Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 177 (2007) 1302–1309
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Short Communication

The single machine batching problem with identical family setup times to minimize maximum lateness is strongly NP-hard ☆

L.F. Lu, J.J. Yuan *

Department of Mathematics, Zhengzhou University, Zhengzhou, Henan 450052, PR China

Received 3 March 2005; accepted 28 December 2005
Available online 20 February 2006

Abstract

In this paper, we consider the single machine batching problem with family setup times to minimize maximum lateness. Recently, Cheng et al. [T.C.E. Cheng, C.T. Ng, J.J. Yuan, The single machine batching problem with family setup times to minimize maximum lateness is strongly NP-hard, Journal of Scheduling 6 (2003) 483–490] proved that this problem is strongly NP-hard. This answers a long-standing open problem posed by J. Bruno and P. Downey [Complexity of task sequencing with deadlines, setup times and changeover costs, SIAM Journal on Computing 7 (1978) 393–404]. By a modification of the proof in Cheng et al. (2003), we show that this problem is still strongly NP-hard when the family setup times are identical.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Scheduling; Batching; Due-dates; Maximum lateness; Multi-operation jobs

1. Introduction and problem formulation

In the last decade, there has been significant interest in scheduling problems that involve an element of batching. In this context, the motivation for batching jobs is a gain in efficiency: it may be cheaper or faster to process jobs in a batch than to process them individually.

One situation where benefits may result from batching occurs when machines require setups if they are to process jobs that have differing characteristics. The setup may reflect the need to change a tool or to clean the machine. In a family scheduling model, jobs are partitioned into families according to their similarity, so that no setup is required for a job if it belongs to the same family of the preciously processed job. However, a setup time is required at the start of the schedule and on each occasion when the machine switches from processing

☆ Project supported by NSFC (10371112), NSFHN (0411011200) and SRF for ROCS, SEM.
* Corresponding author.
E-mail address: yuanjj@zzu.edu.cn (J.J. Yuan).

0377-2217/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2005.12.027

jobs in one family to jobs in another family. In this model, a batch is a maximal set of jobs that are scheduled contiguously on a machine and share a setup.

There are two variants of the family scheduling model depending on when the jobs become available. Under batch availability, a job only becomes available when the complete batch to which it belongs has been processed. An alternative assumption is job availability (usually known in the literature as item availability), in which a job becomes available immediately after its processing is completed. In this paper, we adopt the assumption of job availability.

In the single machine, family jobs, batch scheduling problem (see [1,6]), we have n jobs $J_1, J_2, \ldots, J_n$ and F families of jobs $\mathscr{F}_1, \mathscr{F}_2, \ldots, \mathscr{F}_F$, which partition the job set $\{J_1, J_2, \ldots, J_n\}$. Each job J_j has a processing time p_j and a due-date d_j, and each family $\mathscr{F}_f$ has an associated setup time s_f. The jobs in a family are processed in batches and each batch of family $\mathscr{F}_f$ will incur a setup time s_f. In the literature, the problem is denoted by $1|s_f|V$, where V is the objective function to be minimized. Note that we adopt the assumption of job availability. So, if $B = \{J_{x_1}, J_{x_2}, \ldots, J_{x_k}\}$ is a batch of a family $\mathscr{F}_f$ in a certain schedule π, the jobs in B are processed in the order $(J_{x_1}, J_{x_2}, \ldots, J_{x_k})$ in π and the starting time (of the setup) of batch B under π is t, then the completion time of J_{x_i}, $1 \leqslant i \leqslant k$, under π is

$$C_{x_i}(\pi) = t + s_f + p_{x_1} + p_{x_1} + \cdots + p_{x_i}.$$

For a given schedule π, we define $U_j(\pi) = 0$ if $C_j(\pi) \leqslant d_j$, and $U_j(\pi) = 1$ if $C_j(\pi) > d_j$, $1 \leqslant j \leqslant n$. Hence, a job J_j is tardy if and only if $U_j(\pi) = 1$. We also define the lateness of a job J_j as $L_j(\pi) = C_j(\pi) - d_j$, $1 \leqslant j \leqslant n$. The objective considered in this paper is to find a schedule π that minimizes the maximum lateness $L_{\max}(\pi) = \max_{1 \leqslant j \leqslant n} L_j(\pi)$.

As a example, we consider an instance with six jobs that are partitioned into two families defined by $\{J_1, J_2, J_3\}$ and $\{J_4, J_5, J_6\}$, respectively. Let $s_1 = s_2 = 3$. The processing times are $\{5, 7, 7\}$ and $\{3, 10, 10\}$, and the due-dates are $\{8, 24, 31\}$ and $\{14, 44, 54\}$, respectively. Clearly, we can find that, in the optimal schedule π, both families are split in two batches, resulting in the sequence $(\{J_1\}, \{J_4\}, \{J_2, J_3\}, \{J_5, J_6\})$ such that $L_{\max}(\pi) = 0$.

Bruno and Downey [1] first considered the maximum lateness scheduling problem $1|s_f|L_{\max}$ in 1978. They proved that the problem $1|s_f|L_{\max}$ is binary NP-hard. The best algorithm for the problem $1|s_f|L_{\max}$ is a dynamic programming algorithm given by Ghosh and Gupta [4] with a time bound $O(F^2N^F)$, where $N = \frac{1}{F}\sum_{1 \leqslant f \leqslant F} |\mathscr{F}_f| + 1$. Correspondingly, it is shown by Gerodimos et al. [3] that, the problem $1|s_f$,assembly$|L_{\max}$ is binary NP-hard, and can be solved by applying the dynamic programming algorithm given by Ghosh and Gupta [4] with a time bound $O(F^2n^F)$. The strongly NP-hardness of the problem $1|s_f|L_{\max}$ was proved by Cheng et al. [5] in 2003. This answers a long-standing open problem posed by Bruno and Downey [1] in 1978. However, whether the problem $1|s_f = s|L_{\max}$ is strongly NP-hard is still an open problem.

To clarify the arguments, we will use the notation of the single machine, multi-operation jobs, assembly scheduling problem for the discussion.

As introduced by Gerodimos et al. [3], the single machine, multi-operation jobs, assembly scheduling problem arises in a food manufacturing environment. It can be stated as follows: Let n multi-operation jobs $J_1, J_2, \ldots, J_n$ and a single machine that can handle only one job at a time be given. Each job consists of several operations that belong to different families. There are F families $\mathscr{F}_1, \mathscr{F}_2, \ldots, \mathscr{F}_F$. We assume that each job has at most one operation in each family. If job J_j has an operation in family $\mathscr{F}_f$, then we denote this operation by (f,j) and its associated processing time by $p_{(f,j)} > 0$. The processing time of each job J_j is defined by $p_j = \sum_{(f,j) \in \mathscr{F}_f} p_{(f,j)}$. Each family $\mathscr{F}_f$ has an associated setup time s_f. If in a schedule the operations of a family $\mathscr{F}_f$ are processed in batches, then each batch will incur a setup time s_f. The completion time of an operation is calculated by the assumption of item availability. A job completes when all of its operations have been processed. Hence, the completion time of the job J_j under a schedule π is

$$C_j(\pi) = \max\{C_{(f,j)}(\pi) : (f,j) \text{ is an operation of job } J_j\},$$

where $C_{(f,j)}(\pi)$ is the completion time of the operation (f,j). Suppose that the due-date of each job J_j is d_j, $1 \leqslant j \leqslant n$. The objective is to find a schedule π that minimizes the maximum lateness $L_{\max}(\pi) = \max_{1 \leqslant j \leqslant n} L_j(\pi)$. We call this problem the single machine, multi-operation jobs, maximum lateness scheduling problem. Following [3], we denote the problem by

$$1|s_f, \text{assembly}|L_{\max},$$

Download English Version:

https://daneshyari.com/en/article/477856

Download Persian Version:

https://daneshyari.com/article/477856

Daneshyari.com