Discrete Optimization 8 (2011) 513–524

Contents lists available at SciVerse ScienceDirect

Discrete Optimization

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

Coordination mechanisms with hybrid local policies

Kangbok Lee [a], Joseph Y-T. Leung [b,*], Michael L. Pinedo [c]

[a] Department of Supply Chain Management & Marketing Sciences, Rutgers Business School, 1 Washington Park, Newark, NJ 07102, USA

[b] Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA

[c] Department of Information, Operations & Management Sciences, Stern School of Business, New York University, 44 West 4th Street, New York, NY 10012-1126, USA

ARTICLE INFO

Article history:
Received 9 November 2009
Received in revised form 1 April 2011
Accepted 2 May 2011
Available online 17 June 2011

Keywords:
Coordination mechanism
Price of anarchy
Mixed local policy
Makespan
Total completion time
Eligibility constraints

ABSTRACT

We study coordination mechanisms for scheduling n jobs on m parallel machines where agents representing the jobs interact to generate a schedule. Each agent acts selfishly to minimize the completion time of his/her own job, without considering the overall system performance that is measured by a central objective. The performance deterioration due to the lack of a central coordination, the so-called price of anarchy, is determined by the maximum ratio of the central objective function value of an equilibrium schedule divided by the optimal value. In the first part of the paper, we consider a mixed local policy with some machines using the SPT rule and other machines using the LPT rule. We obtain the exact price of anarchy for the problem of minimizing the makespan and some bounds for the problem of minimizing the total completion time. In the second part of the paper, we consider parallel machine scheduling subject to eligibility constraints. We devise new local policies based on the flexibilities and the processing times of the jobs. We show that the newly devised local policies outperform both the SPT and the LPT rules.

1. Introduction

We consider the distributed scheduling of n independent jobs on m parallel machines in which all decisions are interactively done by agents who are in charge of their own jobs. Each agent has a job and selects a machine to process his/her own job so as to maximize his/her own utility, which is defined as the negation of the completion time of his/her own job. Without a central coordination, the jobs assigned to the same machine would compete to finish as early as possible, which will lead to chaos. To resolve the conflict, each machine will announce, in advance, the sequencing rule used by the machine to sequence the jobs assigned to that machine. For example, if the Shortest Processing Time first (SPT) rule is used, jobs assigned to that machine are sequenced in nondecreasing order of their processing times. Similarly, if the Longest Processing Time first (LPT) rule is used, jobs are sequenced in nonincreasing order of their processing times. The sequencing rule used by a machine is usually referred to as the local policy of that machine.

However, there is a central objective function for evaluating the performance of the overall schedule. The central objective may be one of the classical scheduling objectives, such as makespan or total completion time. Due to the lack of a central coordination, the central objective may deteriorate, which is typically referred to as the *Price Of Anarchy* (POA). The POA is closely related to the concept of a Nash equilibrium. A Nash equilibrium schedule is defined as one where no agent can change his/her current decision for a better schedule. In this paper we will only consider pure Nash equilibria which means

* Corresponding author. Tel.: +1 973 596 3387; fax: +1 973 596 5777.
E-mail addresses: kangblee@andromeda.rutgers.edu (K. Lee), leung@cis.njit.edu, leung@oak.njit.edu (J.Y.-T. Leung), mpinedo@stern.nyu.edu (M.L. Pinedo).

1572-5286/$ – see front matter
doi:10.1016/j.disopt.2011.05.001

that each agent can only select one machine to process his/her own job. We assume that each agent is well aware of the local policy of each machine as well as all information about other jobs (including their processing times).

The POA can be formally defined as follows. Let $\mathcal{I}$ be the set of all instances and $\Lambda(I)$ be the set of all equilibrium schedules of instance $I \in \mathcal{I}$. Let Z be the central objective function. Let σ^* be an optimal schedule. Then, the POA is defined as

$$\text{POA} = \max_{I \in \mathcal{I}} \max_{\sigma \in \Lambda(I)} \left\{ \frac{Z(\sigma)}{Z(\sigma^*)} \right\}.$$

Clearly, the local policy of each machine is a critical factor affecting the overall system performance, and hence the POA. In this paper, we consider the design of local policies in different environments.

1.1. Related works

In the literature, different machine environments and different local policies have been considered. Immorlica et al. [1] investigated four different machine environments, where job j has processing time p_{ij} on machine i. The four machine environments are: (1) identical machines—$p_{ij} = p_{kj} = p_j$ for each job j and machines i and k; (2) uniform machines—$p_{ij} = p_j/s_i$, where p_j is the processing requirement of job j and s_i is the speed of machine i; (3) parallel machine scheduling with assignment restrictions—job j can only be scheduled on a subset, M_j, of machines; i.e., $p_{ij} = p_j$ if $i \in M_j$, otherwise, $p_{ij} = \infty$; and (4) unrelated machines—p_{ij} is an arbitrary positive number. As for local policies, they considered SPT, LPT, Randomized and Makespan rules [1].

Most results in the literature are concerned with the makespan as the central objective function. For identical machines, the POA is $2 - \frac{1}{m}$ if the local policy is the SPT rule [2,1], and $\frac{4}{3} - \frac{1}{3m}$ if the local policy is the LPT rule [3,4]. For two uniform machines, the POA is $\frac{1+\sqrt{5}}{2}$ for both the SPT and LPT rules [5]. For three or more uniform machines, the POA is $\Theta(\log m)$ for SPT [1,6], while that of LPT lies within the interval $[1.54, 1 + \frac{\sqrt{3}}{3}]$ [7,8]. For parallel machine scheduling with assignment restrictions, the POA is $\Theta(\log m)$ for both SPT and LPT [9,1]. For unrelated machines, the POA for SPT lies within the interval $[\log m, m]$ [5,10], while that for LPT is unbounded.

Christodoulou et al. [3] consider a hybrid local policy for two identical machines, where the first machine uses the SPT rule and the second machine uses the LPT rule. They show that the POA of this policy has a lower bound of $\frac{4}{3}$. As we shall see in Section 2, we will show that $\frac{4}{3}$ is also an upper bound for this special case.

If the central objective is the total completion time and the machine environment consists of parallel identical machines, the POA is 1 for SPT and $\frac{n}{m}$ for LPT [11]. If the central objective is the total weighted completion time and the machine environment consists of identical machines in parallel, then the POA is $\frac{\sqrt{2}+1}{2}$ for the Weighted Shortest Processing Time first (WSPT) rule [12,13].

1.2. Our contribution

In this paper we present new local policies for improving job utility or system performance. The paper consists of two parts: (1) the first part deals with a mixed local policy that will avoid starvation of either long jobs or short jobs, and (2) the second part analyzes a local policy for parallel machine scheduling subject to eligibility constraints so as to improve the system performance.

In the first part we focus on mixed local policies. In the literature, all studies assume that all machines use the same local policy. For example, either all machines use SPT or all machines use LPT. The drawback of this approach is that either the long jobs will be disadvantaged (if SPT is used) or the short jobs will be disadvantaged (if LPT is used). To overcome this drawback, we can consider a policy where some machines use SPT, while other machines use LPT. In this way, the long jobs can choose the machines that use LPT, while the short jobs can choose the machines that use SPT. We call this a *Mixed* local policy and let *Mixed*(h) denote the policy where h machines use SPT and $m-h$ machines use LPT, $1 \le h \le m-1$. We assume that all machines use the job index as a tie-breaking rule just in case two jobs have the same processing time; i.e., the job with the smaller job index will be scheduled before the job with the larger job index. In this paper we analyze the POA of Mixed local policies considering the makespan and the total completion time objectives on identical parallel machines.

In the second part of the paper, we consider parallel machine scheduling subject to eligibility constraints. The set of eligible machines for job j, M_j, is called the *eligible set* of job j. Previous results have been obtained under the assumption of arbitrary eligible sets [9,1]. However, in practice, eligible sets may have structural properties that depend on the application. We consider two special types of eligibility constraints, namely *Grade of Service (GoS) eligibility* and *Interval (Itrvl) eligibility*. For applications of these two eligibilities, the reader is referred to the survey paper by Leung and Li [14]. For GoS eligibility, each job j has a grade g_j and it is only allowed to be processed by machine i, where $1 \le i \le g_j$. Interval eligibility is a more general version of GoS eligibility in that each job j has a pair of indexes u_j and v_j such that it can be processed by a machine i, where $u_j \le i \le v_j$. We analyze the POA of traditional local policies such as the SPT and LPT rules. We propose new local policies and show that they outperform the traditional ones.