Ordonnancement cumulatif multi-ressources de tâches à intensités et durées variables avec fenêtres de temps

Christian Artigues[1], Emmanuel Hébrard[1], Alain Quilliot[2] , Hélène Toussaint[2]

[1] LAAS-CNRS, Université de Toulouse, CNRS, INP, UPS, Toulouse, France

{artigues,ehebrard}@laas.fr

[2] LIMOS CNRS/UCA, Clermont-Ferrand France

quilliot@isima.fr, helene.toussaint@uca.fr

Mots-clés : *Ordonnancement cumulatif, multi-ressources, intensités variables, fenêtres de temps, complexité, procédure de séparation et d'évaluation, génération de coupes, insertion.*

Présentation, formulation et complexité du problème

Ce travail considère un problème $\mathcal{P}$ d'ordonnancement d'un ensemble de tâches $\mathcal{J}$ à durées et intensités variables. Chaque tâche $i \in \mathcal{J}$ est associée à un travail total à réaliser P_i, une date de lancement r_i et une date d'échéance d_i. Un ensemble $\mathcal{R}$ de ressources de type cumulatif est considéré, chaque ressource $k \in \mathcal{R}$ possédant une capacité C_k. A chaque tâche i est associé un ensemble de resources $\mathcal{R}_i$ de telle sorte que la tâche doit être exécutée simultanément sur toutes les ressources de $\mathcal{R}_i$ avec une intensité $s_i \in [s_i^{\min}, s_i^{\max}]$ et pendant une durée p_i, toutes deux variables mais devant vérifier $p_i s_i = P_i$. $\mathcal{J}_k$ est l'ensemble des tâches i telles que $k \in \mathcal{R}_i$. La somme des intensités des tâches en cours d'exécution à chaque instant sur une resource k ne doit pas excéder la capacité R_k. L'objectif est la maximisation de la marge de sécurité, c'est à dire l'écart entre la date de fin de la tâche et sa date d'échéance, ce qui est équivalent à minimiser un plus grand retard négatif. Nous proposons un programme non linéaire en nombres entiers pour modéliser le problème. Il est basé les variables continues de début S_i, de fin C_i d'intensité s_i des tâches i mais aussi sur des variables de flots $\Phi_{i,j}^k$ indiquant la quantité de ressource k transférée de la tâche i vers la tâche j si i finit avant que j ne commence, ce qui est indiqué par une variable binaire $x_{i,j}$. Si 0 et $|\mathcal{J}|+1$ représente une tâche initiale (resp. finale) de durée nulle et d'itensité R_k, le modèle s'écrit comme suit. On remarque la contrainte non linéaire (3) de lien entre la date de début d'une tâche, sa date de fin et son intensité.

$$\mathcal{P}: \text{maximiser} \quad z \tag{1}$$
$$\text{s.t.} \quad z \le d_j - C_j \qquad \forall j \in \mathcal{J} \tag{2}$$
$$C_j = S_j + P_j/s_j \qquad \forall j \in \mathcal{J} \tag{3}$$
$$r_j \le S_j \le C_j \le d_j \qquad \forall j \in \mathcal{J} \tag{4}$$
$$s_j^{\min} \le s_j \le s_j^{\max} \qquad \forall j \in \mathcal{J} \tag{5}$$
$$\sum_{j \in \mathcal{J} \cup \{|\mathcal{J}|+1\}} \Phi_{0,j}^k = \sum_{j \in \mathcal{J} \cup \{0\}} \Phi_{j,|\mathcal{J}|+1}^k = R_k \qquad \forall k \in \mathcal{R} \tag{6}$$
$$\sum_{i \in (\mathcal{J}_k \cup \{0\}) \setminus \{j\}} \Phi_{i,j}^k = \sum_{i \in (\mathcal{J}_k \cup \{|\mathcal{J}|+1\}) \setminus \{j\}} \Phi_{j,i}^k = s_j \qquad \forall k \in \mathcal{R}, \forall j \in \mathcal{J}_k \tag{7}$$
$$\Phi_{i,j}^k \le \min(s_i^{\max}, s_j^{\max}) x_{i,j} \qquad \forall i,j \in \forall k \in \mathcal{R}, \forall i,j \in \mathcal{J}_k, i \ne j \tag{8}$$
$$S_j \ge C_i - (d_i - r_j)(1 - x_{i,j}) \qquad \forall i,j \in \mathcal{J}, i \ne j \tag{9}$$
$$x_{i,j} \in \{0,1\} \qquad \forall i,j \in \mathcal{J}, i \ne j \tag{10}$$

Nous établissons (Tab. 1) une cartographie de complexité de cas particuliers du problème à une seule ressource en fonction de la présence d'intensité minimales ($s^{\min}$), maximales ($s^{\max}$) des dates de lancement (r) et d'échéance (d).

Contraintes	Complexité	Contraintes	Complexité
$s^{\max}$	P	$s^{\min}, s^{\max}$	NP-difficile
$r, s^{\min}$	P	$r, s^{\max}$	NP-difficile
$s^{\min}, d$	P	$s^{\max}, d$	NP-difficile
		r, d	NP-difficile

Tab 1. Complexité du problème à une ressource

Résolution par méthodes exactes et approchées

Notons qu'en raison des non linéarités, le problème peut ne pas admettre de solution optimale (ni même réalisable) entière (ni même rationnelle). Considérons l'instance à une ressource de capacité 2 et deux tâches définie par $r_1 = 3$, $P_1 = 6$, $d_1 = 11$, $r_2 = 4$, $P_2 = 4$, $d_2 = 8$, $s_1^{\min} = s_2^{\min} = 0$, $s_1^{\max} = 1$ et $s_2^{\max} = 2$. La solution optimale unique démarre la tâche 1 à $S_1 = 3$ avec une intensité de $s_1^* = \frac{9-\sqrt{33}}{4}$ et la tâche 2 à $S_2 = 4$ avec $s_2^* = \frac{\sqrt{33}-1}{4}$. La marge de sécurité optimale est $z^* = \frac{48-8\sqrt{33}}{9-\sqrt{33}}$. Toutefois la contrainte (3) est convexe, ce qui autorise différentes approches de résolution. Le problème $\mathcal{P}$ peut ainsi être résolu comme un problème linéaire en nombre entiers par séparation et génération de coupes en remplaçant dans (1–10) les contraintes (3) par l'ensemble infini de contraintes $C_j - S_j \geq (-s_j + w)P_j/w^2 + P_j/w \quad \forall j \in \mathcal{J}, \forall w \in \mathbb{Q}$. Par ailleurs en fixant les variables $x_{i,j}$, le problème, noté $\mathcal{P}_x$, devient un programme linéaire et peut être résolu sans branchement par génération de coupes. Considérons une relaxation de $\mathcal{P}$, notée $\tilde{\mathcal{P}}$ qui autorise la préemption et une intensité différente à chaque redémarrage d'une tâche et possiblement plus faible que $s_i^{\min}$. Pour une marge z fixée, on cherche une solution réalisable en modifiant les dates d'échéances $d_i \leftarrow d_i - z$ pour toute tâche i, ce qui donne le problème $\tilde{\mathcal{P}}(z)$. Considérons l'ensemble ordonné $(t_q)_{q \in Q}$ des dates de lancement et d'échéance modifiées différentes. On peut montrer que la préemption peut être limitée à n'intervenir qu'à ces dates et obtenir le programme linéaire suivant pour résoudre $\tilde{\mathcal{P}}(z)$, basée sur la variable s_i^q donnant l'intensité de la tâche i dans l'intervalle $[t_q, t_{q+1}]$:

$$\text{maximiser} \sum_{j \in \mathcal{J}} \sum_{q \in Q} (t_{q+1} - t_q) s_j^q \tag{11}$$

$$\text{s.t.} \quad \sum_{q \in Q} (t_{q+1} - t_q) s_j^q = P_j \qquad \forall j \in \mathcal{J} \tag{12}$$

$$\sum_{j \in \mathcal{J}_k} s_j^q \leq R_k \qquad \forall k \in \mathcal{R}, \forall q \in Q \tag{13}$$

$$s_j^q = 0 \qquad \forall j \in \mathcal{J}, \forall q \in Q, t_{q+1} \leq r_j \wedge t_q \geq \tilde{d}_j \tag{14}$$

$$0 \leq s_j^q \leq s_j^{\max} \qquad \forall j \in \mathcal{J}, \forall q \in Q \tag{15}$$

Le problème $\tilde{\mathcal{P}}(z)$ est réalisable si ce programme linéaire a une solution optimale égale à $\sum_{j \in \mathcal{J}} P_j$. Une borne supérieure de la marge optimale de $\mathcal{P}$ peut être obtenue en trouvant le z maximal par dichotomie. Nous proposons de comparer deux méthodes exactes pour résoudre $\mathcal{P}$: la résolution de (1–10) par séparation et génération de coupe et la résolution par une procédure de séparation et d'évaluation spécifique. Celle ci définit une structure de nœud de l'arbre de recherche par un quintuplet $\Big((\hat{r}_j)_{j \in \mathcal{J}}, (\hat{d}_j)_{j \in \mathcal{J}}, \mathcal{M}, (\underline{m}_j)_{j \in \mathcal{M}}, (\overline{m}_j)_{j \in \mathcal{M}}\Big)_n$ les deux premiers éléments étant les dates de lancement et d'échéances modifiées par branchement et les deux derniers éléments définissant des intervalles "milieu" dans lesquels certaines tâches (ensemble $\mathcal{M}$) sont contraintes d'être exécutées, également par branchement. A chaque nœud, une borne inférieure est obtenue en résolvant la relaxation préemptive du problème $\tilde{\mathcal{P}}$. Des heuristiques sont également définies, tentant de trouver une solution réalisable induite par la solution de la relaxation $\tilde{\mathcal{P}}$ et pouvant améliorer la borne inférieure au cours de la recherche arborescente. Notamment, on peut déduire de la solution relâchée un ordre relatif des tâches et résoudre le problème $\mathcal{P}_x$. Enfin, une heuristique basée sur l'insertion progressive de tâches dans le flot selon un ordre donné avec un algorithme polynomial est proposée pour initialiser la borne inférieure. Les méthodes sont comparées sur des instances de $\mathcal{P}$ provenant de jeux de données correspondant à des problèmes de planification d'évacuation en cas de feux de forêts [**?**] étudiés dans le cadre du projet européen H2020 RISE 691161 GEO-SAFE.

Références

[1] Alain Quilliot, Christian Artigues, Emmanuel Hebrard, and Hélène Toussaint. Models and algorithms for natural disaster evacuation problems. In *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019, Leipzig, Germany, September 1-4, 2019.*, pages 143–146, 2019.