Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Multiprofessor scheduling[☆]

Gyorgy Dosa[a], Zsolt Tuza[b,c,*]

[a] Department of Mathematics, University of Pannonia, Veszprém, Hungary
[b] Department of Computer Science and Systems Technology, University of Pannonia, Veszprém, Hungary
[c] Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, Hungary

ARTICLE INFO

Article history:
Received 31 May 2015
Received in revised form 22 November 2015
Accepted 27 January 2016
Available online xxxx

Keywords:
Multiprofessor scheduling
Minimum makespan
Algorithmic complexity

ABSTRACT

We introduce a new model called "Multiprofessor Scheduling". This problem is a generalization of a number of previous models. On a set $\mathcal{P} = \{P_1, \dots, P_m\}$ of professors and a set $\mathcal{L} = \{L_1, \dots, L_n\}$ of lectures (with given, equal or different durations), a problem instance is specified by two kinds $\mathcal{C}$ and $\mathcal{C}^*$ of conditions given with a list of pairs in the following way: $(P_i, L_j) \in \mathcal{C}$ means that professor P_i can deliver lecture L_j if it is assigned to him, while $(P_s, L_t)^* \in \mathcal{C}^*$ means that P_s has to be present when L_t is delivered (by any *other* professor who will deliver the lecture). The optimization problem asks for the shortest possible time within which all lectures can be delivered.

In this paper we take the first steps to study the problem. We restrict our investigations here to the offline setting, i.e. all information about the problem instance is given in advance. We consider mainly the case where all lectures have unit length. For this special case we prove that the optimum value together with an optimal schedule can be determined in polynomial time as a function of n if m is fixed, or in time linear in m if n is fixed; but is NP-complete if both m and n are unbounded, even in the restricted case where each lecture can be given by just one specific professor (i.e., when $\mathcal{C} = \{(P_i, L_i) \mid i = 1, 2, \dots, n\}$) and any other P_j (with index in the range $n < j \le m$) is involved in only two conditions of $\mathcal{C}^*$.

We also consider the case of different durations, where the durations are between 1 and b for some integer b. For this special case we give a polynomial-time approximation algorithm with approximation ratio b.

The paper mostly deals with non-preemptive scheduling, but the preemptive scenario is also considered to some extent. Moreover, we introduce and study a third variant that we call interruptive scheduling. It is more restricted than preemptive, and less restricted than non-preemptive.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

We define a new problem, called Multiprofessor Scheduling Problem (or MPS for short). A set $\mathcal{P} = \{P_1, \dots, P_m\}$ of professors and a set $\mathcal{L} = \{L_1, \dots, L_n\}$ of lectures are given. In the terminology of scheduling theory, professors correspond to machines and lectures correspond to jobs. Each lecture has a known duration, denoted by p_j, for $1 \le j \le n$, and any lecture must be delivered exactly once. A problem instance will be specified completely by two kinds $\mathcal{C}$ and $\mathcal{C}^*$ of conditions given with a list of pairs: $(P_i, L_j) \in \mathcal{C}$ means that professor P_i can deliver lecture L_j if it is assigned to him, while $(P_s, L_t)^* \in \mathcal{C}^*$

☆ Research supported in part by TÁMOP-4.2.2.B-15/1/KONV-2015-0004 and by the Hungarian Scientific Research Fund NKFIH/OTKA grant SNN 116095.
* Corresponding author at: Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, Hungary.
E-mail addresses: dosa@almos.vein.hu (G. Dosa), tuza@dcs.uni-pannon.hu (Zs. Tuza).

http://dx.doi.org/10.1016/j.dam.2016.01.035
0166-218X/© 2016 Elsevier B.V. All rights reserved.

Please cite this article in press as: G. Dosa, Zs. Tuza, Multiprofessor scheduling, Discrete Applied Mathematics (2016), http://dx.doi.org/10.1016/j.dam.2016.01.035

means that professor P_s has to be present when L_t is delivered. In this latter case we assume that the lecture L_t will be delivered by some *other* professor, who is assigned to this lecture. We assume that $\mathcal{C} \cap \mathcal{C}^* = \varnothing$, and also that each lecture can be delivered by at least one professor.

In this way, if all lectures have the same duration, the problem instance I can be given by the 4-tuple $(\mathcal{P}, \mathcal{L}, \mathcal{C}, \mathcal{C}^*)$; otherwise this would extend to a 5-tuple together with the vector of lecture durations. The optimization problem asks for the shortest possible time within which all lectures can be delivered. We use the notation $OPT(I)$ for the optimum value on instance I. Moreover, we write $OPT_p(I)$ for the optimum in case of preemptive scheduling, i.e. where the lectures are allowed to be suspended and continued at a later time.

We shall often use the terms *job*, *machine*, and *processing time* for a lecture, a professor, and the duration of a lecture, respectively.

Interruptive scheduling. We also introduce here the following novel type of scheduling which seems to have never been studied before. By *interruptive scheduling* we mean preemptive scheduling under the restrictive condition that each job has to be executed on just one machine, although its execution may be suspended an unbounded, finite number of times during the schedule. The minimum makespan of an interruptive scheduling on problem instance I will be denoted by $OPT_i(I)$. The inequalities $OPT_p(I) \leq OPT_i(I) \leq OPT(I)$ are valid by definition, but we note already at this early point that strict inequality may hold on either side (as our examples will show later).

If machines are located in different rooms (or different buildings, etc.), then moving a job from one machine to another may be infeasible. Also, if a lecture has been suspended, it is reasonable to be completed by the same professor who started it. In such situations the interruptive model seems to be more appropriate than the preemptive one.

Some applications. An application, or the explanation (for why another professor P_s must be present when someone else delivers lecture L_t) is that P_s is in fact a demonstrator who is not able (yet) to deliver L_t but it is planned that in the future he/she will also be able to deliver the lecture, so he/she attends the lecture and learns.

Here is another application. A manufacturing system is given with several machines. The machines are of two types. The first machine is a "supermachine", while the other machines are of type "simple lathe machine". Also, given a set of jobs to be processed by the simple lathe machines, and for any job it is specified which of the lathe machines are able to handle it. Each lathe machine has a turning and a blade. The blades need to be sharpened from time to time, and it can be done by the supermachine only. During the sharpening the simple machine must be idle. A typical situation can be that in each shift, any machine needs a sharpening of its blade exactly once, that may be done at any time during the shift. In a more general scenario, there can be more than one supermachine.

Another, similar situation is the next: There are m_1 workers in a weaving factory (the workers correspond to the professors or supermachines), and m_2 automatic looms (corresponding to the demonstrators or simple lathe machines). The thread breaks occasionally several times as the process of production goes. At such occasions a worker must go to the machine and solve the problem. During this the machine is idle. This latter example is a typical *online* model while the previous one is an *offline* model. As the thread breaks cannot be predicted in advance, in the weaving model there are two types of jobs: the regular jobs are where the machines work without any problem, and the unexpected jobs are the interruptions, where some help of a worker is needed.

A further *online* model [9] is the following one. Accidents happen in an unpredictable way in a city, and the injured people are taken into a hospital to perform the necessary operations for them. These operations are the jobs. For any operation, according to the nature of the injury, a special team is needed, the members of the team play the role of the machines. Let us consider one operation. It is possible that the presence of some doctors is indispensable. For example only one doctor can make the anesthesia, so he/she will surely be there during the operation. Also, there is an expert, the only one who can make a special kind of operation. So both of them will be there. Moreover there are also several nurses who are free at that time, and either of them can be chosen as the one who helps the doctors.

Offline and online cases. If complete information (all relevant data) is known before starting the process of optimization, the problem is said to be offline. In practice this often happens to be not the case, since we get to know the input only sequentially in parts. In the online setting the jobs arrive one by one, according to a list L which is unknown in advance, and a decision has to be made before the arrival of the next job.

Measuring the efficiency. Let $I = (\mathcal{P}, \mathcal{L}, \mathcal{C}, \mathcal{C}^*)$ be an arbitrary problem instance. Recall that $OPT(I)$ denotes the (offline, non-preemptive) optimum value on instance I. Moreover, for any offline or online (non-preemptive, preemptive, or interruptive) algorithm A that solves the problem on I, we denote by $A(I)$ the value of the solution delivered by A. The absolute approximation ratio of A (offline) or competitive ratio (online) is

$$r_{abs} = \sup_I \frac{A(I)}{OPT(I)}.$$

Furthermore, the asymptotic approximation ratio of A is defined as

$$r_{as} = \lim \sup_{n\to\infty} \left\{ \max_I \frac{A(I)}{OPT(I)} \mid OPT(I) = n \right\}.$$

These ratios depend on the choice of A; but in the present paper we omit A from the notation because the algorithm will be understood from the context.

Download English Version:

https://daneshyari.com/en/article/6871745

Download Persian Version:

https://daneshyari.com/article/6871745

Daneshyari.com