

# Ordonnancement avec opérations de préparation complexes : programmation par contraintes et optimisation hiérarchique

Adriana Pacheco, Cédric Pralet, Stéphanie Roussel

ONERA / DTIS, Université de Toulouse, F-31055 Toulouse, France  
{adriana.pacheco, cedric.pralet, stephanie.roussel}@onera.fr

**Mots-clés :** *Programmation par contraintes, décomposition des problèmes, ordonnancement.*

## 1 Introduction

Nous nous intéressons aux problèmes d’ordonnancement impliquant des ressources disjonctives qui doivent effectuer des opérations de préparation complexes entre les tâches principales qu’elles réalisent. Ce type de problème a été largement étudié pour traiter des applications dans différents domaines. Comme exemple, nous considérons une application où plusieurs robots doivent être déployés sur un terrain et naviguer (opérations de préparation) à travers un réseau partagé, pour faire des observations (tâches principales) à différents emplacements.

## 2 Ordonnancement itératif

Pour prendre en compte toutes les tâches de préparation qui utilisent des ressources partagés entre les robots telles que des points de passage et des liens de navigation reliant ces points (voir Figure 2), une première option est de définir un problème d’ordonnancement unique (modèle global) contenant toutes les opérations qui pourraient être utilisées dans un planning optimal (un grand nombre de tâches). Pour diminuer la complexité de résolution de ce type de problèmes, nous introduisons deux mécanismes de décision, simples mais efficaces, qui utilisent de façon itérative deux couches d’ordonnancement (voir Figure 1). Une approche en Programmation Par Contraintes (PPC) est considérée pour ces deux couches de décision, qui alternent synthèse rapide d’ordonnancements de haut niveau (couche L1), et production d’ordonnancements plus précis (couche L2). À haut niveau par exemple, les opérations de préparation de plus bas niveau et les ressources qu’elles consomment peuvent être approximées tout simplement comme des durées minimales à laisser entre tâches principales. Le but de ce processus n’est pas d’obtenir une solution optimale pour l’ensemble du problème, mais d’obtenir des solutions de bonne qualité dans un court délai de calcul, ce qui est plus crucial que de trouver une solution optimale dans de nombreux cas d’étude. Nous avons proposé deux stratégies de feedback pour ajuster le modèle de la couche de décision de haut niveau L1 en fonction des résultats produits par la couche L2.

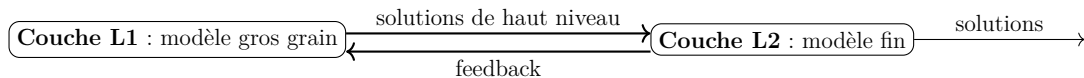


FIG. 1 – Interactions entre les couches de décision

**Stratégie inspirée des modèles d’approximation [3].** Cette première stratégie considère un mécanisme d’interaction générique entre couches, proche des travaux sur les modèles d’approximation (ou modèles de substitution) pour l’optimisation boîte noire [2]. Chaque fois

qu'un nouveau planning détaillé est disponible, le modèle de substitution est mis à jour en fonction des temps réels de préparation qui tiennent compte des interférences entre les opérations de préparation détaillées (voir Figure 3). Ainsi, L1 apprend itérativement une meilleure approximation du contenu de L2, l'objectif étant de converger très rapidement vers de meilleures solutions complètes. Des bornes inférieures des durées réelles des opérations de préparation peuvent également être utilisées pour évaluer la distance maximale à une solution optimale.

**Stratégie de type Décomposition de Benders avec coupes heuristiques [4].** Dans une deuxième stratégie qui peut être considérée comme une sorte de “Logic-Based Benders Decomposition” [1], la couche de bas niveau contient un module d'explication capable de fournir des contraintes pertinentes sur les variables de décision de haut niveau. Ces contraintes (ou coupes) tiennent compte des interférences liées aux conflits d'utilisation des ressources lors des déplacements des robots sur le réseau partagé. Par exemple, dans la Figure (3), les robots  $r_1$  et  $r_2$  ne peuvent pas utiliser simultanément, ni les points de passage 1 et 2, ni les liens  $l_3$  et  $l_4$  dans leurs déplacements respectifs indiqués par des flèches. Dans la plupart des références existantes en robotique, ces interférences ne sont pas prises en compte lors de la planification et l'utilisation de mécanismes anti-collision est supposée au moment de l'exécution. Quatre variantes incomplètes de la stratégie de génération de coupes ont été étudiées.

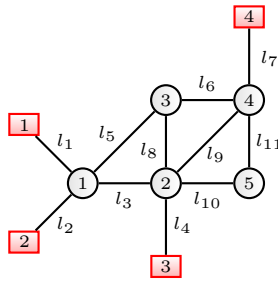


FIG. 2 – Terrain

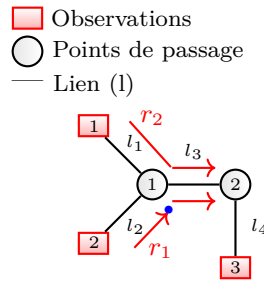


FIG. 3 – Interférence

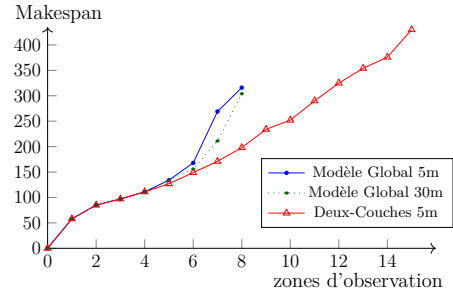


FIG. 4 – Résultats

### 3 Expérimentations

L'approche itérative bi-niveaux a été comparée à la stratégie globale de résolution en PPC, sur des benchmarks représentatifs d'une application multi-robots. Les problèmes d'ordonnement ont été résolus à l'aide d'IBM ILOG CP Optimizer, avec un temps de calcul de 5 ou 30 minutes pour le modèle global, et seulement 5 minutes pour l'approche itérative. Des résultats représentatifs sont présentés sur la Figure (4). Ils montrent l'efficacité des approches et la complémentarité des schémas de génération de coupes mis en place. Les approches itératives proposées sont à la fois simples et beaucoup plus efficaces que le modèle complet pour obtenir très rapidement des solutions de bonne qualité même pour des problèmes de grande taille.

### Références

- [1] John N. Hooker and Greger Ottosson. *Logic-based Benders decomposition*. Mathematical Programming 96(1) : 33-60, 2013.
- [2] Vu Khac Ky, Claudia D'Ambrosio, Youssef Hamadi and Leo Liberti. *Surrogate-based methods for black-box optimization*. International Transactions in Operational Research, 2016.
- [3] Adriana Pacheco, Cédric Pralet and Stéphanie Roussel. *Constraint-Based Scheduling with Complex Setup Operations : An Iterative Two-Layer Approach*. IJCAI, 2019.
- [4] Adriana Pacheco, Cédric Pralet and Stéphanie Roussel. *Decomposition and Cut Generation Strategies for Solving Multi-Robot Deployment Problems*. CP, 2019.