Decomposition approach for fixed jobs multi-agent scheduling problem on parallel machines with renewable resources

B. Zahout, A. Soukhal and P. Martineau

Université de Tours, France
LIFAT EA 6300, CNRS, ROOT ERL CNRS 7002
boukhalfa.zahout,ameur.soukhal,patrick.martineau@univ-tours.fr

Keywords: Competing multi-agent scheduling, fixed job scheduling, resource allocation, parallel machine, MILP, decomposition approach, ε-constraint.

1 Introduction

A scheduling problem involving several actors, where each has its own decision-making autonomy, in charge of executing its subset of jobs on the same resources (the jobs are competing for the use of the same machines), can be assimilated to a multi-agent scheduling problem, where a new type of compromise must be achieved. We define the term "agent" as an entity associated with a subset of jobs. Each agent aims at minimizing his own criterion that depends only on his own jobs. These agents are in competition since they share the same resources (Agnetis, Billaut, Gawiejnowicz, Pacciarelli and Soukhal 2014) We are therefore looking for the best compromised solutions. These problems are close to the multi-objective optimization problems and cooperative game theory (Agnetis, Pascale and Pranzo 2009).

To illustrate our approaches, we focus on the case of two agents A and B. In this paper, all the developed results can be generalized to L agents. Agent A (resp. B) is associated with the set of n_A (resp. n_B) jobs, denoted by $\mathcal{N}^A = \{J_1, J_2, ..., J_{n_A}\}$ (resp. $\mathcal{N}^B = \{J_{n_A+1}, J_{n_A+2}, ..., J_n\}$), where $n = n_A + n_B$.

The n independent jobs should be scheduled without preemption on m identical parallel machines. Additional renewable resources are however necessary to process each job. Several types of such resources, denoted $R_k, k = 1 \ldots K$, are needed. Hence, at execution time of job j, r_{jk} units of resource R_k are required. For each job j, the start date s_j and its finished date f_j $(j = 1, \ldots, n)$ are fixed where its processing time $p_j = f_j - s_j$. w_j is the weight of job j. Dealing with each type of resources k, the machine can process more than one job at a time provided the resource consumption does not exceed a given value R_k $(k = 1 \ldots K)$. We assume that the machines are continuously available during the time interval $[0, \infty)$. All data are assumed positive integers. Without lost of generality, we assume that: $s_j < f_j$ and $r_{jk} \leq R_k$ for all $j = 1, \ldots, n$ and $k = 1, \ldots, K$. The objective of each agent is to find a feasible solution with maximum total weighted number of scheduled jobs. Let x_{ij} be the binary decision variable where $x_{ij} = 1$ if machine i processes job J_j; 0 otherwise. We denote the maximum total weighted number of scheduled jobs of Agent A and of agent B by: $Z^A = \sum_{i=1}^{m} \sum_{j=1}^{n_A} w_j \; x_{ij}$ and $Z^B = \sum_{i=1}^{m} \sum_{j=n_A+1}^{n} w_j \; x_{ij}$, respectively.

In this study, ε-constraint approach is used to determine one Pareto optimal solution (one objective function is minimized and the other one is bounded by Q). By modifying the value Q iteratively, it is possible to obtain the whole set of strict Pareto optimal solutions.

According to the three-field notation of multiagent scheduling problems introduced in (Agnetis, Billaut, Gawiejnowicz, Pacciarelli and Soukhal 2014), the addressed problems are denoted by $Pm|CO, s_j, f_j, r_{jk}, Q_B|\varepsilon(Z^A/Z^B)$ (computation of one Pareto optimal solution) and $Pm|CO, s_j, f_j, r_{jk}|\mathcal{P}(Z^A, Z^B)$ (computation of the optimal Pareto front).

This problem is $\mathcal{NP}$-hard even if only one agent is considered (mono-criterion case) (Zahout, Soukhal and Martineau 2017).

The studied problem can be met in a Data center for example, where the goal is to optimize the objective function of each user (agent). Jobs (applications) submitted by the users should be executed on the cluster defined by m identical parallel machines. Each application is executed in one container virtualized by Docker software, for example. The machines own certain limited types of renewable resources CPU, MEMORY and STORAGE, with capacities Qu_1 of CPU, a certain quantity of memory Qu_2 and a certain storage capacity Qu_3. In this case, to execute $Application_j$, a number of virtual CPUs r_{j1}, virtual memory r_{j2} and hard drives r_{j3} are needed.

The mono-criterion case has been addressed in (Angelelli, Bianchessi and Filippi 2014) where the authors consider only one additional resource (memory) and develop exact and heuristics methods to determine one feasible solution. In the context of grid computing, (Cordeiro, Dutot, Mounié and Trystram 2011) consider the multi-agent scheduling problem with global objective function. Each agent (including the global agent who is dealing with whole set of jobs) aims to minimize his makespan. They study the organizations that share clusters to distribute peak workloads among all the participants. The authors propose a 2-approximation algorithm for finding collaborative solutions.

2 Exact methods

To compute an optimal Pareto solution, we propose an integer programming formulation. Unfortunately, the linear relaxation of such a model is rather poor. For this reason, we develop a `Dantzig-Wolfe` decomposition scheme leading to a `Branch and Price` based on `column generation` scheme. Firstly, let us introduce the definition of the Maximal Subsets.
Maximal Subsets: Let L be the subset of overlapping jobs where $\bar{L} = \{j : \bigcap_{j\in L}[s_j, f_j) \neq \emptyset\}$. L is maximal if it is not included in any other subset of overlapping jobs. Let $\mathcal{L}$ be the set of all maximal subsets, $\mathcal{L} = \{L_1, \ldots, L_h, \ldots, L_H\}$. Dealing with L_h we can choose arbitrarily a sample time t_h that belongs to the processing interval of every job in L_h, i.e $t_h \in \bigcap_{j\in L_h}[s_j, f_j)$. There is a total ordering of the maximal subsets with respect to sample times. According to this ordering, when we pass from a maximal subset to the next one, at least one job finishes its processing and at least one new job starts its processing. As a consequence, there are at most $(n = n_A + n_B)$ maximal subsets. Hence, we have: $1 \leq H \leq n$ and the maximal subsets can be efficiently detected in $O(n^2)$ by using interval graph recognition algorithm introduced in (Habib, McConnell, Paul and Viennot 2000).

2.1 Integer programming formulation

Based on maximal job subsets, we propose the following integer linear programming (MILP) where x_{ij} is a binary variable equal to 1 if machine i processes job J_j; 0 otherwise.

Constraints (2) allow job j to be assigned to at most one machine. Constraints (3) allow the assignment of at most R_k resources from machine i to the jobs. The constraint (4) express the ε-approach bounds.

MILP has mn binary variables and $n + mKH + 1$ constraints, with $1 \leq H \leq n$.
Remarks: If $H = n$ then every maximal job subset contains only one job, and the problem is trivial to solve; And if $H = 1$ then all jobs overlap and the problem reduces to a multidimensional knapsack problem MKP (Martello 1990).

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

$$\text{subject to:} \quad \sum_{i=1}^{m} x_{ij} \leq 1 \qquad j=1,\ldots,n \tag{2}$$

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

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

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

Given Q_B, to compute a strict Pareto solution, we first solve $Pm|CO, s_j, f_j, r_{jk}, Z^B \geq Q_B|Z^A$. Let $(\hat{Z^A}, \hat{Z^B})$ be the obtained optimal solution. We then solve the inverse problem $Pm|CO, s_j, f_j, r_{jk}, Z^A \geq \hat{Z^A}|Z^B$. The computed solution is then optimal Pareto solution, denoted by $(\hat{Z^A}, \hat{Z^{B\prime}})$.

Decomposition approach: We cannot expect the linear relaxation of MILP is good, since it neglects the fact that the resource units are divided among the different machines. We then apply a `Dantzig-Wolfe` decomposition scheme to model MILP, following the *Caprara et al.*'s approach used to solve Resource Allocation Problem *RAP* (Caprara, Furini and Malaguti 2013). The master problem of the resulting column generation approach may have a huge number of columns, so a branch-and-price approach for its solution seems reasonable. These algorithms are essentially branch-and-bound algorithms where, at each node of the search tree, variables (columns) of the problem are generated by applying the column generation technique to address the linear relaxation of the problem, eventually augmented by branching constraints (see (Desaulniers, Desrosiers and Solomon 2006) for a complete survey of column generation methods).

3 Pareto set enumeration

ε-constraint approach with different values of $Q_B \in \{0,\ldots,Q_B\}$ is used to generate the set of strict Pareto solutions. With each value of Q_B we compute $(\hat{Z^A}, \hat{Z^{B\prime}})$, and we add this solution to the set of strict solutions $\mathcal{S}$. We then set $Q_B = \hat{Z^{B\prime}} + 1$ and iterate. If no feasible solution is obtained then stop and $\mathcal{S}$ is the exact Pareto front.

Proposition 1. The number of strict Pareto optimal solutions is bounded by $O(W)$ where $W = max(\sum_{j\in\mathcal{N}^A} w_j; \sum_{j\in\mathcal{N}^B} w_j)$.

4 Computational experiments

We implement our algorithms in $C++$ language and experiments have been driven on a workstation with a 2.2 Ghz Intel Core i7 processor and 16 GB of memory and a time limit of 3600 seconds (1 hour). We use IBM ILOG CPLEX Optimization Studio version 12.8 to solve the MILP, Master and Pricing Model.

Algorithms under study have been carried out with 60 instances (adapted benchmark proposed of Angelelli et al. (Angelelli, Bianchessi and Filippi 2014)). Number of jobs $n \in \{100, 120, 150, 180, 200\}$, number of parallel machines $m \in \{4, 7, 10, 15\}$, without lost of generality, we normalize the capacity of each renewable resource to $R_1 = R_2 = R_3 = \{2, 4, 6\}$, for a total of $5\times4\times3 = 60$ instances. 50% of n are jobs of agent A. By choosing 50% of jobs for each agent, we are therefore interesting in solving the most difficult problems.

Fig. 1 gives the computational time in seconds: a simple call to CPlex using the MILP and the Branch&Price to compute one strict Pareto optimal solution. We can conclude that Branch&Price is very efficient. For example, with 200 jobs, 15 machines and 3 types of resources, Branch&Price needs 2 sec where MILP model needs 768 sec.

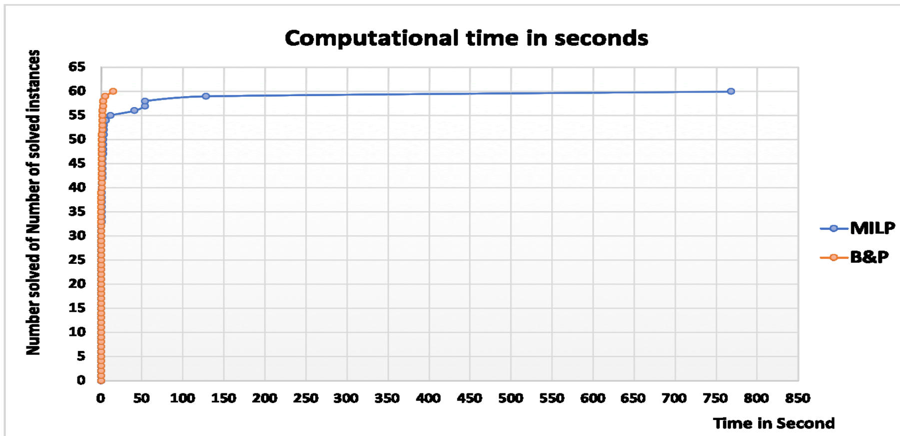

Fig. 1. Computational time in seconds

The proposed methods can be easily generated to the case of more than two agents.

This work is in progress and it will be interesting to develop lower bound to speed up the convergence of the Branch&Price. Other methods such as Meta-heuristics and heuristics providing high-quality solutions with a low computational running time to solve large size instances will be developed.

References

Martello, S., 1990, "Knapsack problems: algorithms and computer implementations", *Wiley-Interscience series in discrete mathematics and optimization.*

Habib, M., R. McConnell, C. Paul, L. Viennot, 2000, "Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing", *Theoretical Computer Science*, Vol. 234, pp. 59-84.

Desaulniers, G., J. Desrosiers, M.M. Solomon, 2006, "Column generation", *Springer Science & Business Medi*, Vol. 5.

Agnetis, A., G. de Pascale, M.Pranzo, 2009, "Computing the Nash solution for Scheduling Bargaining Problems", *International Journal of Operational Research*, Vol. 1, pp. 54-69.

Caprara, A., F. Furini, E.Malaguti, 2013, "Uncommon Dantzig-Wolfe reformulation for the temporal knapsack problem", *INFORMS Journal on Computing*, Vol. 25, pp. 560-571.

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

Angelelli E., N. Bianchessi, C. Filippi, 2014, "Optimal interval scheduling with a resource constraint", *Computers & Operations Research*, Vol. 51, pp. 268-281.

Cordeiro D., P.-F. Dutot, G. Mounié, D. Trystram, 2011, "Tight Analysis of Relaxed Multi-Organization Scheduling Algorithms", *In Proceedings of the 25th IEEE International Parallel & Distributed Processing Symposium (IPDPS), IEEE Computer Society*, Anchorage, AL, USA, pp. 1177-1186.

Zahout B., A. Soukhal, P. Martineau, 2017, "Fixed jobs scheduling on a single machine with renewable resources", *MISTA'2017*, Kuala Lumpur, Malaysia, pp. 1-9.