Omega 63 (2016) 41–47

Contents lists available at ScienceDirect

Omega

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

Just-in-time scheduling with two competing agents on unrelated parallel machines ☆

Yunqiang Yin [a,*], Shuenn-Ren Cheng [b], T.C.E. Cheng [c], Du-Juan Wang [d,*], Chin-Chia Wu [e]

[a] Faculty of Science, Kunming University of Science and Technology, Kunming 650093, China
[b] Graduate Institute of Business Administration, Cheng Shiu University, Kaohsiung County, Taiwan
[c] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[d] School of Management Science and Engineering, Dalian University of Technology, Dalian 116023, China
[e] Department of Statistics, Feng Chia University, Taichung, Taiwan

ARTICLE INFO

Article history:
Received 1 August 2014
Accepted 3 September 2015
Available online 23 October 2015

Keywords:
Scheduling
Two agents
Unrelated parallel machines
Just-in-time scheduling
FPTAS

ABSTRACT

This paper considers two-agent just-in-time scheduling where agents A and B have to share m unrelated parallel machines for processing their jobs. The objective of agent A is to maximize the weighted number of its just-in-time jobs that are completed exactly on their due dates, while the objective of agent B is either to maximize its maximum gain (income) from its just-in-time jobs or to maximize the weighted number of its just-in-time jobs. We provide a bicriterion analysis of the problem, which seek to find the Pareto-optimal solutions for each combination of the two agents' criteria. When the number of machines is part of the problem instance, both the addressed problems are *NP*-hard in the strong sense. When the number of machines is fixed, we show that the problem of maximizing agent A's weighted number of just-in-time jobs while maximizing agent B's maximum gain can be solved in polynomial time, whereas the problem of maximizing both agents' weighted numbers of just-in-time jobs is $\mathcal{NP}$-hard. For the latter problem, we also provide a pseudo-polynomial-time solution algorithm, establishing that it is $\mathcal{NP}$-hard in the ordinary sense, and show that it admits a fully polynomial-time approximation scheme (FPTAS) for finding an approximate Pareto solution.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

In classical just-in-time scheduling, the objective is to minimize the earliness–tardiness cost of completing jobs with respect to their due dates. Most papers in this area consider the objective of minimizing the sum of the earliness and tardiness penalties (see, e.g., [9,24,27,28]). In some situations, the earliness and tardiness penalties depend on whether the jobs are early or tardy, rather than how early or how late they are. Lann and Mosheiov [15] introduce a new objective of minimizing the weighted number of jobs that are early or tardy, which corresponds to maximizing the weighted number of jobs that are completed exactly on their due dates. We refer to any scheduling problem with the objective of maximizing the weighted number of just-in-time jobs as a just-in-time scheduling problem.

Lann and Mosheiov's study has recently received growing attention from the scheduling research community and just-in-time scheduling has been studied in various machine settings. For the single-machine case, Lann and Mosheiov [15] show that the just-in-time scheduling problem is solvable in $O(n^2)$ time, where n is the number of jobs. For the two-machine flowshop case, Choi and Yoon [8] prove that it is $\mathcal{NP}$-hard, but they leave an open question whether the problem is $\mathcal{NP}$-hard in the ordinary sense or in the strong sense. In addition, they show that the unweighted version of the problem can be solved in $O(n^4)$ time for the two parallel-machine case and is $\mathcal{NP}$-hard in the strong sense for the three parallel-machine case. Shabtay and Bensoussan [20] show that the open problem left in Choi and Yoon [8] is $\mathcal{NP}$-hard in the ordinary sense by developing a pseudo-polynomial-time algorithm and a fully polynomial-time approximation scheme (FPTAS) for the problem. Elalouf et al. [10] suggest another pseudo-polynomial-time algorithm for the same problem, which can be converted into a new FPTAS that reduces Shabtay and Bensoussan's complexity result. Shabtay [19] studies the just-in-time problem in the flowshop setting under four different scenarios. For each scenario, he either presents a polynomial-time algorithm or develops an efficient pseudo-polynomial-time algorithm. Shabtay et al. [21] address a two-machine flowshop scheduling problem where the job processing time is controllable by varying the allocation of a resource to the job operations. They adopt a

☆This manuscript was processed by Associate Editor Tkindt.

* Corresponding authors.
E-mail addresses: yinyunqiang@126.com (Y. Yin), wangdujuan@dlut.edu.cn (D.-J. Wang).

http://dx.doi.org/10.1016/j.omega.2015.09.010
0305-0483/© 2015 Elsevier Ltd. All rights reserved.

bicriterion analysis of the problem in which the first objective is to maximize the weighted number of just-in-time jobs while the second objective is to minimize the total resource consumption cost. They develop a pseudo-polynomial-time algorithm for the problem and convert it into a two-dimensional FPTAS. In the parallel-machine setting, Carlisle and Lloyd [6] consider the unweighted version of the just-in-time scheduling problem on m identical parallel machines and show that the problem can be solved in $O(n \log n)$ time. Other solution algorithms for the same problem can be found in Čepek and Sung [7], Frank [11], Yannakakis and Gavril [26], and Hsiao et al. [13]. Arkin and Silverberg [2] develop an $O(n^2 \log n)$ time solution algorithm for the weighted case on m identical parallel machines by converting the problem into a minimum cost flow problem. Bouzina and Emmonss [5], and Carlisle and Lloyd [6] present more efficient minimum cost flow algorithms with an $O(mn \log n)$ running time for the same problem by modelling it on a network that has only $O(n)$ arcs. Kovalyov et al. [14] show that the just-in-time scheduling problem on unrelated parallel machines is equivalent to that of maximizing the weighted m legally colourable vertices in a given interval graph. Both Arkin and Silverberg [2], and Sung and Vlach [23] prove that the just-in-time scheduling problem on m unrelated parallel machines can be solved in $O(mn^{m+1})$ time, which is polynomial when m is fixed. However, when m is arbitrary, they show that the problem becomes $\mathcal{NP}$-hard in the strong sense. Leyvand et al. [16] consider just-in-time scheduling on a set of m machines with controllable processing times, where the objectives are to maximize the weighted number of just-in-time jobs and to minimize the total resource allocation cost. They consider four different models for treating the two criteria. For each model, they either provide a polynomial-time solution algorithm or develop a pseudo-polynomial-time solution algorithm and an FPTAS.

All the above papers focus on the traditional case of just-in-time scheduling with a single agent. In recent years researchers have increasingly considered scheduling with multiple competing agents, which was initially investigated by Agnetis et al. [1] and Baker and Smith [3]. In this case, multiple agents need to process their own sets of jobs, competing for the use of a common resource. Each agent wants to optimize a certain objective function, which depends on the completion times of its jobs only. Variants of the scheduling problem with multiple agents have found many applications in areas such as manufacturing, supply chain management, telecommunication services, project scheduling, etc. A recent survey of multi-agent scheduling research is given in Perez–Gonzalez and Framinan [17]. With a view to modelling a realistic production system, this paper combines the two sub-fields into a unified framework. Specifically, we focus on the innovative just-in-time scheduling model on unrelated parallel machines in the two-agent setting. The purpose of this paper is twofold. One is to investigate this unexplored scheduling model. Another is to ascertain the computational complexity status and provide solution procedures, if viable, for the problems under consideration.

The rest of the paper is organized as follows: In Section 2 we formulate the problem and present a common property of the optimal schedules for the two problems under consideration. In Section 3 we show that the Pareto-optimization problem with a fixed number of machines where agent A's objective is to maximize the weighted number of its just-in-time jobs while agent B's objective is to maximize its maximum gain can be solved in polynomial time. In Section 4 we show that the Pareto-optimization problem with a fixed number of machines where both agents' objectives are to maximize their weighted numbers of just-in-time jobs is $\mathcal{NP}$-hard in the ordinary sense by developing a pseudo-polynomial-time algorithm for the problem and we convert the algorithm into an FPTAS. In the last section we provide some concluding remarks and suggest topics for future research.

2. Problem formulation

We formally describe the problem under study as follows: There are two competing agents (called agent A and agent B, respectively) that have to schedule two families of independent and non-preemptive jobs on m unrelated parallel machines $M_1, M_2, \ldots, M_m$. Agent A has to perform the job set $J^A = \{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$, while agent B has to perform the job set $J^B = \{J_1^B, J_2^B, \ldots, J_{n_B}^B\}$. We call the jobs of agents A and B the A-jobs and B-jobs, respectively. All the jobs are available for processing from time zero onwards. Let $X \in \{A, B\}$ and let $n = n_A + n_B$ denote the total number of jobs. Denote d_j^X as the due date of job J_j^X, w_j^X as the gain (income) from completing job J_j^X just-in-time (i.e., exactly at time d_j^X), and p_{ij}^X as the processing time of job J_j^X on machine M_i for $i = 1, \ldots, m$ and $j = 1, \ldots, n_X$. The jobs that are completed exactly on their due dates in some schedule are called just-in-time jobs. We assume, without loss of generality, that all the d_j^X, w_j^X, and p_{ij}^X values are positive integers, and let $W^A = \sum_{J_k^A \in J^A} w_k^A$ and $W^B = \sum_{J_k^B \in J^B} w_k^B$.

For any given solution, let E_i be the set of just-in-time jobs allocated to machine M_i for $i = 1, \ldots, m$, with $E = E_1 \cup E_2 \cup \cdots \cup E_m$, and let $T = (J^A \cup J^B) \backslash E$ be the set of the other jobs. A partition of set $J^A \cup J^B$ into two disjointed subsets E and T is considered to be a feasible partition (or a feasible schedule) if it is possible to schedule the jobs belonging to set E on the m unrelated parallel machines such that they are all completed just in time. Following Lann and Mosheiov [15], in a feasible schedule, it is assumed that the jobs in T need not be executed, which means that they are rejected. We also denote by E^A and E^B the sets of just-in-time A-jobs and B-jobs, respectively.

Each agent wants to optimize a certain objective function depending on the completion times of its jobs only. Specifically, agent A wants to maximize $Q^A(S) = \sum_{J_k^A \in E^A} w_k^A$ (the weighted number of just-in-time A-jobs, i.e., the total gain from completing the jobs in set E^A just-in -time), while agent B wants to maximize $Q^B(S) = \max_{J_k^B \in E^B} w_k^B$ (the maximum gain from completing the jobs in set E^B just-in-time) or to maximize $Q^B(S) = \sum_{J_k^B \in E^B} w_k^B$ (the weighted number of just-in-time B-jobs, i.e., the total gain from completing the jobs in set E^B just-in-time). Since increasing the objective value of agent A will decrease the objective value of agent B, and vice versa, we need to consider the trade-off between the two objective functions carefully to achieve the best scheduling outcome. For such kind of bicriterion problem, we focus on finding the set of all the Pareto-optimal schedules (points) (Q^A, Q^B), where a schedule S with $Q^A = Q^A(S)$ and $Q^B = Q^B(S)$ is called Pareto-optimal (or efficient) if there does not exist another schedule S' such that $Q^A(S') \geq Q^A(S)$ and $Q^B(S') \geq Q^B(S)$ with at least one of these inequalities being strict. Using the three-field notation proposed by Graham et al. [12] and extended to multicriteria scheduling problems by T'kindt and Billaut [25], we denote the problems under consideration by $Rm||(\sum_{J_k^A \in E^A} w_k^A, \max_{J_k^B \in E^B} w_k^B)$ and $Rm||(\sum_{J_k^A \in E^A} w_k^A, \sum_{J_k^B \in E^B} w_k^B)$, respectively, when the number of machines m is fixed. Note that the criterion in the classical three-field notation is to be minimized, but in this paper the criterion is to be maximized.

Sung and Vlach [23] have shown that the just-in-time scheduling problem on unrelated parallel machine is NP-hard in the strong sense if the number of machines is part of the problem instance. In fact, if the due dates of the B-jobs are made very large, our problems reduce to their problem, so our problems are NP-hard in the strong sense when the number of machines is part of the problem instance, too. Hence, in what follows, we focus only on the case where the number of the machines is fixed.

The following lemma provides an easy-to-prove property for the problems under consideration.

Download English Version:

https://daneshyari.com/en/article/1032383

Download Persian Version:

https://daneshyari.com/article/1032383

Daneshyari.com