Single machine scheduling to minimize the setup time and the earliness

Chin-Chia Wu*

Wen-Chiung Lee

Department of Statistics
Feng Chia University
Taichung 407
Taiwan
R.O.C.

Abstract

We consider the problem of scheduling customer orders on a single machine. It is assumed that each order consists of several jobs that can be clustered into several groups. A setup time is only required when processing switches from jobs of one group to jobs of another group. This problem is considered with the total setup time and the total earliness as measures of performance. A numerous of elimination properties are established, and a branch-and-bound algorithm is proposed to identify all feasible schedules for the bi-criteria problem.

Keywords : *Setup time, earliness, single machine.*

Introduction

In the traditional production process, most research assumes that the setup time is sequence independent and thus includes the setup time in the processing time. However, in many practical production environments, setup time may depend on the current job as well as the one last processed. Hence, there is still some research dealing with the sequence-dependent setup time. In particular, they consider the situation that jobs are classified into several groups, and a setup task is only

* *E-mail*: cchwu@fcu.edu.tw

Journal of Information & Optimization Sciences
Vol. 27 (2), No. 2, pp. 499–510
 0252-2667/06 $2.00 + 0.25

required if jobs of one group switch to jobs of another group. Therefore, a frequent changeover of jobs between groups might result in high production cost. On the other hand, jobs completed earlier than their due dates must be held until the delivery takes place. Thus, earliness reflects the fact that there may be a penalty for finishing activities before they are due. That is, high inventory can cause the high cost of holding inventory, the requirement of valuable space for storage, the premature use of valuable capacity, and deterioration costs of finished good inventory. In this paper, we consider a single-machine scheduling problem to minimize both the total setup time and the total earliness simultaneously.

In the literature, there are only a few attempts to tackle with single-machine scheduling problems involving the setup time, since they are closely related to the salesperson problems. Monma and Potts [1] address about the complexity of scheduling problems with setup times. They also show that under the structure of "triangle inequality" of the setup time, minimizing the maximum completion time, maximum lateness, total weighted completion time, or the number of late jobs can be efficiently solved for a fixed number of classes. Ahn and Hyun [2] modify Psaraftis's [3] dynamic programming algorithm to minimize the mean flow time. Besides that, they also develop an efficient heuristic method for solving large-sized problems. Later, Mason and Anderson [4] consider the average weighted flow time, and develop branch-and-bound method for this problem.

Cheng and Kahlbacher [5] present a study related to the earliness, which is one of the criteria we will consider. However, they assume that all jobs in the same batch are delivered together upon the completion of the last job in the batch. Thus, the earliness of a job is defined as the difference between the delivery time of the batch to which it belongs and the job completion time. Their objective is to find simultaneously the optimal number of batches and the optimal job sequence so as to minimize the sum of delivery and earliness costs. They show that this problem is NP-hard, but under the condition of equal unit earliness penalties, they develop an efficient algorithm to solve the problem. Webster and Baker [6] provide a review of the works on scheduling group of jobs on a single-machine to minimize the total weighted flowtime or the maximum lateness.

Feo et al. [7] use the Greedy Randomized Adaptive Search Procedure method to develop an efficient heuristic algorithm for the problem with sequence-dependent setup costs and linear delay penalties. They show that their method is competitive with the tabu search and successful for problems up to 165 jobs. Pan and Su[8] study a single-machine scheduling problem with sequence-independent setup times to minimize the maximum lateness of jobs. They derive several properties and develop a branch-and-bound method to solve this problem. Meanwhile, Ghosh and Gupta[9] address the same problem using a dynamic programming method. Liaee and Emmons[10] deal with scheduling families of jobs with sequence-dependent setup times on single or parallel machines, with or without the assumption that the jobs in each family must be scheduled together. Their objectives are the total completion time, the maximum completion time, the maximum lateness, the weighted number of late jobs or the maximal cost which is a nondecreasing function of the completion time.

Cheng et al.[11] study the problem of scheduling jobs on a single-machine where a constant setup time is required before the first job of each batch is processed. Their objective is to find an optimal number of batches and a schedule to minimize the sum of the mean batch delivery time and total weighted job earliness, where the earliness is the same as the one which Cheng and Kahlbacker[5] consider. They prove this problem is strongly NP-hard. They also give a dynamic programming method and a heuristic algorithm for the problem.

Liao and Liao[12] consider the problem of minimizing the mean flow time, where jobs can be grouped into different classes and these classes can be further grouped into different families. There are a major setup and a minor setup when jobs are switched from one family to another there is only a minor setup when jobs are switched from one class to another within the same family. They provide a dynamic programming and a heuristic method for this problem.

The works mentioned above is either a single criterion or a linear combination of several criteria. To the best of our knowledge, Liao and Chung[13] are the first authors to include the due date and to study the bi-criteria problems. They consider two problems of scheduling customer orders on a single machine where each order consists of jobs that can be clustered into several groups. One problem is to minimize the total setup time and the number of tardy orders. The other is to minimize the total setup time and the maximum tardiness. They propose branch-and-bound algorithms to identify the set of non-dominated schedules. According to the classification, the problem we study here is also a bi-criteria problem.

Problem formulation

Consider a set of L orders to be processed on a single machine one at a time where each order consists of several jobs that can be clustered into m groups according to the similarity of their production requirements. It is assumed that n jobs are to be processed and available at time zero. It is also assumed that each order cannot have more than one job in a group. In addition, job-splitting and machine idle time are not permitted. A setup time is required when processing switches from jobs of one group to jobs of another group. Each order has an associated due date. It is assumed that jobs in the same order must be shipped together. Hence, a job which is completed before its corresponding order due date must be held in the shop until the order delivery takes place.

Let $C_{[i]}(S)$ be the completion time of the job scheduled in the ith position of sequence S and $E_{[i]}(S)$ denote its earliness. That is, $E_{[i]}(S) = \max\{0, d_l - C_{[i]}(S)\}$ where the ith processed job is in order l and dl is its associated due date. Let $TS(S)$ be the total setup time and $TE(S)$ be the total earliness of sequence S. A sequence S is said to be feasible with respect to total setup time and earliness criteria if there does not exist another sequence S' such that

$$TS(S') \leq TS(S) \quad \text{and} \quad TE(S') \leq TE(S)$$

with at least one of the above holding as a strict inequality. If there is such a sequence S', then we say that S' dominates S and S is an infeasible sequence. The objective of this paper is to find all the feasible sequences with respect to the setup time and the earliness on a single machine. It is denoted as $n/1$, $g/(TS, TE)$.

Properties

In this section, we state some essential properties that will be used later in the branch-and-bound algorithm. Let S and S' be two sequences, and the difference between these two sequences is a pairwise interchange of two adjacent jobs J_x and J_y. That is, $S = (\alpha J_x J_y \beta)$ and $S' = (\alpha J_y J_x \beta)$, where α and β denote subsequences of S and S'. Suppose that job J_x

and J_y have processing time p and q, setup time s_u and s_v, respectively. Besides that, job J_x belongs to order s and job J_y belongs to order t. It is also assumed that J_w is the last processing job in subsequence α with a completion time of A, and J_z is the first job processed in subsequence β. According to whether job J_x and J_y are completed before their due dates in S and S', we divide the situations into the following six cases.

Case 1. We consider a situation in which the corresponding due dates of job J_x and J_y satisfy that $d_s \geq C_{[k]}(S)$ and $d_t \geq C_{[k+1]}(S)$ in S. Meanwhile, the corresponding due dates also satisfy that $d_t \geq C_{[k]}(S')$ and $d_s \geq C_{[k+1]}(S')$ in S'.

Proposition 1a. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 1, S *dominates* S' *if* J_x *and* J_y *are in the same group, and* $p > q$.

Proof. Since J_x and J_y belong to the same group, it follows that

$$C_{[k]}(S) = A + s_u I_{\{J_w, J_x\}} + p\,,$$
$$C_{[k+1]}(S) = A + s_u I_{\{J_w, J_x\}} + p + q\,,$$
$$C_{[k]}(S') = A + s_v I_{\{J_w, J_y\}} + q\,,$$
$$C_{[k+1]}(S') = A + s_v I_{\{J_w, J_y\}} + q + p\,,$$

where $I_{\{J_x, J_y\}} = 0$ if J_x and J_y are in the same group. Otherwise, $I_{\{J_x, J_y\}} = 1$. Since $s_u I_{\{J_w, J_x\}} = s_v I_{\{J_w, J_y\}}$, it implies

$$C_{[k+1]}(S) = C_{[k+1]}(S')\,.$$

It is clear that the total setup times are equal in both sequences. To calculate the difference of the total earliness between sequence S and S', we only need to take the earliness of job J_x and J_y into consideration. Thus, we have

$$[E_{[k]}(S) + E_{[k+1]}(S)] - [E_{[k]}(S') + E_{[k+1]}(S')] = q - p < 0\,.$$

Therefore, S dominates S'. □

Proposition 1b. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 1, S *dominates* S' *if* J_w, J_x *and* J_y *are in different groups, so are* J_x, J_w *and* J_z, *and* $s_v + q < s_u + p$.

Proposition 1c. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 1, S *dominates* S' *if* J_x, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups, and* $s_v + s_u + q < p$.

Proposition 1d. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 1, S *dominates* S' *if* J_y, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups and* $q < s_u + s_v + p$.

Case 2. We consider a situation in which the corresponding due dates of job J_x and J_y satisfy that $d_s \geq C_{[k]}(S)$ and $d_t \geq C_{[k+1]}(S)$ in S. However, $d_t \geq C_{[k]}(S')$ but $d_s < C_{[k+1]}(S')$ in S'. While in S', the kth job is completed before its due date, but the $(k+1)$th job is completed after its due date.

Proposition 2a. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 2, S *dominates* S' *if* J_x *and* J_y *are in the same group, and* $d_s < A + s_u I_{\{J_w, J_x\}} + 2p$.

Proposition 2b. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 2, S *dominates* S' *if* J_w, J_x *and* J_y *are in different groups, so are* J_x, J_y *and* J_z, *and* $d_s < A + 2s_u + 2p$.

Proposition 2c. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 2, S *dominates* S' *if* J_x, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups, and* $d_s < A + 2p$.

Proposition 2d. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 2, S *dominates* S' *if* J_y, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups, and* $d_s < A + 2s_u + s_v + 2p$.

Case 3. We consider a situation in which the due dates of job J_x and J_y satisfy that $d_s \geq C_{[k]}(S)$ and $d_t < C_{[k+1]}(S)$ in S. Meanwhile, the due dates of job J_y and J_x also satisfy that $d_t \geq C_{[k]}(S')$ and $d_s < C_{[k+1]}(S')$ in S'.

Proposition 3a. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 3, S *dominates* S' *if* J_x *and* J_y *are in the same group, and* $d_s - p < d_t - q$.

Proposition 3b. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 3, S *dominates* S' *if* J_w, J_x *and* J_y *are in three different groups, so are* J_x, J_y *and* J_z, *and* $d_s - (s_u + p) < d_t - (s_v + q)$.

Proposition 3c. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 3, S *dominates* S' *if* J_x, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups, and* $d_s - p < d_t - (s_v + q)$.

Proposition 3d. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 3, S *dominates* S' *if* J_y, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups and* $d_s - (s_u + p) < d_t - q$.

Case 4. We consider a situation in which the due dates of job J_x and J_y satisfy that $d_s < C_{[k]}(S)$ and $d_t < C_{[k+1]}(S)$ in S. Meanwhile, the due dates of job J_y and J_x satisfy that $d_t \geq C_{[k]}(S')$ and $d_s < C_{[k+1]}(S')$ in S'.

Proposition 4a. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 4, *if* J_x *and* J_y *are in the same group, then* S *dominates* S'.

Proposition 4b. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 4, *if job* J_w, J_x *and* J_y *are in three different groups, and* J_y, J_x *and* J_z *are also in different groups, then* S *dominates* S'.

Proposition 4c. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 4, *if job* J_x, J_w, *and* J_z *are in the same group but* J_x *and* J_y *are in different groups, then* S *dominates* S'.

Proposition 4d. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 4, *if job* J_y, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups, then* S *dominates* S'.

Case 5. We consider a situation in which the due dates of job J_x and J_y satisfy that $d_s \geq C_{[k]}(S)$ and $d_t < C_{[k+1]}(S)$ in S. Meanwhile, the due dates of jobs J_y and J_x also satisfy that $d_t \geq C_{[k]}(S')$ and $d_s \geq C_{[k+1]}(S')$ in S'.

Proposition 5a. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 5, S *dominates* S' *if* J_x *and* J_y *are in the same group, and* $d_t > A + s_v I_{\{J_w, J_y\}} + 2q$.

Proposition 5b. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 5, S *dominates* S' *if* J_w, J_x *and* J_y *are in three different groups, so are* J_y, J_x *and* J_z, *and* $d_t > A + 2s_v + 2q$.

Proposition 5c. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 5, S *dominates* S' *if* J_x, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups and* $d_t > A + s_u + 2s_v + 2q$.

Proposition 5d. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 5, S *dominates* S' *if* J_y, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups and* $d_t > A + 2q$.

Case 6. We consider a situation that the due dates of job J_x and J_y satisfy that $d_s < C_{[k]}(S)$ and $d_t \geq C_{[k+1]}(S)$ in S. Meanwhile, the due dates of job J_y and J_x also satisfy that $d_t \geq C_{[k]}(S')$ and $d_s < C_{[k+1]}(S')$ in S'.

Proposition 6a. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 6, *if* J_p *and* J_q *are in the same group, then* S *dominates* S'.

Proposition 6b. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 6, *if* J_w, J_x *and* J_y *are in different groups, and* J_y, J_x *and* J_z *are also in different groups, then* S *dominates* S'.

Proposition 6c. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 6, *if* J_x, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups, then* S *dominates* S'.

Proposition 6d. *For the* $n/1$, $g/(TS, TE)$ *problem, under* Case 6, *if* J_y, J_w *and* J_z *are in the same group, but* J_x *and* J_y *are in different groups, then* S *dominates* S'.

The search procedure for feasible solutions can speed up if the schedule of the remaining jobs can be easily determined. Let PS denote the partial sequence in which the order of jobs processed has been determined and GS denote the subsequence in which the remaining unscheduled jobs are arranged group by group, starting from the same group as of the last job in PS if it exists.

Proposition 7. *If all the due dates of the unscheduled jobs are less than the completion time of the last job in* PS, *then sequence* $S = (PS, GS)$ *dominates sequence of the type* $(PS, -)$ *with respect to total setup time and total earliness.*

Since the due dates of the remaining jobs are less than the completion time of the last job in PS, the unscheduled jobs are all tardy no matter how to arrange them. Thus, the total earliness remains the same. In order to reduce the setup time, the remaining jobs must be processed group by group so that setup task occurs once. Moreover, if we start from the same group as of the last job in PS, then the setup time for that group can even be omitted, and it is certainly less than the total setup time of other arrangement.

Lower bounds

In this section, we provide lower bounds that will be used later in the branch-and-bound algorithm to curtail the size of the branching tree. Suppose that the order of jobs in partial sequence PS has been determined and we wish to calculate lower bounds on the total setup time and total earliness of the remaining unscheduled jobs. This can then be used to calculate the lower bounds for the complete sequence.

To calculate a lower bound of the setup time, we first notice that the setup time of the remaining group is needed once if the jobs are processed group and group. Clearly, if there are jobs which belongs to the same group as of the last job in PS, then they should be scheduled immediately after subsequence PS, eliminating the additional setup time for that group. Thus, the lower bound can be expressed as the sum of the setup time in partial sequence PS, and the setup time calculated from the unscheduled jobs if they are processed group by group excluding the setup time for the group of the last job in PS if it exists.

To compute a lower bound of the total earliness for the remaining jobs, we first add the associated setup time to the processing time of each job, and derive their completion time by applying the longest processing time rule. Next, among those corresponding due dates of the unscheduled jobs, we choose the smallest one, and denote it by $d_{\min}$. If we assume all the remaining jobs have a due date $d_{\min}$, then the total earliness calculated is clearly a lower bound. Thus, the lower bound for the complete sequence can be obtained by adding the total earliness of the partial sequence PS.

Computational results

In order to evaluate the performance of the proposed algorithm, a computational experiment is conducted. The algorithm is coded in Fortran 77 and run on a 586 personal computer. The experimental design follows Liao and Chung's[13] framework. The job processing times are generated from a discrete uniform distribution between 11 and 30, and the setup time are also generated from another uniform distribution on the integers between 1 and 10. The due date of order i is randomly selected from a uniform distribution on the integers between TP_i and $TP_i + C * TP$, where TP_i is the total processing time of jobs in order i, TP is the total processing time of all jobs, and C is a constant. In order to test the effect of the due date, C takes on the values of 0.4, 0.6 and 0.8 in the experiments. In addition, to test the effect of the number of orders and the number of groups, we use two types of job patterns as shown in Figure 1. There are six orders and four groups in Data Set I and five orders and six groups in Data Set II.

$$\begin{pmatrix} & J_1 & J_2 & J_3 \\ J_4 & J_5 & J_6 & \\ & J_7 & J_8 & J_9 \\ J_{10} & J_{11} & J_{12} & \\ & J_{13} & J_{14} & J_{15} \\ J_{16} & J_{17} & J_{18} & \end{pmatrix} \qquad \begin{pmatrix} J_1 & & J_7 & & J_{13} & \\ J_2 & & J_8 & & J_{14} & \\ J_3 & J_4 & J_9 & J_{10} & J_{15} & J_{16} \\ & J_5 & & J_{11} & & J_{17} \\ & J_6 & & J_{12} & & J_{18} \end{pmatrix}$$

Data Set I Data Set II

Figure 1
The job patterns for two data sets

The algorithm is tested over three different problem size of $n = 12, 15$ and 18. As a consequence, 18 experiment conditions are examined and 10 random problems are generated for each condition. We present the summary of our findings in Table 1, stating the average and the maximum time required by each procedure, the average and the maximum number of feasible solutions in each problem, and the average and the maximum number of nodes in each branching tree. In general, Data Set II required more computational effort than Data Set I, and the difference becomes more significant when the value of C or the problem size is larger. For both of the data sets, the number of feasible solutions, the number of branching nodes and the computational time increase as the value of C becomes larger. This is because the larger value of C makes the due date of each order larger, and implicitly weakens the dominance criteria, especially Proposition 7. Therefore, the problem becomes harder to solve. In addition, the problem is also harder solve as the problem size is larger. Since the larger the problem size, the larger the total processing time of all jobs. Hence, the values of the due dates also become larger, since they are generated from a discrete uniform distribution with a range depending on the total processing time of all jobs.

Conclusions

In this article, we have considered a single machine problem with respect to minimizing both the total setup time and the total earliness. We have derived some elimination properties for each different condition, and have developed a branch-and-bound algorithm to identify the feasible solutions for the problem. However, the branch-and-bound procedure can provide optimal solutions only for small-and-medium-sized problems. Therefore, one possible extension of our work is to develop a heuristic procedure for large-sized problems. Another possible extension is to consider the problem with machine idle time allowed.

Table 1
Computational results for the TS and TE problem

Data Set	No. of jobs	C	Computation Time Mean	Computation Time Max	Feasible Sol. Mean	Feasible Sol. Max	Branching nodes Mean	Branching nodes Max
I	12	0.4	0.081	0.222	4.8	7	2742	5795
		0.6	0.915	5.889	8.4	13	18927	121224
		0.8	0.773	2.060	11.3	21	18401	39469
	15	0.4	0.286	0.501	7.1	12	8447	15448
		0.6	2.264	10.686	8.6	13	39735	137445
		0.8	57.960	515.000	13.8	22	655487	5335624
	18	0.4	1.511	4.105	7.4	13	20923	49410
		0.6	22.224	76.505	13.4	21	254694	759099
		0.8	2146.816	17283.730	22.0	32	23580586	210073669
II	12	0.4	0.051	0.166	5.7	11	1820	6055
		0.6	0.828	4.727	8.0	15	18098	128825
		0.8	1.404	4.750	14.0	21	36162	131710
	15	0.4	1.393	3.639	11.0	17	23229	57301
		0.6	3.927	13.807	13.9	20	56772	191555
		0.8	210.412	1849.653	19.0	36	1820790	15171355
	18	0.4	26.363	225.575	14.5	24	272650	2281939
		0.6	22.889	97.493	19.7	31	282053	1272637
		0.8	11352.219	84789.000	26.2	42	40997672	302471434

References

[1] C. L. Monma and C. N. Potts (1989), On the complexity of scheduling with batch setup times, *Operations Research*, Vol. 37, pp. 798–804.

[2] B. Ahn and J. Hyun (1990), Single facility multi-class job scheduling, *Computers & Operations Research*, Vol. 17, pp. 265–272.

[3] H. N. Psaraftis (1980), A dynamic programming approach for sequencing groups of identical jobs, *Operations Research*, Vol. 28, pp. 1347–1359.

[4] A. J. Mason and E. J. Anderson (1991), Minimizing flow time on a single machine with job classes and setup times, *Naval Research Logistics*, Vol. 38, pp. 333–350.

[5] T. C. E. Cheng and H. G. Kahlbacher (1993), Scheduling with delivery and earliness penalties, *Asia-Pacific Journal of Operational Research*, Vol. 10, pp. 145–152.

[6] S. Webster and R. K. Baker (1995), Scheduling groups of jobs on single machine, *Operational Research*, Vol. 47, pp. 175–180.

[7] T. A. Feo, K. Sarathy and J. McGahan (1996), A grasp single machine scheduling with sequence dependent setup costs and linear delay penalties, *Computers & Operations Research*, Vol. 23 (9), pp. 881–895.

[8] J. C. H. Pan and C. S. Su (1997), Single machine scheduling with due dates and class setups, *Journal of the Chinese Institute of Engineers*, Vol. 20 (5), pp. 561–572.

[9] J. B. Ghosh and J. N. D. Gupta (1997), Batch scheduling to minimize maximum lateness, *Operations Research Letters*, Vol. 21, pp. 77–80.

[10] M. M. Liaee and H. Emmons (1997), Scheduling families of jobs with setup times, *International Journal of Production Economics*, Vol. 51, pp. 165–176.

[11] T. C. E. Cheng, M. Y. Kovalyov and B. M. T. Lin (1997), Single machine scheduling to minimize batch delivery and job earliness penalties, *SIAM Journal Optimal*, Vol. 7, pp. 547–559.

[12] C. J. Liao and L. M. Liao (1997), *Single facility scheduling with major and minor setups, Computers & Operations Research*, Vol. 24 (2), pp. 169–178.

[13] C. J. Liao and C. H. Chung (1996), Sequencing with setup time and order tardiness trade-offs, *Naval Research Logistics*, Vol. 43, pp. 971–984.

Received August, 2005