Information Processing Letters 116 (2016) 723–728

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On some special cases of the restricted assignment problem

Chao Wang [a,1], René Sitters [b,c,*]

[a] East China University of Science and Technology, 200237 Shanghai, China
[b] VU University Amsterdam, 1081HV Amsterdam, The Netherlands
[c] Centrum Wiskunde & Informatica (CWI), 1098XG Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 21 April 2015
Received in revised form 17 March 2016
Accepted 16 June 2016
Available online 17 June 2016
Communicated by M. Chrobak

Keywords:
Scheduling
Restricted assignment problem
Design of algorithms

ABSTRACT

We consider some special cases of the restricted assignment problem. In this scheduling problem on parallel machines, any job j can only be assigned to one of the machines in its given subset M_j of machines. We give an LP-formulation for the problem with two job sizes and show that it has an integral optimal solution. We also present a PTAS for the case that the M_j's are intervals of the same length. Further, we give a new and very simple algorithm for the case that $|M_j| = 2$ (known as the graph balancing problem) with ratio 11/6.

1. Introduction

A classic algorithm by Lenstra, Shmoy and Tardos [1] gives a 2-approximation for minimizing the makespan on unrelated parallel machines (denoted by $R||C_{\max}$). In this problem, we are given n jobs and m machines and processing times p_{ij} which give the time needed to process job j on machine i. The goal is to assign the jobs to the machines such that the maximum load (total processing time) over all machines is minimized. The same paper shows that approximating the problem within a factor 3/2 is NP-hard. Today, no better upper or lower bound is known and improving one of the two bounds is considered one of the main open problems in the scheduling theory. Even for the so called *restricted assignment problem* no better ratios are known. Here, each job j has a processing time p_j and *processing set* $M_j \subseteq \{1, 2, \ldots, m\}$ and $p_{ij} = p_j$ for $i \in M_j$ and $p_{ij} = \infty$ otherwise.

A breakthrough was made by Svensson [2], who showed that the integrality gap of the configuration LP for the restricted assignment problem is at most 1.942. However, the proof does not give a polynomial time approach for constructing a corresponding schedule. An interesting special case in which $|M_j| \leqslant 2$ for all j was considered by Ebenlendr et al. [3] who called this the *graph balancing problem*. An instance may be seen as a multigraph $G = (V, E)$ where each edge $e_j \in E$ has a weight p_j. The edges should be oriented such that the maximum total weight of incoming edges is minimized. The authors give a 1.75-approximation by LP-rounding and also show that the integrality gap of their LP is 1.75.

Based on the work of Ebenlendr et al. [3], some special cases of graph balancing have been studied. Lee, Leung and Pinedo [4] gave an FPTAS if the graph structure is restricted to a tree. Verschae and Wiese [5] showed that even the configuration-LP has an integrality gap of 1.75 for the graph balancing problem and 2 for the unrelated version where each job may have different job processing times on its two machines.

* Corresponding author at: VU University Amsterdam, 1081HV Amsterdam, The Netherlands.

E-mail addresses: c.wang@mail.ecust.edu.cn (C. Wang), r.a.sitters@vu.nl (R. Sitters).

1 This research done while visiting the VU University Amsterdam and was supported in part by the China Scholarship Council under grant No. 201306740038.

http://dx.doi.org/10.1016/j.ipl.2016.06.007

Glass and Kellerer [6] explored the restricted assignment problem with only two different processing times: 1 and $k \leqslant 2$, and gave an approximation algorithm with factor $2 - 1/k$. In the same paper, they considered the restricted assignment problem with the additional structure that the M_j's are nested and gave a $2 - 1/m$ approximation by list scheduling. For totally ordered sets a 3/2-approximation was designed. Huo and Leung [7] improved the nested case by giving an algorithm with worst-case bound of 5/3, moreover they also studied so called *tree-hierarchical processing sets*. Here, machines are the nodes of a rooted tree and processing sets correspond with a path from some vertex to the root. For the latter problem, they gave a 4/3-approximation algorithm. Muratore et al. [8] improved this nested case by designing a polynomial time approximation scheme (PTAS). Epstein and Levin [9] also studied the nested and tree-hierarchical cases, and they designed a PTAS for the two cases respectively.

For two arbitrary job sizes, Kolliopoulos and Moysoglou [10] gave a 1.883-approximation algorithm for computing the optimal value based on Svensson's [2] result. If the jobs can only be assigned on at most two machines, the ratio reduces to 1.652.

1.1. Our results

We give several results for special cases of the restricted assignment problem. First, we briefly discuss the 2-approximation algorithm for $R||C_{\max}$ given by Shmoys and Tardos [11]. We show that this approach leads to a very simple 1.88-approximation algorithm for the graph balancing problem. Although this ratio is worse than the 1.75-approximation given by Ebenlendr et al. [3], the algorithm and analysis are very easy, given the framework of [11]. Further, we study the restricted assignment problem on intervals. Here, the machines can be ordered such that each processing set is a consecutive set of machines.

We show that the special case of two different processing times s, b can be solved exactly in polynomial time, assuming that $s \leqslant b$ and $b > \text{OPT}/2$. We obtain this by formulating an exact LP, i.e., with an integrality gap of 1. Interestingly, our LP is stronger than the configuration LP which has an integrality gap of at least 3/2 in this case.

As mentioned above, the interval case has been well-studied and several polynomial time approximation schemes are known. We extend these results by giving a PTAS for the problem of equal length intervals and by simplifying the analysis of known approximation schemes by using the framework of [11].

2. Preliminaries

The 2-approximation algorithm for $R||C_{\max}$ by Lenstra, Shmoys, and Tardos [1] is well-known. A slightly different algorithm and analysis was given a few years later by Shmoys and Tardos [11]. They introduced a new rounding technique which does not require the fractional solution to be a vertex of the linear programming relaxation. An advantage is that it can therefore be applied to any fractional assignment and for variants of the problem as we shall do in this paper. Below, we give a short explanation of this rounding technique that we will use several times.

The algorithms in [1] and [11] work as follows. First, a guess T for the optimal value is made. The algorithm produces a schedule of length at most $2T$ if indeed $\text{OPT} \leqslant T$. Using a binary search on T, this will give a 2-approximation algorithm. Denote by $x_{ij} \geqslant 0$ the fraction of job J_j assigned to machine M_i, $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$. If there exists a schedule of length at most T, then the following LP has a feasible solution.

$$\begin{array}{ll} \sum_{i=1}^{m} x_{ij} = 1 & \text{for all } j \\ \sum_{j=1}^{n} x_{ij} p_{ij} \leqslant T & \text{for all } i \\ x_{ij} = 0 & \text{if } p_{ij} > T, \text{ for all } i, j \\ x_{ij} \geqslant 0 & \text{for all } i, j \end{array}$$

In [1], the authors conclude that any extreme optimal LP-solution has at most m fractional jobs and there is a perfect matching of these jobs to machines in the support graph. Hence, this gives a schedule of length at most $2T$. In [11] the approach is as follows. For each machine i, define a complete ordering $\prec_i$ such that $j \prec_i k$ if $p_{ij} \geqslant p_{ik}$. Given a feasible fractional solution x, let $y_i = \lceil \sum_j x_{ij} \rceil$ be the number of jobs (rounded up) assigned to machine i. For each machine, order the fractions assigned to it by $\prec_i$ and call this the LP-schedule. Then split this schedule up into slots such that the fractions in each slot (except for the last) add up to exactly 1. Thus, a job fraction may be split and be assigned to two contiguous slots, and there are y_i slots for machine i. Define a bipartite graph $G = (\{W, V\}, E)$ as follows. For each job j there is a vertex w_j and for each machine i there are vertices $v^i_1, \dots, v^i_{y_i}$. There is an edge between w_j and v^i_z if job j is processed (partially) in the z-th slot of machine i. It follows directly from Hall's theorem that G has a matching that contains all vertices W, i.e., all jobs.

It is easy to see that this matching yields a 2-approximation for $R||C_{\max}$. Consider an arbitrary machine i. For any slot z, let p^i_z be the maximum processing time p_{ij} among all jobs j scheduled in slot z on machine i. (In other words, let $p^i_z = \max\{p_{ij} \mid (w_j, v^i_z) \in E\}$.) Then, the job matched to v^i_z has processing time at most p^i_z. Let L^i_z be the length of slot z on machine i in the LP-schedule. Since jobs are ordered by $\prec_i$, the job matched to v^i_z has processing time at most

$$p^i_z \leqslant L^i_{z-1} \text{ for all } z \geqslant 2. \tag{1}$$

Hence, the total load of machine i in the schedule defined by the matching is at most

$$p^i_1 + \sum_{z=1}^{y_i - 1} L^i_z \leqslant p^i_1 + \sum_{z=1}^{y_i} L^i_z = p^i_1 + \sum_{j=1}^{n} x_{ij} p_{ij} \leqslant 2T.$$

3. A simple 11/6-approximation for graph balancing

We show how the approach by Shmoys and Tardos [11], shown above, can be used to give a simple approximation algorithm for the graph balancing problem with ratio