à s_j qui est filtrée de $\mathcal{S}_j$. Pour une tâche donnée, l'algorithme se termine lorsqu'une date de début n'est pas filtrée et l'algorithme est lancé pour toutes les tâches.

Premiers résultats Nous avons effectué des tests préliminaires de la version naïve de notre contrainte pour valider notre raisonnement. Pour ce faire, nous avons utilisé CP Optimizer 20.1.0 sur 80 des instances de Pan et Shi [5] avec un nombre de tâches allant de 20 à 60 tâches et des paramètres variés. La limite de temps est fixée à 60 secondes.

Trois modèles sont résolus : le modèle classique sans notre contrainte, le modèle avec vérification et le modèle avec filtrage.

Le modèle classique trouve toujours une solution et résout à l'optimal 22% des instances. Le modèle avec vérification résout à l'optimal 98% des instances et ce plus rapidement que le modèle classique. Le modèle avec filtrage, dans sa version naïve, ne trouve pas toujours une solution mais permet de résoudre à l'optimal 41% des instances.

De plus, le modèle avec filtrage permet de réduire considérablement le nombre de nœuds par rapport aux deux autres modèles pour les instances résolues à l'optimalité.

Conclusions et perspectives Nous avons proposé une version naïve d'une contrainte globale permettant d'optimiser le délai moyen pour le problème $1|r_j; d_j| \sum C_j$.

Nos premiers résultats sont prometteurs car notre contrainte réduit le nombre de nœuds dans l'arbre de recherche. Ce premier développement a permis de valider le filtrage.

Une perspective à court terme est de comparer les qualités des bornes obtenues par la relaxation utilisée dans notre contrainte par rapport à celles obtenues par la relaxation utilisée par Kovács et Beck dans leur contrainte. Une perspective suivate à plus long terme est d'améliorer les performances en termes de temps de notre contrainte en nous inspirant de l'incrémentalité des travaux de Kovács et Beck.

Références

[1] Philippe Baptiste, Claude Le Pape et Wim Nuijten. *Constraint-Based Scheduling : Applying Constraint Programming to Scheduling Problem.* International Series in Operations Research & Management Science, Vol. 39, p. xiii–198, Verlag Springer US, 2001.

[2] András Kovács et J. Christopher Beck. *A global constraint for total weighted completion time for unary resources.* Constraints, Vol. 16, No. 1, p. 100–123, 2011.

[3] Peter Brucker. *Scheduling Algorithms.* Chap. 4, p. 61–106, Springer, 2006.

[4] Jan K. Lenstra, Alexander H. G. Rinnooy Kan, Peter Brucker. *Complexity of machine scheduling problems.* Annals of discrete mathematics, Vol. 1, Elsevier, p. 343–362, 1977.

[5] Yunpeng Pan et Leyuan Shi. *New Hybrid Optimization Algorithms for Machine Scheduling Problems.* IEEE Transactions on Automation Science and Engineering, Vol. 5, p. 337–348, 2008.