Information Processing Letters 113 (2013) 593–598

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

# Online scheduling on a single machine with rejection under an agreeable condition to minimize the total completion time plus the total rejection cost ☆

Ran Ma [a,b], Jinjiang Yuan [a,*]

[a] School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China
[b] School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454000, China

ARTICLE INFO

*Article history:*
Received 21 September 2012
Received in revised form 7 May 2013
Accepted 9 May 2013
Available online 14 May 2013
Communicated by B. Doerr

*Keywords:*
Scheduling
Online algorithms
Total completion time
Rejection

ABSTRACT

We consider the single-machine online scheduling with job rejection to minimize the total completion time of the scheduled jobs plus the total rejection cost of the rejected jobs under an agreeable condition on the processing times and rejection penalties of the jobs. In the problem, a set of independent jobs arriving online over time has to be scheduled on the machine with the flexibility of rejecting some of the jobs, where preemption is not allowed and the information of each job $J_j$, including its processing time $p_j$, release date $r_j$ and rejection penalty $e_j$, is not known in advance. The agreeable condition means that, for every two jobs $J_i$ and $J_j$, $p_i < p_j$ implies $e_i \geqslant e_j$, and furthermore, $p_i = p_j$ and $r_i < r_j$ imply $e_i \geqslant e_j$. For this problem, we provide an online algorithm with the best possible competitive ratio of 2. As a consequence, the online algorithm is also best possible when the jobs have identical rejection penalties.



## 1. Introduction

In the scheduling problem with job rejection, we have a set of $n$ jobs $\{J_1, \dots, J_n\}$ and a single machine. Each job $J_j$ has a processing time $p_j > 0$, a release date $r_j \geqslant 0$ and a rejection penalty $e_j \geqslant 0$. Job $J_j$ is either rejected, in which case a rejection penalty $e_j$ has to be paid, or accepted and processed (scheduled) on the machine. Jobs are only revealed at their release times. The objective is to minimize the total completion time of the scheduled jobs plus the sum of the penalties of the rejected jobs. Up to our knowledge, no online algorithm is available for the general problem. Then we focus on the online scheduling under an agreeable condition on the processing times and rejection penalties of the jobs. Under the agreeable condition, for every two jobs $J_i$ and $J_j$, $p_i < p_j$ implies $e_i \geqslant e_j$, and furthermore, $p_i = p_j$ and $r_i < r_j$ imply $e_i \geqslant e_j$. This agreeable condition can be satisfied in many applications [14], since it simply means that, for shorter jobs or the earlier jobs with the same processing requirement the supplier has to pay larger rejection penalties. This coincides with the typical customer expectation that it should be easier to accept the production of a job with a shorter processing time or an earlier job with the same processing time. In our research, we further assume that each job $J_j$ can be accepted or rejected at a time instant greater than or equal to $r_j$.

For a given schedule $\pi$, $A = A(\pi)$ and $R = R(\pi)$ are used to denote the set of the scheduled (accepted) jobs and the set of the rejected jobs, respectively. Then the total completion time of the scheduled jobs under $\pi$ is given by $\sum_{J_j \in A(\pi)} C_j(\pi)$, and the sum of the penalties of the rejected jobs under $\pi$ is given by $\sum_{J_j \in R(\pi)} e_j$. For simplicity, we use $C + R$ to denote the objective function, where $C$ stands for $\sum_{J_j \in A} C_j$ and $R$ stands for $\sum_{J_j \in R} e_j$. Then the problem studied in this paper can be written in the three-field notation of Graham et al. [7] as

☆ This work was supported by NSFC (11271338, 11171313) and NSF Henan (132300410392).

* Corresponding author.
*E-mail address:* yuanjj@zzu.edu.cn (J. Yuan).

0020-0190/$ – see front matter 
http://dx.doi.org/10.1016/j.ipl.2013.05.006

$1|\text{online}, r_j, \text{rej, agreeable}|C+R$, where "rej" means the job rejection assumption and "agreeable" means the agreeable condition.

Scheduling with rejection was first introduced by Bartal et al. [1]. For problem $P|\text{rej}|C_{\max}+R$, in the online over-list version, they presented an algorithm with the best possible competitive ratio of $\frac{\sqrt{5}+3}{2} \approx 2.618$. In the off-line version, they provided a fully polynomial-time approximation scheme for problem $Pm|\text{rej}|C_{\max}+R$, and a polynomial-time approximation scheme for problem $P|\text{rej}|C_{\max}+R$. Following their research, scheduling problems with rejection have received more and more attention.

For the off-line version, Hoogeveen et al. [8] concentrated on the multiprocessor scheduling with rejection where preemption is allowed. They proved that this problem is APX-hard and designed a 1.58-approximation algorithm. Engels et al. [5] studied the single-machine scheduling with rejection to minimize the sum of the weighted completion times of the accepted jobs and the total penalty of the rejected jobs. Cao et al. [2] considered the scheduling problems with rejection or with discretely compressible times. For the problem $1|r_j|C_{\max}+R$, Zhang et al. [16] showed that this problem is binary NP-hard and provided a pseudo-polynomial-time algorithm and a fully polynomial-time approximation scheme. Lu et al. [11,9] dealt with the unbounded and bounded parallel-batch machine scheduling problems with release dates, respectively. Cheng and Sun [3] investigated the single-machine scheduling with deterioration and rejection, in which the processing time of a job is a linear function of its starting time. For more works on offline scheduling with rejection, see a recent survey on offline scheduling with rejection in Shabtay et al. [13].

For the online version, Epstein et al. [6] focused on the online over-list scheduling of unit-time jobs with rejection to minimize the total completion time of the accepted jobs plus the sum of the penalties of the rejected jobs. They presented an online algorithm with a competitive ratio of $\frac{2+\sqrt{3}}{2} \approx 1.866$, and proved that there does not exist an online algorithm with a competitive ratio of less than 1.63784. Seiden [12] considered the problem of online over-list scheduling with rejection in a multiprocessor setting for minimizing the makespan of the scheduled jobs and the sum of the penalties of the rejected jobs and presented an online algorithm with a competitive ratio of $\frac{\sqrt{10}+4}{3} \approx 2.387$ if preemption is allowed to all the accepted jobs. Dosa and He [4] studied the online over-list scheduling by taking the machine cost and rejection into account. In their model, no machine is initially provided and a certain machine cost has to be paid if a new machine is purchased. The objective is to minimize the sum of the makespan, the cost for purchasing machines, and the total penalty of all rejected jobs. For the small job case, they presented a best possible online algorithm with a competitive ratio of 2. A best possible online algorithm with a competitive ratio of 2 was proposed by Lu et al. [10] for single-machine scheduling with rejection to minimize the makespan of the scheduled jobs plus the sum of the penalties of the rejected jobs when the jobs arrive over time.

Online scheduling has been a hot research topic in the last two decades. Although there are several online models in the literature, "online" in this paper means that the jobs will arrive online over time. That is, each job becomes available at its release date, and its characteristics, i.e., processing time and rejection cost, become known at its release date. The quality of an online algorithm is calculated by its competitive ratio. For a minimization problem, the competitive ratio $\rho_{\mathcal{A}}$ of an online algorithm $\mathcal{A}$ is defined to be

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

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

When the penalty of each job is infinite, problem $1|\text{online}, r_j, \text{rej, agreeable}|C+R$ degenerates to $1|\text{online}, r_j|\sum C_j$. By Vestjens [15], any online algorithm for problem $1|\text{online}, r_j|\sum C_j$ has a competitive ratio of at least 2. Hence, any online algorithm for problem $1|\text{online}, r_j, \text{rej, agreeable}|C+R$ also has a competitive ratio of at least 2.

For problem $1|\text{online}, r_j|\sum C_j$, Vestjens [15] presented the following online algorithm DSPT (Delayed SPT) and showed that the algorithm has a best possible competitive ratio of 2.

A job $J_j$ is said to be available at time $t$, if $J_j$ has been released by time $t$ and has not been scheduled before time $t$. We use $p_{\min}(t)$ to denote the minimum processing time of the available jobs at time $t$. $J_{\min}(t)$ is used to denote an available job at time $t$ with processing time $p_{\min}(t)$.

**DSPT.** At the present time $t$, if the machine is idle, there are available jobs and $t \geqslant p_{\min}(t)$, then schedule $J_{\min}(t)$ starting at time $t$. If there is a choice, take the job with the smallest release date. Otherwise, do nothing but wait.

In this paper, by generalizing algorithm DSPT, we present an online algorithm, called Delayed SPT with Rejection and written in the short form as DSPTR. Then we show that DSPTR is a best possible online algorithm with a competitive ratio of 2 for problem $1|\text{online}, r_j, \text{rej, agreeable}|C+R$. As a consequence, the online algorithm is also best possible for problem $1|\text{online}, r_j, \text{rej}, e_j = e|C+R$.

## 2. Algorithm and analysis

In this section, we say a job is available at a time $t$ if the job has been released by time $t$ and has not been scheduled or rejected before time $t$. Let $U(t)$ be the set of all available jobs at time $t$. Denote $A(t) = \{J_j \in U(t)\colon t+p_j < e_j\}$ and $R(t) = \{J_j \in U(t)\colon t+p_j \geqslant e_j\}$. In our algorithm, the jobs in $R(t)$ will be rejected at time $t$. Note that, if a job $J_j$ is rejected by the algorithm, then $t = \max\{r_j,$