Applied Mathematics and Computation 273 (2016) 570–583

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Online scheduling with linear deteriorating jobs to minimize the total weighted completion time

Ran Ma [a,c], Jiping Tao [b], Jinjiang Yuan [c,*]

[a] School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo, Henan 454000, PR China
[b] Department of Automation, Xiamen University, Xiamen, Fujian 361005, PR China
[c] School of Mathematics and Statistics, Zhengzhou University, Zhengzhou, Henan 450001, PR China

ARTICLE INFO

Keywords:
Scheduling
Online algorithms
Total weighted completion time
Linear deterioration

ABSTRACT

In this paper, we study the online scheduling of linear deteriorating jobs on a single machine to minimize the total weighted completion time. In the problem, a set of n independent linear deteriorating jobs arriving online over time has to be scheduled on a single machine, where the information of each job including its deterioration rate and weight is unknown in advance. Linear deterioration means that the processing time p_j of a job J_j is a linear function of its starting time s_j. In this paper, we assume that $p_j = \alpha_j(A + Bs_j)$, where A and B are nonnegative with $A + B > 0$ and $\alpha_j \geq 0$ is the deterioration rate of J_j. The goal is to minimize the total weighted completion time, i.e., $\sum w_j C_j$. For this problem, we provide a best possible online algorithm with a competitive ratio of $1 + \lambda(A) + \alpha_{\max} B$, where $\alpha_{\max} = \max_{1 \leq j \leq n} \alpha_j$ and $\lambda(A) = 0$ or $\lambda(A) = 1$ depending on whether $A = 0$ or $A > 0$.

1. Introduction

Traditional scheduling assumes that the processing time of a job is fixed. However, in the real world, there are numerous situations in which the processing time increases (deteriorates) as the starting time increases. For example, to schedule maintenance or cleaning, a delay often requires additional effort to accomplish the task. Other examples can be found in fire fighting, steel production, and financial management [13,21]. Scheduling of deteriorating jobs was first introduced by Browne and Yechiali [3] and Gupta and Gupta [10], independently. Following their research, this topic has received more and more attention. Related research can be found in [1,4–7,15,23,29–32] among many others.

Online scheduling has been a hot research topic in the last few decades. In contrast to the off-line version, an online algorithm must produce a sequence of decisions based on past events without any information about the unreleased jobs. The lack of knowledge of the future does not generally guarantee the optimality of the schedule generated by an online algorithm. Thus a natural issue is how to evaluate different online algorithms for a same scheduling problem. A widely used criterion to evaluate an online algorithm is its competitive ratio. For a minimization problem, the competitive ratio $\rho_{\mathcal{A}}$ of an online algorithm $\mathcal{A}$ is defined to be

$$\rho_{\mathcal{A}} = \sup\{\mathcal{A}(I)/\mathrm{OPT}(I) : I \text{is an instance with } \mathrm{OPT}(I) > 0\}.$$

* Corresponding author. Tel.: +86 13838147625; fax: +86 13838147625.
E-mail addresses: sungirlmr@hpu.edu.cn (R. Ma), taojiping@xmu.edu.cn (J. Tao), yuanjj@zzu.edu.cn (J. Yuan).

http://dx.doi.org/10.1016/j.amc.2015.10.058
0096-3003/

Here, for an instance I, $\mathcal{A}(I)$ is used to denote the objective value of the schedule generated by the online algorithm $\mathcal{A}$, and OPT(I) is the objective value of an (off-line) optimal schedule. The closer the competitive ratio approaches 1, the better the online algorithm we have. An online algorithm $\mathcal{A}$ is called best possible if no online algorithm has a competitive ratio less than $\rho_{\mathcal{A}}$.

Problem formulation:. In this paper, we focus on the online over time scheduling of linear deteriorating jobs to minimize the total weighted completion time on a single machine. Here *linear deterioration* means that the processing time of job J_j is given by $p_j = \alpha_j(A + Bs_j)$, where $A \geq 0$, $B \geq 0$, at least one of A and B is nonzero, $\alpha_j \geq 0$ is called the deterioration rate of job J_j, and $s_j \geq 0$ is the starting time of job J_j. We use r_j to denote the release date of a job J_j. In the problem, the characteristics of a job J_j, including its deterioration rate α_j and weight w_j, become known at its release date r_j. Then the released jobs will be processed on a single machine. Letting C_j be the completion time of job J_j in a given schedule, we are interested in minimizing the *total weighted completion time*: $\sum w_jC_j$. In the case that $A = 0$, for each job J_j, we have $p_j = \alpha_j(A + Bs_j) = \alpha_jBs_j$, so in order to avoid the trivial situation in which all jobs start and complete at time 0, we assume that all jobs are released at least at a given time $t_0 > 0$. If $A > 0$, we just define $t_0 = 0$. In the scheduling notation of Graham et al. [9], the online scheduling problem studied in this paper can be denoted by $1|\text{online},\ r_j \geq t_0,\ p_j = \alpha_j(A + Bs_j)|\sum w_jC_j$.

This problem has two interesting special versions. In the first special version, we have $B = 0$ and $A > 0$, and so, by scaling, we may assume that $p_j = \alpha_j$. Then the problem degenerates to the classical online scheduling problem $1|\text{online},\ r_j|\sum w_jC_j$.

In the second special version, we have $A = 0$ and $B > 0$, and so, by scaling, we may assume that $p_j = \alpha_js_j$ and call it *simple linear deterioration*. Then the problem is denoted by $1|\text{online},\ r_j \geq t_0,\ p_j = \alpha_js_j|\sum w_jC_j$.

Most related work:. There has been an enormous amount of work on scheduling of deteriorating jobs. Until recently most research has focused on the off-line setting. We do not intend to do a complete review of results in the area and only restrict our attention to some most relevant work on online scheduling problems and scheduling problems of deteriorating jobs directly related to the matter of this paper.

For problem $1|\text{online}, r_j|\Sigma C_j$, Vestjens [28] showed that no online algorithm has a competitive ratio of less than 2. Vestjens [28] and Philips et al. [22] presented distinct online algorithms with a best possible competitive ratio of 2. In the online algorithm Delayed SPT (DSPT) proposed by Vestjens [28], whenever the machine is available at the present time t, an available job J_j with the smallest processing time is scheduled if $p_j \leq t$. The technique "quasi-schedule" was introduced in their competitive ratio analysis. The same idea of shifted release times was used by Stougie (cited in Vestjens [28]) who obtained a third algorithm with a best possible competitive ratio of 2. Following their research, the technique "quasi-schedule" in [28] was also applied in [16–18,33,34]. Alternatively, in the online algorithm presented by Philips et al. [22], an optimal online preemptive schedule SRPT (Shortest Remaining Processing Time) is constructed. Whenever a job is completed in the virtual online preemptive schedule, it joins a queue of jobs waiting to be processed in the nonpreemptive schedule.

However, for problem $1|\text{online},\ r_j|\sum w_jC_j$, "quasi-schedule" and the conversion idea provided respectively in [28] and [22] are no longer practical. The crucial point is that no simple rule can solve the off-line version $1|r_j|\sum w_jC_j$, which is strongly NP-hard [14] in general. As far as we know, the first deterministic online algorithm for $1|\text{online},\ r_j|\sum w_jC_j$ was presented by Hall et al. [11]. They designed a $(3 + \epsilon)$-competitive algorithm by taking advantage of a general online framework. Using the idea of α-points and mean-busy-time relaxation, Goemans et al. [8] designed a deterministic 2.4143-competitive algorithm and a randomized 1.6853-competitive algorithm for $1|\text{online},\ r_j|\sum w_jC_j$. In 2004, for problem $1|\text{online},\ r_j|\sum w_jC_j$, Anderson and Potts [2] presented a best possible deterministic online algorithm, called Delayed SWPT (DSWPT), with a competitive ratio of 2 under the assumption that all release dates and processing times are integers. The main technique used in the competitive ratio analysis in Anderson and Potts [2] is to create a "doubled problem" and an "extended problem".

For the off-line version that all jobs are released at time t_0 and under simple linear deterioration of processing times, Mosheiov [21] showed that the schedule in which the jobs are arranged in the non-decreasing order of the growth rates α_j (Smallest Growth Rate or SGR for short) can minimize the total completion time. Moreover, Mosheiov [21] also showed that the schedule in which the jobs are arranged in the non-decreasing order of the ratios $\alpha_j/((1 + \alpha_j)w_j)$ can minimize the total weighted completion time. Using the idea of SGR, for problem $1|\text{online},\ r_j \geq t_0,\ p_j = \alpha_js_j|\sum C_j$, Liu et al. [16] first introduced an optimal scheduling rule for the corresponding preemptive-resumption model. They further showed that $1 + \alpha_{\max}$ is a lower bound of competitive ratio for the non-preemptive problem, where $\alpha_{\max} = \max_j \alpha_j$ is the maximum deterioration rate of all jobs, and presented an online algorithm named D-SGR (Delayed Smallest Growth Rate). By using the technique "quasi-schedule", Liu et al. [16] showed that D-SGR is best possible for problem $1|\text{online},\ r_j \geq t_0,\ p_j = \alpha_js_j|\sum C_j$ with a competitive ratio of $1 + \alpha_{\max}$. For problem $1|\text{online},\ r_j \geq t_0,\ p_j = \alpha_js_j|\sum C_j^{\beta}$, where $\beta > 0$ is the general index of completion times, Yu and Wong [34] showed that no deterministic online algorithm can be better than $(1 + \alpha_{\max})^{\beta}$-competitive. Also using the technique "quasi-schedule", they gave a best possible online algorithm for $1|\text{online},\ r_j \geq t_0,\ p_j = \alpha_js_j|\sum C_j^{\beta}$ with a competitive ratio of $(1 + \alpha_{\max})^{\beta}$.

By Kononov [12], the off-line scheduling problem $1|p_j = \alpha_j(A + Bs_j)|\sum w_jC_j$ is solvable in $O(n\log n)$ time by scheduling jobs in the non-decreasing order of $\alpha_j/((1 + \alpha_jB)w_j)$ values. To the best of our knowledge, there are no results for online scheduling with linear deteriorating jobs to minimize the total weighted completion time.

Methodology and our contribution:. Tao et al. [24] first introduced the technique called "instance reduction" in their research for an online scheduling problem. The technique was also applied in [25–27]. Ma and Yuan [19] extended the technique "instance reduction" in their research for the online scheduling with job rejection to minimize the total weighted completion time plus