Méthodes exactes et approchées pour l'ordonnancement des travaux concurrents sur des machines parallèles multi-ressources

Boukhalfa ZAHOUT, Ameur SOUKHAL, Patrick MARTINEAU

Université de Tours, LIFAT EA 6300, CNRS, ROOT ERL CNRS 7002.
64, Avenue Jean Portalis 37200 Tours, France
{boukhalfa.zahout,ameur.soukhal,patrick.martineau}@univ-tours.fr

Mots-clés : *Ordonnancement Multiagent, Allocation de ressources, Machines parallèles, travaux à intervalle fixe, PLNE, Génération de colonnes, NSGA-II.*

1 Introduction

Dans cette étude, nous nous intéressons à un problème d'ordonnancement multiagent où des compromis doivent être obtenus. L'ensemble des travaux est réparti dans des sous-ensembles disjoints, dits agents. Chaque agent a sa propre mesure de performance sur son sous-ensemble de travaux, celle-ci étant uniquement fonction de l'ordonnancement de ses travaux. Les agents sont tous en concurrence pour l'utilisation des ressources communes. Ces problèmes d'ordonnancement sont aussi appelés *ordonnancement avec agents en compétition* [1].

La démarche proposée est valable pour un nombre d'agents quelconque mais illustrée, dans ce résumé, pour 2 agents pour faciliter la lecture. Soient donc deux agents A et B. Agent A (resp. B) est associé à l'ensemble des n_A (resp. n_B) travaux, noté par $\mathcal{N}^A = \{J_1, J_2, ..., J_{n_A}\}$ (resp. $\mathcal{N}^B = \{J_{n_A+1}, J_{n_A+2}, ..., J_n\}$), où $n = n_A + n_B$. Les n travaux indépendants doivent être ordonnancés sans préemption sur m machines parallèles identiques. Différents types de ressources additionnelles et renouvelables sont nécessaires pour traiter chaque travail. On note par $R_k, k = 1 \dots K$ la quantité disponible de ressource k. On note par r_{jk} la quantité de ressource en k requise pour exécuter le travail $j, j = 1, \dots, n$. Pour chaque travail j, sa date de début s_j et sa date de fin f_j sont fixées ; sa durée opératoire est donc $p_j = f_j - s_j$ (j s'exécute sur toute la durée de l'intervalle $[s_j, f_j]$). w_j est le poids du travail J_j. A un instant t donné, chaque machine peut traiter plusieurs travaux à la fois, sous contrainte de la disponibilité des quantités des ressources requises. Sans perte de généralité, nous supposons que : $s_j \leq f_j$ et $r_{jk} \leq R_k$ pour tout $j = 1, \dots, n$ et $k = 1, \dots, K$. Toutes les données sont supposées entières et positives. L'objectif de chaque agent est de trouver une solution réalisable qui maximise la valeur totale pondérée de ses travaux ordonnancés. Soit x_{ij} une variable binaire égale 1 si le travail J_j est ordonnancé sur la machine M_i ; 0 sinon. Nous notons la valeur totale maximale pondérée des travaux ordonnancés de l'agent A (resp. B) par : $Z^A = \sum_{i=1}^{m} \sum_{j=1}^{n_A} w_j \; x_{ij}$ (resp. $Z^B = \sum_{i=1}^{m} \sum_{j=n_A+1}^{n} w_j \; x_{ij}$).

Dans cette étude, l'approche ε-contrainte est utilisée pour déterminer une solution de Pareto optimale et pour générer le front de Pareto. Selon la notation des problèmes d'ordonnancement multiagents introduite dans [1], les deux problèmes étudiés sont respectivement notés : $Pm|CO, s_j, f_j, r_{jk}, Q_B|\varepsilon(Z^A/Z^B)$ et $Pm|CO, s_j, f_j, r_{jk}|\mathcal{P}(Z^A, Z^B)$, où CO indique le scénario "COMPETITION". Ces problèmes sont $\mathcal{NP}$-difficile, même dans le cas d'une seule machine avec un seul agent [4].

Le problème étudié se rencontre dans certains systèmes de production de biens ou de services. Considérons l'exemple d'un Data center où l'objectif est d'exécuter les travaux soumis par les utilisateurs (agents) dans le respect du contrat de chacun, i.e optimiser la fonction

objectif de chaque agent. Les applications (travaux) doivent être exécutées sur le cluster défini par m machines parallèles identiques. Chaque application est exécutée dans un conteneur virtualisé par le logiciel *Docker*. Les machines possèdent trois types de ressources renouvelables, en quantité limitée, avec R_{CPU}, R_{RAM} et R_{STOCK}. L'exécution d'un J_j, nécessite un certain nombre de *cpu* virtuels $r_{j\,cpu}$, de *ram* virtuelle $r_{j\,ram}$ et d'espace de stockage $r_{j\,stock}$. Le cas d'un seul type de ressource additionnelle (mémoire) avec un seul agent (mono-critère) a fait l'objet d'étude dans [2], où les auteurs ont proposé des méthodes exactes et approchées pour trouver une solution réalisable.

2 Approches de résolution exactes et approchées

Selon une modélisation par un graphe d'intervalle, soit le sous-ensemble maximal des travaux (sommets) ayant une intersection non vide des intervalles de temps d'exécution, noté par $L_h = \{j : \bigcap_{j=1}^{n}[s_j, f_j] \neq \varnothing\}$. On note par $\mathcal{L} = \{L_1, \ldots, L_h, \ldots, L_H\}$ l'ensemble des $H = |\mathcal{L}|$ sous-ensembles maximaux, i.e. L_h n'est inclus dans aucun autre sous-ensemble de sommets. Cependant, deux sous-ensembles maximaux peuvent avoir des sommets en commun. L'ensemble $\mathcal{L}$ peut être déterminé en $O(n^2)$ [3]. En se basant sur l'ensemble $\mathcal{L}$, nous proposons le PLNE suivant :

$$\text{Maximiser :} \quad \sum_{i=1}^{m}\sum_{j=1}^{n_A} w_j\, x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} x_{ij} \leq 1 \qquad j = 1, \ldots, n \tag{2}$$

$$\sum_{j \in L_h} r_{jk} x_{ij} \leq R_k \qquad i = 1, \ldots, m; \quad k = 1, \ldots, K; \quad h = 1, \ldots, H \tag{3}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n_B} w_j\, x_{ij} \geq Q_{\mathcal{B}} \tag{4}$$

$$x_{ij} \in \{0, 1\} \qquad i = 1, \ldots, m; \quad j = 1, \ldots, n \tag{5}$$

Les contraintes (2) indiquent que le travail est exécuté sur une seul machine, sinon il est rejeté. Les contraintes (3) définissent les capacités maximales de chaque type de ressource R_k par machine. La contrainte (4) exprime l'approche ε-contrainte.

Un *Branche&Price* basé sur la génération de colonne et un NSGA-II ont été développés.

Toutes ces méthodes ont été testées sur des instances générées aléatoirement. Par exemple, pour une instance de 200 travaux, 15 machines et 3 types de ressources, 13mn (resp. 2sec) sont en moyenne nécessaires pour le PLNE (resp. B&P) pour calculer une solution de Pareto optimale. Le NSGA-II donne de très bons résultats lorsque les poids des travaux sont identiques ($\forall j, w_j = 1$). D'autres tests sont en cours, particulièrement pour des poids quelconques.

Tous ces résultats peuvent être généralisés au cas de plusieurs agents. L'ensemble des résultats expérimentaux sera présenté lors du congrès.

Références

[1] A. Agnetis, J.-C. Billaut, S. Gawiejnowicz, D. Pacciarelli, and A. Soukhal. *Multiagent Scheduling, Models and Algorithms*. Springer-Verlag, Berlin Heildelberg New York, 2014.

[2] E. Angelelli, N. Bianchessi, and C. Filippi. Optimal interval scheduling with a resource constraint. *Computers & Operations Research*, 51 :268–281, 2014.

[3] M. Habib, R. McConnell, Ch. Paul, and L. Viennot. Lex-bfs and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234(1-2) :59–84, 2000.

[4] B. Zahout, A. Soukhal, and P. Martineau. Fixed jobs scheduling on a single machine with renewable resources. In *MISTA'2017*, pages 1–9, Kuala Lumpur, Malaysia, 2017.