Restricted Assignment Scheduling with Resource Constraints

Gyorgy Dosa[*]   Hans Kellerer (Speaker) [†]   Zsolt Tuza[‡]

1 Introduction

We are given a set $N = \{1, \ldots, n\}$ n of independent jobs that are to be scheduled on m parallel machines $M_1, \ldots, M_m$. In the *Restricted Assignment problem* (RA-problem, for short) each job j can be executed on a specific subset $\mathcal{M}(j)$ of the machines, and on those machines the processing time of job j is p_j. The objective is to minimize the makespan. In the three field notation, we abbreviate this problem by $R|p_{ij} \in \{p_j, \infty\}|C_{\max}$.

Assume that additionally there are μ renewable resources $R_1, \ldots, R_\mu$. Let Λ_k be the set of jobs which require resource R_k, and let λ_k denote the cardinality of set Λ_k, $k = 1, \ldots, \mu$. Job j requires simultaneous availability of all resources in the set $\mathcal{R}(j) \subseteq \{R_1, \ldots, R_\mu\}$ for processing; we denote by ρ_j the cardinality of $\mathcal{R}(j)$, $j = 1, \ldots, n$. Any resource can be used by only one job at any time. It means that two jobs which require the same resource cannot be processed simultaneously. This problem is abbreviated by $R|p_{ij} \in \{p_j, \infty\}, res\mu 11|C_{\max}$. In the following, we will call it *Restricted Assignment with Resources problem*, briefly RAR-problem. The *degree* of the problem is defined as the quantity $B = \max_{j=1,\ldots,n} \rho_j$, that is the maximum number of resources required by a job.

To the best knowledge of the authors, restricted assignment and resources were not considered previously together. The two types of conditions, however, have been studied separately.

The restricted assignment problem can be considered as a special case of the classical unrelated machine problem $R|\cdot|C_{\max}$, where job j on machine M_i has processing time p_{ij}. The paper of Lenstra et al. [3] contains a polynomial-time 2-approximation algorithm for $R|\cdot|C_{\max}$. On the negative side it is proven that one cannot get any worst-case ratio better than $3/2$ for $R|\cdot|C_{\max}$ unless $\mathsf{P} = \mathsf{NP}$ holds.

There are many papers considering Multiprocessor Scheduling With Resources, MPSR for short. If each task requires at most one resource, i.e. $B = 1$, then MPSR with unit-time jobs admits a polynomial-time algorithm for an arbitrary number of processors, even with prescribed release times and deadlines [1]. On the other hand, for a variable number of resources the problem of minimizing maximum lateness becomes NP-hard, still with unit processing times of all the tasks, and already on just two processors [2].

---

[*]`dosagy@almos.vein.hu`. Department of Mathematics, University of Pannonia, H-8200 Veszprém, Egyetem u. 10, Hungary.

[†]`hans.kellerer@uni-graz.at`. Institut für Statistik and Operations Research, Universität Graz, Universitätsstraße 15, 8010 Graz, Austria.

[‡]`tuza@dcs.uni-pannon.hu`. Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, H-1053 Budapest, Reáltanoda u. 13–15, Hungary.

2 Our Results

We prove inapproximability results and design approximation algorithms for the RAR-problem. Our main negative result is that the problem with unit-time jobs is APX-hard, already on three machines. In the case that each job requires only a bounded number of resources, we design approximation algorithms with constant worst-case bound, without any restrictions on processing times. For some special cases (e.g., unit-time jobs with degree $B = 1$) we design optimal algorithms with polynomial running time.

To derive the main negative result, we prove a theorem on graph coloring, which seems to be of interest on its own right, too. It states APX-hardness of the chromatic number on a restricted class of graphs.

Our results can be stated as follows:

Theorem 1 *The RAR-problem with unit-time jobs and two machines can be solved in polynomial time.*

Theorem 2 *The RAR-problem with n unit-time jobs and m machines can be solved in $O((m^3 + m^2 n) \log n)$ time for $B = 1$.*

Theorem 3 *There is a polynomial-time $(2 - 1/m + B)$-approximation algorithm for the RAR-problem on m machines with arbitrary processing times.*

Theorem 4 *There is a PTAS for the RAR-problem with a fixed number of machines and $B = 1$.*

Theorem 5 *The following optimization problems are APX-complete:*

(i) *the* CHROMATIC NUMBER *problem restricted to graphs of independence number 3,*

(ii) *the* MINIMUM CLIQUE COVER *problem restricted to graphs whose clique number is 3,*

(iii) *even more restrictively the* MINIMUM CLIQUE COVER *problem on graphs whose clique number is 3 and maximum degree is 4.*

Theorem 6 *The RAR-problem is APX-hard, even when it is restricted to the following type of instances: there are only three machines, all jobs have unit time, any job can be processed on any machine, and each resource is required only for two jobs.*

References

[1] J. BLAZEWICZ (1979). *Deadline scheduling of tasks with ready times and resource constraints*, Inform. Process. Lett., 8 (2), 60–63.

[2] J. BLAZEWICZ, J. BARCELO, W. KUBIAK AND H. ROCK (1986). *Scheduling tasks on two processors with deadlines and additional resources*, Europ. J. Oper. Res., 26, 364–370.

[3] J. K. LENSTRA, D. SHMOYS AND E. TARDOS (1990). *Approximation algorithms for scheduling unrelated parallel machines*, Math. Progr., 46, 259–271.