Computers & Operations Research 36 (2009) 2387 – 2393

Contents lists available at ScienceDirect

Computers & Operations Research

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

Preemptive scheduling in the presence of transportation times

Mourad Boudhar*[,1], Amina Haned

Faculty of Mathematics, USTHB University, BP 32 Bab-Ezzouar, El-Alia 16111, Algiers, Algeria

ARTICLE INFO

Available online 1 October 2008

Keywords:
Identical parallel machines
Makespan
Preemption
Transportation time
Migration delay
Heuristics
Complexity

ABSTRACT

In this paper, we consider the problem of scheduling n independent jobs preemptively on m identical parallel machines, to minimize the total completion time (makespan). Each job J_i $(i=\overline{1,n})$ has a processing time p_i and the transportation of an interrupted job from a machine M_j to another machine $M_{j'}$ requires $d_{jj'}$ units of time. We propose a linear programming formulation in real and binary decision variables and we prove that the problem is NP-hard. Some subproblems are analyzed and solved by polynomial algorithms. Finally we present some heuristics and give some lower bounds of the makespan.

1. Introduction

The problem that we consider is that of scheduling a set of n independent and preemptable jobs $J_1, J_2, \ldots, J_n$ on m identical parallel machines $M_1, M_2, \ldots, M_m$. Each job J_i $(i=1,\ldots,n)$ has a processing time p_i. Each machine can process at most one job at a time, and each job can be processed by at most one machine at a time. If a job J_i processed on the machine M_j is interrupted and moved to another machine $M_{j'}$ its transport requires $d_{jj'}$ units of time, which is called the transportation time. The goal is to schedule the jobs so as to minimize the makespan.

By using the three-field notation as in [1], the problem thus defined is denoted $\alpha/pmtn(\gamma)/C_{\max}$, where $\alpha \in \{Pm, P\}$ and γ is either *delay* (arbitrary transportation times) or *delay* $= d$ (identical transportation times).

Scheduling problems with preemption are largely studied in literature. Preemption of a job means that processing may be interrupted and resumed at a later time, even on another machine. A job may be interrupted finitely many times. The first work concerning the preemptive scheduling problems on parallel machines started at the end of the fifties with Mc Naughton. This problem is denoted by $P/pmtn/C_{\max}$ and can be solved in linear time using Mc Naughton's algorithm.

Since then, several cases have been studied, some are polynomials (using linear programming or specific methods), we cite: $Q/pmtn, r_i/C_{\max}$, $P/pmtn, p_i = 1/\sum w_i C_i$, $Q/pmtn/L_{\max}$, $Q/pmtn, r_i/L_{\max}$ and $R/pmtn, r_i/L_{\max}$. Others are NP-hard, we cite as example: $P2/pmtn, r_i/\sum C_i$, $P2/pmtn/\sum w_i C_i$, $P2/pmtn, r_i/\sum U_i$, $P/pmtn/\sum U_i$ and $Q2/pmtn/\sum w_i U_i$ (see [2–7] for all details).

In [8] the problem which consists in scheduling n independent jobs, where each job has processing time p_i, a release time r_i and simultaneously requires at every time $m_i \leqslant m$ machines for its processing, is studied for both of the preemptive and the non-preemptive cases. Such problems are denoted by $P/pmtn, r_i, m_i/C_{\max}$ and $P/r_i, m_i/C_{\max}$, respectively.

Among the problems that are considered in the literature are the ones with setup time constraints (called also changeover times). They express the necessary time needed to pass from the processing of one job to another. This time may represent the time used for the machines adjustment, change of tools, cleaning of some parts of the machines or the control of the machines. In the literature these problems are referred to as "scheduling problems with changeover times" or "scheduling problems with transportation times", for several cases where we assume that there is sufficient transportation capacity to handle all jobs to be transported simultaneo usly.

In the literature dealing with those problems with changes depending on the sequence of the jobs processed on the same machine, there is an analogy between these problems for the minimization of $C_{\max}$ and the vehicles routing problem.

Chu and Yalaoui [9] presented a heuristic solving the problem that minimizes the makespan on identical parallel machines, in which the transportation times between jobs and the possibility of job splitting are considered. Boustta [10] studied the case of a multi-arm injection machine with multiple adjustments to minimize the maximum flow time. He related the problem to the one with identical parallel machines with adjustment times.

* Corresponding author.
E-mail addresses: mboudhar@yahoo.fr (M. Boudhar), amina_haned@yahoo.fr (A. Haned).

[1] Partially supported by LAID3 laboratory, USTHB, Algiers.

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.09.006

Other problems with the same type of constraints are studied for different criteria. For example, one can cite Adjallah et al. [11] who considered the problem with family setup times, which means that the jobs form different families and the passage from one family to another on the same machine requires a setup time independent from the jobs sequence, in the aim of minimizing the weighted sum of the completion times. They have proposed three heuristics for the resolution and have given one application to the tasks management of preventive maintenance.

The aspect of transportation of preempted jobs in scheduling on identical parallel machines has been studied in [1]. In this paper, all transportation times called migration delays are identical and equal to d. Authors showed that the subproblem of $P/pmtn(delay = d)/C_{\max}$ with delay d at most $\max\{\max_{1 \leqslant i \leqslant n}\{p_i\}, (1/m)\sum_{i=1}^{n} p_i\} - \max_{1 \leqslant i \leqslant n}\{p_i\}$ can be solved in linear time. Further, they showed that for any constant $\varepsilon > 0$ the subproblem of $P/pmtn(delay = d)/C_{\max}$ with d larger than $\max\{\max_{1 \leqslant i \leqslant n}\{p_i\}, (1/m)\sum_{i=1}^{n} p_i\} - \max_{1 \leqslant i \leqslant n}\{p_i\}$ is NP-hard in the strong sense. They also gave initial results supporting a conjecture that there always exists an optimal schedule in which at most $m-1$ jobs migrate. Then they gave a $O(n)$ time $O(1 + 1/\log_2 n)$-approximation algorithm for $m = 2$, and showed that there is a polynomial time approximation scheme (PTAS) for arbitrary m. Also, a preemptive version of the flexible job shop problem [12], which is a generalization of the classical job shop scheduling problem in which for every operation there is a group of machines that can process it, has been studied with migration of jobs and a $(2+\varepsilon)$-approximation algorithm has been presented.

In classical scheduling models the transportation (or migration) of a job is done without any delay constraint. However, in [1] it is mentioned that in production planning, for example, it is natural to allow some time for the transition of a product from one machine to another. However, the transportation of the product takes time, and also technical issues might make it necessary to wait for some time; a heated product might need to cool down first, or a product needs to dry before its next operation starts. Thus, in production workshops where a preempted job is processed on two different machines can require a considerable transportation time which is not taken into account in the classical models. It can also be found in the case of the multi-site manufactures where the treatment of a semi-finite product is realized on distant sites, the near-completed products are transported from one site to another in a well-determined span of time; here a site is considered as a machine.

This article is organized as follows: the second section is devoted to the mathematical formulation and the proof of the NP-hardness of the problem. In the third section we propose lower and upper bounds and consider some subproblems that can be solved polynomially. As for the fourth section some heuristics are presented for the resolution of the problem and some numerical tests are carried out to test the performance of different heuristics in the last section. Finally, a conclusion is given at the end of this article.

2. Modeling and complexity

We propose a linear programming formulation with binary decision variables. For this, we define the following variables:

- q_{ij}: the shared part of processing time of the job J_i on the machine M_j, $q_{ij} \geqslant 0$ for all $i = \overline{1,n}$ and all $j = \overline{1,m}$.
- Binary variables x_{ij} defined by: $x_{ij} = 1$ if and only if the job J_i is processed on the machine M_j for $q_{ij} \neq 0$ units of time, for all $i = \overline{1,n}$ and all $j = \overline{1,m}$.
- t_{ij}: represents the processing starting time of the job J_i on the machine M_j, $t_{ij} \geqslant 0$ for all $i = \overline{1,n}$ and all $j = \overline{1,m}$.
- Binary variables $\alpha_{iji'j'}$ defined as follows: $\alpha_{iji'j'} = 1$ if and only if $t_{ij} \leqslant t_{i'j'}$, the variable $\alpha_{iji'j'}$ expresses that a job J_i processed on the machine M_j begins its processing before the job $J_{i'}$ processed on the machine $M_{j'}$.

  $\alpha_{iji'j'}$ variables are only defined for all $i, i' = \overline{1,n}$ $(i \neq i')$ and all $j = j' = \overline{1,m}$ (all possible pairwise jobs on each machine) and for all $i = i' = \overline{1,n}$ and all $j, j' = \overline{1,m}$ $(j \neq j')$ (all possible pairwise machines for each job).

Let y be the length of the scheduling to be minimized.

In a given feasible solution for the model, only t_{ij} variables where $x_{ij} = 1$ are interesting and are used to find the optimal solution, all the others are insignificant and their values cannot be considered by the user. A similar remark is also valid for $\alpha_{iji'j'}$ variables: only $\alpha_{iji'j'}$ variables where $x_{ij} = 1$ and $x_{i'j'} = 1$ are interesting and are used to determine the optimal solution.

We must take into account the following constraints:

- The sum of all shared parts of processing time of each job must be equal to its processing time, that is: $\sum_{j=1}^{m} q_{ij} = p_i$ for all $i = \overline{1,n}$.
- If a job J_i begins its processing at the time t_{ij} on a machine M_j for q_{ij} units of time, it must finish before the date y, therefore: $t_{ij} + q_{ij} \leqslant y$ for all $i = \overline{1,n}$ and all $j = \overline{1,m}$.
- For any pair of different jobs J_i and $J_{i'}$ processed on the "same machine" M_j, either the job J_i precedes the job $J_{i'}$ or the inverse, thus we have either $\alpha_{iji'j} = 1$ and $t_{ij} + q_{ij} \leqslant t_{i'j}$ or $\alpha_{iji'j} = 0$ and $t_{i'j} + q_{i'j} \leqslant t_{ij}$. Hence we will have:

$$\begin{cases} t_{ij} + q_{ij} - t_{i'j} \\ \quad \leqslant (1 - \alpha_{iji'j})B \\ t_{i'j} + q_{i'j} - t_{ij} \leqslant \alpha_{iji'j}B \\ \alpha_{iji'j} + \alpha_{i'jij} = 1 \end{cases} \quad \text{for all } i, i' = \overline{1,n}\ (i \neq i') \text{ and all } j = \overline{1,m},$$

  where B is a large enough number.
- For an interrupted job J_i on the machine M_j and transported to "another machine" $M_{j'}$, either the job begins its processing on the machine M_j and continues on the machine $M_{j'}$ or the inverse, thus we have either $\alpha_{ijij'} = 1$ and $t_{ij} + q_{ij} + d_{jj'} \leqslant t_{ij'}$ or $\alpha_{ijij'} = 0$ and $t_{ij'} + q_{ij'} + d_{j'j} \leqslant t_{ij}$; we then obtain

$$\begin{cases} t_{ij} + q_{ij} + x_{ij}d_{jj'} \\ \quad -t_{ij'} \leqslant (1 - \alpha_{ijij'})B \\ t_{ij'} + q_{ij'} + x_{ij'}d_{j'j} \\ \quad -t_{ij} \leqslant \alpha_{ijij'}B \\ \alpha_{ijij'} + \alpha_{ij'ij} = 1 \end{cases} \quad \text{for all } i = \overline{1,n} \text{ and all } j, j' = \overline{1,m}\ (j \neq j').$$

- For any job J_i and any machine M_j, if $q_{ij} = 0$ then the job J_i is not processed on the machine M_j, as a result x_{ij} will obligatorily be equal to 0, therefore: $x_{ij} \leqslant q_{ij}B$ for all $i = \overline{1,n}$ and all $j = \overline{1,m}$.
- For any job J_i and any machine M_j, if $q_{ij} \neq 0$ then the job J_i is processed on the machine M_j $(x_{ij} = 1)$ for q_{ij} units of time, thus we will have: $x_{ij}B \geqslant q_{ij}$ for all $i = \overline{1,n}$ and all $j = \overline{1,m}$.

The number of variables and the number of constraints of a linear model are two indexes by which we can measure the dimension and the effectiveness of the given formulation. The number of variables of our model is $nm + nm + nm + (n(n-1)m + nm(m-1)) + 1 = nm(n + m + 1) + 1$ and the number of its constraints is equal to $n + nm + 3(n(n-1)m) + 3(nm(m-1)) + nm + nm = n + 3nm(n + m - 1)$ and