Discrete Applied Mathematics 164 (2014) 61–67

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Matching with sizes (or scheduling with processing set restrictions)

Péter Biró[a,*], Eric McDermid[b]

[a] Institute of Economics, Hungarian Academy of Sciences, H-1112, Budaörsi út 45, Budapest, Hungary

[b] Department of Computer Science, University of Wisconsin-Milwaukee, United States

ARTICLE INFO

Article history:
Received 27 July 2010
Received in revised form 27 October 2011
Accepted 2 November 2011
Available online 1 December 2011

Keywords:
Couples
Hospitals/Residents problem
Scheduling
Processing set restrictions
Computational complexity
Approximation algorithms

ABSTRACT

Matching problems on bipartite graphs where the entities on one side may have different sizes are intimately related to scheduling problems with processing set restrictions. We survey the close relationship between these two problems, and give new approximation algorithms for the (NP-hard) variations of the problems in which the sizes of the jobs are restricted. Specifically, we give an approximation algorithm with an additive error of one when the sizes of the jobs are either 1 or 2, and generalise this to an approximation algorithm with an additive error of $2^k - 1$ for the case where each job has a size taken from the set $\{1, 2, 4, \ldots, 2^k\}$ (for any constant integer k). We show that the above two problems become polynomial-time solvable if the processing sets are nested.

1. Introduction

In this paper, we investigate bipartite matching problems where the entities on one side may have different sizes. This type of problem can arise in at least two interesting problem domains. The first is the realm of matching markets in which *couples* are present, where a couple is an inseparable pair of agents. The second significant application area is the realm of scheduling problems in which jobs of different lengths need to be allocated to machines for which they are eligible.

The US Navy assignment process involves hundreds of thousands of sailors to be assigned, commands seeking sailors, and detailers who advise both sailors and commands. According to recent studies [27,22,28], the three most important requirements to be satisfied, in order to achieve an optimal assignment of sailors to billets, are the following.

Size of the matching: All the sailors should be matched, and there are certain critical billets that cannot go unfilled. Stability: Sailors should not be forced into assignments that they have neither asked for nor desire. Moreover, there should be no sailor who would prefer to be matched to a particular billet if the commanding officer responsible for that billet would also request this sailor. Presence of couples: The need to assign married couples to the same location.

If we relax the stability condition by requiring only that a sailor should not be matched to a billet which is not acceptable for her/him, then we get a setting that fits our basic problem, *matching with couples*, that is, the problem of finding a maximum size matching in a bipartite graph where each vertex on one side has size 1 or 2 and each vertex on the other side has an integer capacity. We will define this problem more precisely in Section 2.

The presence of couples is quite natural in other two-sided job markets as well. In fact, in some matching schemes, such as the National Resident Matching Program (NRMP) [21] and the Scottish Foundation Allocation Scheme [25] (schemes that allocate junior doctors to hospitals in the US and in Scotland, respectively), couples are allowed to apply for pairs of positions. The reason for this possibility is that a couple may be prepared to accept a pair of positions only if they are in

* Corresponding author. Tel.: +36 309505368; fax: +36 13092650.
E-mail addresses: birop@econ.core.hu (P. Biró), mcdermid@uwm.edu (E. McDermid).

0166-218X/$ – see front matter
doi:10.1016/j.dam.2011.11.003

the same hospital or geographically close, for example. The detailed description of the mechanism underlying the current matching scheme of the NRMP can be found in the paper of Roth [24], and some remaining problems caused by the presence of couples were studied by Klaus et al. [12].

In the above two-sided markets, where the participants on both sides have preferences and they may reach a private agreement outside the matching scheme (in contrast with the military services), the stability of the solution is considered as a first priority in most existing matching schemes. If couples are not present in the market, we have the classical *hospitals/residents* (or *college admissions*) problem, for which a stable matching can always be found in linear time by the algorithm of Gale and Shapley [5]. However, Ronn [23] showed that the problem of deciding whether a stable matching exists, given a hospitals/residents problem with couples, is NP-complete. Manlove and McDermid [15] proved that the hardness result still holds even if each couple accepts an assignment only if they are both allocated to the same hospital.

In other applications, such as the allocation of students to dormitories or the assignment of papers to reviewers [6], the preferences may be on one side only. Here also we can imagine that the entities to be matched have different sizes, since a couple (or an even larger group) may want to be allocated to the same place, and the organisers of a conference may call for different kinds of papers (e.g. short and regular). In both of the applications mentioned it is natural to seek a complete (or maximum-size) solution, such that the *loads* of the dormitories or the reviewers are *balanced*, leading again to our main problem of matching with couples. Alternatively, we obtain the more general problem *matching with sizes* when the sizes are positive numbers (not only from the set $\{1, 2\}$). We note that, if the organisers of the matching scheme take the preferences into account, then as a second priority they may try to find a maximum-weight or a *rank-maximal* matching (among the maximum cardinality load balanced matchings). Further details of the latter appear in [11].

Finally, in many assignment problems there may be no preferences at all. A major area of applications is scheduling where we need to allocate, say, jobs of different lengths to machines. In this paper, we only take into account the eligibility requirements; that is, for each job we suppose that there is a set of machines that are capable of processing that job. In this case, we get the *parallel machine* (or multiprocessor) *scheduling problem with processing set* (or machine eligibility, or job assignment) *restrictions*. The problem that we focus on is to minimise the makespan (i.e., the time at which the processing of the last job is finished). We will define the problem more precisely in Section 2. Meanwhile, we refer to a survey [14] on scheduling with processing set restrictions and some papers [7,8,10,18] that contain results closely related to those presented in this paper (we will specify these in Section 2).

The outline of this paper is the following. In Section 2, we define the matching with sizes problem and the parallel machine scheduling problem with processing sets, and survey the known results and relationship between these two problems. In Section 3, we give an approximation algorithm for the (NP-complete) problem matching with couples, which finds a solution in $O(ne)$ time (n is the number of jobs and e is the number of edges in the graph) with an additive error 1. Also, we extend our algorithm for the more general problem where the lengths of the jobs are taken from the set $\{1, 2, 4, \ldots, 2^k\}$ for some constant integer k, and we show that in this case a solution with an additive error $2^k - 1$ is guaranteed to be found in $O(2^k ne)$ time. Finally, in Section 4, we give a polynomial-time algorithm to solve the latter problem in the case when the processing sets form a nested set system.

2. Problem statements and related results

In this section we first define the problem matching with sizes and the parallel machine scheduling problem with processing set restrictions. Next, we describe the correspondence between these problems and survey the related literature.

Matching with sizes (MS): We are given a bipartite graph $G(U \cup V, E)$, where $U = \{u_1, u_2, \ldots, u_n\}$, $V = \{v_1, v_2, \ldots, v_m\}$ and $e = |E(G)|$. Each $u_i \in U$ has a *size* $s(u_i) \in \mathbb{R}^+$ and each $v_j \in V$ has a *capacity* $c(v_j) \in \mathbb{R}^+$. A matching M is a set of edges such that the capacity constraints are satisfied; i.e., $\sum_{u_i:\{u_i,v_j\}\in M} s(u_i) \leq c(v_j)$ for each $v_j \in V$. The *size of a matching* M is $s(M) = \sum_{\{u_i,v_j\}\in M} s(u_i)$. The problem is to find a matching of maximum size. The decision problem related to an instance of MS, denoted by D-MS, is to decide whether there exists a *feasible* matching M, defined to be a matching which covers U.

Furthermore, given an instance I of MS and a matching M, let $l_M(v_j) = \sum_{u_i:\{u_i,v_j\}\in M} s(u_i)$ be the *load* of v_j. The *load vector* of M is defined as $\bar{l}_M = (l_M(v_1), l_M(v_2), \ldots, l_M(v_m))$. The quality of the feasible matching may be measured by the L_p norm of the load vector.

$$\|\bar{l}_M\|_p = \left(\sum_{v_j\in V} |l_M(v_j)|^p\right)^{1/p}.$$

Note that $\|\bar{l}_M\|_1$ is equal to the size of M, and $\|\bar{l}_M\|_\infty$ is the maximum load in M. Normally we want to maximise the former and minimise the latter.

Scheduling jobs to machines with processing set restrictions: We have a set of jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ and a set of parallel machines $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$. Each job J_j has processing time p_j and a set of machines $\mathcal{M}_j \subseteq \mathcal{M}$ to which it can be assigned, the *processing set* of J_j. (So we suppose that the machines are *identical* in the sense that a job J_j requires the same amount of time on each machine which is eligible for J_j.) In a *schedule* with the above processing set restrictions, each job J_j is assigned to one of the machines in $\mathcal{M}_j$. In this paper, we will focus on the problem of finding a schedule such that the