Journal of the Korean
OR/ MS Society
Vol. 20, No. 1, April 1995

A Constrained Single Machine Scheduling Model with Earliness / Tardiness and Flow Time Measures

Un Gi Joo* · Chang Sup Sung**

Abstract

This paper considers a single machine nonpreemptive scheduling problem with a given common due date. In the problem, the optimal job sequence is sought to minimize the sum of earliness/tardiness and flow time measures in the situation where all jobs are available at time zero, and weights per unit length of earliness/tardiness and flow time are V and W, respectively.

Some dominant solution properties are characterized to derive both an optimal starting time for an arbitrary sequence and sequence improvement rules. The optimal schedule is found for the case $W \geq V$. By the way, it is difficult to find the optimal schedule for the case $W < V$. Therefore, the derived properties are put on together to construct a heuristic solution algorithm for the case $W < V$, and its effectiveness is rated at the mean relative error of about 3% on randomly generated numerical problems.

1. Introduction

This paper considers a single machine nonpreemptive scheduling problem with a given common due date for all jobs, where all the jobs are available at time zero. The objective is to find the optimal schedule which minimizes sum of flow time and earliness/tardiness with respect to the due date.

* Telecommunication Technology Division, ETRI
** Department of Industrial Engineering, KAIST

Many researches have been studied for the earliness/tardiness problem on a single machine. For example, Panwalkar *et al.* [9] have considered an earliness/tardiness problem including flow time measure to find the optimal sequence and due date. Cheng[2] and Dickman *et al.* [5] have studied a single machine earliness/tardiness scheduling problem with a common due date to be determined and with each job penalty imposed only on a certain range of its completion time deviation from the common due date. Sung and Joo[10] have considered a scheduling problem to minimize the sum of earliness/tardiness and starting time penalties for all jobs, where all the jobs are not necessarily available at time zero. De *et al.* [4] and Mittenthal and Raghavachari[8] have found sequences for minimizing both mean completion tme and variance of job completion times, where the minimization of variance of completion times equals to that of quadratic early-tardy penalties under the unconstrained common due date. The excellent survey work for the earliness/tardiness problem has been performed by Baker and Schdder[1].

This paper considers the same type problem as Panwalker *et al.* [9] which the due date is considered as a decision value. By the way, in this paper, the due date is a given value. Notice that, as described by Hall *et al.* [7], the earliness/tardiness problem is NP-complete, if the due date is a restrictively given value. Therefore, a heuristic algorithm may be required to find good schedule efficiently for the problem with restrictively given due date.

The basic motivation of consideration on earliness and tardiness problem is derived in JIT (Just-In-Time) system. And the minimization of flow time represents the reduction of in-process-inventory or operating cost. The objective of this paper represents an aim for delivering the jobs to prevent loss of goodwill and for minimizing in-process-inventory simultaneously.

Therefore, there occur tradeoffs between amount of inventory and penalty for loss of goodwill on production and sales departments, respectively.

This paper is composed as follows: In Section 2, problem formulation is described. In Section 3, some dominant solution properties are characterized and then a heuristic solution algorithm is presented in Section 4. In Section 5, computational experience is discussed. Finally, concluding remarks are added in Section 6.

2. Model Formulation

Let us define the following notations:

n=total number of jobs to be processed

p_i=processing time of job $i(i=1, 2, \cdots, n)$

d=a given common due date

C_i=completion(flow) time of job i in a sequence

s=processing starting time for the first job in a sequence

W=flow time penalty per unit time

V=earliness/tardiness penalty per unit time

The objective is to find the optimal schedule, S^*, which minimizes

$$Z(S) = \sum_{i=1}^{n} WC_i + \sum_{i=1}^{n} V|C_i - d|.$$

The first term represents total flow time and the second term represents the earliness/tardiness penalty. This objective function may be applicable for situations of manufacturing large goods via splitting with several sub-lots, or perishable goods, and it is further applicable in shipping schedule.

3. Dominant Solution Properties

For analysis, some additional notations are introduced:

A_k=set of jobs following job k in a given sequence(schedule)

B_k=set of jobs preceding job k in a given sequence

E=set of jobs completed early

T=set of jobs completed late

$|A|$=cardinalty of set A

It is well known that the first and second terms in $Z(S)$ are minimized by SPT(Shortest Processing Time) and V-shaped sequencing without intermediate idle time, respectively.

In this section, some properties are characterized to minimize earliness/tardiness and flow time measures simultaneously.

Proposition 1. A sequence without intermediate idle time is dominant.

Proof. It is obvious for a tardy job set since $Z(S)$ becomes a regular measure for the job set (Conway *et al.* [3]). And, for an early job set, it can be easily shown that shifting right of a sequence so that the sequence have no intermediate idle time is not worse when $W<V$. Similarly, shifting left to make a sequence without intermediate idle time is not worse when $W \geq V$.

This completes the proof.

Proposition 1 implies that a schedule of a sequence is determined when processing starting time of the first job is determined for the given sequence. And notice that the processing starting time for the first job is important because of incorporation of flow time. The following proposition describes the optimal starting time for an arbitrary sequence.

Proposition 2. Consider a given sequence S ordered as $1\text{-}2\text{-}\cdots\text{-}n$ and satisfying the relations $\sum_{j=1}^{r} p_j \leq d$ and $\sum_{j=1}^{r+1} p_j > d$ for a job r, $(r \leq n)$, where if $\sum_{j=1}^{n} p_j < d$, then let $r=n$. Let l be the largest integer smaller than or equal to $n(V+W)/2V$. Then, the optimal starting time, s^*, of the first job is determined as

$$s^* = \begin{cases} 0, & \text{if } r < n(V+W)/2V, \\ \text{any point on the closed domain } [0,\ d-\sum_{j=1}^{l} p_j], & \text{if } r = n(V+W)/2V, \\ \text{any point on the closed domain } [d-\sum_{j=1}^{l-1} p_j,\ d-\sum_{j=1}^{l} p_j], & \text{if } r > n(V+W)/2V = l, \\ d-\sum_{j=1}^{l+1} p_j, & \text{if } r > n(V+W)/2V \neq l. \end{cases}$$

Proof. For a given schedule S, the job index k can be defined so that either $C_k < d < C_{k+1}$ or $C_k = d$. The proof will be accomplished for each case. Consider a schedule S_1 such that $C_k < d < C_{k+1}$, where it is noticed that $|E| = k$. And consider schedules S^1 and S^2 resulted from shifting right and left the sequence S_1 so that $C_k^1 = d$ and $C_{k+1}^2 = d$, respectively. Then, the differences of objective values are

$Z(S_1) - Z(S^1) = -(d-C_k)[n(V+W) - 2V|E|]$, and

$Z(S_1) - Z(S^2) = -(C_{k+1}-d)[n(V+W) - 2V|E|]$.

Therefore, $Z(S_1) \geq Z(S^1)$ if $|E| \geq n(V+W)/2V$ and $Z(S_1) \geq Z(S^2)$ if $|E| \leq n(V+W)/2V$.

Now, consider a schedule S_2 such that $C_k = d$, where it is noticed that $|E| = k$. And consider schedules S^3 and S^4 resulted from shifting left and right the schedule S_2 so that $C_k^3 < d < C_{k+1}^3$ and $C_k^4 > d$, respectively. Then, the differences of objective values between S_2, S^3, and S^4 are

$Z(S_2) - Z(S^3) = -(d-C_k)[n(V+W) - 2V|E|]$, and

$Z(S_2) - Z(S^4) = -(C_k-d)[n(V+W) - 2V|E|+2V]$.

Therefore, $Z(S_2) \geq Z(S^3)$ if $|E| \leq n(V+W)/2V$ and $Z(S_2) \geq Z(S^4)$ if $|E| \geq n(V+W)/2V+1$.

From the above results, the following facts are derived:

Case (1). If $|E| < n(V+W)/2V$, then the left shifted schedules, S^2, and S^3 are better than original schedules S_1 and S_2.

Case (2). If $|E| = n(V+W)/2V$, then the schedules, S_1, S_2, S^1, S^2, and S^3, have the same objective value.

Case (3). If $|E| > n(V+W)/2V=l$, then the schedule S_1 can be improved by shifting right to make S^1, and S^1(which is same as S_2) can also be improved by shifting right to make S^4 when $|E| \geq =l+1$. Then, it becomes the Case (2). This fact can be applied for S_2 similarly.

Case (4). If $|E| > n(V+W)/2V \neq l$, then the schedule S_1 can be improved by shifting right as similar way Case (3). However, $d-\sum_{j=1}^{l} p_j$ can not be optimal since a situation of $|E|=n(V+W)/2V+1$ can not be occurred in this case. Notice that this fact is also applicable for the job index r such that $\sum_{j=1}^{r} p_j \leq d$ and $\sum_{j=1}^{r+1} p_j > d$.

This completes the proof.

Notice that proposition 2 characterizes the optimal starting time for a given sequence and the optimal number of early jobs is at most r, and the value of $n(V+W)/2V$ is larger than or equal to n if $W \geq V$, and $n/2 \leq n(V+W)/2V < n$ if $W<V$. It is also noticed that $s^*=0$ when $r<n(V+W)/2V$, and schedules such as completion time of a job coinciding with the due date are sufficient for the optimal schedule when $r \geq n(V+W)/2V$. However, we do not yet know what sequence is optimal, and the optimal starting time for $Z(S)$(not for a given sequence) is not characterized.

The folowing corollary describes a special case for finding the optimal starting time of $Z(S)$.

Corollary 3. For the given problem $Z(S)$, $\sum_{j=1}^{l} p_j \leq d$ from the SPT sequencing, then $s^*=0$ in optimum, where l is the largest integer smaller than or equal to $n(V+W)/2V$.

Proof. It is noticed that the total number of early jobs in a schedule is maximum when the schedule is resulted from SPT sequencing. Therefore, if the maximum number of early jobs are less than or equal to l, $s^*=0$ is optimal by Proposition 2.

This completes the proof.

The above two properties characterize the optimal starting time for a sequence. The characterization of which sequence is optimal is now described according to the relative size of W and V.

Proposition 4. If $W \geq V$, then the optimal schedule is obtained by SPT sequencing with starting time $s^*=0$.

Proof. Notice that an SPT sequencing is optimal for the tardy job set about both of earliness/tardiness and flow time measures.

It remains to show that the SPT sequencing is also optimal for early job set, and the smallest processing time in tardy job set is not smaller than that of associated jobs in early job set. For the proof, consider a schedule S with adjacent two jobs i and j in early, and another schedule S^1 which is the same as S except the positions of jobs i and j are interchanged. Then, $Z(S)-Z(S^1)=(p_i-p_j)(W-V)$. Therefore, SPT is also optimal for early job set.

Secondly, to show that the smallest processing time among the tardy job set is not smaller than that of associated jobs in early job set in optimum, consider a schedule S such that a job i is in the early job set and a job j is in the tardy job set, and another schedule S^1 which is the same as S except the positions of the jobs i and j are interchanged. Then, the objective value $Z(S)$ minus $Z(S^1)$ is negative when $p_i \leq p_j$.

This completes the proof.

For example, consider seven jobs with processing times $p_i=5$, 10, 12, 30, 31, 40, 45, and $d=90$, $W=10$, $V=8$. Then, since $W>V$, $s^*=0$ is optimal and the optimal sequence is SPT order by Proposition 4, i.e., 1-2-3-4-5-6-7 is the optimal sequence.

For the remaining case $W<V$, the shape of optimal sequence is characterized as following proposition.

Proposition 5. If $W<V$, then the optimal sequence is a V-shape, that is, the LPT(Longest Processing Time) and SPT sequencing for early and tardy job set are optimal, respectively.

Proof. Suppose that a schedule is sequenced as $[1]$-$[2]$-$\cdots$-$[n_1-1]$-$[n_1]-(n_2)-(n_2-1)$-$\cdots$-(2)-(1) with processing starting time s of the first job, where n_1 and n_2 denote total number of the early and tardy jobs in the schedule, respectively $(n_1+n_2=n)$. Then, the objective function $Z(S)$ transforms as follows :

$$Z(S)=nWs+\Delta(n_1-n_2)+\sum_{i=1}^{n}[W+i(V-W)-V]/p_{[i]}+\sum_{i=}^{n}[i(V+W)-W]p_{(i)}, \text{ where } \Delta = d-C_{[n]} \geq 0.$$

Therefore, an LPT sequencing are optimal for early and tardy job set, respectively.

This completes the proof.

Though it is characterized that the V-shape sequence is optimal for the case $W<V$ by Prop-

osition 5, it is not easy to find the optimal schedule because there are many V-shape schedules with different objective values. The next section describes how to find the optimal schedule for the case $W<V$.

4. Solution Algorithm for the Case of W<V

Though the optimal sequence for the case $W \geq V$ can be constructed by using Propositions 2 and 4, the problem for the case of $W<V$ is not easy to find the optimal solution.

Notice that the optimal schedule of the problem with $W<V$ can be obtained by full enumeration of $n!$ sequences by Proposition 1 if their corresponding optimal starting times are determined, where the optimal starting time of a sequence can be found for its partial sub-sequence by Proposition 2. Therefore, the optimal schedule can be found by using a branch-and-bound algorithm. In the branching step, some of possible branches can be fathomed by Proposition 5.

In fact, by result of Proposition 5, the dominating sequences are at most $2^{[n/2]}$ since a job can be sequenced at either earlier or later than the due date, where $[n/2]$ denotes the largest integer less than or equal to $n/2$. Notice that the branch-and-found algorithm requires partial sub-sequences to determine the optimal starting time, so that the algorithm seems have high computational load. And, it is impossible to sequence by only using positional weights due to starting time incorporation, hence, it seems to be NP-complete problem as shown in Hall *et al.* [6]. Therefore, a heuristic algorithm may be required for a good solution with computational efficiency.

To develop improving rules for an existing sequence, a pairwise interchange rule and two movement rules are derived in the following propositions.

Proposition 6. Consider a sequence S where two jobs i and j are sequenced early and tardy, respectively. And consider another V-shaped sequence S^1 which has the same jobs in each early and tardy job set as S except the positions of the jobs i and j are interchanged. Then, the sequence S^1 is not worse than S if $|B_i| \geq (V+W)/[(V-W)(|A_j+1)]$ and $p_i > p_j$, or $|B_i| \leq (V+W)/[(V-W)(|A_j+1)]$ and $p_i < p_j$.

Proof. Let $a=p_i-p_j$. And consider a sequence S^2 which has the same jobs in each early and tardy job set as S except the positions of the jobs i and j are interchanged so that $C_j^2=C_i$ and $C_i^2=C_j+p_i-p_j$. Then, the difference between objective function values of S and S^2 is

$$Z(S)-Z(S^2) = \begin{cases} a[(V-W)|B_i|-(V+W)|A_j|-(V+W)], \text{ if } p_i>p_j, \\ a[(V+W)|A_j|-(V-W)|B_i|-(V+W)], \text{ if } p_i<p_j. \end{cases}$$

The V-shaped sequence S^1 is obtained by LPT and SPT sequencing for the early and tardy job set, respectively. And S^1 is not worse than S^2 by Proposition 5.

This completes the proof.

Notice that the positions of jobs i and j can only be interchanged when $s \geq p_j - p_i$ if $p_i < p_j$.

The following two propositions describe conditions for moving a job from early position to tardy position or vice versa.

Proposition 7. Consider a sequence S where two jobs i and j are sequenced early and tardy position, respectively. And consider another V-shaped sequence S^1 which has the same jobs in each early and tardy job set as S except the job i in early position is moved toward tardy position. Then, if $(V+W)[(|A_j|+1)p_i+C_j]+C_i(V-W)-2dV \leq |B_i|(V-W)p_i$ or $(V+W)[(|A_j|+1)p_i+C_j-p_j+p_i]+C_i(V-W)-2dV) \leq |B_i|(V-W)p_i$, then S^1 is not worse than S.

Proof. Consider a sequence S^2 which is the same as S except the position of job i is moved immediately after the job j so that $C_i^2=C_j+p_i$. Then, the difference of objective function values of S and S^2 is $Z(S)-Z(S_2)=|B_i|(V-W)p_i-|A_j|(V+W)p_i+(2d-C_i+C_j+p_i)V+(C_i-C_j-p_i)W$. Therefore, if $(V+W)[(|A_j|+1)p_i+C_j]+C_i(V-W)-2dV \leq |B_i|(V-W)p_i$, then $Z(S) \geq Z(S^2)$.

Consider other sequence S^3 which is the same as S except the position of job i is moved immediately before the job j so that $C_i^3=C_j-p_j+p_i$. Then, the difference between objective function values of S and S^3 is $Z(S)-Z(S^3)=|B_i|(V-W)p_i-|A_j|(V+W)p_i-(C_j-p_j+p_i)(V+W)-C_i(V-W)+2dV$. Therefore, if $(V+W)[(|A_j|+1)p_i+C_j-p_j+p_i]+C_i(V-W)-2dV \leq |B_i|(V-W)p_i$, then $Z(S) \geq Z(S^3)$. The V-shaped sequence S^1 is obtained by sorting the each early and tardy job set in S^2 or S^3, and is then not worse than S.

This completes the proof.

Proposition 8. Consider a sequence S where two jobs i and j are sequenced early and tardy, respectively. And consider another V-shaped sequence S^1 which has the same jobs in each early and tardy job set as S except the job j in tardy set is moved toward early position. Then, if
$(V+W)[(|A_j|p_j+C_j)]+(C_i-p_i)(V-W)-2dV \geq |B_i|(V-W)p_j$ or
$(V+W)[(|A_j|p_j+C_j)]+C_i(V-W)-2dV \geq (|B_i|+1)(V-W)p_j$,
then the sequence S^1 is not worse than S.

Proof. Consider a sequence S^2 which is the same as S except the job j is moved immediately before the job i so that $C_j^2 = C_i - p_i$. Then, the difference of objective values between S and S^2 is $Z(S) - Z(S_2) = |A_j|(V+W)p_j - |B_i|(V-W)p_j + C_j(V+W) + (C_i - p_i)(V-W) - 2dV$.

Therefore, if $(V+W)[(|A_j|p_j + C_j)] + (C_i - p_i)(V-W) - 2dV \geq |B_i|(V-W)p_j$, then $Z(S) \geq Z(S^2)$.

Consider other sequence S^3 which is the same as S except the job j is moved immediately after the job i so that $C_j^3 = C_i$. Then, the difference of objective values between S and S^3 is $Z(S) - Z(S^3) = |A_j|(V+W)p_j - |B_i|(V-W)p_j + C_j(V+W) + C_i(V-W) - p_j(V-W) - 2dV$. Therefore, if $(V+W)[|A_j|p_j + C_j] + C_i(V-W) - 2dV \geq (|B_i|+1)(V-W)p_j$, then $Z(S) \geq Z(S^3)$. And the V-shaped sequence S^1 is not worse than S^2 and S^3.

This completes the proof.

Notice that the job j in tardy position can only be moved to early position when $s - p_j \geq 0$. As commented above, the scheduling problem for the case of $W < V$ is NP-complete. Thus, all the solution properties shall now be put together to derive a heuristic algorithm as follows :

Step 1. Sequence the jobs to be V-shape as an initial sequence.

Step 2. Find the optimal starting time for the current sequence by using Proposition 2, and then calculate the objective value of the resulting schedule.

Step 3.

(1) For each pair of two jobs i and j, $i \in E$ and $j \in T$, apply the results of Propositions 6, 7, and 8 to improve the objective value.

(a) Check if the positions of i and j can be interchanged by using the results of Proposition 6.

(b) Check if the job i can be moved to a tardy position by using the results of Proposition 7.

(c) Check if the job j can be moved to an early position by using the results of Proposition 8.

(2) If there exists any pair of two jobs i and j improving the objective value, then rearrange the sequence maintaining V-shape and then go to Step 2. Otherwise, stop and accept the current sequence as a heuristic solution.

In the algorithm, an initial feasible solution sequence is obtained at Step 1. The initial sequence is constructed to be V-shape by assigning the jobs in early position according to Hall's algorithm[6] until the total processing time of assigned early jobs is less than or equal to d.

A schedule is determined by calculating the optimal starting time at Step 2. Finally, this sequence with its optimal starting time is improved by rearranging the sequence at Step 3. If rearranging is occurred, it is required to modify the starting time to improve the schedule. This step-by-step interactive procedure continues until no further improvement is possible with maintaining V-shape.

5. Computational Experience

The heuristic procedure is coded in PASCAL on the IBM PC/AT, and the test problems are generated with three different due dates of $d=0.2\ TP$, $0.6\ TP$, $1.0\ TP$(TP denoting total processing time) and three different values of $W/V=1/3$, $1/5$, $1/10$. And p_i are randomly generated from a discretely uniform distribution from 1 to 100.

To evaluate the effectiveness of the proposed heuristic algorithm, a lower bound(LB) is developed. It is easily seen that the optimal solution of the $Z(S)$ with unrestricted size on due date but without flow time measuring term provides a lower bound. Similar to Szwarc[11], a lower bound (LB_1) can be calculated by using the optimal objective value of the unrestricted model of $Z(S)$ without considering explicitly the flow time measuring term from the SPT sequencing as

$$LB_1=\begin{cases}(V+W)[k(p_1+p_2)+(k-1)(p_3+p_4)+\cdots+(P_{2k-1}+P_{2k}), & \text{if } n \text{ is odd},\\ (V+W)[kp_1+(k-1)(p_2+p_3)+\cdots+(p_{2k-2}+p_{2k-1}), & \text{if } n \text{ is even}.\end{cases}$$

where k is the longest integer less than or equal to $n/2$.

It can easily be verified that LB_1 is a lower bound for $Z(S^*)$ by result of Proposition 2 since $n/2\le n(V+W)/(2V)<n$ when $W<V$.

And another lower bound(LB_2) can be found from the SPT sequencing as

$$LB_2 = W\sum_{i=1}^{n}(n-i+1)p_i$$

Therefore, the lower bound(LB) is obtained as

$$LB=\text{Min}\{LB_1,\ LB_2\}.$$

To evaluate the effectiveness of the proposed heuristic algorithm, 450 test problems are generated in total, ten problems for each combination of d, W/V, and $n=5$, 8, 10, 12, 15. These problems are examined in terms of the mean relative errors(MRE) over the optimal solution (obtained by a branch-and-bound algorithm separately constructed only for small-sized problem by use of Propositions 1 through 8) shown in Table 1. In the optimal solution search process, the test problem set having any problem required more than 30 minutes to solve is cut off from further consideration.

All the test problems show the effectiveness of the proposed heuristic algorithm, measured for each combination of d, W/V, and n as in Table 1 and average at the MRE, 3.22%. It can be seen from these test results that the error is larger when $d=0.6\ TP$ than other cases, that it gets increasing trend as the W/V becomes larger[smaller] when $d=0.6\ TP$ [$d=0.2\ TP$], and that it gives nondecreasing trend as the problem size n increases. It is noticed that the test results associated with due date $d=0.6\ TP$ may require any better procedure for sequencing, however, the test results associated with $d=0.6\ TP$ are not satisfactory even if a V-shaped sequence for indefinite due date(or sufficiently large due date to be unrestricted model) is considered as its initial solution incorporating the fact of which the V-shape sequencing is optimal when the flow time measuring term is not considered and the due date d is approximately larger than or equal to 0.5 TP(Szwarc[11]).

It is also shown in Table 1 that the MRE's of LB with respect to the optimal solution show decreasing trend as n gets larger.

For the efficiency test and conjecturing the quality of heuristic solution in large problem size n, another set of the total 450 test problems(ten problems for each combination of d, W/V, and $n=20$, 30, 50, 90, 170) are measured in CPU time as listed in Table 2. In the problem set, optimal solutions are not found readily so that sequences with $s=0$ are considered to evaluate the relative superiority of the heuristic result, based on the fact the any(random) sequence with $s=0$ can be feasible to any variation of the given problem with different due dates. The overall mean deviation of all the randomly-generated schedule(sequences) from the heuristic solution is then computed at 29.44%. This implies that the solution properties derived in Sections 3 and 4 may contribute lot to finding a good solution efficiently.

The mean deviation of LB from the heuristic solution, measure with all the test problems of the two set given in Tables 1 and 2, show decreasing trend as n is larger. This further implies that the heuristic algorithm may work well with even larger n. In conclusion, the heuristic algorithm is considerably good for the situations where due date is extremely large or small with computational efficiency as listed average CPU times in Tables 1 and 2.

〈Table 1〉 Mean relative deviations(n=5, 8, 10, 12, 15)

n	d	W/V 1/3	W/V 1/5	W/V 1/10	CPU time(second) optimal	CPU time(second) heuristic
5	0.2 *TP*	0.0031	0.0042	0.0058	0.043	0.014
		0.5625	0.5520	0.5226		
		1.2959	1.2450	1.1106		
	0.6 *TP*	0.0934	0.0534	0.0630	0.066	0.030
		0.3894	0.3513	0.2427		
		0.7984	0.6319	0.4136		
	1.0 *TP*	0.0285	0.0107	0.0137	0.062	0.031
		0.5344	0.4418	0.3020		
		1.2191	0.8163	0.4548		
9	0.2 *TP*	0.0201	0.0284	0.0365	0.796	0.020
		0.5021	0.4877	0.4739		
		1.0505	1.0094	0.9719		
	0.6 *TP*	0.0648	0.0439	0.0424	1.332	0.024
		0.3651	0.2803	0.1902		
		0.6843	0.4544	0.2896		
	1.0 *TP*	0.0060	0.0102	0.0074	0.810	0.042
		0.5354	0.4413	0.3030		
		1.1747	0.8142	0.4475		
10	0.2 *TP*	0.0226	0.0325	0.0857	24.558	0.024
		0.4832	0.4674	0.4310		
		0.9817	0.9415	0.9020		
	0.6 *TP*	0.1108	0.0709	0.0684	63.652	0.024
		0.3312	0.2557	0.1596		
		0.6747	0.4660	0.2750		
	1.0 *TP*	0.0046	0.0037	0.0050	5.580	0.060
		0.5468	0.4531	0.3146		
		1.2462	0.8511	0.4722		

MRE of heuristic solution to optimal solution(top)

MRE of lower bound to optimal solution(middle)

MRE of heuristic solution to lower bound(bottom)

〈Table 1〉 Mean relative deviations(Continued)

n	d	W/V			CPU time(second)	
		1/3	1/5	1/10	optimal	heuristic
12	0.2 *TP*	0.0211	0.0274	0.0337	454.959	0.029
		0.4622	0.4468	0.4318		
		0.8991	0.8575	0.8197		
	0.6 *TP*	0.0572	0.0536	0.0449	696.489	0.030
		0.3178	0.2346	0.1298		
		0.5522	0.3780	0.2156		
	1.0 *TP*	0.0028	0.0092	0.0018	47.217	0.071
		0.5160	0.4228	0.2896		
		1.0806	0.7525	0.4117		
15	0.2 *TP*	0.0579	0.0595	〉	1214.298	0.029
		0.4501	0.4339	〉		
		0.9240	0.8719	0.8245		
	0.6 *TP*	〉	〉	〉		0.040
		〉	〉	〉		
		0.5356	0.3646	0.2092		
	1.0 *TP*	0.0021	0.0041	0.0034	898.323	0.115
		0.5188	0.4255	0.2922		
		1.0858	0.7498	0.4184		

〉 indicating problems requiring more than 30 minutes to solve

MRE of heuristic solution to optimal solution(top)

MRE of lower bound to optimal solution(middle)

MRE of heuristic solution to lower bound(bottom)

〈Table 2〉 Mean relative deviations(n=20, 30, 50, 90, 170)

n	d	W/V 1/3	W/V 1/5	W/V 1/10	average CPU time (second)
20	0.2 *TP*	0.2731	0.282	0.2830	0.037
		0.8703	0.8172	0.7690	
	0.6 *TP*	0.1764	0.2112	0.2480	0.048
		0.5349	0.3616	0.2037	
	1.0 *TP*	0.2429	0.3612	0.4787	0.164
		1.1248	0.7704	0.4278	
30	0.2 *TP*	0.2565	0.2633	0.2698	0.054
		0.8831	0.8255	0.7732	
	0.6 *TP*	0.1842	0.2211	0.2598	0.076
		0.5196	0.3488	0.1933	
	1.0 *TP*	0.2576	0.3766	0.4935	0.332
		1.1309	0.7765	0.4335	
50	0.2 *TP*	0.2707	0.2780	0.2850	0.110
		0.8341	0.7763	0.7238	
	0.6 *TP*	0.1997	0.2404	0.2825	0.154
		0.5190	0.3471	0.1908	
	1.0 *TP*	0.2701	0.3910	0.5071	0.753
		1.1588	0.7947	0.4471	
90	0.2 *TP*	0.2729	0.2802	0.2871	0.251
		0.8311	0.7733	0.7208	
	0.6 *TP*	0.1951	0.2325	0.2716	0.383
		0.5046	0.3367	0.1841	
	1.0 *TP*	0.2694	0.3877	0.5072	2.192
		1.1525	0.7984	0.4430	
170	0.2 *TP*	0.2687	0.2761	0.2831	0.674
		0.8251	0.7673	0.7148	
	0.6 *TP*	0.1928	0.2296	0.2679	1.137
		0.4964	0.3310	0.1807	
	1.0 *TP*	0.2673	0.3910	0.5101	4.987
		1.1612	0.7934	0.4404	

MRE of heuristic solution to initial solution(top)

MRE of heuristic solution to lower bound(bottom)

6. Conclusion

This paper considers a single machine nonpreemptive scheduling problem to find the optimal schedule which minimize the sum of earliness/tardiness and flow time measures for all jobs, where all the jobs are available at time 0.

For the problem, some dominant solution properties are characterized with respect to the relative size of weights W and V. At first, it is found that a SPT sequencing with $s^*=0$ is optimal if $W \geq V$.

For the case of $W<V$, it is characterized that a V-shaped sequence is optimal. However, a heuristic algorithm is developed incorporating derived properties about optimal starting time and improving rules for a current schedule since this problem is NP-complete. The improving rules are composed of a pairwise interchange rule and two moving rules from early to tardy position or vice versa.

The computational experience with the heuristic algorithm shows great potential(preference) with short or long distant due date or large number of jobs.

For further study, it may be extended to a problem with jobs having different weights. And the problem of jobs with different due dates is another subject. Problem with multiple machines may be even more interesting but tough.

References

[1] Baker, K. R. and G. D. Scudder, "Sequencing with Earliness and Tardiness Penalties : A Review," *Operations Research,* Vol. 38(1990), pp. 22-36.

[2] Cheng, T. C. E., "Optimal Common Due-Date with Limited Completion Time Deviation," *Computers and Operations Research,* Vol. 15(1988), pp. 91-96.

[3] Conway, R. W., W. L. Maxwell, and L. W. Millen, *Theory of Scheduling,* Addison-Wesley Publishing Company, 1967.

[4] De, P., J. B. Ghosh, and C. E. Wells, "On the Minimization of Completion Time Variance with a Bicriteria Extension." *Operations Research,* Vol. 40(1992), pp. 1148-1155.

[5] Dickman, B., Y. Wilamowsky, and S. Epstein, "Optimal Common Due-Date with Limited Completion Time," *Computers and Operations Research,* Vol. 18(1991), pp. 125-1271.

[6] Hall, N. G., "Single-and Multiple-Processor Models for Minimizing Completion Time Variance," *Naval Research Logistics,* Vol. 33(1986), p. 49-54.

[7] Hall, N. G., W. Kubiak, and S. P. Sethi, "Earliness-Tardiness Scheduling Problem, Ⅱ : Deviation of Completion times about a Restrictive Common Due Date," *Operations Research,* Vol. 39(1991), pp. 847-859.

[8] Mittenthal, J. and M. Raghavachari, 1993. "Stochastic Single machine Scheduling with Quardratic Early-Tardy Penalties," *Operations Research*, Vol. 41(1993), pp. 786-796.

[9] Panwalkar, S. S., M. L. Smith, and A. Seidmann, "Common Due Date Assignment to Minimize Total Penalty for the One Machine Scheduling Problem," *Operations Research*, Vol. 30(1982), pp. 391-399.

[10] Sung, C. S and U. G. Joo, "A Single Machine Scheduling Problem, with Earliness/Tardiness and Starting Time Penalties under a Common Due Date," *Computers and Operations Research*, Vol. 19(1992), pp. 757-766.

[11] Szwarc, W., "Single-Machine Scheduling to Minimize Absolute Deviation of Completion Times from a Common Due Date," *Naval Research Logistics,* Vol. 36(1989), p. 663-673.