Research Article

A Note on Two-Agent Scheduling with Resource Dependent Release Times on a Single Machine

Peng Liu¹ and Lini Duan²

¹School of Management, Shenyang University of Technology, Shenyang 110870, China
²School of Economics & Management, Shenyang University of Chemical Technology, Shenyang 110142, China

Correspondence should be addressed to Peng Liu; liup7802@163.com

Received 6 October 2014; Accepted 30 November 2014

Academic Editor: Shuenn-Ren Cheng

Copyright © 2015 P. Liu and L. Duan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider a scheduling problem in which both resource dependent release times and two agents exist simultaneously. Two agents share a common single machine, and each agent wants to minimize a cost function dependent on its own jobs. The release time of each A-agent’s job is related to the amount of resource consumed. The objective is to find a schedule for the problem of minimizing A-agent’s total amount of resource consumption with a constraint on B-agent’s makespan. The optimal properties and the optimal polynomial time algorithm are proposed to solve the scheduling problem.

1. Introduction


time. Lu et al. [17] consider a single-machine earliness-
tardiness scheduling problem with due-date assignment, in
which the processing time of a job is a function of its position
in a sequence and its resource allocation.

However, to the best of our knowledge, no work has been
done on models with both aspects of resource dependent
release times and multiagent in the literature. These two
categories of scheduling problems have been extensively
and separately researched over the last two decades. In
this paper, we study the two-agent scheduling problems on
a single machine with resource dependent release times,
where the goal is to find a schedule that minimizes the
objective function of one agent with the restriction that
the objective function of the other agent cannot exceed a
given bound. The problems under consideration fall into the
category of scheduling problems with resource consumption
and multiple agents. Such a scheduling problem commonly
arises in the steel industry. Janiak [18] describes a practical
scheduling problem with resource dependent release times
in steel mills, where batches of ingots have to be preheated
before they can be hot rolled in a blooming mill, and the ingot
preheating time is inversely proportional to the total amount
of resources consumed.

The remainder of this paper is organized as follows. In
Section 2, we describe the proposed problem. In Section 3, we
develop the optimal polynomial time algorithm for the two-
agent single-machine scheduling problem. Section 4 gives
some concluding remarks.

2. Problem Description

We now describe our problem formally. There are two fami-
lies of independent and nonpreemptive jobs $J^A = \{J^A_1, J^A_2, \ldots, J^A_{n_A}\}$ and $J^B = \{J^B_1, J^B_2, \ldots, J^B_{n_B}\}$ to be processed on a common
single machine. The jobs in $J^A$ and $J^B$ are called A-agent’s
jobs and B-agent’s jobs, respectively. Associated with each job
$J^A_h$, let $p^A_h$ denote the processing time and $h = 1, 2, \ldots, n_A$. The release time $r^A_h$ is related to the amount of the resource
$f^A_h$ consumed on job $J^A_h$. A strictly decreasing continuous function is given $f^A : R^+ \rightarrow R^+$. We refer to $f^A$ as the
resource consumption function. We assume that $f^A_h = f(r^A_h)$. Each A-agent’s job can start at any time after the release
time of the job, and idle time between jobs is allowed. Since
the consumption function $f^A$ is strictly decreasing continuously,
we may assume that each job starts as soon as it becomes
available. That is, we can take $s^A_h = r^A_h$, where $s^A_h$ is the starting
time of job $J^A_h$. Associated with each job $J^B_k$, let $p^B_k$ and $r^B_k$
denote the processing time and the release times, respectively,
and $k = 1, 2, \ldots, n_B$. Let $\pi$ indicate a feasible schedule of the
$n = n_A + n_B$ jobs. Let $C^B(\pi)$ denote the completion time of
B-agent’s job $J^B_k$ under schedule $\pi$. The objective function
of agent A is to minimize the total amount of resource
consumption $\sum_{h=1}^{n_A} f(r^A_h)$. The objective function of agent B is
to minimize the makespan $C^B_{\max}$ of agent B with the restriction that

the makespan $C^B_{\max}$ of agent B cannot exceed a given bound
$U$. If the value $U$ is too small, an instance of the scheduling
problem may not have feasible solutions. If there is at least one
feasible solution, we say that the instance is feasible. According
to the three-field notation $\psi_1 | \psi_2 | \psi_3$ of Graham et
al. [19], the scheduling problem is denoted as $1\| \sum_{h=1}^{n_A} f(r^A_h) : C^B_{\max} \leq U$.

3. Main Results

In this section, we develop an optimal polynomial time
algorithm to solve the problem $1\| \sum_{h=1}^{n_A} f(r^A_h) : C^B_{\max} \leq U$.

Given a sequence $\pi = \{J^A_1, J^B_1, \ldots, J^A_{n_A}, J^B_{n_B}\}$, for
each B-agent’s job, the completion time $C^B_k$ may be completed
recursively as $C^B_1 = r^B_1 + p^B_1, C^B_k = \max\{r^B_k, C^B_{k-1} + p^B_k\}, k = 2, \ldots, n_B$.

Thus the completion time of job $J^B_k$ may also be taken as

$C^B_k = \max\{r^B_k + \sum_{l=j}^{k} p^B_l\}$.

Moreover, let the maximum job completion time be

denoted by $C^B_{\max} = \max_{1 \leq k \leq n_B} C^B_k$; then $C^B_{\max} = \max_{1 \leq k \leq n_B} (r^B_k + \sum_{l=j}^{k} p^B_l)$.

Lemma 1. Given a sequence $\pi = \{J^A_1, J^B_1, \ldots, J^A_{n_A}, J^B_{n_B}\}$ and a constant $U$, define $C^A(\pi) = \max_{1 \leq k \leq n_B} (r^B_k + \sum_{l=j}^{k} p^B_l)$. Then, if $U < C^A$, the sequence $\pi$ corresponds to an infeasible schedule.

Now we can define bounds for the constraint $U$. Define

$U = \max_{1 \leq k \leq n_B} (r^B_k + \sum_{l=j}^{k} p^B_l)$. The analysis in the following
section will be confined to the case in which $U \leq U$.

Lemma 2. An optimal schedule exists in which the A-agent’s jobs are processed in the nonincreasing order of processing times
$p^A_h$.

Proof. The resource consumption function $f^A$ is a strictly
decreasing continuous function to A-agent’s jobs. Since
releasing A-agent’s jobs sooner consumes more resource, A-
agent’s jobs should be released as late as possible. Hence A-
agent’s jobs should be released in nonincreasing order of
processing times $p^A_h$. $\square$

Lemma 3. An optimal schedule exists in which the B-agent’s jobs are processed in the nondecreasing order of release times
$r^B_k$.

Proof. The makespan of agent B is the maximum completion
time of B-agent’s jobs on the single machine; that is, the
makespan of agent B is the completion time of the last B-
agent’s job. Using a pairwise job interchange argument, we
can process B-agent’s jobs in the nondecreasing order of
release times $r^B_k$. $\square$

Next, an algorithm to determine an optimal schedule of the
problem $1\| \sum_{h=1}^{n_A} f(r^A_h) : C^B_{\max} \leq U$ is developed as follows.
Algorithm 4.

Step 1. Arrange the A-agent’s jobs as \( \{J_1^A, J_2^A, \ldots, J_{n_A}^A\} \) according to the nonincreasing order of \( r_h^B \) and denote all B-agent’s jobs sequenced by the nondecreasing order of \( r_k^B \) as a dummy job \( B_1 \).

Step 2. Define sequence \( S = \{B_1, J_1^A, J_2^A, \ldots, J_{n_A-1}^A, J_{n_A}^A\} \) and calculate \( C_h^B \) for agent \( B \). The sequence \( S \) is an optimal schedule and the starting times of A-agent’s job are given by \( s_1^A = U, s_h^A = s_{h-1}^A + p_{h-1}^A = s_1^A + \sum_{i=1}^{h-1} p_i^A, h = 2, 3, \ldots, n_A \).

Theorem 5. Algorithm 4 generates an optimal schedule for the problem \( \sum_{h=1}^{n_A} f(r_h^A) : C_{\text{max}}^B \leq U \) in \( O(n_A \log n_A + n_B \log n_B) \) time.

Proof. The proof of optimality is straightforward from the results of Lemmas 1–3. We now turn to time complexity. The time to sequence the jobs of set \( J^A \) according to the nonincreasing order of \( r_h^B \) is \( O(n_A \log n_A) \). The time to sequence the jobs of set \( J^B \) according to the nondecreasing order of \( r_h^B \) is \( O(n_B \log n_B) \). Creating dummy job \( B_1 \) incurs \( O(n_B) \) operations. So the overall computational complexity of Algorithm 4 is bounded by \( O(n_A \log n_A + n_B \log n_B) \). This completes the proof.

4. Conclusions

In this paper, we combine two important issues in scheduling that recently have received increasing attention from researchers: resource dependent release times and multiple agents. Our goal is to find a schedule for the problem of minimizing A-agent’s total amount of resource consumption with a constraint on B-agent’s makespan. We propose the optimal properties and the optimal polynomial time algorithm for the considered scheduling problem.

The future research may be directed to analyze the problems with other objective functions such as minimizing the number of late jobs, the total weighted completion time and tardiness. An interesting research topic is also to analyze the scheduling problem with more than two agents or in other machine environments.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to the editor and reviewers for their helpful comments and suggestions. This research is supported by the National Natural Science Foundation of China (Grant no. 71001074) and the Program for Liaoning Excellent Talents in University (Grant no. WJQ2013003).

References


Submit your manuscripts at http://www.hindawi.com