Computers & Industrial Engineering 91 (2016) 42–47

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Machine scheduling with deteriorating jobs and DeJong's learning effect

Min Ji [a,*], Xiaoying Tang [a], Xin Zhang [a], T.C.E. Cheng [b]

[a] School of Computer Science and Information Engineering, Contemporary Business and Trade Research Center, Zhejiang Gongshang University, Hangzhou 310018, PR China
[b] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 22 March 2015
Received in revised form 18 October 2015
Accepted 23 October 2015
Available online 28 October 2015

Keywords:
DeJong's learning effect
Job deterioration
Scheduling
FPTAS

ABSTRACT

We consider parallel-machine scheduling with deteriorating jobs and DeJong's learning effect. We focus on the problems to minimize the total completion time and the makespan. We show that the former is polynomially solvable, while the latter is *NP*-hard, for which we provide a fully polynomial-time approximation scheme.

1. Introduction

Scheduling with deteriorating jobs was first considered by Gupta and Gupta (1988) and Brown and Yechiali (1990) independently. Later, Mosheiov (1994) studied a special case proposed by Brown and Yechiali (1990) (referred to as simple linear deterioration) and gave polynomial-time algorithms for several classical scheduling criteria. Cheng and Ding (1998) derived some relationships between scheduling problems involving time-dependent jobs with increasing/decreasing linear processing times. General results on the relations between scheduling problems involving jobs with time-dependent processing times can be found in Gawiejnowicz, Kurc, and Pankowska (2009a, 2009b), Gawiejnowicz and Kononov (2014), and Rustogi and Strusevich (2012, 2014). Focusing on scheduling problems involving jobs with time-dependent processing times, Gawiejnowicz (2008) gave a detailed survey of such scheduling problems and discussed advanced topics such as application of matrix methods; for the latest results on such scheduling problems, the reader may refer to Yin, Cheng, and Wu (2014, 2015). Furthermore, recent papers that consider scheduling with deteriorating jobs include Ji and Cheng (2009, 2010), Lee, Wu, and Liu (2009), Lee, Wang, Shiau, and Wu (2010), Gawiejnowicz and Kononov (2010), Gawiejnowicz and Lin (2010), Wang, Wang, and Ji (2011), Li, Ng, Cheng, and Yuan (2011), Ji, Hsu, and Yang (2013), and Yin, Wu, Cheng, and Wu (2015), among others. There are many applications of scheduling models where the processing time of a job is an increasing function of its start time. These include repayment of multiple loans, derusting of operations, control of queues in communication systems, etc. For a list of applications of scheduling models with deteriorating jobs, the reader may refer to Gawiejnowicz (2008).

On the other hand, it is necessary and reasonable to consider the learning effect in scheduling. Motivated by observations in the aircraft industry, Wright (1936) initiated research on the learning effect in manufacturing. Gawiejnowicz (1996) first considered learning in scheduling research, which was later popularized by Dondeti and Mohanty (1998), Biskup (1999), and Cheng and Wang (2000). Subsequently, many scheduling researchers have devoted a great deal of effort to this stream of research and proposed a large variety of position-based learning effect models (see, e.g., Agnetis, Billaut, Gawiejnowicz, Pacciarelli, & Soukhal, 2014; Cheng, Kuo, & Yang, 2013; Low & Lin, 2011; Lu, Wei, & Wang, 2012; Wang & Wang, 2013; Yin, Xu, & Wang, 2010). Biskup (2008) and Agnetis et al. (2014) provided comprehensive reviews of this line of research. Extending the work to the multi-agent scheduling context, Agnetis et al. (2014) provided details on multi-agent scheduling research. Almost all of these models considering the position-dependent learning effect suffer a common drawback that when a job's position is large enough in a schedule, its actual processing time is close to zero. Following Wright's (1936) notion of learning, DeJong (1957) proposed a new learning model $T_s = T_1(M + (1 - M)/s^m)$, where T_s is the time required for the sth cycle of the batch, T_1 is the time required for the first cycle of a batch, s stands for the sth cycle, M represents

* Corresponding author. Tel./fax: +86 571 28008303.
E-mail addresses: jimkeen@163.com (M. Ji), Edwin.Cheng@polyu.edu.hk (T.C.E. Cheng).

http://dx.doi.org/10.1016/j.cie.2015.10.015
0360-8352/

the "factor of incompressibility" ($0 \leqslant M \leqslant 1$), and m is the exponent of reduction ($0 < m < 1$). It is clear that when s increases, the time required for the sth cycle will naturally fall and approach a certain limit, i.e., MT_1. So DeJong's learning model overcomes the drawback associated with the log-linear learning model. Subsequent studies have empirically validated DeJong's learning model, e.g., Yelle (1979), Badiru (1992), Okolowski and Gawiejnowicz (2010), Ji, Yao, Yang, and Cheng (2014), etc.

Scheduling problems considering the effects of learning and job deterioration at the same time has been extensively studied in the literature (see, e.g., Huang, Wang, & Ji, 2014; Kuo & Yang, 2011; Toksari & Güner, 2010; Wang, 2009a, 2009b; Wang, Huang, Wang, Yin, & Wang, 2009; Wang & Wang, 2014; Wu, Wang, & Wang, 2011; Yang & Kuo, 2010). However, to the best of our knowledge, most studies assume that the position-dependent learning curve approaches 0 when a job's position is large enough in a schedule, which is unrealistic because this implies no further improvement can be made after some amount of production in manufacturing. DeJong's learning curve can overcome this drawback, but it has seldom been combined with deteriorating jobs despite its relevance in practice. The exception is Wang (2009a), who considered two models that combine DeJong's learning effect with deteriorating jobs, i.e., $p_{jr} = p_j\alpha(t)(M + (1 - M)r^a)$ and $p_{jr} = p_j(\alpha(t) + M + (1 - M)r^a)$, where $0 \leqslant M \leqslant 1$ is an incompressible factor. He provided polynomial-time algorithms to solve some single-machine problems. The phenomenon of learning and job deterioration occurring simultaneously can be found in many real-life situations. For example, as manufacturing becomes increasingly competitive, in order to provide customers with greater product variety, organizations are moving towards shorter production runs and frequent product changes, which give rise to the phenomenon of learning and deterioration in performing operational activities. Considering both the learning and forgetting effects in measuring productivity should be helpful in improving the accuracy of production planning and productivity estimation (see, e.g., Nembhard & Osothsilp, 2002).

In this paper we combine DeJong's learning effect with deteriorating jobs for scheduling as follows: $p_j = \bar{p}_j[M + (1 - M)r^a] + \alpha t$ ($0 \leqslant M \leqslant 1$), where a is the non-positive learning index and α is the non-negative job deterioration rate. Note that if $M = 0$, the model reduces to the model $p_{jr} = p_j r^a + \alpha t$ considered by Yang and Kuo (2010).

The rest of the paper is organized as follows: In Section 2 we provide the problem description. In Section 3 we present some useful preliminary results. In Section 4 we provide a fully polynomial-time approximation scheme (FPTAS) to minimize the makespan, and propose polynomial-time algorithm to minimize the total completion time in the parallel-machine setting. In Section 5 we conclude the paper and suggest topics for future research.

2. Notation and problem formulation

We describe the scheduling problem as follows. There is a set of n independent jobs $\{J_1, J_2, \ldots, J_n\}$ waiting to be processed on m parallel machines. Each machine can handle only one job at a time and the processing of a job cannot be interrupted. All the jobs are available for processing at time zero. In this paper we study a scheduling model that considers both learning and job deterioration simultaneously as follows:

$$p_j = \bar{p}_j[M + (1 - M)r^a] + \alpha t, \tag{1}$$

where, given a schedule, p_j is the actual processing time and $\bar{p}_j$ is the normal processing time of job J_j, $0 \leqslant M \leqslant 1$ is the incompressible factor, r is the position of J_j, $a \leqslant 0$ is the common learning index, $\alpha \geqslant 0$ is the common job deterioration rate, and t is the starting time of job J_j in the schedule.

The objectives are to minimize the makespan $C_{\max} = \max_{j=1,2,\ldots,n} C_j$ and the total completion time $\sum_{j=1}^{n} C_j$, where C_j denotes the completion time of job J_j in a given schedule. Using the three-field notation of Graham, Lawler, Lenstra, and Rinnooy Kan (1979) for describing scheduling problems, we denote the problems under study as $P_m|p_j = \bar{p}_j[M + (1 - M)r^a] + \alpha t|C_{\max}$ and $P_m|p_j = \bar{p}_j[M + (1 - M)r^a] + \alpha t|\sum C_j$.

3. Preliminary results

In this section we give some useful preliminary results for solving the scheduling problems under consideration. Given m parallel machines, we assume that the number of jobs allocated to machine i is n_i ($i = 1, 2, \ldots, m$). Then the allocation of n jobs to m machines can be expressed as $P(n, m) = (n_1, n_2, \ldots, n_m)$ with $\sum_{i=1}^{m} n_i = n$. Let $[ij]$ denote the job that occupies the jth position on machine i in a given schedule, and $p_{[ij]}$ and $\bar{p}_{[ij]}$ denote the actual processing time and the normal processing time of the job, respectively.

Lemma 1. *The completion time of the job scheduled in the jth position on machine i, i.e., $C_{[ij]}$, for $i = 1, 2, \ldots, m, j = 1, 2, \ldots, n_i$, is equal to $\sum_{k=1}^{j} \bar{p}_{[ik]} Q_{jk}$, where $Q_{jk} = (1 + \alpha)^{j-k}[M + (1 - M)k^a]$ for $k = 1, 2, \ldots, j$.*

Proof (By mathematical induction). When $j = 1$, $C_{[i1]} = \bar{p}_{[i1]}[M + (1 - M)1^a]$, so the proposition is true for $j = 1$. Assume that the proposition is true for $j = l$, i.e., $C_{[il]} = \sum_{k=1}^{l} \bar{p}_{[ik]} Q_{lk}$, where $Q_{lk} = (1 + \alpha)^{l-k}[M + (1 - M)k^a]$ for $k = 1, 2, \ldots, l$. When $j = l + 1$,

$$\begin{aligned} C_{[i,l+1]} &= C_{[il]} + \bar{p}_{[i,l+1]}[M + (1 - M)(l + 1)^a] + \alpha C_{[il]} \\ &= \sum_{k=1}^{l} \bar{p}_{[ik]}(1 + \alpha)^{l+1-k}[M + (1 - M)k^a] \\ &\quad + \bar{p}_{[i,l+1]}(1 + \alpha)^{l+1-(l+1)}[M + (1 - M)(l + 1)^a] \\ &= \sum_{k=1}^{l+1} \bar{p}_{[ik]}(1 + \alpha)^{l+1-k}[M + (1 - M)k^a] = \sum_{k=1}^{l+1} \bar{p}_{[ik]} Q_{l+1,k}, \end{aligned}$$

where $Q_{l+1,k} = (1 + \alpha)^{l+1-k}[M + (1 - M)k^a]$ for $k = 1, 2, \ldots, l + 1$.

Hence, the proposition is also true for $j = l + 1$. By the principle of mathematical induction, the lemma is established. □

Lemma 2. *If $|z_1^a - z_2^a| \leqslant \delta \min\{z_1^a, z_2^a\}$, for $0 < \delta \leqslant 1$, $a \leqslant 0$, and z_1, z_2 are positive integers, then $|(z_1 + 1)^a - (z_2 + 1)^a| \leqslant \delta \min\{(z_1 + 1)^a, (z_2 + 1)^a\}$.*

Proof. Without loss of generality, we assume $z_1 \leqslant z_2$. Noting that $z_1^a - z_2^a \leqslant \delta z_2^a$, we have

$$\left(\frac{z_1 + 1}{z_2 + 1}\right)^a \leqslant \left(\frac{z_1}{z_2}\right)^a \leqslant (1 + \delta)(a \leqslant 0). \tag{2}$$

i.e.,

$$(z_1 + 1)^a - (z_2 + 1)^a \leqslant \delta(z_2 + 1)^a. \tag{3}$$

Similarly, if $z_1 > z_2$, we have

$$(z_2 + 1)^a - (z_1 + 1)^a \leqslant \delta(z_1 + 1)^a. \tag{4}$$

From (3) and (4), we conclude that $|(z_1 + 1)^a - (z_2 + 1)^a| \leqslant \delta \min\{(z_1 + 1)^a, (z_2 + 1)^a\}$. □