



HAL
open science

Algorithme glouton et recherche locale arborescente pour le MSPSP avec préemption partielle

Oliver Polo Mejia, Christian Artigues, Pierre Lopez

► To cite this version:

Oliver Polo Mejia, Christian Artigues, Pierre Lopez. Algorithme glouton et recherche locale arborescente pour le MSPSP avec préemption partielle. 20e congrès annuel de la société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2019), Feb 2019, Le Havre, France. <hal-02352265>

HAL Id: hal-02352265

<https://laas.hal.science/hal-02352265v1>

Submitted on 6 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Algorithme glouton et recherche locale arborescente pour le MSPSP avec préemption partielle

Oliver Polo-Mejía^{1,2}, Christian Artigues², Pierre Lopez²

¹ CEA, DEN, DEC, SETC, St Paul lez Durance, France
oliver.polomejia@cea.fr

² LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
{artigues, lopez}@laas.fr

Mots-clés : *ordonnancement de projets, multi-compétences, préemption partielle*

1 Introduction

Dans le problème d'ordonnancement de projet multi-compétences avec préemption partielle (MSPSP-PP), l'objectif est de trouver un ordonnancement qui minimise la durée totale du projet (C_{\max}) sur des ressources renouvelables multi-compétences avec capacité limitée, pouvant répondre à plus d'une exigence de compétences par activité. Trouver une solution consiste à déterminer les périodes pendant lesquelles chaque activité est exécutée ainsi que les ressources qui l'exécuteront à chaque période, tout en respectant la capacité des ressources et les caractéristiques des activités. La principale caractéristique du MSPSP-PP est la façon dont est gérée la préemption des ressources pendant les périodes d'interruption. Au lieu de libérer toutes les ressources, comme dans les problèmes d'ordonnancement préemptifs classiques, nous autorisons de ne libérer qu'un sous-ensemble de ressources, les ressources complémentaires restant occupées par l'activité.

Cette variante a été originalement proposée pour modéliser le processus d'ordonnancement dans une des installations nucléaires de base du Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA). Les résultats expérimentaux, présentés dans [2], montrent les difficultés de la programmation linéaire mixte en nombres entiers et de la programmation par contraintes pour traiter des instances de taille réaliste (plus de 30 activités). C'est pourquoi nous devons développer des méthodes heuristiques efficaces comme celle proposée dans cette communication.

2 Algorithme glouton : schéma de génération en série

Le MSPSP-PP peut être vu comme un problème composé de deux sous-problèmes couplés : un problème d'ordonnancement d'activités lié à un problème d'allocation des techniciens. Pour résoudre le MSPSP-PP, nous proposons un schéma de génération en série, où les activités sont ordonnancées une à une au plus tôt, suivant une règle de priorité.

Pour améliorer la qualité de la solution, nous proposons de construire l'ensemble des activités à ordonnancer (L) comme suit : $L = \{NPA, SPA, PA\}$ où NPA est le sous-ensemble des activités non préemptives, SPA les activités partiellement préemptives et PA les activités préemptives. NPA , SPA et PA sont triés selon la règle de priorité choisie. Cette approche exploite la capacité des activités préemptives et partiellement préemptives pour combler les espaces laissés libres après ordonnancement des activités non préemptives.

Une fois défini l'ordre dans lequel les activités seront traitées, nous devons choisir les techniciens qui les effectueront. En s'inspirant des travaux de Bellenguez-Morineau [1], ce problème d'allocation peut être traité comme un problème de flot maximum à coût minimum (MCMF)

sur un graphe biparti $G_i = (X, U)$, $X = S \cup P$, où S représente l'ensemble des compétences requises par l'activité A_i et P est le sous-ensemble des techniciens disponibles pendant la période t et qui maîtrisent au moins une des compétences requises par l'activité A_i . On associe à chaque arc de U entre un technicien P_j et une compétence, un coût unitaire $CT_{i,j}$ qui est une fonction du degré de demande de ce technicien par les activités non encore ordonnancées.

3 Recherche locale arborescente

Nous proposons d'utiliser un algorithme de recherche locale, partiellement inspiré des méthodes « Limited Discrepancy Search » et « Branch-and-Greed », pour améliorer les résultats de l'algorithme glouton. Pour chaque règle de priorité, il existe un grand nombre d'ordonnancements possibles définis par les affectations de techniciens effectuées. En effet, pour chaque période pendant laquelle nous planifions une unité de durée d'une activité, il peut y avoir un grand nombre d'allocations. En raison de l'explosion combinatoire, énumérer toutes les solutions peut être prohibitif. Un arbre de recherche binaire incomplet peut alors être intéressant. Comme l'algorithme glouton, l'arbre binaire part d'une liste d'activités, les ordonnance et leur attribue des techniciens. Toutefois, pour un nœud correspondant à l'affectation d'un ensemble de techniciens à une activité à une période donnée, on génère deux nœuds fils : à gauche la meilleure allocation obtenue en résolvant le MCMF, à droite, la deuxième meilleure solution (obtenue en résolvant à nouveau le MCMF avec des contraintes supplémentaires). Pour les activités non préemptives, un seul nœud sera généré (pour garantir que les mêmes techniciens exécutent toute l'activité), tandis que pour les préemptives et partiellement préemptives, il faut générer autant de nœuds que d'unités de durée de l'activité.

Nous devons limiter encore plus le nombre de branches visitées, spécialement pour les instances industrielles (plus de 30 activités) ayant une grande proportion d'activités préemptives et partiellement préemptives. De la manière dont la solution est construite dans l'algorithme glouton, nous pouvons en déduire que la probabilité d'erreur de l'heuristique diminue à mesure que nous ajoutons plus d'activités au planning partiel. On peut ensuite diminuer le nombre de branches examinées en donnant à chaque nœud une probabilité décroissante en fonction de la profondeur dans l'arbre pour examiner la branche de droite. Pour explorer l'arbre, on utilise une recherche en profondeur d'abord de gauche à droite, en éliminant toutes les solutions (ou solutions partielles) n'améliorant pas le meilleur C_{\max} .

4 Conclusion et perspectives

L'algorithme glouton proposé nous permet d'obtenir un gap moyen de 4,98% pour les instances de taille industrielle. L'algorithme de recherche locale basé sur une arborescence réduit ce gap à 2,88%. Les deux algorithmes semblent donner de meilleurs résultats lorsque la proportion d'activités non préemptives dans l'instance est élevée.

Dans la suite, nous devons mieux étudier comment choisir habilement la probabilité de visiter la branche de droite dans l'arbre de recherche, afin de trouver un compromis entre la qualité des solutions et le temps requis pour les obtenir. Nous devons aussi étudier différentes approches pour gérer les nœuds de l'arbre de recherche afin d'améliorer les temps de résolution.

Références

- [1] Odile Bellenguez-Morineau. Methods to solve multi-skill project scheduling problem. *4OR*, 6(1):85–88, March 2008.
- [2] Oliver Polo-Mejía, Marie-Christine Anselmet, Christian Artigues, and Pierre Lopez. Mixed-integer and constraint programming formulations for a multi-skill project scheduling problem with partial preemption. In *12th International Conference on Modelling, Optimization and Simulation*, 8 pages, Toulouse, France, June 2018.