Bornes pour un problème d'ordonnancement avec allocation et stockage d'énergie et coûts linéaires par morceaux

Nabil Absi, Christian Artigues, Safia Kedad-Sidhoum, Sandra Ulrich Ngueveu, Félix Goupil

Bornes pour un problème d'ordonnancement avec allocation et stockage d'énergie et coûts linéaires par morceaux

Nabil Absi[1] Christian Artigues[2] Safia Kedad-Sidhoum[3]
Sandra U. Ngueveu[2] Felix Goupil[2]

[1]Mines Saint-Etienne and UMR CNRS 6158 LIMOS, Gardanne, France
[2]LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
[3]CNAM-CEDRIC, Paris, France

Mots-clés : *Ordonnancement, énergie, stockage, coûts linéaires par morceaux, programmation mixte, lot-sizing, programmation dynamique, génération de colonnes, matheuristique.*

Présentation du problème et programme linéaire en variables mixtes

Ce papier traite d'un problème d'ordonnancement d'un ensemble $\mathcal{A}$ de tâches préemptives, sur un horizon $\mathcal{T}$ discrétisé en périodes. Chaque tâche doit être exécutée à l'intérieur d'une fenêtre temporelle $[r_i, d_i[$ sur une durée p_i, $\forall i \in \mathcal{A}$, et consommatrices d'électricité selon une demande individuelle b_i sur chaque période d'exécution de la tâche. L'énergie demandée par l'ensemble des tâches dans une période est la somme des énergies demandées individuellement par les tâches. Elle peut être apportée au moyen de deux sources d'énergie. Une source d'énergie réversible, de type batterie, peut stocker et restituer de l'énergie avec un stock initial s_0 et une capacité de stockage $\overline{Q}$. Une source d'énergie non-réversible, de type fournisseur d'électricité, ne peut que fournir de l'énergie moyennant un coût linéaire par morceaux dépendant du temps $\rho_t(b)$ à payer sur chaque période de temps t pour une consommation totale b. Dans un travail précédent nous avons proposé des approches de programmation linéaire en nombres entiers pour un problème (ORDO) similaire mais sans source réversible. Nous avons montré que le problème était déjà NP-difficile au sens fort et proposé un algorithme de Branch & Price (BP) pour le résoudre efficacement [2]. A l'inverse, lorsque l'ordonnancement des tâches est fixé, on obtient une demande énergétique prédéterminée. Le problème de couverture de la demande de chaque période par les deux sources est équivalent à un problème de lot-sizing (LS), NP-difficile au sens faible. Un algorithme de programmation dynamique (PD) en $O(|\mathcal{T}|^2\bar{q}\bar{d})$ a été proposé pour des demandes entières où $\bar{d}$ désigne la demande moyenne et $\bar{q}$ est le nombre moyen de points de rupture de la fonction ρ.

Un programme linéaire en variables mixtes (PLVM) de taille pseudo-polynomiale peut être proposé en utilisant les variables de décisions binaires x_{it}, $\forall i \in \mathcal{A}$, $t \in \mathcal{T}$ signifiant que la tâche i est en cours d'exécution à la période t. Une variable continue $s_t \geq 0$ donne le stock dans la ressource réversible à la période t et une variable continue w_t donne la quantité d'énergie fournie par la source non-réversible à la période t. Avec ces variables on doit minimiser la fonction $\sum_{t\in\mathcal{T}} \rho_t(w_t)$. Une contrainte de respect des durées s'écrit $\sum_{t=r_i}^{d_i-1} x_{it} \geq p_i$, $\forall i \in \mathcal{A}$. Le respect dans la demande et l'allocation d'énergie pour toute période $t \in \mathcal{T}$ s'écrit $w_t - \sum_{i\in\mathcal{A}} b_i x_{it} + s_{t-1} - s_t = 0$. On impose par ailleurs que le stock initial soit restitué ($s_{|\mathcal{T}|} \geq s_0$) et que le stock ne dépasse pas la capacité ($s_t \leq \bar{Q}$, $\forall t \in \mathcal{T}$). Enfin on pose $x_{it} = 0$ pour tout $t \notin [t_i, d_i[$.

Matheuristique

Le PLVM montre ses limites et ne parvient pas à résoudre optimalement plusieurs instances à 30 et 60 tâches en moins de 600s. Or comme nous disposons de méthodes exactes efficaces pour résoudre exactement (ORDO) et (LS) séparément, nous proposons une matheuristique basée sur l'enchaînement itératif des deux résolutions. Nous commençons (étape 1) par résoudre (ORDO) sans stockage, ce qui fixe une demande énergétique par période. Ensuite (étape 2), nous résolvons (LS) par programmation dynamique. Avant de revenir à l'étape 1, nous modifions les fonctions ρ_t afin de tenir compte des entrées/sorties dénergie dans la source réversible : si à une période t la batterie fournit de l'énergie ($\Delta_t = s_{t-1} - s_t > 0$) alors nous décalons la fonction ρ_t à droite de Δ_t et créons un morceau $[0, -\Delta_t]$ de coût constant $\rho_t(0)$ pour modéliser la gratuité des Δ_t unités. Si à l'inverse $\Delta_t < 0$ la fonction est décalée à gauche de Δ_t et la partie négative est ignorée, pour représenter le coût minimum $\rho_t(-\Delta_t)$ à payer. L'étape 1 est ainsi résolue à nouveau et le processus est répété jusqu'à ce que le coût ne soit pas amélioré d'une itération sur l'autre. Pour l'implémentation, des arrondis de la demande doivent être effectués en entrée puis en sortie de (PD) qui nécessite des demandes entières. (ORDO) est résolu soit par (BP) soit par une heuristique que nous proposons. Elle consiste à trier les tâches dans une liste puis à insérer dans cet ordre les p_i tranches de chaque tâche à la meilleure date par rapport à l'augmentation de coût résultante. Ensuite une recherche locale supprime et réinsère toutes les tranches d'une tâche sélectionnée comme étant celle conduisant à la meilleure solution

Une nouvelle formulation étendue

Le principe de la formulation étendue proposée dans [2] définit une variable par ensemble de tâches exécutées simultanément. Sans stockage, le coût d'un ensemble est connu. En présence de stockage, le coût à payer à une période où un ensemble de tâches est exécuté dépend de l'utilisation de la batterie. Aussi, nous proposons une définition modifiée d'une colonne, comme un ensemble de tâches associé à un des points de rupture $k \in \mathcal{K}$ de la fonction ρ_t. La variable continue y_{lkt} donne la quantité d'énergie consommée par l'ensemble l sur le point k à la période t. Le lien entre les variables y_{lkt} et x_{it} est alors donné pour chaque période t par la contrainte $\sum_{l\in\mathcal{L}_t}\sum_{k\in\mathcal{K}_l}(o_k - g^{\text{in}}_{lk} + g^{\text{out}}_{lk})y_{lkt} - \sum_{i\in\mathcal{A}} b_i x_{it} = 0$ où o_k est la quantité d'énergie du point k, g^{in}_{lk} (g^{out}_{lk}) est la quantité d'énergie entrant (sortant) de la source réversible, $\mathcal{L}_t$ les ensembles de tâches compatibles avec t, $\mathcal{K}_l$ les points compatibles avec l. Le coût total est donné par $\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_k \sum_{l\in\mathcal{L}_t} y_{lkt}$ où c_k est le coût du point k.

Résultats

La table de gauche ci dessous montrent, sur un ensemble de 72 instances non résolues optimalement en 600s par le modèle pseudo-polynomial (colonne CPLEX), les résultats de la matheuristiques en terme d'écart à la meilleure borne supérieure, le nombre de solutions réalisables trouvées et le nombre moyen d'itération. Différents nombres d'itérations maximum (10, 6, 3) sont testés dans un temps limite de 600s. La table de droite donne pour quelques instances la borne obtenue par génération de colonnes sur le modèle étendu (le sous problème est un sac-à-dos avec sélection de capacité) en comparaison de la borne à la racine du modèle pseudo-polynomial. Dans les deux cas des bornes inférieures et supérieures meilleures que celles trouvées par CPLEX sont obtenues.

		Matheuristique			CPLEX
		10×60 s	6×100 s	3×200 s	600s
écart	moy	**0.19%**	0.23%	0.42%	2.59%
	min	0 %	0 %	0 %	0.02%
	max	1.45 %	1.07 %	1.82 %	21.26%
réal.	#	55	62	**69**	62
it	moy	6.34	5.13	2.94	/

	génér. col.		CPLEX	
inst	val	cpu(s)	val	cpu(s)
38	**270.192**	304.5s	268.186	**3.59s**
85	**1325.20**	26.03s	1298.29	**0.40s**
134	**6562.83**	421.30s	5660.63	**1.72s**
134	**1059.33**	11.35s	1042.65	**0.23s**

Références

[1] Absi N., C. Artigues, S. Kedad-Sidhoum, S.U. Ngueveu and O. Saadi, Lot-sizing models for energy management, IWLS'2017, p.45-48. 2017

[2] Sandra Ulrich Ngueveu, Christian Artigues, Pierre Lopez Scheduling under a non-reversible energy source : An application of piecewise linear bounding of non-linear demand/cost functions. *Discrete Applied Mathematics* 208 : 98–113, 2016.