



Une méthode exacte pour le problème du jobshop cyclique robuste

Idir Hamaz, Laurent Houssin, Sonia Cafieri

► To cite this version:

Idir Hamaz, Laurent Houssin, Sonia Cafieri. Une méthode exacte pour le problème du jobshop cyclique robuste. ROADEF 2018, 19ème congrès de la société Française de Recherche Opérationnelle et d'Aide à la Décision, Feb 2018, Lorient, France. hal-01737765

HAL Id: hal-01737765

<https://hal-enac.archives-ouvertes.fr/hal-01737765>

Submitted on 19 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Une méthode exacte pour le problème du jobshop cyclique robuste

Idir Hamaz¹, Laurent Houssin¹, Sonia Cafieri²

¹ LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France
{ihamaz, lhoussin}@laas.fr

² ENAC, Université de Toulouse, F-31055 Toulouse, France
sonia.cafieri@enac.fr

Mots-clés : *ordonnancement cyclique, optimisation robuste, branch and bound.*

1 Description du problème

Le problème d'ordonnancement cyclique de base (BCSP) est défini par un ensemble $\mathcal{T} = \{1, \dots, n\}$ de n tâches génériques. Chaque tâche $i \in \mathcal{T}$ a une durée p_i et doit être exécutée une infinité de fois. On note $\langle i, k \rangle$ la $k^{\text{ème}}$ occurrence de la tâche i . Un ordonnancement σ est une affectation des dates de début $t(i, k)$ pour chaque occurrence $\langle i, k \rangle$. Un ordonnancement est dit *périodique* avec un temps de cycle α si

$$t(i, k) = t(i, 0) + \alpha k, \quad \forall i \in \mathcal{T}, \forall k \geq 1. \quad (1)$$

Afin de simplifier la notation, nous définissons $t_i = t(i, 0)$ pour chaque tâche i dans \mathcal{T} . D'après (1), un ordonnancement périodique peut donc être défini seulement avec les dates de début des premières occurrences $(t_i)_{i \in \mathcal{T}}$ ainsi que le temps de cycle α .

Les différentes tâches sont liées par un ensemble de contraintes de précédence dites *contraintes uniformes*. Ces contraintes sont données par :

$$t(i, k) + p_i \leq t(j, k + H_{ij}), \quad \forall i, j \in \mathcal{T}, \forall k \geq 0. \quad (2)$$

où i et j sont deux tâches génériques, et H_{ij} un entier naturel, appelé *hauteur*, qui représente le décalage entre les occurrences des tâches i et j .

L'objectif du problème est de trouver un ordonnancement σ_{opt} qui satisfait les contraintes uniformes (2) et minimise le temps de cycle α .

Un graphe orienté $G = (V, E)$ peut être associé à ce problème. Un nœud du graphe G représente une tâche et un arc (i, j) est présent si une contrainte de précédence existe entre les deux tâches correspondantes. Dans ce cas, l'arc (i, j) , appelé *arc uniforme*, est étiqueté par deux valeurs, une longueur $L_{ij} = p_i$ et une hauteur $H_{ij} \in \mathbb{Z}$.

Le temps de cycle minimum est donné par le poids moyen maximum du graphe G . Celui-ci est donné par

$$\alpha = \max_{c \in \mathcal{C}} \frac{\sum_{(i,j) \in c} L_{ij}}{\sum_{(i,j) \in c} H_{ij}}$$

Où \mathcal{C} est l'ensemble des circuits du graphe G .

Le circuit c donnant le poids moyen maximum est appelé *circuit critique*. Différents algorithmes sur le calcul de circuits critiques peuvent être trouvés dans la littérature [3].

Le problème du jobshop cyclique peut être vu comme un problème d'ordonnancement cyclique de base avec des contraintes de ressources. Nous considérons un ensemble $\mathcal{M} = \{1, \dots, m\}$ de m machines tel que $m < n$. Chaque tâche $i \in \mathcal{T}$ est exécutée sans interruption sur la machine $M_{(i)}$. Les tâches sont organisées sous forme de job et celles appartenant au même job

sont liées par des contraintes de précédence (2). Les tâches s'exécutant sur une même machine sont soumises à des contraintes de ressources qui s'expriment comme suit :

$$\forall i, j \in \mathcal{T} \text{ t.q. } M_{(i)} = M_{(j)}, \forall k, l \in \mathbb{N} : t(i, k) \leq t(j, l) \Rightarrow t(i, k) + p_i \leq t(j, l) \quad (3)$$

De la même manière que le BCSP, le problème du jobshop cyclique peut être modélisé par un graphe. Les arcs uniformes sont construits comme décrit précédemment, et les arcs disjonctifs, i.e., les arcs représentant les contraintes disjonctives, sont construits comme suit : pour chaque paire de tâches i et j exécutées sur une même machine, une paire d'arcs (i, j) et (j, i) est introduite. Dans ce cas, l'arc (i, j) est étiqueté par une longueur $L_{ij} = p_i$ et une hauteur K_{ij} et l'arc (j, i) est par une longueur $L_{ji} = p_j$ et une hauteur K_{ji} . Ces hauteurs sont des variables à déterminer, de plus, nous avons la propriété suivante : $K_{ij} + K_{ji} = 1$.

Résoudre le problème du jobshop cyclique revient à trouver les valeurs des décalages K_{ij} minimisant le temps de cycle α . Autrement dit, trouver un ordre d'exécution des occurrences des tâches s'exécutant sur les mêmes machines de sorte à minimiser le temps de cycle. Notez qu'une fois les décalages événementiels déterminés, le problème revient à résoudre un problème d'ordonnancement cyclique de base.

Dans ce travail, nous considérons que les durées des tâches sont incertaines et appartiennent à un ensemble d'incertitude. En nous basant sur les travaux de Bertsimas et Sim [1], nous considérons l'ensemble d'incertitude suivant :

$$\mathcal{U}^\Gamma = \left\{ (p_i)_{i \in \mathcal{T}} \in \mathbb{R}^n : p_i = \bar{p}_i + \hat{p}_i \xi_i, \forall i \in \mathcal{T}; \xi_i \in \{0, 1\}; \sum_{i \in \mathcal{T}} \xi_i \leq \Gamma \right\}$$

Où \bar{p}_i et \hat{p}_i représentent respectivement la valeur nominale et la déviation de la durée de la tâche i , et Γ un entier positif représentant une borne supérieure sur le nombre de durées des tâches autorisées à dévier de leurs valeurs nominales.

L'objectif du problème est de trouver un ordre d'exécution des tâches sur les différentes machines, i.e., les valeurs des décalages événementiels ainsi que la valeur du temps de cycle α tel que, pour chaque $p \in \mathcal{U}^\Gamma$, il existe un vecteur $(t_i(p))_{i \in \mathcal{T}}$ qui respecte les contraintes de précédence.

2 Approche de résolution

Différentes approches pour la résolution du problème d'ordonnancement cyclique de base où les durées des tâches appartiennent à l'ensemble d'incertitude \mathcal{U}^Γ ont été déjà proposées dans [2]. Parmi ces approches, la version de l'algorithme de Howard adapté à l'ensemble d'incertitude \mathcal{U}^Γ semble être très efficace et insensible à la variation du budget d'incertitude Γ . Afin de résoudre notre problème, nous proposons un algorithme de Branch and Bound. Les branchements sont effectués sur les valeurs possibles des décalages K_{ij} et, une fois ces valeurs fixées, nous déterminons le temps de cycle minimum associé, garantissant la faisabilité pour chaque $p \in \mathcal{U}^\Gamma$ en utilisant l'algorithme de Howard adapté. Des résultats préliminaires sur des instances générées aléatoirement montrent des temps de résolution modérés pour des instances allant de 10 jusqu'à 30 tâches (toutes les instances ont été résolues en moins de 100 secondes) et une insensibilité par rapport à la variation du budget d'incertitude.

Références

- [1] Bertsimas Dimitris and Melvyn Sim. *The price of robustness*. *Operations research*, 52(1) : 35–53, 2004.
- [2] Idir Hamaz, Laurent Houssin, and Sonia Cafieri. Robust Basic Cyclic Scheduling Problem. *Manuscrit soumis pour publication*.
- [3] Dasdan Ali. *Experimental analysis of the fastest optimum cycle ratio and mean algorithms*. *ACM Transactions on Design Automation of Electronic Systems*, 9(1) :385–418, 2004.