数理解析研究所講究録
945 巻 1996 年 104-114

Single Machine Scheduling with Generalized Due Dates

田中 圭介 (Keisuke Tanaka) ミラン ブラッハ (Milan Vlach)

School of Information Science
Japan Advanced Institute of Science and Technology–Hokuriku
Tatsunokuchi, Ishikawa 923-12, Japan
(tanaka, vlach)@jaist.ac.jp

1 Introduction

We are concerned with scheduling n independent jobs $J_1, J_2, \ldots, J_n$ on a single machine so as to minimize a given objective function involving generalized due dates. We make the following assumptions about feasibility of schedules.

1. The scheduling period is the interval $[0, \infty)$.
2. The machine is continuously available from the beginning, and it cannot process more than one job at a time.
3. The processing times $p_1, p_2, \ldots, p_n$ of jobs $J_1, J_2, \ldots, J_n$ are positive numbers known in advance, and they are independent of schedules.
4. Preemption is not permitted, that is, each job, once started, must be completed without interruption before another job is started.
5. All jobs are available for processing from the beginning.

The objective functions we are interested in involve generalized due dates proposed by Hall [7]. To illustrate the difference between the traditional view of due dates and Hall's view, consider the concept of lateness of a job in a schedule. In the traditional view, each job J_i has associated with it not only a processing time p_i but also a due date d_i. All due dates $d_1, d_2, \ldots, d_n$ are known in advance and they are independent of schedules. In Hall's view, no job has its own due date in advance. Instead, only a non-decreasing sequence

$$\delta_1 \leq \delta_2 \leq \cdots \leq \delta_n$$

of numbers, called generalized due dates, is given. In both cases, for each $1 \leq i \leq n$, every schedule S determines uniquely

1. the job $J_{S(i)}$ in the ith position of schedule S, that is, an order (sequence)

$$(S(1), S(2), \ldots, S(n))$$

   in which the jobs are processed on the machine, and
2. the completion time $C_i(S)$ of job J_i in schedule S.

The lateness of the ith job in S, that is, the lateness $L_{S(i)}(S)$ of job $J_{S(i)}$ in S under the traditional view is given by

$$L_{S(i)}(S) = C_{S(i)}(S) - d_{S(i)},$$

whereas the lateness $L^H_{S(i)}(S)$ of job $J_{S(i)}$ in S according to Hall's view is given by

$$L^H_{S(i)}(S) = C_{S(i)}(S) - \delta_i.$$

For example, if

$$\begin{aligned} p_1 &= 3, \quad p_2 = 2, \quad p_3 = 5, \\ d_1 &= 4, \quad p_2 = 7, \quad p_3 = 10, \\ \delta_1 &= 4, \quad \delta_2 = 7, \quad \delta_3 = 10, \end{aligned}$$

then, for the permutation schedule given by the sequence (J_1, J_3, J_2), we have

$$\begin{aligned} L_1 &= -1, \quad L_2 = 3, \quad L_3 = -5, \\ L^H_1 &= -1, \quad L^H_2 = 0, \quad L^H_3 = -2, \end{aligned}$$

Several authors [2, 7, 11] describe situations in which generalized due dates arise quite naturally. These include public utility planning, survey design and some types of flexible manufacturing. Obviously, the concept of generalized due dates was proposed with the aim of allowing for job independent due dates. It may, however, be also useful to consider generalized due dates as numbers through which the sequence dependent due dates are determined. Then we obtain the traditional concept and Hall's concept as two (very) special cases of sequence dependent due dates $D_1(S), D_2(S), \ldots, D_n(S)$. Taking constant (sequence independent) functions

$$D_i(S) = d_i \quad \text{for } 1 \leq i \leq n,$$

we obtain the traditional concept, taking

$$D_i(S) = \delta_{S^{-1}(i)} \quad \text{for } 1 \leq i \leq n,$$

we obtain Hall's concept.

Now it is clear that we may expect a variety of changes in results concerning the generalized due date counterparts of the traditional scheduling problems. The following table, in which the notation[1] proposed by Graham et al. [6] is used, shows that problems involving generalized due dates may be easier, harder, or equally difficult as their traditional counterparts. The table suggests that the max-problems tends to be harder and the sum-problems tend to be easier for the problems involving generalized due dates (see [8] for further details).

[1]This notation will be used throughout this paper.

Problem (notation for traditional view)	Traditional view	Hall's view
$1\|\|L_{\max}$	Polynomially solvable	Polynomially solvable
$1\|prec\|L_{\max}$	Polynomially solvable	NP-hard
$1\|r_j\|L_{\max}$	NP-hard	NP-hard
$1\|\|\sum U_j$	Polynomially solvable	Polynomially solvable
$1\|prec, p_j = 1\|\sum U_j$	NP-hard	Polynomially solvable
$1\|r_j\|\sum U_j$	NP-hard	NP-hard
$1\|\|\sum T_j$	NP-hard	Polynomially solvable
$1\|prec, p_j = 1\|\sum T_j$	HP-hard	Polynomially solvable
$1\|r_j\|\sum T_j$	NP-hard	NP-hard

Most research in scheduling involving generalized due dates has been concerned with establishing the complexity status of the problems whose traditional counterparts have regular objective functions. Little is known about the problems whose traditional counterparts have non-regular objective functions, and about approximation algorithms for the problems involving generalized due dates.

In what follows, we are concerned with several single machine problems involving non-regular objective functions and generalized due dates. The objective functions we are interested in are defined as follows.

Traditional model	Hall's model
$L_{\max}(S) = \max_{1\le i\le n} L_i(S)$	$L^H_{\max}(S) = \max_{1\le i\le n} L^H_i(S)$
$L_{\min}(S) = \min_{1\le i\le n} L_i(S)$	$L^H_{\min}(S) = \min_{1\le i\le n} L^H_i(S)$
$\Delta L(S) = L_{\max}(S) - L_{\min}(S)$	$\Delta L^H(S) = L^H_{\max}(S) - L^H_{\min}(S)$
$L_{\text{abs}}(S) = \max\|L_i(S)\|$	$L^H_{\text{abs}}(S) = \max\|L^H_i(S)\|$

Main results can be summarized as follows. First, we show that the problems of minimizing the maximum absolute lateness and range of lateness are NP-hard in the strong sense, both with and without allowing for machine idle time. Second, for all of these problems, we give simple efficient approximation algorithms based on the first-fit strategy. We show that they achieve the performance ratios of n for the problems of minimizing the maximum absolute lateness and of $(n+1)/2$ for the problems of minimizing the range of lateness.

2 Complexity

In this section, we begin with surveying the maximum and minimum lateness problems. For these problems, simple sequencing rules give optimal schedules. Then, we investigate the complexity of the problems of minimizing the maximum absolute lateness and minimizing the range of lateness both under and without the requirement that machine idle time is forbidden.

Following the notation of Hoogeveen [9], we use *nmit* to indicate that no machine idle time is allowed. Recall that a permutation schedule is a non-preemptive schedule in which the machine starts processing at time $t = 0$ and continues without any inserted idle time until $t = \sum_{j=1}^{n} p_j$. Such schedules are usually specified by sequencing rules. We use the following rules.

Shortest processing time (SPT) rule: sequence the jobs in non-decreasing order of processing times

$$p_{S(1)} \leq p_{S(2)} \leq \cdots \leq p_{S(n)}$$

Longest processing time (LPT) rule: sequence the jobs in non-increasing order of processing times

$$p_{S(1)} \geq p_{S(2)} \geq \cdots \geq p_{S(n)}$$

Earliest due date (EDD) rule: sequence the jobs in non-decreasing order of (traditional) due dates

$$d_{S(1)} \leq d_{S(2)} \leq \cdots \leq d_{S(n)}$$

Minimum slack time (MST) rule: sequence the jobs in non-decreasing order of slack times

$$d_{S(1)} - p_{S(1)} \leq d_{S(2)} - p_{S(2)} \leq \cdots \leq d_{S(n)} - p_{S(n)}$$

Proposition 1 *Each permutation schedule generated by the*

1. *EDD rule is optimal for the* $1||L_{\max}$ *problem,*
2. *MST rule is optimal for the* $1|nmit| - L_{\min}$ *problem,*
3. *SPT rule is optimal for the* $1||L^H_{\max}$ *problem,*
4. *LPT rule is optimal for the* $1|nmit| - L^H_{\min}$ *problem.*

Proof. All these results can easily be proved by a straightforward adjacent pairwise interchange argument. The first case is also known as Jackson's rule [10]. The second case, a mirror image of Jackson's rule, was already presented in [1, 3]. ∎

Now, we investigate the problem of minimizing the maximum absolute lateness. Garey, Tarjan, and Wilfong [5] proved the following proposition.

Proposition 2 *The problems*

$$1||L_{\text{abs}} \quad \textit{and}$$
$$1|nmit|L_{\text{abs}}$$

can be solved in polynomial time.

On the other hand, we have the following theorem concerning Hall's model.

Theorem 1 *The problems*

$$1||L^H_{\text{abs}} \quad \textit{and}$$
$$1|nmit|L^H_{\text{abs}}$$

are NP-hard in the strong sense.

Proof. First, we consider the problem $1|nmit|L^H_{\text{abs}}$. We use 3-partition problem, which is known to be NP-hard in the strong sense [4]. We transform 3-partition problem to the maximum absolute lateness problem with the constraint *nmit*.

Suppose we have an instance of 3-partition problem, i.e., suppose we have a positive integer B and a family $A = \{a_1, a_2, \ldots, a_{3n}\}$ of positive integers such that $\sum_{j=1}^{3n} a_j = nB$ and $B/4 < a_j < B/2$ for $1 \leq j \leq 3n$. We construct an instance of the lateness problem under consideration with

1. $4n$ jobs $J_1, J_2, \ldots, J_{4n}$ whose processing times $p_1, p_2, \ldots, p_{4n}$ are

$$p_i = a_i \quad \text{for } 1 \leq i \leq 3n,$$
$$p_i = B + 1 \quad \text{for } 3n + 1 \leq i \leq 4n, \text{ and}$$

2. Hall's due dates $\delta_1, \delta_2, \ldots, \delta_{4n}$ given by

$$\delta_i = (2B+1)\lceil i/4 \rceil - B/2 \quad \text{for } 1 \leq i \leq 4n,$$

where $\lceil x \rceil$ is the smallest integer no less than x.

We now show that the problem above has a schedule with the maximum absolute lateness $B/2$ iff A has a desired partition. Suppose that A has a partition $A = A_1 \cup A_2 \cup \cdots \cup A_n$ such that $\sum_{a_j \in A_i} a_j = B$ for $1 \leq i \leq n$. Further suppose that $A_i = \{a_{3i-2}, a_{3i-1}, a_{3i}\}$ for $1 \leq i \leq n$. It is easy to check that the schedule

$$(J_{3n+1}, J_1, J_2, J_3, J_{3n+2}, J_4, J_5, J_6, \ldots, J_{4n}, J_{3n-2}, J_{3n-1}, J_{3n})$$

is feasible and gives the maximum absolute lateness $B/2$.

On the other hand, suppose that there exists a schedule with the maximum absolute lateness $B/2$. Notice that all feasible schedules give the maximum lateness no less than $B/2$ and the minimum lateness no greater than $-B/2$, thus, the maximum absolute lateness no less than $B/2$.

We consider the positions of $J_{3n+1}, J_{3n+2}, \ldots, J_{4n}$. We call jobs $J_1, J_2, \ldots, J_{3n}$ a-type and jobs $J_{3n+1}, J_{3n+2}, \ldots, J_{4n}$ b-type. For each $1 \leq i \leq n$, a b-type job must be scheduled in the $(4i-3)$rd position from $(2B+1)(i-1)$ to $(2B+1)i - B$. If a-type job J_a is scheduled in the $(4i-3)$rd position, the lateness of the job in the $(4i-4)$th position is greater than $B/2$, or the lateness of J_a is less than $-B/2$.

If b-type job J_b in the $(4i-3)$rd position is not scheduled from $(2B+1)(i-1)$ to $(2B+1)i-B$, the lateness of the job in the $(4i-4)$th position is greater than $B/2$, or the lateness of J_b is less than $-B/2$.

The positions of b-type jobs divide the whole time interval where the machine runs into n time intervals of the same length B. This implies A has a partition $A = A_1 \cup A_2 \cup \cdots \cup A_n$ such that $\sum_{a_j \in A_i} a_j = B$ for $1 \leq i \leq n$, where the jobs from the $(4i-2)$nd position to the $(4i)$th correspond to the elements in A_i for each $1 \leq i \leq n$.

Next, we consider the problem $1||L^H_{\text{abs}}$. Notice that if the machine idle time is inserted, the maximum lateness of $B/2$ cannot be attained, neither can the maximum absolute lateness. So, even if we allow the machine to be idle, no optimal schedule has the machine idle time. This enable us to apply a similar argument as for the *nmit* version of the problem. ∎

Next, we investigate the complexity of the problems of minimizing the range of lateness. Hoogeveen [9] proved the following proposition.

Proposition 3 *The problems*

$$1||\Delta L \quad \textit{and}$$
$$1|nmit|\Delta L$$

can be solved in polynomial time.

On the other hand, we have the following two theorems concerning Hall's model.

Theorem 2 *The problem* $1|nmit|\Delta L^H$ *is NP-hard in the strong sense.*

Proof. We transform 3-partition problem to the range of lateness problem with the constraint *nmit* by constructing the same lateness problem as in the proof of Theorem 1.

Notice that all feasible schedules give the maximum lateness no less than $B/2$ and the minimum lateness no greater than $-B/2$, thus, the range of lateness no less than B. By a similar argument as in the proof of Theorem 1, we can show that the problem above has a schedule with the range B of lateness (i.e., the maximum lateness $B/2$ and the minimum lateness $-B/2$) iff A has a desired partition. ∎

Theorem 3 *The problem $1||\Delta L^H$ is NP-hard in the strong sense.*

Proof. We transform 3-partition problem to the range of lateness problem by constructing the same lateness problem as in the proof of Theorem 1.

We now show that the problem above has a schedule with the range B of lateness iff A has a desired partition. Suppose that A has a partition. It is easy to check that the following schedule gives the range B of lateness: the order of the jobs is the same as in the proof of Theorem 1 and there is no machine idle time in the schedule.

On the other hand, suppose that there exists a schedule with the range B of lateness. Let C be the sum of the intervals where the machine is idle. Then, all schedules give the the maximum lateness no less than $B/2+C$, and the minimum lateness no greater than $-B/2+C$, thus, the range of lateness no less than B. We call jobs $J_1, J_2, \ldots, J_{3n}$ a-type and jobs $J_{3n+1}, J_{3n+2}, \ldots, J_{4n}$ b-type.

We show the sum of the processing times of the jobs in these positions is exactly B for all $1 \le i \le n$. Suppose the sum of the processing times of the jobs in the $(4i-2)$nd position to the $(4i)$th is greater than B for some $1 \le i \le n$. Then, the difference between the completion time of the job in the $(4i-3)$rd position and that in the $(4i)$th is greater than B, which prevents us from attaining the range B of lateness. Thus, there are only a-type jobs in the $(4i-2)$nd position to the $(4i)$th.

On the other hand, suppose the sum of the processing times of the jobs in the $(4i-2)$nd position to the $(4i)$th is less than B for some $1 \le i \le n$. Then, that in the $(4j-2)$nd position to the $(4j)$th is greater than B for some $1 \le j \le n$, $j \ne i$. This implies A has a partition $A = A_1 \cup A_2 \cup \cdots \cup A_n$ such that $\sum_{a_j \in A_i} a_j = B$ for $1 \le i \le n$, where the jobs from the $(4i-2)$st position to the $(4i)$rd correspond to the elements in A_i for each $1 \le i \le n$. ∎

3 Approximation algorithms

In this section, we present two simple approximation algorithms for the problems $1||L^H_{\mathrm{abs}}$, $1|nmit|L^H_{\mathrm{abs}}$, $1||\Delta L^H$, and $1|nmit|\Delta L^H$. The algorithms are based on the first-fit strategy.

First, we introduce Algorithm A which works for the problems of minimizing L^H_{abs} and minimizing ΔL^H without allowing for the machine idle time.

The algorithm returns the resulting schedule A as a permutation, i.e., A returns the index $A(i)$ of the job in the ith position for each i, $1 \le i \le n$.

```
Algorithm A(p_1, p_2, ..., p_n, δ_1, δ_2, ..., δ_n)
  δ_0 ← 0
  for i = 1 to n do
    a_i = δ_i − δ_{i−1}
  I ← {1, 2, ..., n}
  J ← {1, 2, ..., n}
```

```
while I ≠ ∅ do
  Choose i such that a_i = min_{k∈I} a_k
  Choose j such that p_j = min_{k∈J} p_k
  A(i) ← j
  I ← I \ {i}
  J ← J \ {j}
od
Output(A)
end
```

The time complexity of Algorithm A is $O(n \log n)$, if we use a fast sorting scheme. First, we show the following lemma which plays an important role in the proofs of establishing the performance guarantees.

Lemma 1 *For each schedule S, we have*

$$\max_i\{p_{A(i)} - a_i\} \le \max_i\{p_{S(i)} - a_i\}$$

and

$$\min_i\{p_{A(i)} - a_i\} \ge \min_i\{p_{S(i)} - a_i\}.$$

Proof. We only verify the validity of the first inequality. The proof of the second one is analogous. Without loss of generality, we assume that $p_1 \le p_2 \le \cdots \le p_n$. The proof is by contradiction. Suppose that there exists a schedule S such that $p_{A(j)} - a_j > p_{S(k)} - a_k$, where j and k are such that $p_{A(j)} - a_j = \max_i\{p_{A(i)} - a_i\}$ and $p_{S(k)} - a_k = \max_i\{p_{S(i)} - a_i\}$.

Since $p_{A(j)} - a_j > p_{S(k)} - a_k \ge p_{S(j)} - a_j$, we have $p_{A(j)} > p_{S(j)}$, consequently, $A(j) > S(j)$. There are at most $(A(j) - 2)$ i's such that $i \ne j$ and $p_{S(i)} < p_{A(j)}$. But, there are at least $(A(j) - 1)$ i's such that $i \ne j$ and $a_i \le a_j$. Therefore, there exists i, $i \ne j$ such that $a_i \le a_j$ and $p_{S(i)} \ge p_{A(j)}$. Hence,

$$\begin{aligned} p_{S(k)} - a_k &\ge p_{S(i)} - a_i \\ &= p_{S(i)} - a_j + a_j - a_i \\ &\ge p_{A(j)} - a_j + a_j - a_i \\ &\ge p_{A(j)} - a_j. \end{aligned}$$

This contradicts the assumption. ∎

Then, we analyze the performance of the algorithm concerning the problem $1|nmit|L^H_{\text{abs}}$. Let OPT be an optimal schedule for this problem. We obtain the following bound on the performance of Algorithm A.

Theorem 4

$$L^H_{\text{abs}}(A) \le n \times L^H_{\text{abs}}(OPT).$$

Proof. By Lemma 1, we have

$$\begin{aligned} L^H_{\text{abs}}(OPT) &= \max\{L^H_{\max}(OPT), -L^H_{\min}(OPT)\} \\ &\ge \frac{1}{2} \times (L^H_{\max}(OPT) - L^H_{\min}(OPT)) \\ &\ge \frac{1}{2} \times \max_i |L^H_{OPT(i)} - L^H_{OPT(i-1)}| \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2} \times \max_i |C_{OPT(i)} - \delta_i - (C_{OPT(i-1)} - \delta_{i-1})| \\
&= \frac{1}{2} \times \max_i |p_{OPT(i)} - a_i| \\
&\geq \frac{1}{2} \times \max_i |p_{A(i)} - a_i|.
\end{aligned}
$$

Let $I = \{1, 2, \ldots, n\}$, and let J and K be the set of the indices such that $p_{A(i)} \geq a_i$ for all $i \in J$ and $p_{A(i)} < a_i$ for all $i \in K$, respectively. From the definition of L^H_{abs} and a_i, it follows that

$$
\begin{aligned}
L^H_{\text{abs}}(A) &= \max\{L^H_{\max}(A), -L^H_{\min}(A)\} \\
&\leq \max\{\sum_{i\in J}(p_{A(i)} - a_i), -\sum_{i\in K}(p_{A(i)} - a_i)\}.
\end{aligned}
$$

First, we assume that $|J| = n/2$. Then we have

$$
\begin{aligned}
L^H_{\text{abs}}(A) &\leq \max\{|J| \times \max_{i\in J}\{p_{A(i)} - a_i\}, -|K| \times \min_{i\in K}\{p_{A(i)} - a_i\}\} \\
&\leq \max\{\frac{n}{2} \times \max_i\{p_{A(i)} - a_i\}, -\frac{n}{2} \times \min_i\{p_{A(i)} - a_i\}\} \\
&\leq \frac{n}{2} \times \max_i |p_{A(i)} - a_i|.
\end{aligned}
$$

Next, we assume that $|J| \leq (n-1)/2$. Then we have

$$
\begin{aligned}
L^H_{\text{abs}}(A) &\leq \max\{\sum_{i\in J}(p_{A(i)} - a_i), -\sum_{i\in I\setminus J}(p_{A(i)} - a_i)\} \\
&= \max\{\sum_{i\in J}(p_{A(i)} - a_i), \sum_{i\in J}(p_{A(i)} - a_i) - \sum_{i\in I}(p_{A(i)} - a_i)\} \\
&\leq \sum_{i\in J}(p_{A(i)} - a_i) + |\sum_{i\in I}(p_{A(i)} - a_i)| \\
&\leq |J| \times \max_{i\in J}\{p_{A(i)} - a_i\} + |C_{A(n)} - \delta_n| \\
&\leq \frac{n-1}{2} \times \max_i |p_{A(i)} - a_i| + |L^H_{A(n)}|.
\end{aligned}
$$

Finally, we assume that $|J| \geq (n+1)/2$. Then we have

$$
\begin{aligned}
L^H_{\text{abs}}(A) &= \max\{\sum_{i\in I\setminus K}(p_{A(i)} - a_i), -\sum_{i\in K}(p_{A(i)} - a_i)\} \\
&= \max\{\sum_{i\in I}(p_{A(i)} - a_i) - \sum_{i\in K}(p_{A(i)} - a_i), -\sum_{i\in K}(p_{A(i)} - a_i)\} \\
&\leq |\sum_{i\in I}(p_{A(i)} - a_i)| - \sum_{i\in K}(p_{A(i)} - a_i) \\
&\leq |C_{A(n)} - \delta_n| - |K| \times \min_{i\in K}\{p_{A(i)} - a_i\} \\
&\leq |L^H_{A(n)}| + \frac{n-1}{2} \times \max_i |p_{A(i)} - a_i|.
\end{aligned}
$$

To conclude the proof, it is sufficient to observe that

$$
|L^H_{A(n)}| = |L^H_{OPT(n)}| \leq L^H_{\text{abs}}(OPT).
$$

∎

Theorem 4 provides the performance ratio n between the optimal value of L^H_{abs} and the value induced by a schedule found by Algorithm A. The following theorem says that this ratio cannot be improved.

Theorem 5 *There exists an instance satisfying*

$$L^H_{\text{abs}}(A) = n \times L^H_{\text{abs}}(OPT).$$

Proof. Let n be an even number and let $m = n/2$. Consider an instance of the absolute lateness problem with

1. $2m$ jobs $J_1, J_2, \ldots, J_{2m}$ whose processing times $p_1, p_2, \ldots, p_{2m}$ are

$$\begin{aligned} p_1 &= 2, \\ p_i &= 1 \quad \text{for } 2 \le i \le m, \\ p_i &= 5 \quad \text{for } m+1 \le i \le 2m, \end{aligned}$$

2. Hall's due dates $\delta_1, \delta_2, \ldots, \delta_{2m}$ given by

$$\delta_i = 3i \quad \text{for } 1 \le i \le 2m.$$

For the instance above, it follows that $a_i = 3$ for all $1 \le i \le 2m$. So, Algorithm A has the possibility to give the schedule

$$(J_{m+1}, J_{m+2}, \ldots, J_{2m}, J_1, J_2, \ldots, J_m),$$

while an optimal schedule is

$$(J_1, J_{m+1}, J_2, J_{m+2}, \ldots, J_m, J_{2m}).$$

The value of L^H_{abs} of the schedule found by Algorithm A is $2m$, while the optimal value of L^H_{abs} is 1. ∎

Next, we analyze the performance of the algorithm concerning the problem $1|nmit|\Delta L^H$. Now, let OPT be an optimal schedule for this problem. We obtain the following bound on the performance of Algorithm A.

Theorem 6

$$\Delta L^H(A) \le \frac{n+1}{2} \times \Delta L^H(OPT).$$

Theorem 6 provides the performance ratio $(n+1)/2$ between the optimal value of ΔL^H and the value induced by a schedule found by Algorithm A. The following theorem says that this ratio cannot be improved.

Theorem 7 *There exists an instance satisfying*

$$\Delta L^H(A) = \frac{n+1}{2} \times \Delta L^H(OPT).$$

Next, we present an approximation algorithm for the problems $1||L^H_{\text{abs}}$ and $1||\Delta L^H$. To describe a schedule S', which may involve inserted idle times, we use an ordered pair $S' = (S_1, S_2)$, where S_1 is a permutation of jobs and S_2 is a function whose value $S_2(i)$ gives the idle time inserted between jobs $J_{S_1(i-1)}$ and $J_{S_1(i)}$.

```
Algorithm A'(p_1, p_2, ..., p_n, δ_1, δ_2, ..., δ_n)
  A_1 ← Algorithm A(p_1, p_2, ..., p_n, δ_1, δ_2, ..., δ a_n).
  c_0 ← 0
  for i = 1 to n do
    c_i ← c_{i-1} + p_{A_1(i)}
    if c_i < δ_i then
      A_2(i) ← δ_i − c_i
      c_i ← δ_i
    neht else
      A_2(i) ← 0
  od
  Output((A_1, A_2))
end
```

The time complexity of Algorithm A$'$ is $O(n \log n)$, if we use a fast sorting scheme. First, we introduce the following lemma which plays an important role in the proofs of establishing the performance guarantees. This lemma is an immediate consequence of Lemma 1.

Lemma 2 *For each schedule $S' = (S_1, S_2)$, and for each number b, we have*

$$\max_i \{p_{A_1(i)} - a_i + b\} \leq \max_i \{p_{S_1(i)} - a_i + b\}$$

and

$$\min_i \{p_{A_1(i)} - a_i + b\} \geq \min_i \{p_{S_1(i)} - a_i + b\}.$$

Then, we analyze the performance of the algorithm concerning the problem $1||L^H_{\mathbf{abs}}$. Let $OPT' = (OPT_1, OPT_2)$ be an optimal schedule for this problem. We obtain the following bound on the performance of Algorithm A$'$.

Theorem 8

$$L^H_{\mathbf{abs}}(A') \leq n \times L^H_{\mathbf{abs}}(OPT').$$

Theorem 8 provides the performance ratio n. The following theorem says that this ratio cannot be improved.

Theorem 9 *There exists an instance satisfying*

$$L^H_{\mathbf{abs}}(A') = n \times L^H_{\mathbf{abs}}(OPT').$$

Next, we analyze the performance of the algorithm concerning the problem $1||\Delta L^H$. Now, let $OPT' = (OPT_1, OPT_2)$ be an optimal schedule for this problem. We obtain the following bound on the performance of Algorithm A$'$.

Theorem 10

$$\Delta L^H(A') \leq \frac{n+1}{2} \times \Delta L^H(OPT').$$

Theorem 10 provides the performance ratio $(n+1)/2$ between the optimal value of ΔL^H and the value induced by a schedule found by Algorithm A$'$. The following theorem says that this ratio cannot be improved.

Theorem 11 *There exists an instance satisfying*

$$\Delta L^H(A') = \frac{n+1}{2} \times \Delta L^H(OPT').$$

Acknowledgments

We thank Kunihiro Fujiyoshi for drawing our attention to the possible improvement of Theorem 4.

References

[1] Baker, K. R. *Introduction to Sequencing and Scheduling.* Wiley, New York, 1974.

[2] Browne, J., Dubois, D., Rathmill, K., Sethi, S. P., and Stecke, K. Classification of flexible manufacturing systems. *The FMS Magazine* (Apr. 1984), 114–117.

[3] Conway, R. W., Maxwell, W. L., and Miller, L. W. *Theory of Scheduling.* Addison-Wesley, Massachusettes, 1967.

[4] Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[5] Garey, M. R., Tarjan, R. E., and Wilfong, G. T. One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research 13*, 2 (May 1988), 330–348.

[6] Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics 5* (1979), 287–326.

[7] Hall, N. G. Scheduling problems with generalized due dates. *IEE Transactions 18* (1986), 220–222.

[8] Hall, N. G., Sethi, S. P., and Sriskandarajah, C. On the complexity of generalized due date scheduling problems. *European Journal of Operational Research 51* (1991), 100–109.

[9] Hoogeveen, J. A. Minimizing maximum earliness and maximum lateness on a single machine. Tech. Rep. BS-R9001, Centre for Mathematics and Computer Science, Amsterdam, 1990.

[10] Jackson, J. R. Scheduling a production line to minimize maximum tardiness. Tech. Rep. 43, Management Science Research Project, University of California, Los Angeles, 1955.

[11] Stecke, K. E., and Solbert, J. J. Loading and control policies for a flexible manufacturing system. *International Journal of Production Research 19* (1981), 481–490.