Discrete Applied Mathematics 186 (2015) 295–300

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Note

A note on the lower bound for the Price of Anarchy of scheduling games on unrelated machines

Yujie Yan [a], Zhihao Ding [a,b], Zhiyi Tan [a,*]

[a] Department of Mathematics, Zhejiang University, Hangzhou 310027, PR China

[b] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, USA

ARTICLE INFO

Article history:
Received 8 February 2014
Received in revised form 1 January 2015
Accepted 7 January 2015
Available online 11 February 2015

Keywords:
Scheduling game
Coordination mechanism
Price of Anarchy
Worst-case ratio

ABSTRACT

This note presents a lower bound for the Strong Price of Anarchy (SPoA) of coordination mechanisms for unrelated parallel machine scheduling games with social cost of minimizing the makespan. The SPoA of any set of non-preemptive strongly local policies satisfying the IIA property is at least m, the number of machines. Combining with the upper bound of the worst-case ratio of Shortest First algorithm for unrelated parallel machine scheduling problem with objective of minimizing the makespan (Ibarra and Kim, 1977), the SPoA of *SPT* policy, as well as the worst-case ratio of Shortest First algorithm, is exactly m.

1. Introduction

Scheduling games have been popular recently with the development of Internet and network economics. The most common model has a parallel machine setting. Given a job set $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ and a machine set $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$. Each job is processed on one of the machines, and the processing time needed for J_j on M_i is p_{ji}. Unlike classical scheduling problems where jobs are assigned by a central decision maker, each job can individually choose a machine for processing. The choices of all jobs constitute a schedule. Jobs are independent selfish players, each aiming at minimizing its individual cost, which is its own completion time on the selected machine in the schedule. A schedule is a *Nash Equilibrium* (*NE*) if no job can reduce its cost by moving to a different machine [8]. A schedule is a *Strong Equilibrium* (*SE*) if there is no coalition of jobs such that the cost of each job of the coalition will be reduced by migration simultaneously [1]. An *SE* is also an *NE*, but the reverse is not always true.

Apart from individual cost of each job, the *social cost* which reflects utilities of the overall system is also of interest. In this note, we define social cost as the makespan of the schedule, i.e., the maximum completion time of all the jobs, which is natural and common in both theory and application. Due to lack of central coordination, an *NE* or *SE* may not be optimal in terms of the social cost. To measure the inefficiency of an equilibrium, the notions of PoA and SPoA are introduced. The *Price of Anarchy* (PoA) (res., *Strong Price of Anarchy* (SPoA)) is the worst-case ratio between the social cost of any *NE* (res., *SE*) and that of a social optimum [8,1]. Obviously, SPoA $\leq$ PoA by definition.

To reduce the inefficiency of equilibria while not to impose centralized control on the jobs, one can design a *scheduling policy* for each machine [3]. Each job has the privilege to select the machine for processing. However, once all the jobs finish selecting machines, jobs on the same machine are processed according to the policy of that machine. More specifically, let the set of jobs selecting M_i be $\mathcal{J}_i$ and the scheduling policy of M_i be P_i. Once $\mathcal{J}_i$ is determined, P_i takes $\mathcal{J}_i$ as input and outputs the

* Corresponding author.
E-mail address: tanzy@zju.edu.cn (Z. Tan).

http://dx.doi.org/10.1016/j.dam.2015.01.013
0166-218X/

completion times of all the jobs in $\mathcal{J}_i$. The set of scheduling policies of each machine forms a *coordination mechanism* [3]. The central problem on coordination mechanism is to analyze properties of common policies and design effective mechanism with the smallest possible PoA.

Scheduling policies can be classified into different types in terms of how much information they need to determine the completion times [2]. Policy P_i is *local* if it only uses the information of the processing times of $\mathcal{J}_i$. Policy P_i is *strongly local* if it only uses the information of the processing times of $\mathcal{J}_i$ on M_i. A policy is *non-preemptive* if it processes each job in a continuous fashion without any idle time. Policy P_i satisfies the property of the *independence of irrelevant alternatives* (IIA) if the precedence order of any $J_{j_1}, J_{j_2} \in \mathcal{J}_i$ is not influenced by the availability of any other job in $\mathcal{J}_i$. Policy P_i is an ordering policy if M_i gives a global order of all the jobs, and jobs selecting M_i are processed according to it. A non-preemptive policy satisfying IIA is always an ordering policy [2]. In [2] and [4], it is proved that the PoA and SPoA of any set of non-preemptive strongly local policies satisfying the IIA property are at least $\frac{m}{2}$, respectively.

Another line of research concentrates on the inefficiency of certain scheduling policies [7]. Suppose that $\mathcal{J}_i = \{J_{j_1}, J_{j_2}, \ldots, J_{j_l}\}$ and $p_{j_1,i} \le p_{j_2,i} \le \cdots \le p_{j_l,i}$. In the *MAKESPAN* policy, all jobs of $\mathcal{J}_i$ are processed in parallel, and thus the completion time of each job is $\sum_{k=1}^{l} p_{j_k,i}$. Policy *SPT* (res., *LPT*) sequences jobs of $\mathcal{J}_i$ in non-decreasing (res., non-increasing) order of their processing times on M_i, and processes them one by one non-preemptively. Thus the completion time of J_{j_s} under *SPT* (res., *LPT*) policy is $\sum_{k=1}^{s} p_{j_k,i}$ (res., $\sum_{k=s}^{l} p_{j_k,i}$). Clearly, both *SPT* and *LPT* are non-preemptive strongly local policies.

We summarize below some main results on the PoA and SPoA of certain coordination mechanisms. More results for the cases under certain special machine environments can be found in [7]. For notation simplicity, the PoA (res., SPoA) of scheduling policy P is in fact the PoA (res., SPoA) of coordination mechanism where the policy of each machine is P. It has been shown that *NE* always exists under all policies mentioned above [7]. The PoA of *MAKESPAN* policy is unbounded and the SPoA of *MAKESPAN* policy is exactly m [9,1,5]. The PoA of *SPT* policy is obtained in a different way. For the classical unrelated machine scheduling, there is an algorithm called *Shortest First* (Algorithm D of [6]). Shortest First selects the job among all unassigned jobs that can be completed the earliest if it is assigned to the appropriate machine. The worst-case ratio of Shortest First for the unrelated machine scheduling problem with the objective of minimizing the makespan is at most m [6]. In [7], it is proved that the set of *NE* for the *SPT* policy is precisely the set of schedules that can be generated by the Shortest First algorithm. Thus the PoA of *SPT* policy is the same as the worst-case ratio of Shortest First.

In this note, we show that the SPoA of any set of non-preemptive strongly local policies satisfying the IIA property is at least m. Combining the upper bound of Shortest First [6], the PoA and SPoA of *SPT* policy are both m, and the worst-case ratio of Shortest First algorithm for unrelated machine scheduling with objective to minimize the makespan is exactly m. Thus the long-standing gap between the lower and upper bounds on the worst-case ratio of Shortest First algorithm, as well as the PoA of *SPT* policy, is eliminated.

2. Lower bound

In this section, we present a lower bound on the SPoA of any set of non-preemptive strongly local policies satisfying the IIA property. The proof uses the same idea in [2,4] but a new instance, which results in a tight bound. Suppose that P_i is the policy of M_i such that P_i is non-preemptive, strongly local and satisfies the IIA property, $i = 1, \ldots, m$. Thus P_i is an ordering policy. We will construct a job set $\mathcal{J}$ whose SPoA is m.

For a subset $S \subseteq \mathcal{J}$, let $J_{k,i}(S)$ be the job at the kth position among jobs of S under the policy P_i, $k = 1, \ldots, |S|$, $i = 1, \ldots, m$.

Let x be a multiple of $m!$. Define $n_i = \frac{(x+1)(m-1)!x^{m-i-1}}{(i-1)!}$, $1 \le i \le m-1$ and $n_m = 1$. Let $w_0 = 1$, $w_i = \sum_{l=1}^{i} n_l$ for any $i = 1, \ldots, m$, and $|\mathcal{J}| = w_m = \sum_{l=1}^{m} n_l$. We partition $\mathcal{J}$ into m disjoint subsets S_i, $i = 1, \ldots, m$ with $|S_i| = n_i = w_i - w_{i-1}$ as follows (see Fig. 1).

First let $S_m = \{J_{n,m}(\mathcal{J})\}$. Then recursively define $\mathcal{J}^i$ and S_i for $i = m-1, \ldots, 1$ as follows. Let $\mathcal{J}^i$ be the complement set of the union of the $m-i$ subsets $S_{i+1}, \ldots, S_m$, i.e.,

$$\mathcal{J}^i = \mathcal{J} \setminus \bigcup_{l=i+1}^{m} S_l.$$

Note that

$$|\mathcal{J}^i| = |\mathcal{J}| - \sum_{l=i+1}^{m} |S_l| = w_m - \sum_{l=i+1}^{m} n_l = w_i.$$

The subset S_i consists of the n_i jobs which are ordered last among jobs of $\mathcal{J}^i$ under policy P_i, i.e.,

$$S_i = \{J_{w_{i-1}+1,i}(\mathcal{J}^i), J_{w_{i-1}+2,i}(\mathcal{J}^i), \ldots, J_{w_i,i}(\mathcal{J}^i)\}.$$

Now we define the processing time of all jobs. For the first $m-1$ subsets $S_1, S_2, \ldots, S_{m-1}$, jobs of S_i can be processed on M_i and M_{i+1}. The job in the last subset S_m can only be processed on M_m. All the jobs that can be processed on M_i have the same processing time on M_i equaling to $p_i = \frac{(i-1)!x^{i-m}}{(m-1)!}$. It is easy to verify that

$$\frac{p_i}{p_{i+1}} = \frac{1}{ix}, \quad i = 1, \ldots, m-1. \tag{1}$$