Theoretical Computer Science 583 (2015) 57–66

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Single-machine batch scheduling with job processing time compatibility

Shisheng Li [a,*], T.C.E. Cheng [b], C.T. Ng [b], Jinjiang Yuan [c]

[a] Department of Information and Computation Science, Zhongyuan University of Technology, Zhengzhou 450007, PR China
[b] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, PR China
[c] School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, PR China

ARTICLE INFO

Article history:
Received 25 April 2014
Received in revised form 26 February 2015
Accepted 30 March 2015
Available online 3 April 2015
Communicated by D.-Z. Du

Keywords:
Batch machine
Compatibility
Release date
Approximation algorithm

ABSTRACT

We consider scheduling problems with job processing time compatibility on a single unbounded batch machine. Each job's processing time is characterized by an interval. Any number of jobs can be processed in a batch under the condition that the processing time intervals of the jobs in the same batch have a nonempty intersection. The processing time of a batch is given by the left endpoint of the intersection of the processing time intervals of the jobs in the batch. For the makespan minimization problem with individual job release dates, we design a pseudo-polynomial dynamic programming algorithm for the case where the number of distinct release dates is fixed. We also present a class of online algorithms that are 2-competitive and a polynomial-time approximation scheme for the case where the number of release dates is arbitrary. For the problem to minimize the weighted number of tardy jobs under a common due date, we show that it is binary $\mathcal{NP}$-hard and provide a polynomial-time algorithm when the jobs have a common weight.

1. Introduction

We are given a set of jobs $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$, which are to be scheduled and processed on a single unbounded batch machine. The actual processing time of job J_j falls within the interval $[a_j, (1+\alpha)a_j]$, where $\alpha > 0$ is a given number and a_j is the normal processing time of job J_j, $j = 1, 2, \dots, n$. Each job J_j is associated with a due date d_j, a release date r_j, and a weight w_j. The jobs are processed in batches and all the jobs in the same batch have the same starting time and will have to wait until all the jobs in the batch are completed. A batch can include any number of compatible jobs. We assume, as in Bellanger et al. [1], that the jobs are compatible if their processing time intervals have a nonempty intersection. The processing time of a batch $\mathcal{B}$, denoted by $p(\mathcal{B})$, is determined as the left endpoint of the intersection of the corresponding job processing time intervals, i.e., $p(\mathcal{B}) = \max\{a_j : J_j \in \mathcal{B}\}$. Hence, if job J_j is assigned to batch $\mathcal{B}$, then its actual processing time is $p(\mathcal{B})$. The goal is to find a schedule of the jobs so as to minimize the makespan $C_{\max}$ (i.e., the completion time of the last job) or the weighted number of tardy jobs. Using the notation of Bellanger et al. [1], we denote the problems under study as $1|p\text{-}batch, G = \alpha\text{-INT}, r_j|C_{\max}$ and $1|p\text{-}batch, G = \alpha\text{-INT}, d_j = d|\sum w_j U_j$.

The scheduling model described above is closely related to batch scheduling with job compatibility. The original batch scheduling model without job compatibility is motivated by the burn-in operations that are performed in ovens during the final testing stage of circuit board manufacturing (Lee et al. [16] and Uzsoy et al. [26,27]). Other applications of this model

* Corresponding author.
E-mail address: shishengli96@163.com (S. Li).

http://dx.doi.org/10.1016/j.tcs.2015.03.043
0304-3975/

have been found in many real-life industry applications, including numerically controlled routers for cutting metal sheets or printed circuit boards, heat treatment facilities in the steel and ceramic industries, the diffusion area in semiconductor wafer fabrication facilities, a variety of operations in the manufacturing of integrated circuits such as diffusion, oxidation, and certain chemical-vapor deposition processes (see, e.g., Hochbaum and Landy [13], Dobson and Nambimadom [10], and Mathirajan and Sivakumar [20]). Brucker et al. [6] provide an extensive discussion of single-machine batch scheduling problems with various objective functions and assumptions on the batch size. Potts et al. [23] consider two-machine flowshop, job shop, and open shop parallel-batch scheduling problems to minimize the makespan. Extensive surveys of batch scheduling models without job compatibility can be found in [5,7].

Generally, there are two categories of batch scheduling with job compatibility in the literature. The first category is concerned with problems where compatibility is characterized by compatible job families, in which jobs of different families cannot be processed together in the same batch. Dobson and Nambimadom [10], Jolai [15], Malve and Uzsoy [19], and Chakhlevitch et al. [8] consider the situation where all the jobs in each family have the same processing times. Yuan et al. [28], Nong et al. [21], Li and Yuan [18], Fu et al. [12], and Li and Chen [17] consider the situation where all the jobs in each family have distinct processing times. The second category is concerned with problems where compatibility is characterized by compatibility graphs, in which the vertices are the jobs and two vertices are connected by an edge if and only if the vertices (jobs) are compatible, so that every batch has to contain only pairwise compatible jobs. Boudhar and Finke [4] consider batch scheduling problems in which compatibility is characterized by a general acyclic graph. Boudhar [3] considers problems in which compatibility is characterized by a bipartite graph. Cheng et al. [9] analyze batch scheduling problems in which batch (job) processing may need to satisfy precedence relations. They show that the makespan and total completion time minimization problems are both strongly $\mathcal{NP}$-hard. Finke et al. [11] study problems in which compatibility is given by an arbitrary interval graph, motivated by the production of metallic office equipment. They consider makespan minimization problems with given batch capacities and various types of compatibility graphs. Bellanger and Oulamara [2] analyze a two-stage flexible flowshop scheduling problem with batch machines and job compatibility. Oulamara et al. [22] consider a two-machine flowshop problem with conventional and batch machines in the first and second stages, respectively, and arbitrary job compatibility. Bellanger et al. [1] study a scheduling problem with job processing time compatibility, in which the job's actual processing time is given by an interval and jobs are compatible if their processing intervals have a nonempty intersection, motivated by the undertaking of galvanic operations, chemical milling operations, and temperature testing operations in metal processing. They show that the makespan minimization problem is polynomially solvable and the maximum lateness minimization problem is $\mathcal{NP}$-hard even if there are only two distinct due dates, and present a dynamic programming algorithm for the total completion time minimization problem.

In this paper we continue the work of Bellanger et al. [1] and address the batch scheduling problem with job processing time compatibility in the presence of dynamic job arrivals (release dates). This problem has not been examined in the scheduling literature, so it is of considerable theoretical interest. In addition, it is of practical interest for several reasons. First, minimizing the makespan $C_{\max}$ in the presence of dynamic job arrivals can be a surrogate for maximizing machine utilization. Second, minimizing $C_{\max}$ with dynamic job arrivals is equivalent to the problem of minimizing the maximum lateness $L_{\max}$ when all the jobs are available simultaneously. This latter problem is of considerable interest to metal processing where galvanic operations and chemical milling operations are performed on jobs before they are shipped to customers.

The remaining of this paper is organized as follows: In Section 2 we first give some preliminary results for problem $1|p\text{-}batch, G=\alpha\text{-INT}, r_j|C_{\max}$; second, we present a pseudo-polynomial-time dynamic programming algorithm for the case where the number of distinct release dates is fixed, which implies that the equivalent problem $1|p\text{-}batch, G=\alpha\text{-INT}|L_{\max}$ with a fixed number of distinct due dates is binary $\mathcal{NP}$-hard; third, we design a class of online batch compatible largest processing time-based (BCLPT-based) algorithms that are 2-competitive; finally, we provide a polynomial-time approximation scheme for the case where the number of release dates is arbitrary. In Section 3 we show that problem $1|p\text{-}batch, G=\alpha\text{-INT}, d_j=d|\sum w_jU_j$ is binary $\mathcal{NP}$-hard and present an $O(n^2\sum_{j=1}^n w_j)$ algorithm, which implies that the unit-weighted case can be solved in $O(n^3)$ time. In Section 4 we conclude the paper and suggest some topics for future research.

2. Minimizing the makespan with release dates

In this section we consider the scheduling problem $1|p\text{-}batch, G=\alpha\text{-INT}, r_j|C_{\max}$. Bellanger et al. [1] show that problem $1|p\text{-}batch, G=\alpha\text{-INT}|L_{\max}$ is $\mathcal{NP}$-hard even for the case with two distinct due dates, which implies that problem $1|p\text{-}batch, G=\alpha\text{-INT}, r_j|C_{\max}$ is $\mathcal{NP}$-hard for the case with two distinct release dates. However, the strong $\mathcal{NP}$-hardness of problem $1|p\text{-}batch, G=\alpha\text{-INT}|L_{\max}$ remains an open question. We first devise a pseudo-polynomial algorithm for the case where the number of distinct release dates is fixed. Then we give a class of online algorithms that are 2-competitive. Finally, we provide a polynomial-time approximation scheme (PTAS) for the case where the number of release dates is arbitrary.

2.1. Preliminaries

Before explaining our algorithms, we first describe the *batch compatible largest processing time* (**BCLPT**) rule of Bellanger et al. [1], which finds an optimal solution for problem $1|p\text{-}batch, G=\alpha\text{-INT}|C_{\max}$.