Méthode tabou pour un problème de RCPSP multi-mode avec précédences généralisées

Baptistin Carvin, Eva Epoy, Odile Bellenguez, Guillaume Massonnet

Méthode tabou pour un problème de RCPSP multi-mode avec précédences généralisées

Baptistin Carvin[1,2], Eva Epoy[1], Odile Bellenguez[1], Guillaume Massonnet[1]

[1] IMT Atlantique, LS2N, La Chantrerie, 4 rue Alfred Kastler, 44307 Nantes, France
{baptistin.carvin,odile.bellenguez,guillaume.massonnet}@imt-atlantique.fr
[2] Vif, 44240 La Chapelle-sur-Erdre, France

Mots-clés : *recherche tabou, ordonnancement cumulatif, précédences généralisées*

1 Introduction

Nous nous intéressons à un problème de l'industrie agro-alimentaire qui peut se modéliser comme un RCPSP multi-mode [4]. Chaque produit fini est un projet qui nécessite une ou plusieurs étapes de fabrication à effectuer dans un ordre donné. On note $\mathcal{J}$ l'ensemble des tâches à réaliser (pour l'ensemble des projets) et on considère que chaque machine m est spécialisée (ou éligible) pour la réalisation d'un sous-ensemble $\mathcal{J}_m \subseteq \mathcal{J}$ de tâches. Les modes d'une tâche sont ordonnés par durée de fabrication croissante. Une tâche j est contrainte par une date de début au plus tôt r_j et de fin au plus tard d_j. De plus, les tâches sont soumises à des relations de précédence généralisées minimum et maximum.

Pour être réalisée, chaque tâche mobilise exactement une machine à la fois (qui sont considérées comme des ressources du premier type) ainsi que des ressources additionnelles (second et troisième types) définies selon le mode. Sur les premières ressources $\mathcal{U}_k$, de capacité 1, on a de plus des temps de transition, pour effectuer un réglage ou un nettoyage, notés $s_{jj'k} \forall j, j' \in \mathcal{J}, k \in \mathcal{U}_k$. On a également des ressources cumulatives, dont certaines avec un profil de disponibilité variable : Les ressources du second type, tel que les outils (moules...) et la main d'œuvre spécialisée (boulanger...) nécessaire durant la phase de réglage et la phase de fabrication, appartiennent à un ensemble (noté $\mathcal{U}_r$) de ressources renouvelables cumulatives dont la quantité disponible est fixe au cours de la production. Enfin les ressources du troisième type, les ingrédients (matières premières ou produits semi-finis issus d'autres tâches), sont nécessaires uniquement lors de la phase de fabrication. Elles appartiennent à l'ensemble $\mathcal{U}_n$ de ressources non-renouvelables cumulatives dont la quantité disponible varie au cours de la production.

Pour traiter notre problème, nous utilisons une méthode tabou basée sur les travaux de Klein [2]. Nous présentons donc plusieurs règles de priorité utilisées dans la phase d'initialisation ainsi que les trois opérateurs utilisés dans la phase d'exploration du voisinage.

2 Représentation de la solution et initialisation

Pour représenter notre solution, nous utilisons une liste de priorité, qui attribue un index à chaque tâche, ainsi qu'un mode de fabrication. Par défaut, nous affectons les tâches à leur mode 1 (avec la fabrication la plus rapide). L'ordre des tâches dans la liste modifie les tâches ayant accès au ressources en priorité et donc peut modifier l'ordonnancement qu'elle produit. Pour initialiser la liste de priorité, nous utilisons différentes règles de priorité classiques : Shortest Processing Time, Earliest Starting Time, Earliest Finishing Time, Latest Starting Time, Latest Finishing Time, Minimum Slask Time, Greatest Resource Demand et Most Total Successors ; ainsi que d'autres règles spécifiques à notre problème.

Tout d'abord comme les temps de réglage entre les tâches sont comparables à des distances entre l'exécution des tâches, une règle, notée s, est directement adaptée de la méthode du plus proche voisin. En plus de cette première approche, nous étudions également d'autres règles de priorité. La première sélectionne la tâche qui présente la plus petite somme de temps de changement avec les tâches qui la suivent, afin de minimiser le prochain temps de changement en moyenne. Nous pouvons aussi baser la sélection sur la quantité, absolue ou relative à la quantité totale disponible, de ressources non renouvelables consommées.

Pour réaliser un ordonnancement à partir d'une liste de priorité et des modes attribués, nous suivons le Serial Schedule Generation Scheme (SSGS) défini par Kolish et Hartmann [3]. Nous parcourons la liste et ordonnançons les tâches une à une à la date au plus tôt respectant leur mode de fabrication, les précédences et les disponibilités des ressources qui contraignent le placement. Il est donc nécessaire de vérifier les dates d'insertion possibles. Nous avons identifié 3 groupes de dates de début de phase de réglages (S_j) ou de fabrication (T_j) :

- $S_j = \max\{r_j, C_i - s_{ijk}\ \forall i \in Pred_j\}$ avec C_i la date de complétion de i et k la machine utilisée dans le mode 1. Dans le cas où les ressources ne sont pas contraignantes.
- $S_j = C'_j$ avec j' la dernière tâche utilisant la ressource renouvelable manquante.
- $T_j = l - s_{jj'k}$ avec l la date de livraison de la ressource non-renouvelable manquante et j la dernière tâche utilisant la machine k (machine utilisée dans le mode affecté).

Après avoir construit une liste de tâches avec chaque règle de priorité, et l'ordonnancement correspondant, on garde celui de plus petite date d'achèvement comme solution initiale. Nous utilisons ensuite des opérateurs de voisinage qui modifient localement l'ordre des tâches dans la liste, ce qui permet d'explorer le voisinage d'un ordonnancement au sein d'une méthode tabou.

3 Méthode Tabou

Nous proposons une méthode Tabou basée sur 3 opérateurs de voisinages : **insertion**(j, k) qui insère la tâche j à la position k dans la liste de tâches, **swap**(j, j') qui échange les positions des tâches j et j' dans la liste et **mode**(j, m) qui affecte à la tâche j son mode m. A cause des précédences, certains mouvements d'**insertion** et de **swap** sont interdits afin d'éviter que des tâches soit ordonnancées avant leur prédécesseurs. Les mouvements autorisés sont donc :

- **swap**(i, j) tel que $\max\{pos_h : \forall h \in Pred(i)\} < pos_j < \min\{pos_h : \forall h \in Succ(i)\}$ et $\max\{pos_h : \forall h \in Pred(j)\} < pos_i < \min\{pos_h : \forall h \in Succ(j)\}$
- **insertion**(j, k) tel que $\max\{pos_h : \forall h \in Pred(j)\} < k < \min\{pos_h : \forall h \in Succ(j)\}$

Nous rendons tabou les mouvements récents ainsi que leurs mouvements inverses. En cas de stagnation de la méthode, nous diversifions la recherche en recommençant à partir d'une solution admissible issue d'une autre règle de priorité. Les premiers résultats sont prometteurs et nous encouragent à poursuivre les expérimentations entamées en affinant l'utilisation des opérateurs et en testant différents paramétrages sur un plus grand nombre d'instances.

Références

[1] Eva Epoy. Recherche Tabou pour le Multi-mode Resource Constrained Project Scheduling Problem. Rapport de master. Recherche opérationnelle [cs.RO]. 2022. ⟨dumas-03843949⟩

[2] Robert Klein. Scheduling of Resource-Constrained Projects Springer, Boston, MA. 2000.

[3] R. Kolisch and S. Hartmann. Heuristic algorithms for the resource-constrained project scheduling problem : Classication and computational analysis. Węglarz J. (eds) Project Scheduling. International Series in Operations Research Management Science. Springer, Boston, MA", 1999.

[4] Carvin, Baptistin, Odile Bellenguez, and Guillaume Massonnet. Modèle pour un problème d'ordonnancement de type RCPSP multi-mode avec précédences généralisées. 23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision. 2022.