European Journal of Operational Research 256 (2017) 55–61

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

FPTAS for the two identical parallel machine problem with a single operator under the free changing mode

Pierre Baptiste[a], Djamal Rebaine[b,*], Mohammed Zouba[a]

[a] Département de Mathématiques et Génie Industriel, École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3A7, Canada
[b] Département d'Informatique et de Mathématique, Université du Québec à Chicoutimi, 555, boul. de l'Université, Saguenay, Québec G7H 2B1, Canada

ARTICLE INFO

Article history:
Received 15 September 2015
Accepted 31 May 2016
Available online 11 June 2016

Keywords:
FPTAS
Free changing mode
Identical parallel machines
Operator

ABSTRACT

We address in this paper the problem of scheduling a set of independent and non-preemptive jobs on two identical parallel machines with a single operator in order to minimize the makespan. The operator supervises the machines through a subset of a given set of *modi operandi:* the working modes. A working mode models the way the operator divides up his interventions between the machines. The processing times thus become variable as they now depend on the working mode being utilized. To build a schedule, we seek not only a partition of the jobs on the machines, but also a subset of working modes along with their duration. A pseudo-polynomial time algorithm is first exhibited, followed by a fully polynomial time approximation scheme (FPTAS) to generate an optimal solution within the free changing mode.

1. Introduction

In the context of classical scheduling problems, research has mainly concentrated on machine and job characteristics such as availability, processing sequence, and so on. However, in real cases, the processing of a job on a machine may need interventions of operators. Nowadays, most machines are partially automated, and operators act as supervisors to load and unload jobs on the machines, sometimes controlling or doing short setups. Therefore, an operator is not fully assigned to a single machine, but may supervise simultaneously several ones. Unfortunately, when operators are considered to build a solution, they are usually assigned to a single machine. Furthermore, processing times are usually considered as independent of the schedule, except when considering learning curves or skill levels.

To the best of our knowledge, only a little research has been done in the direction that considers that the simultaneous supervision of several machines by a human operator may induce the increase of the execution times of the jobs being processed. The scheduling details of those micro-operations (loading, unloading and controlling) would be a non-sense as most of operator interventions cannot be fully anticipated. Such scheduling problems are so far only studied when micro-operations are limited to loading and unloading, done by robots, in the context of flow-shop models (Abdulkader, ElBeheiry, Afia, & El-Kharbotly, 2013; Brauner & Finke, 2001).

Traditionally, when considering additional (renewable or non-renewable) resources, other than machines, to schedule jobs, such resources are assumed to be of different types but limited in capacity. A subset of them are then used to accompany the processing of the jobs, but do not have an impact on the processing length whatsoever. Furthermore, at any time, a resource is assigned to at most one job (Blazėwicz, Ecker, Pesch, Schmidt, & Weglarz, 2007; Jurish & Kubiak, 1997; Kellerer & Strusevich, 2003; Oulamara, Rebaine, & Serairi, 2013). However, some software give the possibility to associate a portion of an operator to the processing of an operation. Those portions are summed up at each time period and the operator capacity is then checked. This model ignores the impact on the processing times. In fact, if a portion of an operator is necessary for an operation, the real processing time of this operation will depend on what this operator is doing in parallel. Either, the operator is only occupied by this operation then, the processing time of that operation is not affected. Or, this operator shares his activity over one or more operations. In the latter case, the processing times of all involved operations may increase in length. Note that even if the activity ratio of an operator is used at less than 100%, supervising several machines may induce local conflicts and idle periods on the machines.

Cheurfa (2005) introduced a model in which the sharing of the operator over the machines is expressed through the multiplication of the processing times by a vector of constant values (one

* Corresponding author. Tel.: +1 41854550115220; fax: +1 4185455012.
E-mail addresses: pierre.baptiste@polymtl.ca (P. Baptiste), Djamal_Rebaine@uqac.ca (D. Rebaine), mohammed.zouba@polymtl.ca (M. Zouba).

http://dx.doi.org/10.1016/j.ejor.2016.05.062
0377-2217/

value for each supervised machine with a sum that could exceed one). These values may differ from one machine to another. The present paper aims to generalize this model in the sense we assume that, for a given subset of machines, an operator may choose from a given finite set of *modi operandi*, called hereafter *working modes*. During the time interval a working mode is being utilized, the processing times of the jobs being scheduled are multiplied by the same set of multipliers. The set of multipliers can be changed by switching to another working mode. It is in fact a set of ways to attribute priorities to the different supervised machines. As long as the same working mode is utilized, processing times of the jobs are linearly increased by the same set of multipliers. A scheduling problem is thus not only an assignment-sequencing-dating problem for the jobs, but also a choice of working modes along with their duration of use. Let us note that classical scheduling problems are a particular case of this model, in the sense a single working mode is utilized with a set of multipliers all equal to one. The present work ensues from Zouba, Baptiste, and Rebaine (2009) in which the changing of the working mode sets occur at periodic times. We investigate here another type of changing mode: the free changing mode *i.e.* the changeover of the working modes may occur at any time. Note that the model studied in this paper is also discussed in Zouba, Baptiste, Rebaine, and Soumis (2011) in which a geometrical approach is presented to derive a pseudo polynomial algorithm. In the present paper, we improve this algorithm by a factor of ℓ, where ℓ denotes the number of working modes. Indeed, in Zouba, Baptiste, Rebaine, and Soumis (2011), $(\ell - 1)$ knapsack problems were used to derive an optimal solution, whereas in the solution we are presenting in this paper only two knapsack problems are needed. We further develop a fully polynomial time approximation schemes for our problem. Based on a geometrical approach, optimal properties are also presented in Zouba, Baptiste, Rebaine, and Soumis (2012) but for the case of three machines.

This paper is organized as follows. Section 2 describes the problem we are considering on two identical parallel machines. Section 3 presents preliminary results under the free changing mode assumption. In Section 4, we develop a pseudo-polynomial time algorithm that produces an optimal solution with a single operator. In Section 5, we derive a fully polynomial time approximation scheme (FPTAS) for this problem. Section 6 is our conclusion.

2. Description of the problem

The scheduling problem we consider consists in processing a set of n independent and non-preemptive jobs $J = (J_i : i = 1, \dots, n)$ on two identical parallel machines m_1 and m_2. We assume that the jobs and the machines are continuously available from time $t = 0$.

In order for a job to be executed on a machine, the supervision of an operator on that machine is required during a fraction of time. The standard processing time, p_i, of J_i corresponds to the case where an operator supervises exclusively the machine to which this job is assigned, and it is called *the basic processing time* of job J_i. Since there is only one operator in the shop floor, the productivity of the machines is thus affected, and has an impact on the processing times of the jobs since they vary according to the working mode being utilized.

The operator may be associated with several working modes. Accordingly every working mode induces productivity rates on the machines. Let $x_{k,j} \in [0, 1]$ be the productivity rate of machine m_j for a given working mode AT_k. Note that $x_{k,j} = 1$ or 0 if m_j is not supervised or exclusively supervised by the operator, respectively. The k-th working mode AT_k, $k = 1, 2, \dots, \ell$, where ℓ denotes the number of available working modes, is defined as $AT_k = (x_{k,j} : j = 1, 2)$. The real processing time of job J_i on machine m_j, associated with working mode AT_k, now becomes as p_{ikj} defined as $p_{ikj} = p_i / x_{k,j}$ if $x_{k,j} \neq 0$. If $x_{k,j} = 0$ this means that machine m_j is not used. The corresponding processing time has therefore no sense.

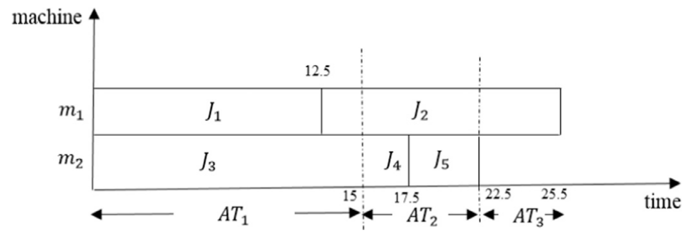

Fig. 1. A nonoptimal solution.

The working mode of an operator may change over time. The case investigated here is the one in which it may be changed at any time. This mode is called the free changing mode; the corresponding model investigated in this paper will be denoted hereafter by SWO. There are also two other types of changes: the calendar changing mode in which the working mode may be only changed at periodic times, and the end-task changing mode in which a working mode may be changed as soon as the supervised task finishes its processing.

With regard to the computational complexity status of SWO problem, it can be easily shown that the corresponding decision problem is $\mathcal{NP}$-complete in the weak sense. Indeed, when the number of working modes is reduced to one, the corresponding problem is clearly equivalent to the two-uniform parallel machine problem, which is known to be $\mathcal{NP}$-hard in the weak sense (see e.g. Pinedo, 2002).

Before closing this section, let us illustrate the problem we are studying by the following example.

Example: Consider the following instance of SWO problem with $n = 5$ jobs and a single operator to whom $\ell = 4$ working modes are associated in a free changing mode. The basic processing times and the four working modes are, respectively, $p_1 = 10$, $p_2 = 8$, $p_3 = 8$, $p_4 = 2$, and $p_5 = 2$, with $AT_1 = (0.8, 0.4)$, $AT_2 = (0.4, 0.8)$, $AT_3 = (1, 0)$, and $AT_4 = (0, 1)$.

Let us assign J_1 and J_2 to m_1 and the rest of the jobs to m_2, and utilize AT_1 for 15 units of time, AT_2 for 7.5 units of time, and AT_3 for the rest of the time. We then get the following: Job J_1 is processed on m_1 when AT_1 is being utilized, and its real processing time is thus $10/0.8 = 12.5$ units of time. Job J_2 is processed for 2.5 units of time when AT_1 is being utilized, for 7.5 units of time when AT_2 is being utilized, and for 3 units of time when AT_3 is being utilized and its real processing time is thus $2.5 + 7.5 + 3 = 13$ units of time. Job J_3 is processed on m_2 for 15 units of time when AT_1 is being utilized, and for 2.5 units of time when AT_2 is being utilized; its real processing time is thus $15 + 2.5 = 17.5$ units of time. Jobs J_4 and J_5 are processed on m_2 within AT_2; the real processing time of each of them is thus $2/0.8 = 2.5$ units of time. The completion times, $C_i(S)$ for J_i, of the jobs, under the above strategy, say S, are $C_1(S) = 12.5$, $C_2(S) = 25.5$, $C_3(S) = 17.5$, $C_4(S) = 20$ and $C_5(S) = 22.5$, with a makespan of value 25.5 as pictured by Fig. 1.

The optimal solution S^* can be obtained by first utilizing AT_1 for 20 units of time, and then AT_2 for the rest of the time, as pictured by Fig. 2. The completion times of the jobs are $C_1(S^*) = 12.5$, $C_2(S^*) = 25$, $C_3(S^*) = 20$, $C_4(S^*) = 22.5$, and $C_5(S^*) = 25$, with a makespan of value 25.

3. Preliminaries

The aim of this section is to present properties of an optimal solution, under the free changing mode environment, that are needed in the next section. Before proceeding further, we first start with notations and definitions.