Theoretical Computer Science 596 (2015) 109–123

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Two agent scheduling with a central selection mechanism

Gaia Nicosia [a,*], Andrea Pacifici [b], Ulrich Pferschy [c]

[a] Dipartimento di Ingegneria, Università degli Studi Roma Tre, Italy
[b] Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università degli Studi di Roma "Tor Vergata", Italy
[c] Department of Statistics and Operations Research, University of Graz, Austria

ARTICLE INFO

Article history:
Received 24 January 2014
Received in revised form 22 May 2015
Accepted 23 June 2015
Available online 29 June 2015
Communicated by P. Widmayer

Keywords:
Agent scheduling
Pareto optimality
Combinatorial games

ABSTRACT

We address a class of deterministic scheduling problems in which two agents compete for the usage of a single machine. The agents have their own objective functions and submit in each round an arbitrary, unprocessed task from their buffer for possible selection. In each round the shortest of the two submitted tasks is chosen and processed on the machine.
We consider the problems under two distinct perspectives: First, we look at them from a centralized point of view as bicriteria optimization problems and try to characterize the set of Pareto optimal solutions. Then, the problems are viewed under the perspective of a single agent. In particular, we measure the worst-case performance of classical priority rules compared to an optimal strategy. Finally, we consider *minimax* strategies, i.e. algorithms optimizing the objective of one agent in the worst case.
© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In *multi-agent scheduling* problems a set of *tasks* has to be processed on some processing *resource* and each task belongs to one decision maker (agent). Each agent is interested in optimizing its own performance measure which only depends on its tasks completion times. Although these problems can be viewed as a special case of multi-objective scheduling problems, their specific properties and applications have spurred a considerable amount of research since the seminal works by Agnetis et al. [2] and Baker and Smith [6]. Most of the recent literature on multi-agent scheduling problems falls into two main streams of research: one focuses on the problem in a multi-objective optimization perspective (see, for example, [10,15]); the other is from the algorithmic game theory point of view. In the latter context, for instance, *mechanism design* has received considerable attention in the recent literature (see, e.g. [5,9]). The goal is to design system-wide rules which, given the selfish decisions of the users, maximize the total social welfare. The degree to which these rules approximate the social welfare in a worst-case equilibrium is known as the price of anarchy of the mechanism. See [1] for an overview of these types of problems.

In this work we address a two-agent scheduling problem introduced in [3]: Two agents, A and B, each owning a set of nonpreemptive tasks (or jobs), require a single (commonly used) machine to process their tasks. Each agent pursues the minimization of a given objective function, such as makespan, total completion time or total weighted completion time. Additionally, a coordination mechanism, aiming at the maximization of the number of processed tasks per time unit, regulates access of agents' tasks to the machine as follows: Each agent submits one task for possible processing. The shortest among the two submitted tasks is selected and scheduled at the end of the current schedule, which is initially empty. When

* Corresponding author.
E-mail addresses: nicosia@ing.uniroma3.it (G. Nicosia), pacifici@disp.uniroma2.it (A. Pacifici), pferschy@uni-graz.at (U. Pferschy).

http://dx.doi.org/10.1016/j.tcs.2015.06.051
0304-3975/© 2015 Elsevier B.V. All rights reserved.

all tasks of one agent have been processed, the remaining tasks of the other agents are appended thereafter in the order they are submitted. In the following we refer to the above steps as *rounds*. We also say that the selected task in one round (and the corresponding agent) is the *winner*, while the other task (and agent) is the *loser*.

We look at the problem in two different settings. In a *centralized perspective* we aim at characterizing the set of Pareto optimal (PO) schedules in terms of size and computational complexity, as in a bicriteria optimization problem. Then, we consider the problem from a single *agent perspective* and search for a *strategy*, i.e., an algorithm that suggests to the agent which task to submit in each round, taking into account its own objective function. In this context we study the effectiveness of natural priority rules, e.g., when the agent submits its tasks in SPT or WSPT order. Moreover, we consider the case in which one agent wants to select a strategy that minimizes its solution cost in the worst possible case, i.e., for any strategy adopted by the opponent, even if the opponent aims only at worsening the first agents' objective. This corresponds to what is usually called *minimax strategy* in game theory.

This twofold approach has also been adopted in [3,4,11,12]. In particular, in [3] the authors introduce a class of multi-agent scheduling problems in which the decision process is organized in rounds and provide some preliminary results for different shop configurations. A detailed analysis of the so-called *linear conveyor* shop configuration is carried out in [4], where a number of properties and solution algorithms are presented taking into account both centralized and single-agent perspectives. The shop configuration of [4] refers to a manufacturing application in which two linear conveyor belts, one for each agent, transport parts to the machine. So, each agent sequences the parts on the conveyor, implying that, at each round, one of the two candidate tasks is the loser of the preceding round. In other words, each task is submitted for possible processing, in the given order, until it wins.

In this paper we consider a different, more general configuration (denoted as *flexible processing* in [3]) in which there are no queues at the machine and at each round any part from the two agents' buffers can be picked up and submitted for possible processing. Hence, in this case, the agents are free to choose any available task for submission at each round, independently from the outcome of the previous round. In particular, the two shop configurations (linear conveyor and flexible processing) produce different sets of feasible schedules. Note that in [3] part of the results discussed in this paper have been already mentioned.

Hereafter, we present the organization of the paper and summarize the main contributions. In Section 2 we introduce the notation and give a formal statement of the addressed problems. Section 3 characterizes the set of PO solutions for various objective functions. In particular, we investigate the complexity of finding Pareto optima and determine their number. We derive bounds on the size of the PO set and thus establish the boundary between objective functions with exponential size PO sets and those whose PO sets have polynomial size. We also show that the problem of deciding whether a feasible schedule with given bounds on the agents objectives exists, is $\mathcal{NP}$-complete for those problems having exponentially many PO solutions.

In Section 4 we consider a single agent perspective and provide results on the worst-case performance of various heuristic strategies. In particular, we evaluate the quality of schedules one agent, say B, can attain when its tasks are sequenced by applying the well known SPT and WSPT rules against any strategy, or against a "reasonably restricted" strategy of the opponent agent. To this purpose we measure the worst-case performance ratio $\rho(H)$ between the value of B's objective obtained by applying a heuristic algorithm H and that of the best possible solution it could attain (a formal definition of $\rho(H)$ is given in Section 4). For most cases we derive tight worst-case ratios.

Section 5 deals with the problem of devising minimax strategies for agent B. We prove that the best solution value B can attain for any regular (i.e. nondecreasing with respect to the completion times) objective function f^B against a malevolent strategy of A, is independent on whether A and B are restricted to stick to a submission sequence or whether they are free to adopt any strategy. In particular, we show that SPT is a minimax strategy for one agent when its objective is the minimization of the makespan or total completion time. On the other hand, finding a minimax strategy when B's objective is the total weighted completion time turns out to be non trivial but it can be done in polynomial time.

2. Formal problem definition

Let A and B denote the two agents. Each agent owns a set of n nonpreemptive tasks to be performed on a single machine which can process only one task at a time. Tasks have nonnegative deterministic processing times $a_1 \leq a_2 \leq \ldots \leq a_n$ for agent A and $b_1 \leq b_2 \leq \ldots \leq b_n$ for agent B. For convenience, we will often refer to tasks by their processing times. Sometimes each task also has a *weight* indicating its importance. We will only need explicit weight values for agent B and thus define a weight w_j for each task b_j, $j = 1, \ldots, n$. All data are known by both agents and all tasks are available at the beginning of the planning process. Each agent wants to optimize its own objective function, which only depends on the completion times of its tasks: $f^A = f^A(C_1^A, \ldots, C_n^A)$ and $f^B = f^B(C_1^B, \ldots, C_n^B)$, where C_j^X is the completion time of task j of agent X ($j = 1, \ldots, n$, $X = A, B$). In this paper we consider the minimization of (i) makespan $f^X = \max\{C_1^X, \ldots, C_n^X\}$, ($ii$) total completion time $f^X = \sum_{j=1}^n C_j^X$, and (iii) total weighted completion time, e.g. $f^B = \sum_{j=1}^n w_j C_j^B$. We denote by (f^A, f^B) the problem where f^A and f^B are the two agents' objective functions.

The decision process is divided into $2n$ rounds each consisting of the following two steps.

Download English Version:

https://daneshyari.com/en/article/433833

Download Persian Version:

https://daneshyari.com/article/433833

Daneshyari.com