Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Scheduling jobs with equal processing times and a single server on parallel identical machines

An Zhang [a,*], Hongjun Wang [a], Yong Chen [a], Guangting Chen [b]

[a] Institute of Operational Research and Cybernetics, Hangzhou Dianzi University, Hangzhou 310018, PR China
[b] Taizhou University, Taizhou 317000, Zhejiang, PR China

ARTICLE INFO

Article history:
Received 4 August 2014
Received in revised form 19 January 2016
Accepted 16 May 2016
Available online xxxx

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

ABSTRACT

This paper studies the parallel-machine scheduling problem with a single server. There is a set of jobs to be processed on a set of m parallel and identical machines. Prior to processing on a machine, each job has to be loaded by a single server, which takes both the server and the machine a certain time. Preemption is not allowed. We consider the objective of minimizing the sum of jobs' completion times. This problem has been shown to be NP-hard even when all jobs have equal processing times (Brucker et al., 2002). We prove in this paper that the SPT algorithm has a worst case ratio of $1+\frac{\sqrt{m-1}}{\sqrt{m}+\sqrt{m-1}} < 1.5$ for the equal-processing-time problem.

1. Introduction

We consider parallel-machine scheduling with a single server. In this problem, each job has to be loaded or set up onto a machine by a single server before being processed. During the setup time, both the server and the machine are busy and unavailable for other jobs. But immediately after it is finished, the server becomes released and the machine has to process the job till completion without preemption. The problem is interesting because it occurs in many scenarios of production environments, like the network computing [5], the textile industry [1], the third-party logistics [11] and so on.

Perhaps the earliest study of algorithmic and computational complexity issues in this environment is due to Hall et al. [6]. Under many specific assumptions with respect to the setup and processing times, they gave a boundary between NP-hard and polynomially solvable problems for a variety of classical objective functions. Following the research of Hall et al. [6], Brucker et al. strengthened the results by proving the complexity of problems with either equal processing times or equal (or even unit) setup times [2]. Kravchenko and Werner settled the complexity of one open problem (the existence of pseudopolynomial algorithms for minimizing the makespan with unit setup times) and the problem of minimizing the forced idle times [9]. Glass et al. [5] considered the single server scheduling on dedicated machines, where both the complexity and the approximation results for problems of minimizing the makespan are provided. The two-machine flow shop scheduling problem is studied in [3,4,12,13], where quite a number of complexity results are obtained. It should be emphasized that most of the above researches focus on whether or under what conditions the problem becomes polynomially solvable or remains (strongly) NP-hard.

Researches deriving approximation algorithms with worst case guaranteed are also found in the literature. The pioneer contribution was made by Hall et al. [6], where scheduling algorithms for minimizing the makespan on m parallel machines

* Corresponding author.
E-mail address: anzhang@hdu.edu.cn (A. Zhang).

http://dx.doi.org/10.1016/j.dam.2016.05.014

Table 1
Approximation results for scheduling with a single server.

Problems	Complexity	Approximation results
$P, S1\|s_i\|C_{max}$	Strongly NP-hard [6]	LS: $3-\frac{2}{m}$; LPT: $2-\frac{1}{m}$ [6]
$P2, S1\|s_i=t_i=1\|C_{max}$	NP-hard [6]	LS: $\frac{12}{7}$; LPT: $\frac{4}{3}$ [8]
$P, S1\|s_i, p_i=p\|C_{max}$	NP-hard [2]	SPT: $2-\frac{1}{m}$ (Theorem 3.2)
$P, S1\|s_i\|\sum w_iC_i$	Strongly NP-hard [6]	$5-\frac{1}{m}$ [14]; $3-\frac{1}{m}$ [7]
$P, S1\|s_i=s\|\sum C_i$	Strongly NP-hard [6]	SPT: $\frac{3}{2}$ [14]
$P, S1\|s_i=1\|\sum C_i$	Strongly NP-hard [2]	$\sum C_j - \sum C_j^* \le n'(m-2)$ [10]
$P, S1\|s_i, p_i=p\|\sum C_i$	NP-hard [2]	SPT: $1+\frac{\sqrt{m-1}}{\sqrt{m-1}+\sqrt{m}}$ (Theorem 3.3)

with a single server is considered. Using the standard scheduling notation, this problem can be denoted as $P, S1|s_i|C_{max}$. They showed that the classical LS and LPT algorithms have a worst case ratio of $3-\frac{2}{m}$ and $2-\frac{1}{m}$ respectively. And the tightness of the bounds even holds for the case of unit setup times [9]. Xie et al. [15] considered the problem in which each job not only needs to set up on but also to remove from the machine. For the case of unit setup and removal times, $P, S1|s_i=t_i=1|C_{max}$, they claimed a bound of $\frac{3}{2}-\frac{1}{2m}$ for the LPT algorithm. Unfortunately, there exists an error pointed out by Jiang et al. in a subsequent work [8]. In the same paper, Jiang et al. further studied the two-machine case and showed that the exact bounds of LS and LPT are $\frac{12}{7}$ and $\frac{4}{3}$, respectively.

The min-sum problem of $P, S1|s_i|\sum w_iC_i$ was considered in [14], where Wang and Cheng provided the first approximation algorithm with worst case ratio of $5-\frac{1}{m}$. More than ten years later, Hasani et al. [7] revisited the problem and presented a new algorithm that improves the bound to $3-\frac{1}{m}$. Other approximation attempts are mainly concerning the model with equal weights, i.e. minimizing the sum of jobs' completion time. For the strongly NP-hard problem $P, S1|s_i=1|\sum C_i$, Kravchenko and Werner [10] proposed an algorithm which creates a schedule with an absolute error bounded by the product of $m-2$ and the number (denoted as n', please see Table 1) of jobs with processing time less than $m-1$. It was shown that the SPT algorithm has a worst case ratio of $\frac{3}{2}$ for the case of equal setup times $P, S1|s_i=s|\sum C_i$, see [14]. Ou et al. [11] considered a multiple-server problem with no setup but equal removal times, $P, S|t_i=t|\sum C_i$, where the SPT algorithm was analyzed and shown to have a worst case ratio of 2. In Table 1, we summarize the approximation results for parallel machines scheduling with a single server, with both the makespan minimization and the min-sum problems included.

In this paper, we study the single-server scheduling problem with arbitrary setup times and equal processing times. The problem has been shown to be NP-hard for both the makespan minimization problem and the min-sum problem in [2]. We analyze the performance of the SPT algorithm and provide worst case ratios with respect to the number of machines. The rest of the paper is organized as follows. Section 2 gives the notations, definitions and lower bounds. In Section 3, we devote ourselves to the analysis of the SPT algorithm. Finally, some concluding remarks are made in Section 4.

2. Notations, definitions and lower bounds

Assume that we are given a single server $S1$, m parallel machines, $M_1, M_2, \ldots, M_m$ and n independent jobs, $J_1, J_2, \ldots, J_n$. Prior to processing on a machine, each job J_i needs to be loaded by the server, which takes both the server and the machine a setup time of s_i. Immediately after that, the server becomes available and the machine has to process the job till completion, which takes the machine a processing time of $p_i = p$. Both the machine and the server can handle one job at a time, and preemption of jobs is not allowed. Usually, the SPT (shortest processing time first) algorithm is a good choice for min-sum problems. For our problem, it can be described as follows (see in Fig. 1 an instance run by SPT).

The SPT algorithm

Sort the jobs in the non-decreasing order of their setup times, then schedule them one by one at the earliest time when the server is available and there is a machine idle.

In reminder of the paper, we always assume that $s_1 \le s_2 \le \cdots \le s_n$. Denote by π and π^* the schedule generated by the SPT algorithm and the optimal schedule, and let C_i and C_i^* be the completion time of job J_i in π and π^*, respectively. Suppose that $J_{[i]}$ is the ith completed job in π^*, then we can get the following lower bound on the completion time of $J_{[i]}$.

Lemma 2.1. *In any optimal schedule, the completion time of job $J_{[i]}$ must satisfy that*

$$C_{[i]}^* \ge \max\left\{ \frac{ip + \sum_{u=1}^{i} s_u}{m}, p + \sum_{u=1}^{i} s_u \right\}.$$