Information Processing Letters 110 (2010) 510–513

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On-line two-machine job shop scheduling with time lags

Xiandong Zhang [a,*], Steef van de Velde [b]

[a] Department of Management Science, School of Management, Fudan University, Shanghai 200433, China
[b] Rotterdam School of Management, Erasmus University, PO Box 1738, 3000 DR Rotterdam, The Netherlands

ARTICLE INFO

Article history:
Received 30 September 2009
Received in revised form 2 April 2010
Accepted 2 April 2010
Available online 8 April 2010
Communicated by M. Yamashita

Keywords:
Scheduling
Job shop
Time lags
Greedy algorithm
Competitive analysis

ABSTRACT

We consider the on-line two-machine job shop scheduling problem with time lags so as to minimize the makespan. Each job consists of no more than two operations and time lags exist between the completion time of the first and the start time of the second operation of any two-operation job. We prove that any greedy algorithm is 2-competitive. For the non-clairvoyant variant of the problem, no on-line algorithm can do better. For the clairvoyant variant, no on-line delay algorithm has a competitive ratio better than $\frac{\sqrt{5}+1}{2} \approx 1.618$, and a greedy algorithm is still the best on-line non-delay algorithm. We also show that the same results hold for the two-machine flow shop problem with time lags.

1. Introduction

We consider the two-machine job shop problem with time lags, where jobs arrive over time, to minimize the makespan. In such a system, there is a set $\mathcal{J}$ of n independent jobs $J_1, \ldots, J_n$ that needs scheduling on two machines M_1 and M_2. Each job $J_j \in \mathcal{J}$ consists of no more than two operations O_{ij} $(i = 1, 2)$, and operation O_{ij} requires processing on machine M_i during an uninterrupted non-negative processing time p_{ij} $(i = 1, 2; j = 1, \ldots, n)$. The sequence of operations for each job is prescribed. Let $\mathcal{J}^1$ be the set containing all jobs for which O_{1j} has to be scheduled before O_{2j} or O_{2j} is missing (hence need not be scheduled), and let $\mathcal{J}^2$ be the set containing all jobs for which O_{2j} has to be scheduled before O_{1j} or O_{1j} is missing, where $j = 1, \ldots, n$. We have $\mathcal{J} = \mathcal{J}^1 \cup \mathcal{J}^2$.

Either machine is available from time 0 onwards and can handle only one job at a time. For each job J_j there is a time lag l_j required between the completion of its first and the start of its second operation. All jobs have release times, which means that the first operation of any job J_j cannot be started before its release time r_j $(j = 1, \ldots, n)$. Preemption of jobs, that is, interrupting a job and resuming in at a later point in time, is not allowed. The objective is to minimize the maximum completion time $C_{\max}$, that is, to find a schedule of minimum length or makespan. Following the standard three-field $\alpha|\beta|\gamma$ scheduling notation (Graham et al. [2]), we denote the problem as $J2|o_j \leqslant 2, r_j, l_j|C_{\max}$, where o_j is the number of operations of job J_j $(j = 1, \ldots, n)$. If $\mathcal{J}^2 = \emptyset$ or $\mathcal{J}^1 = \emptyset$, the problem reduces to the corresponding two-machine flow shop problem, denoted as $F2|r_j, l_j|C_{\max}$.

Time lags have several practical interpretations. They can model the transportation times between machines if the number of vehicles is not restrictive, or if the jobs can travel by themselves, like for example barges sailing between port terminals for loading and unloading containers. Time lags can also model required heating or cooling down times.

The complexity of the *off-line* version of $J2|o_j \leqslant 2, r_j, l_j|C_{\max}$, where all job data are known a priori, is relatively well understood. It is strongly NP-hard, even in the case of unit processing times, since the two-machine flow shop problem $F2|p_{ij} = 1, l_j|C_{\max}$ is already strongly NP-

* Corresponding author. Tel.: +86 021 25011156, fax: +86 021 65642412.
E-mail addresses: xiandongzhang@fudan.edu.cn (X. Zhang), svelde@rsm.nl (S. van de Velde).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2010.04.002

hard (Yu [8]; Yu et al. [9]). Dell'Amico [1] showed that any instance of $J2|o_j \leqslant 2, l_j|C_{\max}$ can be solved by solving two instances of $F2|l_j|C_{\max}$. Therefore, if all time lags are equal or if the solution of $F2|l_j|C_{\max}$ is restricted to the class of permutation schedules, the related two cases of $J2|o_j \leqslant 2, l_j|C_{\max}$ are polynomially solvable. Panwalkar [5] identified another well-solvable special case of the job shop problem with time lags. As far as we know, there exists no approximation algorithm for the general $J2|o_j \leqslant 2, l_j|C_{\max}$ problem.

We study the *on-line* version, where the jobs dynamically arrive at a priori unknown points in time (the so-called *release times*) and the job data are not known a priori. We also do not know the number of jobs to be scheduled. In particular, we study the *non-clairvoyant* variant, in which the processing time of an operation is unknown until it has finished, and the required time lag is unknown until it has elapsed.

The quality of an on-line algorithm is typically measured by its *competitive ratio*, and an on-line algorithm is called ρ-competitive if the objective value of the schedule produced by the on-line algorithm is at most ρ times the value of an optimal off-line solution, for any instance of the problem. An on-line algorithm is called *best possible* if no one-line algorithm has a lower competitive ratio.

Results for on-line job shop and flow shop scheduling problems with time lags are very scarce. For the case with unit execution time and arbitrary time lags without release times, Rayward-Smith and Rebaine [6] present $(2 - \frac{3}{n+2})$-competitive algorithms for $F2|\textit{on-line}, p_{ij} = 1, l_j|C_{\max}$. The competitive ratio is proved to be tight, which means that the ratio holds with equality for specific instances of the problem. For the case without time lags, Sgall [7] shows that no deterministic algorithm is better than 2-competitive for $F2|\textit{on-line}|C_{\max}$. For the on-line two-machine open shop problem with time lags, Zhang and Van de Velde [10] prove that any greedy algorithm has a tight competitive ratio of 2 and this ratio is 5/3 in case of small time lags, that is, if the maximum time lag is no larger than the smallest processing time. *A greedy algorithm* for an on-line scheduling problem with time lags assigns to a machine any available operation as soon as the machine becomes available. Zhang and Van de Velde [10] also prove that no on-line non-delay algorithm can have a better competitive ratio. As far as delay algorithms are concerned, that is, algorithms that allow a machine to be idle while an operation is available for processing, no delay algorithm can do better than a greedy algorithm for the non-clairvoyant variant of the problem. For the clairvoyant variant, no on-line delay algorithm has a competitive ratio better than $\sqrt{2}$.

In this paper, we analyze the performance of a greedy algorithm for the on-line version of $J2|o_j \leqslant 2, r_j, l_j|C_{\max}$ that processes an available operation as soon as possible, with ties broken arbitrarily. Accordingly, the resulting schedule is *non-delay*, that is, no machine is kept idle while an operation is waiting to be processed.

We prove that the competitive ratio of any greedy algorithm is 2, this bound is tight, and no on-line non-delay algorithm can do better. Using an adversary strategy argument, we also prove that no on-line delay algorithm can have a better performance guarantee for the non-clairvoyant variant of the problem. For the clairvoyant version of the problem, we prove that no on-line delay algorithm can have a better competitive ratio than $\frac{\sqrt{5}+1}{2} \approx 1.618$. We prove that these results apply to $F2|\textit{on-line}, r_j, l_j|C_{\max}$ also.

2. Performance analysis of a greedy algorithm

Let G be any greedy algorithm. We prove that G is 2-competitive for the on-line two machine job shop scheduling problem with time lags.

Let r_j be the arrival time of job J_j $(j = 1, \ldots, n)$. For a given instance, let $C^*_{\max}$ denote the minimum makespan and $C^G_{\max}$ denote the makespan of the schedule given by the greedy algorithm G. Due to symmetry of the argument, we can assume without loss of generality that machine M_2 finishes last. For the schedule constructed by G, let S_{ij} and C_{ij} denote the starting and completing time of O_{ij}, respectively $(i = 1, 2;\ j = 1, \ldots, n)$.

For any subset $\mathcal{H} \subseteq \{J_1, \ldots, J_n\}$, we define

$$r(\mathcal{H}) = \min_{J_j \in \mathcal{H}} r_j,$$

$$p_1(\mathcal{H}) = \sum_{J_j \in \mathcal{H}} p_{1j},$$

$$p_2(\mathcal{H}) = \sum_{J_j \in \mathcal{H}} p_{2j},$$

$$C(\mathcal{H}) = \max_{J_j \in \mathcal{H}} (r_j + p_{1j} + l_j + p_{2j}).$$

Clearly, we have that

$$C^*_{\max} \geqslant \max_{\mathcal{H} \in \mathcal{J}} \bigl\{ r(\mathcal{H}) + p_1(\mathcal{H}), r(\mathcal{H}) + p_2(\mathcal{H}), C(\mathcal{H}) \bigr\}. \quad (1)$$

Lemma 1. *If there is no idle time before $C^G_{\max}$ on machine M_2, then $C^*_{\max} = C^G_{\max}$.*

So, in the remainder we suppose machine M_2 has idle time before $C^G_{\max}$.

Let T denote the last point in time such that M_2 is busy throughout the time interval $[T, C^G_{\max}]$ but idle immediately before time T. Consider now the jobs with $S_{2j} \geqslant T$ on machine M_2. We divide these jobs into two disjoint subsets: subset $\mathcal{X}$ contains all the jobs with $r_j < T$, and subset $\mathcal{Y}$ contains all the jobs with $r_j \geqslant T$.

Lemma 2. *If $\mathcal{X} = \emptyset$, we have $C^*_{\max} = C^G_{\max}$.*

Proof. Note that if $\mathcal{X} = \emptyset$, then $\mathcal{Y}$ cannot be empty, and hence we have that

$$C^G_{\max} = T + p_2(\mathcal{Y}) \leqslant r(\mathcal{Y}) + p_2(\mathcal{Y}) \leqslant C^*_{\max}.$$

So, if $\mathcal{X} = \emptyset$, then the greedy algorithm G has returned an optimal schedule. □

Lemma 3. *If $\mathcal{Y} \neq \emptyset$, we have $C^G_{\max} \leqslant 2C^*_{\max}$.*