Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 155 (2007) 2546–2554
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Online scheduling with machine cost and rejection☆

J. Nagy-György[a], Cs. Imreh[b]

[a]Department of Mathematics, University of Szeged, Aradi vértanúk tere 1, H-6720 Szeged, Hungary
[b]Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary

Received 5 July 2005; received in revised form 13 June 2007; accepted 3 July 2007
Available online 27 August 2007

Abstract

In this paper we define and investigate a new scheduling model. In this new model the number of machines is not fixed; the algorithm has to purchase the used machines, moreover the jobs can be rejected. We show that the simple combinations of the algorithms used in the area of scheduling with rejections and the area of scheduling with machine cost are not constant competitive. We present a 2.618-competitive algorithm called OPTCOPY.

Keywords: Online algorithms; Scheduling; Competitive analysis

1. Introduction

In machine scheduling usually there is a fixed set of machines and a given set of jobs must be scheduled on the machines. The scheduling algorithm is not allowed to change the number of machines and it is not allowed to reject jobs. In the last few years some generalized models were investigated where it is allowed to change the set of machines, and also some models where the jobs can be rejected.

The problem of scheduling with machine cost is defined in [9]. In this model the number of machines is not a given parameter of the problem: the algorithm has to purchase the machines, and the goal is to minimize the cost spent for purchasing the machines plus the makespan. In [9] the problem where each machine has cost 1 is investigated. It can be supposed without loss of generality that the machines have cost 1, any constant cost can be reduced to this problem by scaling the processing times. Two online models are defined. In the list model the jobs arrive one by one and the decision maker has to decide in each step whether to buy new machines and then schedule the job on one of the already purchased machines without any information about the further jobs. In this model a $(1+\sqrt{5})/2$-competitive ($\approx$1.618) algorithm is presented for the solution of the problem and it is shown that no online algorithm can have smaller competitive ratio than $\frac{4}{3}$. The problem is also investigated in the Time model, where the jobs have release times and they are not allowed to start before their release time. In the online version we do not even know the existence of a job before its release time. In this model a $(1+\sqrt{1+6/\sqrt{2}})/2$-competitive ($\approx$1.645) algorithm is presented and it is shown that no online algorithm can have smaller competitive ratio than $(\sqrt{33}+9)/12\approx1.229$. Later in [2] the problem

E-mail addresses: Nagy-Gyorgy@math.u-szeged.hu (J. Nagy-György), cimreh@inf.u-szeged.hu (Cs. Imreh).
☆ This research has been supported by the Hungarian National Foundation for Scientific Research, Grant F048587.

0166-218X/$ - see front matter
doi:10.1016/j.dam.2007.07.004

in the list model is further investigated and a 1.5798-competitive algorithm is presented. A semi-online version of the list model is investigated in [7], and a lower bound for the possible competitive ratios of randomized algorithms is given in [11]. The scheduling problem with machine cost where the preemption of the jobs is allowed is investigated in [10]. A more general version where the cost of purchasing the machines is described by machine cost functions is investigated in [8].

The problem of scheduling with rejection is defined in [1]. In this model, it is possible to reject the jobs. The jobs are characterized by a processing time and a penalty. The goal is to minimize the makespan of the schedule for the accepted jobs plus the sum of the penalties of the rejected jobs. In the offline case an FPTAS is presented for fixed number of machines, and a PTAS in the case where the number of machines is part of the input. In the online case a 2.618-competitive algorithm is given for arbitrary number of machines, and a 1.618-competitive algorithm in the case of two machines. Matching lower bounds are also presented. The preemptive version of online scheduling with rejection is studied in [12]. In [12] a generalized version of the reject total penalty algorithm (see [1]) is analyzed, and it is proven that this generalized algorithm is 2.387-competitive for arbitrary number of machines. A general lower bound of 2.124, and a lower bound of 2.33 for the class of obliviously scheduling algorithms (the accepted jobs are scheduled without knowledge of the rejection penalties) are also proven. In [4] the offline scheduling problem with rejection is investigated in some more complex machine models, in [5] an FPTAS is given for scheduling with rejection on related parallel machines.

In this paper we consider a more general model where the machines are not given to the algorithm in advance but the algorithm must purchase them, and the jobs can be rejected. The goal is to minimize the makespan plus the cost of purchasing the machines plus the sum of the penalties of the rejected jobs. We call the total cost of purchasing the machines, machine purchasing cost.

It is easy to see that the offline problem is NP-complete. It is a generalization of an NP-complete problem (it is reduced to the problem of scheduling with machine cost if each penalty is ∞). On the other hand the offline version is not very interesting, we can check each value of m from 1 to n, and any offline α-approximation algorithm for the scheduling problem with rejection on m-machines yields an offline α-approximation algorithm for the more general version. A semi-online version of the problem, where the size of the jobs is bounded by 1 is investigated in [3]. In [3] a 2-competitive algorithm is given for this special case. Furthermore the authors observe that the problem is a generalization of the well-known ski rental problem, therefore it follows that no algorithm with smaller competitive ratio than 2 exists for its solution.

We consider the online problem. The jobs arrive one by one, and after the arrival of a job the decision maker can decide to purchase new machines and then it has to reject the job or schedule it on one of the already purchased machines. The problem is online thus the decision maker has to make his decisions without any information on the following jobs. For the problem we measure the performance of the algorithms by the competitive ratio. An online algorithm is called *c-competitive* if for each input the cost of the schedule produced by the algorithm is at most c times larger than the cost of the optimal schedule. The smallest c for which the algorithm is c-competitive is the competitive ratio of the algorithm.

The paper is organized as follows. In the next section we introduce the basic notations and recall some results and algorithms from the areas of scheduling with machine cost and scheduling with rejection which will be used later. In Section 3 we present the developed online algorithms for the problem. First we consider some algorithms which are the combinations of the algorithms used in the simpler models and we show that these algorithms are not constant competitive. We present an improved algorithm which we call OPTCOPY, and we prove that it is $(3+\sqrt{5})/2$-competitive ($\approx$2.618). We close the paper by summarizing the results and listing some related open questions.

2. Preliminaries

In the problem each job j has a processing time p_j and a penalty which is the cost of rejecting it, denoted by w_j. For a set $H \subseteq J$ we make use of the notations $P_H = \sum_{j \in H} p_j$ and $W_H = \sum_{j \in H} w_j$. As a shorthand we denote $P_{\{1,\dots,\ell\}}$ with simply writing P_ℓ. Moreover, for every m we denote the set of jobs with penalty $w_j \leqslant p_j/m$ by B_m.

For an arbitrary list J of jobs and an algorithm A, we denote by $A(J)$ the cost of the schedule produced by algorithm A on list J, the cost of the optimal schedule is denoted by $OPT(J)$. Therefore we say that an algorithm is c-competitive if $A(J) \leqslant c \cdot OPT(J)$ is valid for every J.

As subroutines we will use some known algorithms, we collect them and the related results below.

During the solution of the problem we have to schedule the accepted jobs on the already purchased machines. In the scheduling part our goal is to minimize the makespan. Since the jobs have no release time we obtain that scheduling the jobs without idle time on each machine, the maximal completion time is the total processing time of the jobs assigned to the machine. Therefore the algorithms do not have to schedule the jobs, only to assign them to the machines. Several algorithms are developed for the online scheduling problem on n identical machines (see the survey [13]), we will use the classical, greedy online scheduling algorithm LIST [6]. This algorithm always schedules greedily the arriving job on a least loaded machine.

Since in the problem the number of machines is not fixed, we need to give strategies for the problem of purchasing machines. We suppose that each machine has cost 1. In [9] the following class of purchasing strategies is defined. For an increasing sequence $\varrho = (0 = \varrho_1, \varrho_2, \ldots, \varrho_i, \ldots)$ we can define the following rule. When job j_ℓ is revealed $\mathscr{A}_\varrho$ purchases machines (if necessary) so that the current number of machines i satisfies $\varrho_i \leqslant P_\ell < \varrho_{i+1}$. An algorithm $\mathscr{A}_\varrho$ uses the above purchasing strategy and List scheduling for the schedule which means that it assigns job j_ℓ to the least loaded machine.

We also have to define some rules for the rejection of jobs. In [1] the following rule called RTP(α) (reject total penalty) is presented. If a job j is contained in B_m we reject it. Otherwise we denote by W_{j-1} the total penalty paid for the rejection of jobs rejected earlier which are not contained in B_m. If $W_{j-1} + w_j \leqslant \alpha p_j$ we reject the job, otherwise we accept it.

3. Algorithms

In this section we develop and analyze some algorithms for the solution of the problem. Since we have rules for purchasing the machines and for the rejection and scheduling of the jobs it is a straightforward idea to mix these rules and build algorithms for the complex problem. In the first part we show that the simple combinations of these rules are not constant competitive.

3.1. Mixed algorithms

In all of the following algorithms, α is a given constant, $\rho = (0, \rho_2, \ldots, \rho_i, \ldots)$ is an increasing sequence, $B_i = \{j | w_j \leqslant p_j/i\}$, when $i \neq 0$ and $B_0 = B_1$. In the jth step A_j denotes the set of accepted jobs and R_j the set of the rejected ones. In all cases we start with 0 machines.

1st combined algorithm (CA1). jth *step*:

(i) When job j appears, we purchase machines (if necessary) so that the current number of machines i satisfies $\rho_i \leqslant P_{A_{j-1}\cup\{j\}} < \rho_{i+1}$.
(ii) If $j \in B_i$, we reject job j.
(iii) If $j \notin B_i$, and $W_{R_{j-1}\setminus B_i} + w_j \leqslant \alpha p_j$, we also reject it.
(iv) Otherwise, we schedule it on a least loaded machine, according to the List algorithm.

Proposition 1. *There is no such C that algorithm CA1 is C-competitive.*

Proof. Assume that CA1 is C-competitive for some $C > 0$. Let $n > C$, $J = \{1\}$, let $p_1 = \rho_{n+1}$ and $w_1 = 1$. For this job, the optimal schedule rejects it and $OPT(J) = 1$ holds. Algorithm CA1 also rejects it, but it purchases $n + 1$ machines; so its cost $\mathrm{CA1}(J) = n + 2 > n > C \cdot OPT(J)$, because of the constraint $n > C$. From this contradiction it follows that CA1 is not C-competitive. □

We also investigate the following similar algorithm which can handle the counterexample given above.

2nd combined algorithm (CA2). jth *step*:

(i) When job j appears, we compute the number i such that $\rho_i \leqslant P_{A_{j-1}\cup\{j\}} < \rho_{i+1}$ holds.
(ii) If $j \in B_i$, we reject job j.

(iii) If $j \notin B_i$, and $W_{R_{j-1} \setminus B_i} + w_j \leqslant \alpha p_j$, we also reject it.
(iv) Otherwise if necessary, we purchase machines so that the current number of them reaches i; after that, we schedule it on a least loaded machine, according to the List algorithm.

Proposition 2. *There is no such C that algorithm CA2 is C-competitive.*

Proof. Assume that CA2 is C-competitive for some $C > 0$. Let n and k be two integers such that $n > 2C$ and $\rho_2/2 \leqslant n/k < \rho_2$. Furthermore, let $|J| = kn$, and for all $j \in J$ let $p_j = w_j = n/k$. If we purchase n machines and schedule k jobs on each of them, the cost will be $n + k(n/k) = 2n$. From this we can conclude $OPT(J) \leqslant 2n$. Since algorithm CA2 rejects all the jobs, the cost $\mathrm{CA2}(J) = kn(n/k) = n^2$. Because of the constraint $n > 2C$, $n^2 > 2Cn$ holds, so CA2 is not C-competitive. □

3rd combined algorithm (CA3). *jth step:*

(i) Let i be the actual number of the machines. If $j \in B_i$, we reject job j.
(ii) If $j \notin B_i$, and $W_{R_{j-1} \setminus B_i} + w_j \leqslant \alpha p_j$, we also reject it.
(iii) Otherwise if necessary, we purchase machines so that the number of them i satisfies $\rho_i \leqslant P_{A_{j-1} \cup \{j\}} < \rho_{i+1}$. After that, we schedule it on a least loaded machine, according to the List algorithm.

Proposition 3. *There is no such C that algorithm CA3 is C-competitive.*

Proof of Proposition 2 can also be applied to this case.

3.2. Algorithm OPTCOPY

In this section we present a more sophisticated algorithm. The basic idea is that instead of the original problem we consider a relaxed version, where we replace part of the cost of the schedule (purchasing cost of machines plus the makespan) with a lower bound of it.

Suppose that we accepted a set A of jobs, m machines were purchased, and the current makespan is M. Then $Mm \geqslant P_A$, thus $m \geqslant P_A/M$. So we obtain that for the cost of the schedule $M + m \geqslant M + P_A/M$ is valid. Let l_A denote the greatest processing time that belongs to a job in A. We define the following expression:

$$M_A := \begin{cases} \max\{\sqrt{P_A}, l_A\} & \text{if } P_A > 1, \\ 1 & \text{otherwise.} \end{cases}$$

Concerning the value of M_A the following statement follows immediately by the definition.

Lemma 4. *For two arbitrary sets A_1 and A_2 of jobs if $A_1 \subseteq A_2$ then $M_{A_1} \leqslant M_{A_2}$.*

Now for an arbitrary set A of jobs let

$$T_A := \begin{cases} M_A + \dfrac{P_A}{M_A} & \text{if } A \neq \emptyset, \\ 0 & \text{if } A = \emptyset. \end{cases}$$

The geometrical meaning of T_A is the following: if we consider the jobs as rectangles with sides 1 and p_i, then $2T_A$ is the smallest possible perimeter of the rectangles which can be used to pack the rectangles assigned to the jobs. By this interpretation we can prove easily the following statements.

Lemma 5. *For two arbitrary sets A_1 and A_2 of jobs, if $A_1 \subseteq A_2$ then $T_{A_1} \leqslant T_{A_2}$.*

Lemma 6. *Let A be an arbitrary nonempty set, furthermore let $x \geqslant \max\{1, l_A\}$ be an arbitrary positive number, then $x + P_A/x \geqslant T_A$.*

Using Lemma 6 we immediately obtain the following statement for the case where rejection is not allowed which is also proved in [9].

Lemma 7 (*Imreh and Noga [9]*). *The cost of an optimal schedule with machine cost of the jobs from set A when no rejection is allowed is at least T_A.*

In [9] Theorem 2 proves that algorithm $\mathscr{A}_\varrho$ with the sequence $\varrho = (0, 4, \ldots, i^2, \ldots)$ is φ-competitive with $\varphi = (1 + \sqrt{5})/2$ in the model where the rejection of the jobs is not allowed. In the proof the authors show that for an arbitrary set A of jobs $\mathscr{A}_\varrho(A)/OPT(A) \leqslant \varphi$. This is shown by case disjunction, and in each case the inequality $\mathscr{A}_\varrho(A)/T_A \leqslant \varphi$ is proven and by Lemma 7 this shows the required statement. Therefore the same proof proves the following statement:

Lemma 8 (*Imreh and Noga [9]*). *For algorithm $\mathscr{A}_\varrho$ with the sequence $\varrho = (0, 4, \ldots, i^2, \ldots)$ and an arbitrary input set A of jobs when no rejection is allowed*

$$\mathscr{A}_\varrho(A) \leqslant \varphi T_A,$$

where $\varphi = (1 + \sqrt{5})/2$.

Now we can define the relaxed problem. Jobs arrive, each job has a processing time and a penalty. We have to find a solution where the total penalty paid for the rejected jobs plus the value T_A for the set A of accepted jobs is minimal. We call this problem relaxed. For a set J of jobs the cost of the optimal solution of the relaxed problem is denoted by $ROPT(J)$. From Lemma 7 the following statement follows.

Corollary 9. *For an arbitrary set J of jobs $ROPT(J) \leqslant OPT(J)$.*

Proof. Consider an optimal solution of the original problem on input J. Let A be the set of the accepted jobs, R be the set of the rejected jobs. Then by Lemma 7 we obtain that $OPT(J) \geqslant \sum_{j \in R} w_j + T_A$. On the other hand using the sets R and A in the case of the relaxed problem the value of the objective function is $\sum_{j \in R} w_j + T_A$. Therefore we obtain a feasible solution of the relaxed problem with not larger objective function value than $OPT(J)$, thus the statement of the corollary follows. $\square$

To develop algorithm OPTCOPY we have to examine the structure of the optimal solutions of the relaxed problem. For an arbitrary list of jobs J denote J_k the set of the first k jobs of J. Then the following statement is valid.

Lemma 10. *Suppose that A^*_{k-1} is the set which belongs to an optimal solution of the relaxed problem on set J_{k-1}. Then the relaxed problem on set J_k has an optimal solution such that A^*_{k-1} is a subset of the set of the accepted jobs.*

Proof. Assume that there is no such optimal solution. Let A_k be the set of the accepted jobs and R_k the set of rejected jobs in an optimal solution of the relaxed problem on set J_k. As we assumed, $A^*_{k-1} \nsubseteq A_k$. Therefore $A^*_{k-1} \neq \emptyset$. We have to deal with the following two cases: (1) when $k \in R_k$ and (2) when $k \in A_k$.

Case 1: $k \in R_k$.

If we use A_k as the accepted jobs we receive a feasible solution of the relaxed problem on set J_{k-1}, therefore we obtain that

$$ROPT(J_{k-1}) \leqslant W_{R_k \setminus \{k\}} + T_{A_k}.$$

If we substitute the definition of $ROPT(J_{k-1})$ and we increase both side by w_k then we receive that

$$W_{R^*_{k-1}} + w_k + T_{A^*_{k-1}} \leqslant W_{R_k} + T_{A_k}.$$

On the other hand the right side is $ROPT(J_k)$ thus we obtained that

$$W_{R^*_{k-1} \cup \{k\}} + T_{A^*_{k-1}} \leqslant ROPT(J_k).$$

Let $A^*_k := A^*_{k-1}$, that is an optimal solution and naturally satisfies property $A^*_{k-1} \subseteq A^*_k$. This is a contradiction.

Case 2: $k \in A_k$.

Case 2 has two subcases: (a) $M_{A^*_{k-1}} > M_{A_k}$ and (b) $M_{A^*_{k-1}} \leqslant M_{A_k}$.

(a) $M_{A^*_{k-1}} > M_{A_k}$.

We obtain by Lemma 4 that $M_{A^*_{k-1}\cup\{k\}} \geqslant M_{A^*_{k-1}}$. Then using Lemma 6 with the values $x = M_{A^*_{k-1}}$ and $A = A^*_{k-1} \cup \{k\}$ (the conditions of the lemma are satisfied since $M_{A^*_{k-1}} > M_{A_k} \geqslant p_k$) we obtain that

$$T_{A^*_{k-1}\cup\{k\}} \leqslant M_{A^*_{k-1}} + \frac{P_{A^*_{k-1}\cup\{k\}}}{M_{A^*_{k-1}}} = T_{A^*_{k-1}} + \frac{p_k}{M_{A^*_{k-1}}}.$$

On the other hand if we use the sets R_k and $A_k \backslash \{k\}$ we have a feasible solution of the relaxed problem on set J_{k-1}, thus

$$W_{R^*_{k-1}} + T_{A^*_{k-1}} + \frac{p_k}{M_{A^*_{k-1}}} = ROPT(J_{k-1}) + \frac{p_k}{M_{A^*_{k-1}}} \leqslant W_{R_k} + T_{A_k\backslash\{k\}} + \frac{p_k}{M_{A^*_{k-1}}}.$$

Furthermore $p_k/M_{A^*_{k-1}} < p_k/M_{A_k}$ is valid and by Lemma 6 (with values $x = M_{A_k}$ and $A = A_k \backslash \{k\}$):

$$T_{A_k\backslash\{k\}} \leqslant M_{A_k} + \frac{P_{A_k\backslash\{k\}}}{M_{A_k}}$$

follows. Therefore we obtain that

$$W_{R_k} + T_{A_k\backslash\{k\}} + \frac{p_k}{M_{A^*_{k-1}}} < W_{R_k} + M_{A_k} + \frac{P_{A_k\backslash\{k\}}}{M_{A_k}} + \frac{p_k}{M_{A_k}} = ROPT(J_k).$$

Using the chain of inequalities proven above we obtain that

$$W_{R^*_{k-1}} + T_{A^*_{k-1}\cup\{k\}} < ROPT(J_k),$$

which is a contradiction, thus this case is not possible.

(b) $M_{A^*_{k-1}} \leqslant M_{A_k}$.

If we use the sets $R^*_{k-1} \cup (R_k \cap A^*_{k-1})$ and $A_k \cap A^*_{k-1}$ we have a feasible solution of the relaxed problem on set J_{k-1}, thus

$$ROPT(J_{k-1}) \leqslant W_{R^*_{k-1}} + W_{R_k\cap A^*_{k-1}} + T_{A_k\cap A^*_{k-1}}.$$

Then we apply Lemma 6 with the values $x = M_{A^*_{k-1}}$ and $A = A_k \cap A^*_{k-1}$ (the conditions hold since $M_{A^*_{k-1}} \geqslant M_{A_k\cap A^*_{k-1}}$ by Lemma 4), and we obtain that

$$T_{A_k\cap A^*_{k-1}} \leqslant M_{A^*_{k-1}} + \frac{P_{A_k\cap A^*_{k-1}}}{M_{A^*_{k-1}}},$$

therefore

$$ROPT(J_{k-1}) \leqslant W_{R^*_{k-1}} + W_{R_k\cap A^*_{k-1}} + M_{A^*_{k-1}} + \frac{P_{A_k\cap A^*_{k-1}}}{M_{A^*_{k-1}}}. \tag{1}$$

If we use that $ROPT(J_{k-1}) = W_{R^*_{k-1}} + M_{A^*_{k-1}} + P_{A^*_{k-1}}/M_{A^*_{k-1}}$ and $P_{A^*_{k-1}} = P_{A_k\cap A^*_{k-1}} + P_{R_k\cap A^*_{k-1}}$ then by inequality (1) and by the constraint of the subcase it follows that

$$\frac{P_{R_k\cap A^*_{k-1}}}{M_{A_k}} \leqslant \frac{P_{R_k\cap A^*_{k-1}}}{M_{A^*_{k-1}}} \leqslant W_{R_k\cap A^*_{k-1}}. \tag{2}$$

If we use Lemma 6 with the values $x = M_{A_k}$ and $A = A_k \cup A^*_{k-1}$ (the conditions of the lemma hold since $M_{A_k} \geqslant l_{A_k}$, $M_{A_k} \geqslant M_{A^*_{k-1}} \geqslant l_{A^*_{k-1}}$) then we obtain

$$W_{R_k\cap R^*_{k-1}} + T_{A_k\cup A^*_{k-1}} \leqslant W_{R_k\cap R^*_{k-1}} + M_{A_k} + \frac{P_{A_k\cup A^*_{k-1}}}{M_{A_k}}. \tag{3}$$

From inequality (2) we get

$$
\begin{aligned}
&W_{R_k \cap R^*_{k-1}} + M_{A_k} + \frac{P_{A_k \cup A^*_{k-1}}}{M_{A_k}} \\
&\quad = W_{R_k \cap R^*_{k-1}} + M_{A_k} + \frac{P_{A_k \cap A^*_{k-1}} + P_{A_k \cap R^*_{k-1}} + P_{R_k \cap A^*_{k-1}} + p_k}{M_{A_k}} \\
&\quad \leqslant W_{R_k \cap R^*_{k-1}} + W_{R_k \cap A^*_{k-1}} + T_{A_k} = ROPT(J_k).
\end{aligned} \tag{4}
$$

Let $A^*_k := A_k \cup A^*_{k-1}$. Using inequalities (3) and (4) it follows that A^*_k provides an optimal solution and $A^*_{k-1} \subseteq A^*_k$, what is again a contradiction. □

The relaxed problem can be solved in polynomial time. The algorithm which solves the problem is based on the following structural property.

Lemma 11. *For each job j we consider the problem $REL(j)$ which is the restricted relaxed problem where it is given that j is the largest accepted job. Order the set of jobs which are not larger than j by the value p_i/w_i into increasing sequence. Then $REL(j)$ has an optimal solution which is a prefix of this sequence.*

Proof. Consider the problem $REL(j)$ for a job j and let A and R be the sets of the accepted and rejected jobs in an optimal solution. Let $i \neq j$ be the accepted job, where the value p_i/w_i is maximal. Since A and R are the optimal sets we obtain that

$$W_R + T_A \leqslant W_{R \cup \{i\}} + T_{A \setminus \{i\}}.$$

On the other hand $M_A \geqslant M_{A \setminus \{i\}}$ thus by Lemma 6 we obtain that $M_A + (P_A - p_i)/M_A \geqslant T_{A \setminus \{i\}}$. Therefore,

$$W_R + T_A \leqslant W_R + w_i + M_A + \frac{P_A - p_i}{M_A} = W_R + T_A + w_i - \frac{p_i}{M_A}.$$

Thus we obtained that $p_i/w_i \leqslant M_A$.

Now suppose that the solution does not satisfy the property stated in the lemma. Then there exists a job $k \neq j$ with the properties $p_k \leqslant p_j$ and $p_k/w_k \leqslant p_i/w_i$ which is rejected. Consider the feasible solution which also accepts k. Then the value of the objective function is $W_{R \setminus \{k\}} + T_{A \cup \{k\}}$ and by Lemma 6 we obtain that

$$W_{R \setminus \{k\}} + T_{A \cup \{k\}} \leqslant W_R - w_k + M_A + \frac{P_A + p_k}{M_A}.$$

On the other hand $p_k/w_k \leqslant p_i/w_i \leqslant M_A$ thus $p_k/M_A \leqslant w_k$ which yields that $W_{R \setminus \{k\}} + T_{A \cup \{k\}} \leqslant W_R + T_A$. Therefore accepting job k the value of the objective function does not increase and this proves the statement of the lemma. □

By Lemma 11 we can find a polynomial time algorithm which solves the relaxed problem. (We consider the restricted problem $REL(j)$ for each j and we investigate the possible prefixes of the ordered sequences and choose the best solution.) Furthermore by Lemma 10 it follows that we can find in each step such an optimal solution of the relaxed problems for set J_k where the size of the maximal accepted job is increasing. Using such maximal jobs and the prefixes of the ordered sequences in each step we have a polynomial time algorithm which gives such optimal solutions which satisfy Lemma 10. We call this algorithm RELOPT. Denote the sets of the accepted jobs from J_k by A^*_k and the set of rejected jobs by R^*_k. Therefore $A^*_i \subseteq A^*_k$ if $i \leqslant k$. Then the following statement holds.

Lemma 12. *For the above defined sets, the following inequality is valid:*

$$\sum_{j=1}^{n} W_{R^*_{j-1} \cap A^*_j} \leqslant T_{A^*_n}.$$

Proof. We have $R_j^*\backslash\{j\} \subseteq R_{j-1}^*$ by $A_{j-1}^* \subseteq A_j^*$. Therefore

$$ROPT(J_{j-1}) = W_{R_j^*\backslash\{j\}} + W_{R_{j-1}^*\cap A_j^*} + T_{A_{j-1}^*}.$$

On the other hand using the sets $R_j^*\backslash\{j\}$ and $A_j^*\backslash\{j\}$ we get a feasible solution of the relaxed problem on set J_{j-1} thus

$$ROPT(J_{j-1}) \leqslant W_{R_j^*\backslash\{j\}} + T_{A_j^*\backslash\{j\}},$$

so substituting the definition of $ROPT(J_{j-1})$ we obtain that

$$W_{R_{j-1}^*\cap A_j^*} \leqslant T_{A_j^*\backslash\{j\}} - T_{A_{j-1}^*}.$$

Therefore,

$$\sum_{j=1}^{n} W_{R_{j-1}^*\cap A_j^*} \leqslant \sum_{j=1}^{n}(T_{A_j^*\backslash\{j\}} - T_{A_{j-1}^*}).$$

On the other hand by Lemma 5 we obtain $T_{A_j^*\backslash\{j\}} \leqslant T_{A_j^*}$, thus

$$\sum_{j=1}^{n} W_{R_{j-1}^*\cap A_j^*} \leqslant \sum_{j=1}^{n}(T_{A_j^*} - T_{A_{j-1}^*}) = T_{A_n^*},$$

and this is what we have to prove. □

Now we are ready to define the class of algorithms OPTCOPY$_\varrho$. OPTCOPY$_\varrho$ rejects all of the jobs rejected by RELOPT, therefore it does not accept more jobs than the optimal solution of the relaxed problem. On the other hand it may reject more jobs than an optimal solution, but we can prove some bounds on the amount of the rejected jobs.

Algorithm OPTCOPY$_\varrho$. At the arrival of a new job j perform the following steps.

(i) If j is rejected by RELOPT, reject it, otherwise go to step (ii).
(ii) Schedule the job by algorithm $\mathscr{A}_\varrho$, where in the machine purchasing rule only the accepted jobs are taken into account.

We have the following result.

Theorem 13. OPTCOPY$_\varrho$ *with the sequence* $\varrho = (0, 4, \ldots, i^2, \ldots)$ *is* $(3+\sqrt{5})/2$*-competitive.*

Proof. Denote A_n the set of jobs scheduled by OPTCOPY and A_n^* the set of jobs accepted by RELOPT. Since $A_n \subseteq A_n^*$ and because of Lemma 8

$$\text{OPTCOPY}_\varrho(J) = W_{R_n} + \mathscr{A}_\varrho(A_n) \leqslant W_{R_n} + \varphi T_{A_n^*}, \tag{5}$$

furthermore, by the definition of the algorithms OPTCOPY and RELOPT we obtain that

$$R_n = \bigcup_{j=1}^{n} R_j^* = \bigcup_{j=1}^{n-1}(R_j^*\backslash R_{j+1}^*) \cup R_n^* = \bigcup_{j=1}^{n-1}(R_j^* \cap A_{j+1}^*) \cup R_n^*,$$

so applying Lemma 12

$$W_{R_n} = W_{R_n^*} + \sum_{j=1}^{n-1} W_{R_j^*\cap A_{j+1}^*} \leqslant W_{R_n^*} + T_{A_n^*}. \tag{6}$$

Finally applying inequalities (5) and (6), we get

$$\text{OPTCOPY}_Q(J) \leqslant W_{R_n^*} + (1+\varphi)T_{A_n^*} \leqslant (1+\varphi)OPT(J),$$

and this is exactly what we have to prove. □

We note that we could not determine the competitive ratio of the algorithm, we just proved an upper bound on it. On the other hand it is easy to see that the competitive ratio of the algorithm is at least $(2+2\varphi)/(\varphi+1/\varphi)\approx 2,34$. Consider the following sequence of jobs: the first job is $(\varphi N, \varphi N)$, and then N^3 jobs of size $(1/N, \infty)$ followed by one job of size $(\varphi N, \infty)$ follows. (The second part of the example is the same which was used in [9].) Then OPTCOPY will reject the first job and accept the others, it will schedule the first N^3 by purchasing N machines and putting N^2 jobs on each machine. The final job will be placed on an arbitrary machine. Therefore, OPTCOPY's cost will be $N + N + 2\varphi N$. The optimal cost is no more than $\varphi N + \lceil (N+2\varphi)/\varphi \rceil$. So, the competitive ratio of OPTCOPY is at least

$$\frac{(2+2\varphi)N}{\varphi N + \lceil (N+2\varphi)/\varphi \rceil} \overset{N\to\infty}{\longrightarrow} \frac{2+2\varphi}{\varphi + 1/\varphi}.$$

4. Conclusions and further questions

In this paper we introduced a new scheduling model called scheduling with machine cost and rejection which is a common generalization of the well-known models of scheduling with machine cost and scheduling with rejection. We have shown that the straightforward combinations of the known algorithms are not constant competitive, and we presented algorithm OPTCOPY which is $(3+\sqrt{5})/2$-competitive.

Concerning our model many interesting questions arise. In the case of scheduling with machine cost some results are proven for semi-online models, this can also be an interesting question in this case. In the simpler models the versions where preemption is allowed are also investigated, we think so that this could be an interesting question in this more general case.

Acknowledgements

The authors wish to thank the anonymous referees for their helpful and valuable advice and suggestions that helped to improve significantly the first version of this paper.

References

[1] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, L. Stougie, Multiprocessor scheduling with rejection, SIAM J. Discrete Math. 13 (2000) 64–78.
[2] Gy. Dósa, Y. He, Better online algorithms for scheduling with machine cost, SIAM J. Comput. 33 (2004) 1035–1051.
[3] Gy. Dósa, Y. He, Scheduling with machine cost and rejection, J. Combin. Optim. 12 (4) (2006) 337–350.
[4] D.W. Engels, D.R. Karger, S.G. Kolliopoulos, S. Sengupta, R.N. Uma, J. Wein, Techniques for scheduling with rejection, J. Algorithms 49 (2003) 175–191.
[5] L. Epstein, J. Sgall, Approximation schemes for scheduling on uniformly related and identical parallel machines, Algorithmica 39 (2004) 43–57.
[6] R.L. Graham, Bounds for certain multiprocessor anomalies, Bell System Tech. J. 45 (1966) 1563–1581.
[7] Y. He, S.Y. Cai, Semi-online scheduling with machine cost, J. Comput. Sci. Tech. 17 (2002) 781–787.
[8] Cs. Imreh, On-line scheduling with general machine cost functions, Electron. Notes in Discrete Math. 27 (2006) 49–50.
[9] Cs. Imreh, J. Noga, Scheduling with Machine Cost, in: Proceedings of APPROX'99, Lecture Notes in Computer Science, vol. 1761, Springer, Berlin, 1999, pp. 168–176.
[10] Y.W. Jiang, Z. He, Preemptive online algorithms for scheduling with machine cost, Acta Inform. 41 (2005) 315–340.
[11] S.S. Seiden, A guessing game and randomized online algorithms, in: Proceedings of STOC'00, SIGACT, ACM, Portland USA, Avon books, NY, 2000, pp. 592–601.
[12] S.S. Seiden, Preemptive multiprocessor scheduling with rejection, Theoret. Comput. Sci. 262 (2001) 437–458.
[13] J. Sgall, On-line scheduling, in: A. Fiat, G.J. Woeginger (Eds.), Online algorithms: the state of the art, Lecture Notes in Computer Science, vol. 1442, Springer, Berlin, 1998, pp. 196–231.