Doctopic: Algorithms, automata, complexity and games ARTICLE IN PRESS TCS:9749

Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Two-agent single-machine scheduling problem with just-in-time jobs

Byung-Cheon Choi [a], Jibok Chung [b,*]

[a] Department of Business Administration, Chungnam National University, 79 Daehangno, Yuseong-gu, Daejeon 305-704, Republic of Korea
[b] Department of Business Administration, Daejeon University, 96-3 Yongun-dong, Dong-gu, Daejeon 300-716, Republic of Korea

A R T I C L E I N F O

Article history:
Received 4 February 2013
Received in revised form 25 March 2014
Accepted 28 May 2014
Available online xxxx
Communicated by D.-Z. Du

Keywords:
Scheduling
Competing agents
Just-in-time job
Complexity

A B S T R A C T

We consider two scheduling problems on a single machine with two agents. The objective is to optimize the performance measure for agent 1 while maintaining the weighted number of just-in-time jobs for agent 2 at or above a given threshold. The performance measures for agent 1 are the total weighted completion times and the weighted number of tardy jobs. We analyze how the complexities of the two problems change, depending on the conditions of processing times and weights.

1. Introduction

We consider two single-machine scheduling problems with two agents, where each agent is responsible for a set of non-preemptive jobs. All jobs share common resources (machines) and both agents wish to optimize their own objectives applied to optimize their own sets of jobs. This is the case in various real-world applications, such as preventive maintenance (PM) planning [1], rescheduling [2] and a multimedia service environment [3]. The objective is to optimize one agent's objective while satisfying the constraints for the other agent's objective.

The model introduced in this paper represents the situation where a machine manager selects among multiple options, each providing utility to the system for taking the machine down for PM. The machine manager's objective is to minimize the production planning damage from the options selected while maintaining total utility above the given threshold.

We use the concept of a *just-in-time* (JIT) job, defined as a job completed exactly on its due date, to represent the situation where the start and completion times of each PM are given in advance.

The following introduces the formal notation and problem definitions. Agent h has a set of n_h jobs $J^h = \{J_1^h, J_2^h, \dots, J_{n_h}^h\}$, $h = 1, 2$. Associated with J_j^h are a due date d_j^h, a weight w_j^h, and a processing time p_j^h. Let σ be a feasible schedule and $C_j^h(\sigma)$ the completion time of J_j^h in σ. Let $E_j^h(\sigma)$ and $U_j^h(\sigma)$ be indicator variables of J_j^h in σ, defined as

$$E_j^h(\sigma) = \begin{cases} 1 & \text{if } J_j^h \text{ is a JIT job} \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad U_j^h(\sigma) = \begin{cases} 1 & \text{if job } J_j^h \text{ is tardy} \\ 0 & \text{otherwise.} \end{cases}$$

* Corresponding author.
E-mail addresses: polytime@cnu.ac.kr (B.-C. Choi), jbchung@dju.ac.kr (J. Chung).

http://dx.doi.org/10.1016/j.tcs.2014.05.022
0304-3975/© 2014 Published by Elsevier B.V.

Please cite this article in press as: B.-C. Choi, J. Chung, Two-agent single-machine scheduling problem with just-in-time jobs, Theor. Comput. Sci. (2014), http://dx.doi.org/10.1016/j.tcs.2014.05.022

This paper considers two problems, referred to as Problems P1 and P2, respectively:

$$\textit{Problem P1:}\ \text{minimize} \sum_{J_j^1 \in J^1} w_j^1 C_j^1(\sigma) \text{ subject to } \sum_{J_j^2 \in J^2} w_j^2 E_j^2(\sigma) \geq \lambda_1,$$

and

$$\textit{Problem P2:}\ \text{minimize} \sum_{J_j^1 \in J^1} w_j^1 U_j^1(\sigma) \text{ subject to } \sum_{J_j^2 \in J^2} w_j^2 E_j^2(\sigma) \geq \lambda_2,$$

where λ_h, $h = 1, 2$, are given thresholds. Without loss of generality, assume that non-JIT jobs are processed after the completion of the other jobs. Furthermore, in Problem P2, tardy jobs are assumed to be processed after the other jobs are completed.

The rest of this paper is organized as follows. Section 2 discusses the related literature. Sections 3 and 4 establish the complexity of Problems P1 and P2, respectively, and investigate their pseudo-polynomially and polynomially solvable cases. Finally, Section 5 presents our conclusions and future research.

2. Literature review

The literature on single-machine scheduling problems with multiple agents studied the problem of optimizing one agent's objective while satisfying the constraints on those of others. Agnetis et al. [4,5] considered single-machine scheduling problems with two agents in which each agent wants to minimize an objective function, such as a maximum of regular functions, the number of tardy jobs, or total weighted completion times. They established the complexity of most combinations of objectives, except when the objective is to minimize the total completion time of the first agent while maintaining the number of tardy jobs of the other agent at or below a given value. Leung et al. [1] proved this case is NP-hard. Cheng et al. [6] considered single-machine problems with more than two agents, each agent having as objective the total weighted number of tardy jobs. They showed the problem is strongly NP-hard. Lee et al. [7] considered a single-machine scheduling problem with multiple agents in which each agent tries to minimize the total weighted completion time. They showed that the problem has a fully polynomial time approximation scheme, and provided an efficient approximation algorithm with a good worst-case ratio.

The literature on scheduling with JIT jobs studied the problem of maximizing the weighted number of JIT jobs in various machine environment. Lann and Mosheiov [8] considered the problem of minimizing the weighted number of early and tardy jobs on a single machine and showed it is polynomially solvable. Noted that minimizing the weighted number of early and tardy jobs is equivalent to maximizing the weighted number of JIT jobs. Hiraishi et al. [9] showed that an identical-machine scheduling problem with sequence-dependent setup times is polynomially solvable, and Cepek and Sung [10] proposed a more efficient algorithm for the same problem. Sung and Vlach [11] showed that the problem with a fixed number of machines is polynomially solvable on unrelated parallel machines, but strongly NP-hard when the number of machines is part of the input. Choi and Yoon [12] considered a flow shop problem and showed that the two-machine problem is NP-hard while the unweighted version is polynomially solvable. Furthermore, they showed that the unweighted version with three machines is strongly NP-hard.

To the best of our knowledge, however, no paper has examined scheduling problems with multiple agents under a JIT environment except for that of Chung and Choi [13]. They considered two multi-agent scheduling models where each agent tries to maximize a weighted number of JIT jobs. The first objective is to find the optimal schedule for one agent with constraints on the other agents' objectives and the second is to find the efficient set. They showed that for a fixed number of agents, the first problem is NP-hard, even in the single-machine case. Then they presented a polynomial-time algorithm for the unweighted version of the second problem under a two-machine flow shop environment.

3. Problem P1

In this section, we analyze how the complexity of Problem P1 changes, depending on the conditions that whether the processing times of jobs are identical or not, and whether the weights of jobs are identical or not.

3.1. Computational complexity

In this subsection, we show that Problem P1 is NP-complete in the strong sense even if $w_j^h = 1$, $j = 1, 2, \ldots, n_h$, $h = 1, 2$ and $p_j^2 = 1$, $j = 1, 2, \ldots, n_2$, and furthermore, there exists no ρ-approximation algorithm for any fixed value ρ even if $w_j^h = 1$, $j = 1, 2, \ldots, n_h$, $h = 1, 2$.

In order to prove the strongly NP-completeness, we introduce the 3-partition problem, which is stated as follows: Given $3m$ positive integers $a_1, a_2, \ldots, a_{3m}$ such that $D/4 < a_j < D/2$ and $\sum_{j=1}^{3m} a_j = mD$, are there m disjoint sets $S_1, \cdots, S_m$ such that $\sum_{j \in S_i} a_j = D$, $i = 1, \cdots, m$?