Single Machine Scheduling with Generalized Total Tardiness Objective Function

Evgeny R. Gafarov a, Alexander A. Lazarev b

Institute of Control Sciences of the Russian Academy of Sciences,
Profsoyuznaya st. 65, 117997 Moscow, Russia,
email: aaxel73@mail.ru, bjobmath@mail.ru

Frank Werner

Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg,
PSF 4120, 39016 Magdeburg, Germany,
email: frank.werner@mathematik.uni-magdeburg.de

March 18, 2010

Abstract

In this note, we consider a single machine scheduling problem with
generalized total tardiness objective function. An NP-hardness proof
and a pseudo-polynomial time solution algorithm are proposed for a
special case of this problem.

Keywords: Scheduling, Single machine, Total tardiness, Number of
tardy jobs, Complexity, Pseudo-polynomial algorithm

MSC classification: 90 B 35

1 Introduction

Two classical single machine scheduling problems are the problem of
minimizing total tardiness and the problem of minimizing the number
of tardy jobs which can be formulated as follows. We are given a set
N = {1, 2, . . . , n} of n independent jobs that must be processed on a
single machine. Preemptions of a job are not allowed, and at any time
no more than one job can be processed. The processing of the jobs
starts at time 0. For each job j ∈ N, a processing time pj > 0 and a
due date dj are given.

A schedule is uniquely determined by a permutation π = (j1, j2, . . . , jn)
of the jobs of set N. Let Cj(π) = Σi=1 pj be the
completion time of job j in schedule π. If Cj(π) > dj, then job j is
tardy and we have Uj = 1, otherwise Uj = 0. If Cj(π) ≤ dj, then job
j is said to be on-time. Moreover, let Tj(π) = max{0, Cj(π) − dj} be
the tardiness of job j in schedule π. For the problem of minimizing
the number of tardy jobs \(1|| \sum U_j\), the objective is to find an optimal
schedule \(\pi^*\) that minimizes the value \(F(\pi) = \sum_{j=1}^n U_j(\pi)\) and for the
problem of minimizing total tardiness \(1|| \sum T_j\), the objective is to find
an optimal schedule \(\pi^*\) that minimizes the value \(F(\pi) = \sum_{j=1}^n T_j(\pi)\).

Problem \(1|| \sum U_j\) can be solved in \(O(n \log n)\) time by Moore’s al-
gorithm [7]. Problem \(1|| \sum T_j\) is NP-hard in the ordinary sense [1, 3].
A pseudo-polynomial dynamic programming algorithm of time com-
plexity \(O(n^4 \sum p_j)\) has been proposed by Lawler [2]. A summary of
polynomially and pseudo-polynomially solvable special cases can be
found e.g. in [4].

In this note, we consider a generalization of these two problems. In
addition to the above data, a quota of tardiness \(b_j \geq 0\), a coefficient
of normal penalty \(v_j \geq 0\) and a coefficient of abnormal penalty \(w_j \geq 0\)
are given for each job \(j \in N\). We define the generalized tardiness as
follows:

\[
GT_j(\pi) = \begin{cases} 
0, & \text{if } C_j(\pi) - d_j \leq 0, \\
v_j \cdot (C_j(\pi) - d_j), & \text{if } 0 < C_j(\pi) - d_j \leq b_j, \\
w_j, & \text{if } b_j < C_j(\pi) - d_j,
\end{cases}
\]

where \(w_j \geq v_j b_j\) for all \(j \in N\), and we define

\[
GT(\pi) = \sum_{j=1}^n GT_j(\pi).
\]

This means that, from a certain level of tardiness described by pa-
rameter \(b_j\) for job \(j\), the penalty \(w_j\) for exceeding the due date \(d_j\) is constant and does no longer depend on the concrete value of the tar-
diness. The objective is to find an optimal schedule \(\pi^*\) that minimizes
the function \(GT(\pi)\). We will denote this problem by \(1|| \sum GT_j\). It is
obvious that this problem is NP-hard. For the special case of \(b_j = 0\),
we have the classical problem \(1||w_j U_j\) which is NP-hard in the ordi-
ney sense [5]. For problem \(1||w_j U_j\), there exists a pseudo-polynomial
solution algorithm with time complexity \(O(n^2 d_{\text{max}})\) [6], where \(d_{\text{max}}\)
is the maximal due date of the jobs. Moreover, it is easy to show that
already the special case of problem \(1|| \sum GT_j\) with \(b_j \in Z_+\) is also
NP-hard.

In this note, we consider a special case of the generalized total
tardiness problem with

\[
b_j = p_j, \quad v_j = 1, \quad w_j = p_j,
\]
i.e., \(GT_j = \min\{\max\{0, C_j(\pi) - d_j\}, p_j\}\) for all \(j \in N\). We prove that
this problem is NP-hard in the ordinary sense and give a modification
of Lawler’s pseudo-polynomial algorithm [2] for this special case.

2
2 A Complexity Result and a Solution Algorithm

Before presenting the NP-hardness proof, we introduce the Partition problem which is as follows.

Partition problem
Given is a set \( N = \{a_1, a_2, \ldots, a_n\} \) of numbers \( a_1 \geq a_2 \geq \ldots \geq a_n > 0 \) with \( a_i \in \mathbb{Z}_+ \), \( i = 1, 2, \ldots, n \). Let \( A = \frac{1}{2} \sum_{j \in N} a_j \) with \( A \in \mathbb{Z}_+ \). Does there exist a subset \( N' \subset N \) such that
\[
\sum_{i \in N'} a_i = A?
\]

Theorem 1 The special case (1) of problem \( 1||\sum GW_j \) is NP-hard.

Proof. We give a reduction from the Partition problem. Given an instance of the Partition problem, we construct an instance of special case (1) with \( n+1 \) jobs. The processing times are given by \( p_j = a_j, j = 1, 2, \ldots, n \), and \( p_{n+1} = 1 \), i.e., we have \( \sum_{j=1}^{n+1} p_j = 2A+1 \). Additionally, the due dates are given by \( d_j = A, j = 1, 2, \ldots, n \), and \( d_{n+1} = A+1 \). It is obvious that for any schedule \( \pi \), we have \( GT(\pi) \geq A = \sum_{j=1}^{n+1} p_j - d_{n+1} \). Moreover, for all schedules, inequality \( GT(\pi) \leq A + 1 \) holds.

We have an optimal schedule \( \pi^* = (\pi_1, n+1, \pi_2) \) without idle times, where \( C_{n+1}(\pi^*) = A + 1 \), if and only if the answer for the instance of the Partition problem is "YES", i.e., all jobs from the partial schedule \( \pi_1 \) and job \( n+1 \) are on-time, all jobs from the partial schedule \( \pi_2 \) are tardy and we have \( GT(\pi^*) = A \).

Now we present a modification of Lawler’s algorithm for the special case (1).

Lemma 1 Let \( \pi \) be an optimal schedule with respect to given due dates \( d_1, d_2, \ldots, d_n \) and \( C_j \) be the completion time of job \( j \), \( j = 1, 2, \ldots, n \), for this schedule. Moreover, let \( d'_j \) be chosen such that:
- if \( C_j < p_j + d_j \), then
  \[
  \min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\};
  \]
- if \( C_j \geq p_j + d_j \), then
  \[
  d'_j = d_j.
  \]

Then any optimal schedule \( \pi' \) with respect to the due dates \( d'_1, d'_2, \ldots, d'_n \) is also optimal with respect to \( d_1, d_2, \ldots, d_n \).
Proof. Let $GT$ denote the generalized tardiness with respect to $d_1, d_2, \ldots, d_n$ and $GT'$ denote the generalized tardiness with respect to $d'_1, d'_2, \ldots, d'_n$.

Let $\pi'$ be any optimal schedule with respect to $d'_1, d'_2, \ldots, d'_n$ and $C'_j$ be the completion time of job $j$ for this schedule. We can write the objective function values $GT(\pi)$ and $GT(\pi')$ in the form

$$GT(\pi) = GT'(\pi) + \sum_{j=1}^{n} A_j,$$

$$GT(\pi') = GT'(\pi') + \sum_{j=1}^{n} B_j,$$

where $A_j$ and $B_j$ are terms depending on job $j \in N$ which we determine in the following.

We prove that $A_j \geq B_j$ holds for all $j = 1, 2, \ldots, n$. We consider the following two cases (a) and (b).

(a) Let $C_j \leq d_j$. Then we obtain:

1. if $C'_j \leq d'_j$, then $A_j = 0$ and $B_j = 0$;
2. if $d'_j < C'_j \leq d_j$, then $A_j = 0$ and $B_j < 0$;
3. if $d_j < C'_j$, then $GT_j(\pi') \leq GT'_j(\pi')$. Thus, we have $A_j = 0$ and $B_j < 0$.

(b) Let $C_j > d_j$. Then we obtain:

1. if $C'_j \leq d_j$, then $A_j = d'_j - d_j \geq 0$ and $B_j = 0$;
2. if $d_j < C'_j \leq d'_j$, then $A_j = d'_j - d_j \geq 0$ and $B_j = (C'_j - d_j) - 0 < d'_j - d_j = A_j$;
3. if $d'_j < C'_j$ and $C'_j - d_j \leq p_j$, then $B_j = d'_j - d_j = A_j$;
4. if $d'_j < C'_j$ and $C'_j - d_j > p_j$, then $C'_j - p_j > d_j$. We obtain $B_j = GT_j(\pi') - GT'_j(\pi') = p_j - (C'_j - d'_j) = d'_j - (C'_j - p_j) < d'_j - d_j = A_j$.

Thus, inequality $A_j \geq B_j$ holds for all $j = 1, 2, \ldots, n$. Moreover, we get $GT'(\pi) \geq GT'(\pi')$ since $\pi'$ minimizes $GT'$. Therefore, we obtain

$$GT'(\pi) + \sum_{j=1}^{n} A_j \geq GT'(\pi') + \sum_{j=1}^{n} B_j$$

and $GT(\pi) \geq GT(\pi')$, i.e., Lemma 1 has been proven.

□

Two following lemmas are obvious.

4
Lemma 2 Let $\pi$ be an optimal schedule. If we have $C_j(\pi) \geq p_j + d_j$ for a job $j \in N$, then there exists an optimal schedule, where job $j$ is processed on the last position.

Lemma 3 [2] There exists an optimal schedule $\pi$ in which job $i$ precedes job $j$ if $d_i \leq d_j$ and $p_i < p_j$, and in which all on-time jobs are in a non-decreasing order of the due dates.

Now we can present the following major result.

Theorem 2 Let the jobs of set $N$ be ordered such that $d_1 \leq d_2 \leq \ldots \leq d_n$ and $j^*$ be a job with the largest processing time. Then

- there exists a job $k \geq j^*$ such that $\sum_{i=1}^{k} p_i < p_{j^*} + d_{j^*}$ and there exists an optimal schedule, where all jobs $l = 1, 2, \ldots, k$ with $l \neq j^*$ are scheduled before $j^*$ and the remaining jobs are scheduled after $j^*$, or
- there exists an optimal schedule, where job $j^*$ is processed on the last position.

The proof of this theorem can be given in a similar way as in [2].

Thus, we can construct a modification of Lawler’s algorithm [2], where for job $j^*$, we consider independently all positions $k \geq j^*$ such that $\sum_{i=1}^{k} p_i < p_{j^*} + d_{j^*}$ and, additionally, the last position in the schedule. The time complexity of this algorithm remains the same, i.e., this special case of the generalized total tardiness problem under consideration can be solved in $O(n^4 \sum p_j)$ time.

References
