Journal of the Operations Research
Society of Japan
Vol. 36, No. 3, September 1993

HEURISTIC ALGORITHMS FOR A SINGLE-MACHINE COMMON DUE DATE ASSIGNMENT UNDER EARLINESS/TARDINESS MEASURE*

C.S. Sung J.I. Min C.K. Park
Korea Advanced Institute of Science and Technology

(Received November 5, 1991; Final October 26, 1992)

Abstract This paper considers a single-machine scheduling problem with a set of various jobs having different weights where the objective is to find the optimal assignment of a common due date and the associated optimal job sequencing to minimize the weighted mean absolute deviation (WMAD) of job completion times about the common due date. In the problem analysis, several dominant solution properties are characterized to organize two efficient heuristic solution algorithms, for which numerical experiments are then made for illustration and comparative review. Their preferences over the reference works are shown in efficiency and effectiveness senses.

1. Introduction

This paper considers a single–machine scheduling problem where a common due date needs to be assigned and both earliness and tardiness penalties are imposed on violation of not exactly meeting the common due date. It is practically meaningful to incorporate such duedate restrictions in scheduling subject, since due dates are usually set by contract with users (customers) so that any schedule violating such due date contract may incur monetary penalty or severe work delay in series–work environment as especially in assembly line process. In fact, earliness violation can incur unnecessary inventory to carry or opportunity loss. Thus, it is often required to consider both earliness and tardiness penalties in scheduling study. The so–called just–in–time manufacturing piocess control problem may be considered as its immediate application.

There are several recent reference works incorporating both earliness and tardiness penalties in single–machine scheduling problems. Sidney [10] has formulated a single machine scheduling problem with penalties represented by a nondecreasing continuous function of earliness and tardiness measure for which an algorithm was derived to minimize the maximum penalty. Kanet [8] has considered the average deviation of job completion times from a common due date, referred to as the mean absolute deviation (MAD) problem, which has been extended by Hall [7] to the case of parallel identical machines. Emmons [6] has considered the situation where the cost per unit time for completing a job late was not equal to the correspo–

* This work was partly supported by Korean Traders Scholarship Foundation under Grant Number 89–51.

nding earliness penalty. Bagchi et al. [2] have considered a similar problem having the weighted sum of the squared deviations of job completion times from a common due date as its measure and developed a branching procedure for solving the problem.

Min [9] and Cheng [4] have also investigated the single–machine scheduling problem but under the measure of weighted mean absolute deviation (WMAD) between
ob completion time and common due date for jobs having different weights, and characterized the optimal solution properties. In those reference works, two important solution characteristics have been analyzed as follows:

a) One job is completed exactly on the optimal common due date, and
b) The optimal schedule is represented by a weighted V–shape sequence of job processing times with the shape vertex fixed at the optimal common due date in the sense that early jobs (finished prior to due date) are sequenced in weighted longest processing time job first (WLPT) order and tardy jobs (finished after due date) are sequenced in weighted shortest processing time job first (WSPT) order.

Moreover, WMAD problems with the optimal schedules represented by such weighted V–shape sequences can be transformed into not equivalent but similar parallel two–machine weighted mean flow time (WMF) problems. Thus, the associated heuristic algorithm development in this paper is justified in nature.

De et al. [5] have suggested a heuristic approach (called QH) for the problem transformed into a 0–1 quadratic programming where a given full sequence was used as the basis for improvement by trying to reassign every job on early or tardy side of the due date at a better position in the sequence. Thus, the following critical comments can be made on the heuristic method:

a) For job position change, each candidate job selection from among n jobs is made based on its contribution (coefficient) to the objective measure, so that the computational complexity for all the job contributions is in the order of $O(n^2)$,
b) Large memory space is required because all the trying sequences have to be kept along until the solution search stops, and
c) The optimal solution search may not be efficient because each selected candidate job is considered only for its own reassignment but with its associated partial sequence of the remaining (n–1) jobs kept unchanged so that the search speed toward the optimal schedule may slow.

This paper newly suggests a heuristic algorithm to improve the aforementioned QH method by incorporating the following sequencing strategy:

a) Treating the problem as a weighted mean flow time problem (F_w) for which a solution sequence is constructed by selecting candidate jobs (from among the remaining jobs) to add to an intermediate (current) partial sequence individually,
b) Selecting each candidate job in the WSPT order of weighted processing times (p_j/w_j), and
c) Assigning such a selected candidate job into either the early or tardy subsequence of the

intermediate partial sequence on the basis of the corresponding contribution preference. This strategy incorporation can be expected to make several computational contributions to the proposing algorithm effectiveness and efficiency as follows;

a) Storage space is required only for two (early and tardy) partial sequences during the whole solution search,

b) The computational complexity is only in the order of O(n *log* n) because of sorting p_j/w_j's , and

c) The mechanism of each candidate job selection and assignment is designed to make the slope of the weighted V–shape sequence deeper around the associated common due date so that the better solution can be searched.

2. Problem Analysis

For the given scheduling problem, jobs are all nonpreemptive and available for processing at time zero and the scheduling objective is to minimize the weighted mean absolute deviation (WMAD) of job completion times from the due date. This objective measure is selected here to comply with the statement in [2] that, insofar as the cost per unit time for completing each job may be different, the WMAD problem may provide greater flexibility in achieving scheduling objectives than the mean absolute deviation (MAD) problem.

The following notations are now introduced for the problem analysis ;

$N = \{1, 2, \cdots , n\}$ = set of n independent jobs.
p_j= processing time of job j, $j \in N$.
w_j= weighting factor associated with job j, $j \in N$.
c_j= completion time of job j, $j \in N$.
d = common due date.
Π = set of n ! sequences generated with n jobs in the set N.
S = an arbitrary sequence in Π.
E = partial sequence of jobs scheduled to finish before or on due date.
T = partial sequence of jobs scheduled to finish after due date.
$n_1 = |E|$, $n_2 = |T|$, where $|A|$ denotes the number of jobs in A.
$<j>$ = j *th* job in E.
$[j]$ = j *th* job in T.

Then, the problem objective can be described as to find a schedule S that minimizes

$$Z(S) = (\sum_{j \in N} w_j \, |d - c_j| \,) \, / \, (\sum_{j \in N} w_j) \qquad (1)$$

where the denominator is just a normalizing constant. This intends to get jobs as clustered as possible around the due date, possibly with some initial idle times allowed. Note that such a common due date d is often assumed to be large enough to give full freedom in scheduling jobs. Without loss of generality, let the given problem jobs be numbered in order of

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_n}{w_n}. \tag{2}$$

Some dominant solution properties are now characterized accordingly.

Property 1 (P1).

In the optimal schedule, there is no idle time inserted between any two jobs.

Property 2 (P2).

In the optimal schedule, one of the jobs has its completions time exactly at the common due date d.

Property 3 (P3).

A schedule is dominant if its jobs in E are ordered in the WLPT sequence such that

$$\frac{p_{<1>}}{w_{<1>}} \geq \frac{p_{<2>}}{w_{<2>}} \geq \cdots \geq \frac{p_{<n_1>}}{w_{<n_1>}}, \tag{3}$$

and the rest of its jobs are in T and ordered in the WSPT sequence such that

$$\frac{p_{[1]}}{w_{[1]}} \leq \frac{p_{[2]}}{w_{[2]}} \leq \cdots \leq \frac{p_{[n_2]}}{w_{[n_2]}}. \tag{4}$$

Property 4 (P4).

Denoting by d* the optimal common due date and having the relation $|E| = n_1$ so that $d = c_{<n_1>}$, it holds that

$$\sum_{j \in E} w_j \geq (1/2)(\sum_{j \in N} w_j) \text{ and } \sum_{j \in E} w_j - w_{<n_1>} < (1/2)(\sum_{j \in N} w_j). \tag{5}$$

These properties (P1–P4) are proved in Min [9], Cheng [4], and De et al. [5].

Relations (3) and (4) imply that the optimal schedule can be depicted in a V–shape sequence of weighted processing times with the shape vertex fixed at the optimal common due date d*. Therefore, such weighted V–shape sequences only will be paid attention all through the rest of this paper work.

Corollary 1.

The WMAD problem can be transformed into a type of parallel two–machine scheduling problem with the measure of weighted mean flow time (WMF).

Proof. Let E and T correspond to represent the sets of the given problem jobs assigned to two parallel machines in the WSPT orders, respectively, and let $f_i(j)$ be the associated flow time of job j on machine i (where i = 1, 2) in a schedule found under the weighted mean flow time measure.

Suppose that a schedule (S) to the WMAD problem is given as follows;

$$S = (<1>, <2>, \cdots , <n_1>, [1], [2], \cdots , [n_2]) ,$$

where $d = c_{<n_1>}$. In order to consider that the jobs in E are forming a schedule at one of the two parallel machines, the jobs (arranged under the deviation measure from the due date, say $d - c_{<j>}$) in E of the given schedule S should rather be viewed in the reversed order under the weighted mean flow time measure. Thus, for the schedule S,

$$\begin{aligned} Z(S) &= \left[\sum_{j=1}^{|E|} w_{<j>} (d-c_{<j>}) + \sum_{j=1}^{|T|} w_{[j]} (c_{[j]}-d) \right] / (\sum_{j\in N} w_j) \\ &= \left[\sum_{j=1}^{|E|} w_{<j>} f_1(<j>) + \sum_{j=1}^{|T|} w_{[j]} f_2([j]) \right] / (\sum_{j\in N} w_j) \qquad (6) \end{aligned}$$

$$\text{where} \quad f_1(<j>) = \begin{cases} \sum_{h>j} p_{<h>} & 1 \le j \le n_1-1 \\ 0 & j = n_1 \end{cases}$$

$$f_2([j]) = \sum_{h\le j} p_{[h]} \qquad 1 \le j \le n_2$$

Equation (6) implies that the given WMAD problem is written in a type of parallel two–machine WMF problem expression, while each job in E is newly assigned the flow time excluding its own processing time.

Thus, the proof is completed.

By the way, Corollary 1 does not imply that the given WMAD problem is equivalent to a parallel two–machine WMF problem. Therefore, the question "*Is the WMAD problem NP–complete*" is still open, even if the parallel two–machine WMF problem is known NP–complete [10].

Bearing the problem complexity in mind, heuristic solution procedures are pursued for finding good solutions quickly without examining all feasible schedules. They are all exploited based on the dominance schedule properties discussed earlier. It is noticed that the WMAD problem can be solved in polynomial time for each of the following four special cases:

Case 1. Equal weights, $w_j = w$, for all j

Case 2. Equal processing times, $p_j = p$, for all j.

Case 3. Different weights w_e and w_t with respect to the early and late completion jobs, respectively, but they are common for the jobs in the respective sets.

Case 4. Each job weight is represented by its processing time, $w_j = p_j$, for all j.

3. Solution Algorithms

The problem complexity discussion in the preceding section provides the motivation of proposing two new heuristic algorithms for the solution search in this section.

Cheng [4] has suggested a partial search algorithm in a sort of a branch–and–bound approach and De et al. [5] have also investigated two solution algorithms including a branch–and–bound algorithm and a dynamic programming algorithm. The dynamic programming algorithm has been suggested for problems where it was possible to transform problem parameters (processing times or job weights) into integer values. However, these algorithms have been noticed about their solution search burden on computational time and memory space requirement which might be serious as the number of jobs increases. Therefore, De et al. [5] have rather suggested a heuristic algorithm (called QH) where, given a full solution sequence composed of its two subsequences (early–job subsequence and tardy–job subsequence), each job is tried to move out of one subsequence into the other so as to make an improved solution sequence (determined on the basis of the movement decision including its greater contribution to the objective measure), and this sequence improvement continues until no further improvement is possible. This algorithm requires all the n jobs to be tried for their position movement in each of such sequence improvement trial steps so that more than $n^2/2$ moving possibilities may have to be evaluated for the whole improvement procedure, since more than n/2 trial steps may have to be repeated. Moreover, its computational load may become serious as problem size increases, since it is designed to start with its initial solution having all the n jobs assigned on the early–job sequence.

This paper now wants to exploit newly a better heuristic algorithm based on the results of Corollary 1 characterizing that the given WMAD problem can be transformed into a type of parallel two–machine WMF problem.

First, consider the weight parameter which is multiplied by each job processing time in flow time measure problems, and define it to be "position weight". Then, for basic single–machine problems, the position weight of the i *th* job in a flow time measure sequence and that in a weighted flow time measure sequence are represented by (n–i+1) and $\sum_{j=i}^{n} w_{[j]}$, respectively. These imply that in a given sequence, the earlier assigned job has the greater position weight, and accordingly, SPT and WSPT are the optimal sequencing rules for mean flow time and weighted mean flow time problems, respectively. In other words, in a given sequence, each job makes its contribution to the objective measure by its position weight, so that the job with the greater position weight is more contributory than others. This position weight property is adapted in the new algorithm to give the higher job assignment priority to the positioning of the job with the greater position weight in the parallel two–machine WMF sequencing, so that jobs are required to be sequenced in WSPT order at their corresponding machines out of the two machines. Thus, the problem of assigning each job to one of the two machines only remains to solve.

The detailed description of the proposing heuristic procedure is now presented. A complete (full) sequence is found just by adding all the candidate (available for assignment) jobs individually to an intermediate (current) partial sequence for which each candidate job is selected in the SPT order of weighted processing times (p_j/w_j), and two adding procedures (forward method H1 and backward method H2) are tried. In the forward method and the backward method, jobs are added individually in the decreasing order and the increasing order of their p_j/w_j values, respectively. And these two adding procedures are considered here under the following reasonings:

a) At the initiation stage of sequencing, the job with the greatest p_j/w_j may draw attention on its position assignment in the solution sequence due to its objective contribution heavily depending on the choice between the earliness side position and the tardiness side position with which the assignment procedure starts for the rest of the jobs. Likewise, the job with the smallest p_j/w_j may be considered to start with because of a similar objective contribution expected due to its position assignment.

b) For a set of jobs having the larger processing time variance measured on all the n processing times, the position initializations of the job with the greatest p_j/w_j and the job with the smallest p_j/w_j may make the greater impact on their solution contributions, in comparison with others having smaller variances.

This is why this paper gives higher priority to the sequencing of jobs with larger position weights over those of other jobs, and hence selects the job with the largest p_j/w_j first from among all the candidate jobs at the stage of each sequential position assignment in the solution search.

A strong strategic point (great beneficial) in this heuristic is simply to have each candidate job assigned the right next position to the corresponding intermediate partial sequence without changing any job position in the partial sequence so that only one sequence is carried on to update until being ended up with the final solution sequence. However, all the aforementioned reference works have considered the way of changing all the positions of jobs involved in a full sequence, which can incur many instances of additional position changes for each job once it is position–assigned. This requires "carrying–on all the full sequence informations generated since the beginning of the procedure".

Now, the following notations are introduced to characterize the proposed approach:

S_p = partial sequence of p jobs in a weighted V–shape schedule

$E(S_p)$ = subsequence of early jobs in the partial sequence S_p

$T(S_p)$ = subsequence of tardy jobs in the partial sequence S_p

$Z(S_p)$ = objective value of the partial sequence S_p

Then, let's consider the procedure where the job with the greatest p_j/w_j is assigned first. According to Property 3, the procedure can be characterized as making each adding job assigned to a position closer to the due date in a sequence than its preceding job positions. Therefore, the processing job is moved one position further from the due date, since the adding job has p_j/w_j smaller than the processing one. Based on this job assignment rule, Theorem 1 specifies the conditions for each adding job to be assigned to a better position on between earliness and tardiness sides.

Theorem 1.

Let S_{k-1} be a partial sequence of jobs $\{1, 2, \cdots, k-1\}$ scheduled, and S_k' and S_k'' denote the partial sequences with job k added to S_{k-1} on earliness and tardiness sides, respectively. If $\sum_{j\in E(S_{k-1})} w_j \leq \sum_{j\in T(S_{k-1})} w_j + w_k$, then it holds that $Z(S_k') \leq Z(S_k'')$. Otherwise, it holds that $Z(S_k') > Z(S_k'')$.

Proof. According to Property 3, the partial sequence S_{k-1} has (k–1) jobs some being sequenced in WLPT order on its earliness side and the others in WSPT order on its tardiness side. Therefore, the newly adding job k is assigned a position (on either one of earliness and tardiness sides) closer to the due date than any of the (k–1) earlier– assigned jobs, since job k has p_k/w_k smaller than that of any of the (k–1) jobs.

If job k is assigned on earliness side, then it has its completion time exactly on the due date and each of the jobs previously assigned on earliness side has a new completion time pushed away from the due date by p_k time, while all the jobs assigned on tardiness side remain at their original positions without any completion time change. Therefore, the objective measure of the partial sequence S_k' with job k assigned on earliness side can be expressed as

$$Z(S_k') = Z(S_{k-1}) + p_k \sum_{j \in E(S_{k-1})} w_j . \tag{7}$$

On the other hand, if job k is assigned on tardiness side, then it has its starting time on the due date and each of the jobs previously assigned on tardiness side newly has its completion time pushed away from the due date by p_k time. Therefore, the objective measure of the partial sequence S_k'' with job k assigned on tardiness side can be written as

$$Z(S_k'') = Z(S_{k-1}) + p_k \Big(\sum_{j \in T(S_{k-1})} w_j + w_k \Big) . \tag{8}$$

From these two measures of $Z(S_k')$ and $Z(S_k'')$, the result follows.

Thus, the proof is completed.

Theorem 1 describes where each job (selected in a non–increasing order of p_j/w_j values for its position assignment in the solution sequence) can be assigned on earliness or tardiness side. Moreover, for each job position assignment, the job weight informations are required only about the jobs selected earlier such that job k is assigned on earliness side if it holds that $\sum_{j \in E(S_{k-1})} w_j \leq \sum_{j \in T(S_{k-1})} w_j + w_k$, while job k is assigned on tardiness side if it holds that $\sum_{j \in E(S_{k-1})} w_j > \sum_{j \in T(_{k-1})} w_j + w_k$. Based on this property, the H1 procedure is constructed to process in forward approach. In addition, H1 requires only the comparison between two marginal job assignment effects (measure contribution) on earliness and tardiness sides for each selected job until all the n jobs are assigned in the solution sequence, so that its computational complexity is in the order of O(n *log* n). This implies that it searches for a solution efficiently than the heuristic QH. Moreover, it is more effective, since it is based on the utilization of Property 4 in each selected job assignment.

Heuristic 1 (H1 : Forward Method)

Step 0. Set at $E = \phi$, $T = \phi$, and $k = 0$.

Step 1. Set at $k = k + 1$.

Step 2. If $\sum_{j \in E} w_j \leq \sum_{j \in T} w_j + w_k$, then assign the last position in E to job k.
Otherwise, assign the first position in T to job k.

Step 3. If $k = n$, then Stop.
Otherwise, go to Step 1.

Let's now consider another procedure where the job with the smallest p_j/w_j is assigned first. In the procedure, a job to be added to an intermediate partial sequence is assigned a position outside the sequence so as to get away from the associated due date, since it has p_j/w_j larger than any of the jobs assigned earlier. This sequencing rule (procedure) can characterize each job assignment condition as given in Theorem 2.

Theorem 2

Let S_{k-1} be a partial sequence of jobs $\{n-k+2, n-k+3, \cdots, n\}$ scheduled, and S_k' and S_k" denote the partial sequences with job $(n-k+1)$ added to S_{k-1} on earliness and tardiness sides, respectively. If $\sum_{j \in E(k-1)} p_j \leq \sum_{j \in T(k-1)} p_j + P_{n-k-1}$, then it holds that $Z(S_k') \leq Z(S_k'')$. Otherwise, it holds that $Z(S_k') > Z(S_k'')$.

Proof: According to Property 3, the partial sequence S_{k-1} has $(k-1)$ jobs some being sequenced in WLPT order on its earliness side and the others in WSPT order on its tardiness side. Therefore, the newly adding job $(n-k+1)$ is assigned a position (on either one of earliness and tardiness sides) farther away from the due date than any of the $(k-1)$ earlier-assigned jobs, since job $(n-k+1)$ has P_{n-k-1}/W_{n-k-1} larger than that of any of the $(k-1)$ jobs.

If job $(n-k+1)$ is assigned on earliness side, then it has its completion time being away from the due date by $\sum_{j \in E(S_{k-1})} p_j$, while all the $(k-1)$ earlier-assigned jobs remains at their original positions without any completion time change. Therefore, the objective measure of the partial sequence S_k' with job $(n-k+1)$ assigned on earliness side can be expressed as

$$Z(S_k') = Z(S_{k-1}) + W_{n-k-1} \sum_{j \in E(S_{k-1})} p_j . \tag{9}$$

On the other hand, if job $(n-k+1)$ is assigned on tardiness side, then it has its completion time being away from the due date by $\sum_{j \in T(S_{k-1})} p_j + P_{n-k-1}$ time, while each of the $(k-1)$ earlier-assigned jobs remains at its original completion time. Therefore, the objective measure of the partial sequence S_k" with job $(n-k+1)$ assigned on tardiness side can be written as

$$Z(S_k'') = Z(S_{k-1}) + W_{n-k-1} \left(\sum_{j \in T(S_{k-1})} p_j + P_{n-k-1} \right) . \tag{10}$$

From these two measures of $Z(S_k')$ and $Z(S_k'')$, the result follows.

Thus, the proof is completed.

Theorem 2 describes that all the n jobs can be assigned proper positions in a sequence only by considering their processing times to select candidate jobs in a non–decreasing order of p_j/w_j values and assigning each job position on either earliness or tardiness side with respect to their marginal contributions (effects) to the objective measure. Specifically, job (n–k+1) is assigned on earliness side if it holds that $\sum_{j\in E(S_{k-1})} p_j \le \sum_{j\in T(S_{k-1})} p_j + P_{n-k-1}$, while job (n–k+1) is assigned on tardiness side if it holds that $\sum_{j\in E(S_{k-1})} p_j > \sum_{j\in T(S_{k-1})} p_j + P_{n-k-1}$. Based on this property, the H2 procedure is constructed to process in backward approach. This procedure also requires only the comparison between two marginal job assignment contributions on earliness and tardiness sides for each selected job until all the n jobs are assigned in the solution sequence, so that its computational complexity is in the order of O(n *log* n). This implies that it also searches for a solution efficiently than the heuristic QH. Moreover, it is expected from Theorem 2 that the H2 procedure may keep the job position assignment as balanced as possible between earliness and tardiness sides and hence be effective.

Heuristic 2 (H2 : Backward Method)

```
Step 0.   Set at $E = \phi$ , $T = \phi$ , and $k = n+1$ .
Step 1.   Set at $k = k - 1$ .
Step 2.   If $\sum_{j\in E} p_j \le \sum_{j\in T} p_j + p_k$, then assign the first position in E to job k.
          Otherwise, assign the last position in T to job k.
Step 3.   If $k = 1$ , then Stop.
          Otherwise, go to Step 1.
```

4. Computational Experiment

The effectiveness and efficiency of the proposed solution algorithms is evaluated on randomly generated job set of different sizes, in comparison with the dynamic programming algorithm and the heuristic method referred as QH in De et al. [5]. The dynamic programming algorithm and the three heuristic algorithms (including heuristic QH) are tested on HP 3000 Series Workstation computer using Pascal language to solve various problems.

The processing times (p_j) are generated using a uniform random number generator for integers ranged in interval (1, 100), and similarly, the job weights (w_j) are generated in interval (1, 100) and (1, 50). For each job set, 10 test problems are randomly generated.

The effectiveness, e_H, of heuristic algorithm H ($H \in \{QH, H1, H2\}$) is individually measured as $e_H = 100\,(Z_H - Z_o)/Z_o$, where Z_H and Z_o represent the heuristic and the optimal values of the weighted mean absolute deviation of job completion times, respectively.

The results of the computational experiments are shown in Table 1 where for each of the three heuristics (QH, H1, H2) the minimum, median, and maximum gaps between its heuristic solution and the optimal solution are separately listed in percentage value at each job size.

As seen from Table 1, the proposed two algorithms H1 and H2 perform better than the algorithm QH in the effectiveness sense (comparing with the optimal solution). They also outperform in the efficiency measure, since they are characterized to require only 2n times of marginal effect comparisons in searching for a full solution sequence by assigning all the n jobs individually in an simple adding process but the algorithm QH requires $n^2/2$ times of such marginal effect comparisons.

Various comparisons between H1 and H2 can be made one against the other. As discussed earlier, there is the difference between the two in the first job assignment that H1 and H2 algorithms are processed always to assign jobs with the largest and the smallest p_j/w_j first, respectively, on earliness side. And, for problems of jobs with large variances on their processing times, H1 may find better solutions than H2, since H1 assigns the job with the largest p_j/w_j on earliness side and the additional reasons are described as follows:

a) If the job with the largest p_j/w_j is assigned on earliness side, then its processing time can not make any contribution to the objective measure (even if its processing time is relatively large), and

Table 1. Comparative evaluation of different heuristics.

p_j U(1, 100)	w_j U(1, 100)	QH	H1	H2
		3.954	0.868	1.363
n = 5		5.665	1.805	2.060
		7.605	4.281	4.833
		2.291	0.991	1.102
n = 30		3.633	1.635	1.671
		5.284	2.775	2.973
		1.808	0.348	0.674
n = 60		1.939	0.764	0.897
		2.171	0.968	1.012
		0.936	0.329	0.353
n = 100		1.318	0.429	0.449
		1.595	0.515	0.686
p_j U(1, 100)	w_j U(1, 50)	QH	H1	H2
		2.127	0.718	0.847
n = 5		5.414	2.537	2.587
		9.564	5.035	6.092
		2.402	0.762	0.768
n = 30		3.551	1.982	1.697
		6.493	2.707	2.785
		1.290	0.403	0.548
n = 60		1.795	0.635	0.775
		2.719	1.012	1.128
		0.888	0.380	0.400
n = 100		1.222	0.511	0.429
		1.572	0.579	0.545

b) If the job with the largest p_j/w_j , on the other hand, is assigned on tardiness side, then its processing time can make large contribution to the objective measure.

For problems of jobs with small variances on their processing times (where the processing time difference between the two jobs, one having the largest p_j/w_j and the other having the smallest p_j/w_j , is relatively small), a similar comparison can be made as follows:

a) The processing time of the job with the smallest p_j/w_j can be relatively important in the contribution sense.

b) In the job position effect, the job with the largest position weight (or with the smallest p_j/w_j) may make better contribution than the job with the largest p_j/w_j , since the job with the smallest p_j/w_j is assigned a position near to the due date and hence its processing time affects other jobs. In the situation, if the job with the smallest p_j/w_j is assigned on earliness side (as done in H2), then it makes its contribution to the objective measure by the amount of its processing time multiplied by the portion of the position weight excluding its job weight. On the other hand, if it is assigned on tardiness side, then it makes its contribution to the objective measure by the amount of its processing time multiplied by the portion of the position weight including its job weight.

Therefore, H2 may find better solutions than H1, since H2 assigns the job with the smallest p_j/w_j on earliness side.

In summary, for problems of jobs with large variances on their processing times, H1 algorithm (assigning the job with the largest p_j/w_j on earliness side) may find better solutions than H2 algorithm. On the other hand, for problems of jobs with small variances on their processing times, H2 algorithm (assigning the job with the smallest p_j/w_j on earliness side) may find better solutions than H1 algorithm. However, in these H1 and H2 algorithms, each job position assignment is made based on its objective contribution measured with respect to the associated intermediate partial sequence rather than a full sequence, which might be a weak point. Furthermore, the two algorithms are designed to start with the first job assignment made only on earliness side, while they can similarly be tried with tardiness side for the first job assignment. Nevertheless, there are two strong practical remarks to make. One is that H1 and H2 algorithms are effective and easy to adapt to real–time environment, since their computation time requirements are relatively negligible. And the other one is that they may even solve large size problems on personal computer, since their memory (storage) space requirements are almost negligible.

5. Conclusive Remarks

This paper considers a problem of sequencing a given set of jobs on a single machine and assigning a common due date. The objective is to minimize the weighted mean absolute deviation (WMAD) of job completion times about the common due date. This WMAD problem is expressed in not an equivalent but a type of parallel two–machine scheduling problem with the measure of weighted mean flow time. Therefore, this paper exploits two heuristic algorithms based on some dominance schedule properties. And the computational experiment shows that both the heuristics are effective and efficient.

The results of this study can immediately be applied to process–scheduling problems incorporating a class of product parts (jobs) with a common due date (to be assigned) in manufacturing industry, and also to scheduling problems of loading trade commodities on a ship with their shipping date pre–scheduled in transportation business.

For further study, the problem may be extended to incorporate the situations where two classes of jobs are involved with the associated two common due dates or where job arrivals occur dynamically.

Acknowledgements

The authors thank the editor and anonymous referees for their invaluable comments.

References

[1] Bagchi, U., Sullivan, R.S., and Chang, Y.L., "Minimization Mean Absolute Deviation of Completion Times about a Common Due Date," *Naval Research Logistics Quarterly,* **33,** 227–240 (1986).

[2] Bagchi, U., Sullivan, R.S., and Chang, Y.L., "Minimizing Absolute and Squared Deviations of Completion Times with Different Earliness and Tardiness Penalties and a Common Due Date," *Naval Research Logistics Quarterly,* **34,** 739–751 (1987).

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

[4] Cheng, T.C.E., "A Note on a Partial Search Algorithm for the Single–Machine Optimal Common Due–Date Assignment and Sequencing Problem," *Computers and Operations Research,* **17,** 321–324 (1990).

[5] De, P., Ghosh, J.B., and Well, C.E., "CON Due–Date Determination and Sequencing," *Computers and Operations Research,* **17,** 333–342 (1990).

[6] Emmoms, H., "Scheduling to a Common Due Date on Parallel Uniform Processors," *Naval Research Logistics Quarterly,* **34,** 803–810 (1987).

[7] Hall, N.G., "Single– and Multi– Processor Models for Minimizing Completion Time Variance," *Naval Research Logistics Quarterly,* **33,** 49–54 (1986).

[8] Karnet, J.J., "Minimizing the Average Deviation of Job Completion Times about a Common Due Date," *Naval Research Logistics Quarterly,* **28,** 643–651 (1981).

[9] Min, J.I., *A Single Machine Scheduling Problem with Earliness and Tardiness Penalties under Common Due Date.* Unpublished Master Thesis, Korea Advanced Institute of Science and Technology, February 1989.

[10] Sidney, J.B., "Single Machine Scheduling with Earliness and Tardiness Penalties," *Operations Research,* **25** (1), 62–69 (1977).

Chang Sup SUNG:
Department of Industrial Engineering,
Korea Advanced Institute of Science and Technology,
Taejon 305–701,
Korea