Theoretical Computer Science 700 (2017) 37–44

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Online algorithms for scheduling on batch processing machines with interval graph compatibilities between jobs

Qian Wang, Ji Tian *, Ruyan Fu, Xiangjuan Yao

School of Mathematics, China University of Mining and Technology, Xuzhou, Jiangsu 221116, PR China

A R T I C L E   I N F O

Article history:
Received 24 January 2017
Received in revised form 22 July 2017
Accepted 29 July 2017
Available online 3 August 2017
Communicated by L. Cardelli

Keywords:
Online scheduling
Batch machine
Compatibility graph
Competitive ratio

A B S T R A C T

We consider the online (over time) scheduling problem of minimizing the makespan on m unbounded parallel-batch machines, in which jobs in the same batch have to be pairwise compatible. Compatibility is a symmetric binary relation, which is represented by an interval compatibility graph. The processing time of a batch is equal to the maximum processing time of the jobs in it, and all jobs in the same batch start and finish at the same time. For this problem, firstly, we show that there exists no online algorithm with a competitive ratio less than 2. Then we provide an online algorithm with a competitive ratio $2+\frac{m-1}{m+1}$, which is optimal for the case $m=1$. When all jobs have the same processing times, we also give an optimal online algorithm.

1. Introduction

Recently, parallel-batch scheduling problems have been studied extensively. This scheduling model is motivated by operations scheduling in industries such as burn-in operations scheduling in semiconductor manufacturing, heat-treatment operations scheduling in the steel and ceramic industry (see Dobson and Nambimadom [6], Lee et al. [11]). In the literature, a great deal of research work has been done on this subject for various scheduling objectives and additional constraints (see Deng et al. [5], Fu et al. [8], Liu et al. [13], Poon and Yu [14], Potts and Kovalyov [15], Tian et al. [16], Yuan et al. [17], Zhang et al. [18]).

Online scheduling is one of the flourishing scheduling models studied over the past few decades. There are many diverse meanings of the term “on-line”. Here, online means that jobs arrive over time and the scheduler does not know the information of a job until it arrives. The quality of an on-line algorithm is usually measured by its competitive ratio. An on-line algorithm is called ρ-competitive if, for any instance, it produces a schedule with an objective value at most ρ times the objective value that given by an off-line optimal schedule. An online algorithm H is called optimal if no online algorithm has a competitive ratio less than that of algorithm H.

In this paper, we consider online batch scheduling problem under the assumption that the jobs in the same batch have to be compatible, i.e., their physical or chemical properties must be similar. This relation is represented by a compatibility graph $G=(V,E)$, where V is the set of jobs and a pair of jobs is an element of the edge set E if and only if they are compatible. Here, we focus on the case that G is an interval graph, which has much interesting applications in industry. One example occurs in many steel plants. To produce steel plates, metallic materials go through repeated cycles rolling and heating in a rolling-mill. In the process of heating, the metal coils is piled up on a base and then heated together in a bell

* Corresponding author.
E-mail address: jitian@cumt.edu.cn (J. Tian).

http://dx.doi.org/10.1016/j.tcs.2017.07.022
0304-3975/

furnace. The furnace has a finite capacity b, which means that at most b coils can be heated simultaneously. The coils heated together have to be compatible, which means in this case that they have to have similar heights. These compatibilities can be expressed by means of a tolerance height θ. A coil with height H has a tolerance interval $[H-\theta, H+\theta]$. Two metal coils are compatible if and only if their tolerance intervals intersect. Each coil J_j has to be heated for at least p_j time units. The heating time of a furnace is defined as the longest heating time of coils it contains. For a given set of metal coils (jobs), one obtains an interval graph for the job compatibilities, and a clique with size b in this graph defines an admissible batch for the furnace. The objective is to minimize the total heating time $C_{\max}$. More applications can be seen from the literature such as Bellanger et al. [1] and Finke et al. [7]. Recall that an interval graph $G=(V,E)$ is a graph for which the vertex set $V=\{I_1, I_2, \ldots, I_n\}$ can be identified with a set of intervals on the real line, such that I_i and I_j are adjacent in G if and only if $I_i \cap I_j \neq \emptyset$. The online problem studied in this paper can be described as follows:

There is a set $\mathcal{J}=\{J_1, J_2, \ldots, J_n\}$ of jobs to be scheduled on m unbounded batch machines. Each job J_j has an arrival time r_j, a processing time p_j and an interval $I_j=[a_j, b_j]$, where the interval associating with a job is for compatibility. These information of job J_j is not known in advance, but when it arrives we acquire its all information immediately. The batch machine can process any number of jobs in a batch as long as these jobs are pairwise compatible. Two jobs are compatible if and only if their intervals have a non-empty intersection. In this problem, the scheduler has to decide which jobs could form a batch and when each batch starts. In a schedule, a batch B has a processing time $p(B)=\max\{p_j : J_j \in B\}$, a starting time $S(B)$ and a completion time $C(B)=S(B)+p(B)$. All jobs in B start at the same time $S(B)$ and finish at the same time $C(B)$. The objective is to find a schedule of the jobs to minimize the makespan $C_{\max}$. Using the notation of Finke et al. [7], the problem under consideration is denoted by $Pm|\text{online}, r_j, \text{p-batch}, \text{G}=\text{INT}|C_{\max}$, where "G = INT" indicates that the compatibility graph is an interval graph.

In the literature, a lot of work studied scheduling problems on a single batch machine with graph compatibility between jobs. Boudhar and Finke [2] considered the batch scheduling problems with job compatibilities characterized by a general acyclic graph. They proved NP-hardness of the general problem and gave some polynomial algorithms for several special cases. When the compatibility graph is a bipartite graph, Boudhar [3] also showed it is NP-hard and gave some polynomial time algorithms for several special cases. Boudhar [4] studied the problems with job compatibilities for a split graph. He showed that the general problem with dynamic job arrivals is NP-hard and several special cases can be solved in polynomial time. Bellanger et al. [1] studied a scheduling problem with job processing time compatibilities. In this model, each job J_j has a normal processing time p_j, and two jobs J_i and J_j are compatible if $\max\{p_i, p_j\} \le (1+\alpha)\cdot\min\{p_i, p_j\}$, where $\alpha > 0$ is a constant. The compatibility graph is essentially a specific interval graph. When all jobs' arrival times are equal to 0, they showed that it can be solved in $O(n\log n)$ time by the BCLPT (Batch Compatible Largest Processing Time) rule. Li et al. [12] studied the same scheduling problem with dynamic job arrivals. When the number of distinct release dates is fixed, they designed a pseudo-polynomial dynamic programming algorithm, and when the number of distinct release dates is arbitrary, they presented a polynomial time approximation scheme. In online setting, they also provided a class of online algorithms with a competitive ratio of 2. Later, Fu et al. [9] established an optimal online algorithm with a competitive ratio $1+\beta_\alpha$, where β_α is the positive root of equation $(1+\alpha)x^2+\alpha x=1+\alpha$. Finke et al. [7] considered four scheduling problems on a single batch machine with job compatibilities for an arbitrary interval graph. When all jobs' arrival times are equal to 0, they presented a min–max formula and polynomial time algorithms for these problems.

In our paper, we study online scheduling problem on m unbounded batch machines with interval graph compatibilities between jobs. We show that there exists no online algorithm with a competitive ratio less than 2 in Section 2. In Section 3, we provide an online algorithm with a competitive ratio $2+\frac{m-1}{m+1}$, which is optimal for the case $m=1$. In Section 4, when all jobs have the same processing times, we also give an optimal online algorithm. Conclusions are presented in Section 5.

Throughout this paper, we use σ and π to denote the schedules created by an online algorithm and an optimal off-line algorithm, respectively. Denoted by $C_{\max}(\sigma)$ and $C_{\max}(\pi)$ the makespan of schedules σ and π, respectively.

2. A lower bound

Theorem 2.1. *For problem $Pm|\text{online}, r_j, \text{p-batch}, G=\text{INT}|C_{\max}$, there exists no online algorithm with a competitive ratio less than 2. The result still holds even when all jobs have equal processing times.*

Proof. Let K be a sufficiently large positive integer, and ϵ be an arbitrary small positive number. For any online algorithm H, we construct a specific job list L as follows.

At time 0, mK jobs $J_1, J_2, \cdots, J_{mK}$ arrive. All the jobs have a processing time 1 and their intervals are defined as

$$I_j=[j, j+\tfrac{1}{2}],\ \forall j\in\{1,2,\cdots,mK\}.$$

Clearly, $I_i \cap I_j=\emptyset$, for any $1\le i<j\le mK$. By the compatibility constraint, any two jobs from $\{J_1, J_2, \cdots, J_{mK}\}$ are incompatible, and so, they must be processed in distinct batches in any schedule. In the time interval $[0,K)$, once a job J_j from these mK jobs is processed, a new copy of J_j, i.e., a job with processing time 1 and the same interval as J_j, arrives exactly ϵ time later than the starting time of job J_j. Denoted by $\mathcal{A}_j$ the set of job J_j and its replicas, where $j\in\{1,2,\cdots,mK\}$. It is easy to find that there are still mK jobs at time K in σ, and these jobs are pairwise incompatible. Hence, the amount of processing that has to be executed after time K is at least mK, and so $C_{\max}(\sigma)\ge K+\frac{mK}{m}=2K$. Without loss of generality, in time interval $[0,K)$, we assume that all the jobs processed on m machines are from h different