Information Processing Letters 114 (2014) 500–503

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Minimizing total weighted completion time approximately for the parallel machine problem with a single server

Keramat Hasani [a,*], Svetlana A. Kravchenko [b], Frank Werner [c]

[a] *Department of Computer Engineering, Malayer Branch, Islamic Azad University, Malayer, Iran*
[b] *United Institute of Informatics Problems, Surganova St. 6, 220012 Minsk, Belarus*
[c] *Otto-von-Guericke-Universität, Fakultät für Mathematik, PSF 4120, 39016 Magdeburg, Germany*

ARTICLE INFO

Article history:
Received 30 July 2013
Received in revised form 8 April 2014
Accepted 8 April 2014
Available online 13 April 2014
Communicated by M. Chrobak

Keywords:
Parallel machine scheduling
Single server
Approximation algorithm
Worst-case analysis

ABSTRACT

In this paper, we consider the scheduling problem of minimizing total weighted job completion time when a set of jobs has to be processed on a set of m parallel identical machines with a single server. We propose an approximation algorithm with a worst-case ratio $3 - \frac{1}{m}$. This result improves an existing $(5 - \frac{1}{m})$-approximation algorithm given by Wang and Cheng (2001).

1. Introduction

In this paper, we consider a parallel machine scheduling problem with a single server and the total weighted completion time objective function to be minimized, i.e., problem $P, S1||\sum w_j C_j$ using the standard scheduling notation. This problem can be formulated as follows. A set of n jobs $1, \dots, n$ has to be processed by a set of m parallel machines $M_1, \dots, M_m$. Before processing, each job j has to be loaded on the machine, on which this job is processed at once after loading. The loading, which is called a setup, is performed by a server. There is only one server, and the loading procedure requires both the server and the machine for s_j time units. The processing time p_j and the weight w_j for each job j are known in advance. Each job can be processed by an arbitrary machine, and each machine and also the server can perform only one job at a time. We want to find a feasible schedule minimizing the function $\sum w_j C_j$, where C_j denotes the completion time of job j. This problem $P, S1||\sum w_j C_j$ was considered in [5], where a $(5 - \frac{1}{m})$-approximation algorithm was given. In this paper, we propose a $(3 - \frac{1}{m})$-approximation algorithm for the same problem. In fact, we use the same approach as in [5], i.e., we replace the original problem by a relaxed problem, which can be easily solved and thus, an optimal order of the completion times can be found. Then, in the original problem, we schedule the jobs in the order determined and estimate the performance bound.

The interested reader can find a number of papers with descriptions of practical applications of scheduling models with setup times in production, service, or information processing. We mention only some applications in Flexible Manufacturing Systems, where a worker [4] or an automated guided vehicle [2] is shared among several pieces of the equipment to perform the setups.

There are some known results related to our problem, see Table 1. The problem $P, S1||\sum w_j C_j$ is unary NP-hard, since it is known that the problem $P2, S1|s_j = s|\sum C_j$ is unary NP-hard [2]. Some special cases of this problem were considered so far. The problem $P2, S1|p_j = p|\sum C_j$ is binary NP-hard [1], and the problem $P, S1|s_j = 1|\sum C_j$

* Corresponding author. Tel.: +98 8512228093; fax: +98 8512228236.
E-mail address: hasani@iau-malayer.ac.ir (K. Hasani).

http://dx.doi.org/10.1016/j.ipl.2014.04.005
0020-0190/

Table 1
Some known related results.

$P, S1\|\|\sum w_j C_j$	unary NP-hard
$P2, S1\|s_j = s\|\sum C_j$	unary NP-hard
$P2, S1\|p_j = p\|\sum C_j$	binary NP-hard
$P, S1\|s_j = 1\|\sum C_j$	unary NP-hard
$P2, S1\|s_j = 1\|\sum C_j$	$O(n \log n)$
$P3, S1\|s_j = 1\|\sum C_j$	$O(n^7)$
$P, S1\|s_j = 1\|\sum C_j$	$\sum C_j(\tilde{s}) - \sum C_j(s^*) \le n'(m-2)$
$P, S1\|s_j = s\|\sum C_j$	$\frac{\sum C_j(\tilde{s})}{\sum C_j(s^*)} \le \frac{3}{2}$
$P, S1\|\|\sum w_j C_j$	$\frac{\sum C_j(\tilde{s})}{\sum C_j(s^*)} \le 5 - \frac{1}{m}$

is unary NP-hard [1]. There exists a polynomial algorithm for the problem $P2, S1|s_j = 1|\sum C_j$ [2]. For the problem $P3, S1|s_j = 1|\sum C_j$, a polynomial algorithm was developed in [1]. For the unary NP-hard problem $P, S1|s_j = 1|\sum C_j$, an algorithm was proposed which creates a schedule $\tilde{s}$ with the following estimation:

$$\sum_{j=1}^{n} C_j(\tilde{s}) - \sum_{j=1}^{n} C_j(s^*) \le n'(m-2),$$

where s^* denotes an optimal schedule and $n' = |\{j|p_j < m-1\}|$, see [3]. It was shown that the SPT (shortest processing time) schedule is a $\frac{3}{2}$-approximation for the problem $P, S1|s_j = s|\sum C_j$, see [5].

The interested reader is referred to [6] for additional information on server scheduling models.

In the next section, we present the main result, i.e., we propose a $(3 - \frac{1}{m})$-approximation algorithm for the problem $P, S1||\sum w_j C_j$, and finally we give some concluding remarks in the last section.

2. Main result

Recall that for the problem $P, S1||\sum w_j C_j$, we know the set of machines $M_1, \ldots, M_m$, the set of jobs $1, \ldots, n$, and for each job j, we know the non-negative rational numbers p_j, s_j and w_j.

Next, we consider a relaxation of the given problem. In this relaxation, we add one additional machine MS to the set of machines $M_1, \ldots, M_m$. The set of jobs $1, \ldots, n$ is the same. However, each job j consists of two operations as follows. One operation, say $(j, 1)$, has to be processed on (one or several of) the machines $M_1, \ldots, M_m$ for p_j time units. It can be processed with preemptions and moreover, this operation can be performed with overlapping, i.e., it can be processed at the same time on several machines. Another operation, say $(j, 2)$, has to be processed on machine MS for s_j time units without preemptions. Both operations are independent, i.e., both operations can be processed at the same time.

Example. In Fig. 1, there are given two schedules. The first one uses two machines, and it is feasible for the problem $P, S1||\sum w_j C_j$. The second one uses three machines, and it is feasible for the relaxed model described above.

We denote by sa a feasible schedule for the relaxed model, by $CA_j(sa)$ the completion time of job j and by $CA_{j,k}(sa)$ the completion time of the operation (j, k) in the schedule sa, $k = 1, 2$. We set

$$CA_j(sa) = \max\{CA_{j,1}(sa), CA_{j,2}(sa)\}.$$

We want to find a schedule which minimizes the weighted sum of completion times for all jobs. Denote the described relaxed model by $RA||\sum w_j CA_j$.

Lemma 1. *For the model $P, S1||\sum w_j C_j$ and for the corresponding relaxed model $RA||\sum w_j CA_j$, the inequality*

$$\sum w_j CA_j^* \le \sum w_j C_j^*$$

holds, where $\sum w_j C_j^*$ denotes the optimal value of the weighted sum of completion times for the problem $P, S1||\sum w_j C_j$, and $\sum w_j CA_j^*$ denotes the optimal value of the weighted sum of completion times for the corresponding relaxed model $RA||\sum w_j CA_j$.

Proof. Take any feasible schedule s for the problem $P, S1||\sum w_j C_j$. We construct a schedule sa in the following way. We add one additional machine MS and schedule all the jobs $1, \ldots, n$ for $s_1, \ldots, s_n$ time units on machine MS in the same order as in the schedule s. For the set of machines $M_1, \ldots, M_m$, we schedule all the jobs $1, \ldots, n$ for $p_1, \ldots, p_n$ time units at the same time intervals and on the same machines as in the schedule s. Finally, we obtain the schedule sa which is feasible for the model $RA||\sum w_j CA_j$, and the inequality

$$\sum w_j CA_j(sa) \le \sum w_j C_j(s)$$

holds. $\square$

Thus, for the model $RA||\sum w_j CA_j$, we know the set of machines $M_1, \ldots, M_m, MS$, the set of jobs $1, \ldots, n$, each of the jobs j has the weight w_j and consists of two operations: operation $(j, 1)$ has to be processed on the machines $M_1, \ldots, M_m$ for p_j time units with preemptions and overlappings, and operation $(j, 2)$ has to be processed on machine MS for s_j time units.

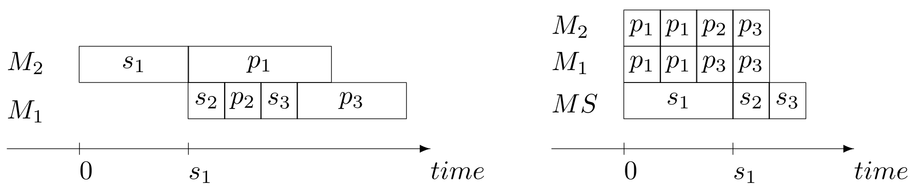

Fig. 1. The left schedule is feasible for the model $P, S1||\sum w_j C_j$, and the right schedule is feasible for the relaxed model $RA||\sum w_j CA_j$.