

Arbres de décision robustes pour le RCPSP multi-mode

Tom Portoleau^{1,2}, Christian Artigues¹, Romain Guillaume²

¹ LAAS-CNRS, Université de Toulouse, CNRS, France,
{tom.portoleau,christian.artigues}@laas.fr

² IRIT-CNRS, Université de Toulouse, France, romain.guillaume@irit.fr

Mots-clés : *ordonnancement sous incertitude, information-based, arbre de décision*

1 Introduction

La prise en compte de l'incertitude dans les problèmes d'ordonnancement a vastement été étudiée dans la littérature. Classiquement, deux familles de méthodes ont émergé. Les approches pro-actives qui ont pour objectif de construire dans une phase hors-ligne des plannings robustes, avec une garantie théorique sur l'objectif mais qui en pratique produisent des solutions très conservatrices. A l'inverse, les approches réactives calculent en-ligne des nouveaux plannings pour faire face rapidement aux incertitudes lorsqu'elles se réalisent. Ces méthodes sont souvent basées sur des règles de priorités et les solutions qu'elles produisent sont plus flexibles, mais n'apportent aucune garantie théorique et sont souvent dans le pire scénario de moins bonne qualité qu'une solution robuste. Pour combiner les avantages des deux familles de méthodes, des approches dite hybrides ont été étudiées. Dans [1], les auteurs ont proposé ce qu'ils appellent l'approche Just In Case, dans laquelle ils calculent un calendrier contingent multiple, avec comme application la planification d'actions pour un problème d'ordonnancement de télescope autonome. Une autre façon de voir cette approche est de la considérer comme un cas très spécifique d'optimisation adaptable robuste multi-étapes, où une partie des données incertaines est révélée séquentiellement, et dont l'objectif idéal est de calculer une solution entièrement adaptable, c'est-à-dire qu'un planning optimal est connu pour chaque réalisation de scénario possible. D'un point de vue computationnel, le calcul d'une telle solution est inenvisageable. Pour pallier cela, plusieurs méthodes d'approximation d'une solution entièrement adaptable ont vu le jour. Une méthode efficace cherche à calculer une bonne partition de l'ensemble des scénarios, puis calculer des solutions sur chacun des sous ensembles de la partition. En s'inspirant de cette idée de partitionnement de l'ensemble des scénarios, nous proposons une approche basée sur la construction d'un arbre de décision pour guider un décideur dans la mise à jour de son planning, et ce appliqué à un problème industriel : l'optimisation de l'utilisation de ressources sur une ligne d'assemblage aéronautique. On modélise le problème comme un RCPSP multi-mode : on doit ordonnancer un ensemble de tâches -dont les durées sont incertaines- de sorte à minimiser l'utilisation d'un sous-ensemble de ressources, tout en respectant un makespan fixé. Une particularité de ce problème est que les tâches peuvent être effectuées dans plusieurs modes, où un mode correspond à une utilisation des ressources et à un temps de traitement différents.

2 Modèle

L'enjeu de notre approche est la construction d'un arbre de décision robuste [2] qui propose au décideur d'adapter de manière en-ligne le planning en cours pour répondre au mieux à la réalisation du scénario en cours. Pour construire un tel arbre, on suppose qu'il existe ce qu'on appelle des moments de décision, durant lesquels le décideur est capable d'observer le déroulement du planning, d'accéder à des informations sur la réalisation du scénario et de modifier la

solution en cours. Chaque moment de décision correspond à un niveau de l'arbre de décision, de sorte que t_1 correspond au premier niveau, t_2 au deuxième, etc. Ainsi, la profondeur de l'arbre est contrôlée par le nombre de moments de décision. A chaque bifurcation à un niveau j de l'arbre, une nouvelle solution partielle, cohérente avec le planning partiel qui a été accompli jusqu'à t_j . A chaque noeud de l'arbre, certaines informations sur le déroulement du scénario sont disponibles. Dans notre modèle, une information correspond à la réponse à la question "la tâche i a-t-elle duré plus ou moins que X ?". Chacune de ces informations permet donc de couper en deux l'ensemble des scénarios, d'un côté ceux pour lesquels la réponse est oui, et ceux pour lesquels la réponse est non. Plus généralement m informations permettent de séparer un ensemble de scénarios en 2^m sous-ensembles. Néanmoins, il est possible que l'accès à ces informations soit coûteux, et que toutes les informations ne soient pas pertinentes pour améliorer la qualité de la solution en cours. Afin de sélectionner des "bonnes" informations on modélise le problème de sélection d'information comme un problème d'optimisation, qui prend en entrée un moment de décision t_j , l'état du planning à ce moment, et les informations disponibles. Les solutions à ce problème sont des sous-ensembles d'informations, à partir desquelles on peut calculer une partition, puis calculer une solution sur chacun des sous-ensembles de ladite partition. Le critère est minimiser lexicographiquement le vecteur $RS(P) = (\min_{s \in S} \max_{\omega \in P} f(\omega, s))_{P \in P}$ où P est une partition, et $f(\omega, s)$ la valeur objectif de la solution s dans le scénario ω . On appelle ce problème le Robust Partition Problem (RPP). Pour chaque sous-ensemble de la partition calculée, on crée une nouvelle branche sous le noeud avec la solution partielle associée.

3 Algorithme, résultats expérimentaux et conclusion

La construction de l'arbre se fait en profondeur, et prend en paramètre la liste $(t_i)_i$ des moments de décision, Q le nombre d'information maximal que l'on peut utiliser à chaque noeud et L la taille maximale de la partition calculée à chaque noeud par la résolution du RPP. L'idée pour résoudre ce problème est d'étendre le RCPSP multi-mode en considérant qu'avoir des informations sur une tâche revient à dédoubler ses différents modes : pour chaque mode initial de la tâche, on crée un nouveau mode fictif où la durée de la tâche est ajustée en fonction de l'information. L'algorithme se base sur un modèle PPC que l'on appelle pour calculer le sous-ensemble d'information qui permet de calculer la solution avec la meilleure valeur minmax. On recommence en interdisant l'utilisation du même sous-ensemble d'information, jusqu'à ce que la limite d'information Q soit atteinte, ou que le nombre de sous-ensemble de scénario calculé soit $L - 1$. Quand c'est le cas on complète la partition avec l'ensemble des scénarios qui n'est pas couvert, et la solution associée est la solution robuste de la branche précédente. Pour évaluer nos arbres, on se compare à un algorithme réactif qui utilise les mêmes informations qu'eux mais de manière aléatoire. Tous les tests sont réalisés sur des instances industrielles. On cherche à montrer que le choix des informations pertinentes permet de produire des solutions de meilleure qualité. On cherche également à compter, sur les mêmes scénarios le nombre de solutions différentes calculées par les deux algorithmes, que l'on considère comme un critère de stabilité. Finalement, on regarde l'impact de : la distribution de probabilités sur le tirage des scénarios, le nombre de contraintes de précédences et la disponibilité des ressources, sur la qualité de nos arbres. Ce travail a été en partie financé par le projet ANR PER4MANCE.

Références

- [1] Mark Drummond, John Bresina, and Keith Swanson. Just-in-case scheduling. In *AAAI*, volume 94, pages 1098–1104, 1994.
- [2] Tom Portoleau, Christian Artigues, and Romain Guillaume. Robust predictive-reactive scheduling : an information-based decision tree model. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 479–492. Springer, 2020.