KYBERNETIKA — VOLUME *48* (2012), NUMBER 2, PAGES 346–356

BATCH SCHEDULING PROBLEM WITH DUE-DATE AND FUZZY PRECEDENCE RELATION

Xuesong Li, Hiroaki Ishii and Minghao Chen

A single-machine batch scheduling problem is investigated. Each job has a positive processing time and due-date. Setup times are assumed to be identical for all batches. All batch sizes cannot exceed a common upper bound. As in many practical situations, jobs have to be subject to flexible precedence constraints. The aim of this paper is to find an optimal batch sequence. The sequence is to minimize the maximal completion time and maximize the minimum value of desirability of the fuzzy precedence. However, there usually exists no batch sequence optimizing both objectives at a time. Therefore, we seek some non-dominated batch sequences after the definition of non-dominated batch sequence. Based on an iterative Procedure HL proposed by Cheng et al., an efficient algorithm is presented to find some non-dominated batch sequences.

Keywords: single-machine, batch scheduling, modified due-date, fuzzy precedence relation, non-dominated batch sequence

Classification: 90B35, 90C29, 90C70, 68Q25

1. INTRODUCTION

In batch scheduling problems, jobs are grouped (each group is called batch) and scheduled in batches, and a setup time is incurred when starting a new batch. Batch availability is assumed here, i. e. the completion time of the batch is the completion time of the final job in the batch. Depending on the calculation of the length of a batch, two types of batching problems exist, denoted by p-batching problems and s-batching problems [1]. For p-batching problems the length of a batch is equal to the largest processing time among all jobs in the batch, while for s-batching problems the length is sum of the processing times of all jobs in the batch.

Till now, there exist many researches on a batch problem [2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 15, 16, 17]. There are also survey papers [13, 14]. This paper treats one model of s-batch problem which considers a single machine batch scheduling problem with due-date and fuzzy precedence constraints. Section 2 formulates the ordinary precedence relation case. Modified due-date is used to make a sequence which is compatible with the precedence constraints. Section 3 proposes an efficient solution procedure for the ordinary precedence relation problem, which is based on the Procedure HL [4] to partition the job sequence into batches. Section 4 presents a numerical example to illustrate how the solution procedure runs. Section 5 formulates the fuzzy version of a single machine batching scheduling problem, i. e. the fuzzy precedence relation, as the main problem of this paper. An efficient algorithm is proposed to solve the bi-criteria problem, i. e. minimize the maximal completion time and maximize the minimum value of desirability of the fuzzy precedence. Since usually there exist no batch sequence optimizing both objectives at a time, we seek some non-dominated batch sequences after the definition of non-dominated batch sequence by the algorithm. Also a numerical example is presented to illustrate how the algorithm runs. Section 6 summarizes results in this paper and discusses further research problems.

2. ORDINARY PRECEDENCE RELATION CASE

There are n simultaneously available jobs $\{J_1, J_2, \ldots, J_n\}$ to be scheduled non-preemptively for processing on a single machine in batches. The machine is continuously available from $t = 0$, and can process only one job at a time. Each job J_i has a positive processing time P_i and due-date d_i. A sequence- and batch- independent setup time, denoted by s,is incurred whenever a batch is formed. Setup times are assumed to be non-anticipatory. All batch sizes cannot exceed a common upper bound b. Jobs have to be subject to a set of precedence constraints, PC. A precedence relation, $J_i \prec J_j$ in PC implies that job J_i must be completed before job J_j starts to be processed. The problem is to find the optimal solution consisting of a batch number and allocation of jobs to batches minimizing the maximum completion time under a common limited batch size.

Under the above setting, the following single machine scheduling problem is considered:

$$
\begin{aligned}
P\colon\quad & \text{Minimize} \quad && C_{\max} = \sum_{j=1}^{n} p_j + ks \\
& \text{Subject to} && \sum_{i=1}^{k} |B_i| = n, \\
& && |B_i| \le b, \quad i = 1, \ldots, k, \\
& && C_j \le d_j, j \in B_i, \quad i = 1, \ldots, k,
\end{aligned}
$$

where jobs are subject to PC, $|B_i|$ denotes the number of jobs in batch B_i, C_j is the completion time of job $J_j,\ j \in B_i,\ i = 1, \ldots, k$ and $C_{\max}$ is the maximum completion time. Let

$$T_i = \{J_j | J_i \prec J_j\} \tag{1}$$

be a job set consisting of jobs that J_i precedes.

Lawler and Moore [8] have shown that there exists a feasible schedule that completes each job until its modified due-date under the precedence relation if and only if there exists a feasible schedule using modified due-date $d_i^{'}$ defined as below without precedence relation.

$$d_i^{'} = \min\{d_i, \min\{d_j | J_j \in T_i\}\}, \quad i = 1, \ldots, n. \tag{2}$$

Note that if all processing times are positive, a sequence ordered as non-decreasing modified due-dates is compatible with the precedence constraints. Without loss of generality, we assume that jobs are indexed in the non-decreasing order of modified due-dates such that

$$d_1^{'} \le d_2^{'} \le \cdots \le d_n^{'} \tag{3}$$

where $p_i \le p_{i+1}$ if $d_i^{'} = d_{i+1}^{'}$ $(i = 1, \dots, n-1)$ and in this indexing if $J_i \prec J_j$, then $i < j$.

Now the problem P is reduced to the problem $P^{'}$ which finds the optimal batch number for the sequence arranged by the modified due-dates.

$$P^{'}: \text{ Minimize } \quad C_{\max} = \sum_{i=1}^{k} \sum_{j \in B_i} p_j + ks$$

$$\text{Subject to } \quad \sum_{i=1}^{k} |B_i| = n,$$
$$|B_i| \le b, \quad i = 1, \dots, k,$$
$$C_j \le d_j^{'}, j \in B_i, \quad i = 1, \dots, k.$$

3. SOLUTION PROCEDURE FOR ORDINARY PRECEDENCE RELATION CASE

Assume that the setup time s and job processing times p_i $(i = 1, \dots, n)$ are fixed. In this case, Cheng et al. [4] suggested the Procedure HL to partition the optimal job sequence into batches. In this paper, **Algorithm 1** is proposed based on the Procedure HL to solve problem $P^{'}$.

Outline of Algorithm 1. The first batch is initiated with the setup time only. At the beginning of iteration $j, j = 1, \dots, n$, jobs $1, \dots, j-1$, are assumed to have been assigned into batches. Job j is assigned as follows. If the constraints of problem $P^{'}$ can be satisfied with the addition of j to the last batch, then do so. Otherwise, if job j can be completed under the constraints of problem $P^{'}$ by starting a new batch, then do so. If neither is possible, then no feasible schedule exists.

Algorithm 1.

Step 0. Set $j = 1, k = 1, B_i = \{J_1\}$. Go to Step 1.

Step 1. If the current batch B_k does not include any predecessor of job j and all jobs can be completed till their modified due-dates by adding job j to B_k, then go to Step 2. Otherwise, go to Step 5.

Step 2. If the completion time of the current batch B_k does not exceed the modified due-dates by adding job j to B_k, then go to Step 3. Otherwise, go to Step 5.

Step 3. If the number of jobs in the current batch B_k does not exceed the upper bound b, then go to Step 4. Otherwise, go to Step 5.

Step 4. Set $B_k \leftarrow B_k \cup \{j\}$. If $j = n$, terminate. Otherwise return to Step 1 after setting $j = j + 1$.

Step 5. Set $k = k + 1$. If $k = n + 1$, terminate as no feasible batch sequence exists. Otherwise return to Step 1 after setting $j = j + 1$.

Theorem 1. A schedule constructed by above **Algorithm 1** is optimal for problem $P^{'}$. Time complexity of the algorithm is $O(n^2)$.

Proof. Let S be a schedule constructed by the algorithm and let S^* be an optimal schedule. Assume that both schedules coincide until $J_1, J_2, \dots, J_{i-1}$. Then we have two situations as shown in Figure 1 and Figure 2. We have $d_i^{'} \leq d_j^{'}$ which follows from the fact that in S job j is scheduled according to (3).

Case 1. J_j and J_i are in the same batch. Let $i_1, \ldots, i_k$ be all jobs scheduled in S^* between J_j and J_i. Furthermore, assume that these jobs are ordered according to starting time. If exchange i with j, there is a feasible schedule $S^{'}$ again. Furthermore, $S^{'}$ is also optimal because $d_i^{'} \leq d_j^{'}$. S and $S^{'}$ coincide till $J_1, J_2, \ldots, J_i$. Now we seek next different job in $S^{'}$ from S. If that job cannot be found, it means that S is also an optimal batch sequence. Otherwise, check case 1 or case 2 as below and continue this process after a finite number of steps, an optimal schedule which coincides with S is obtained.

Case 2. J_j and J_i are in different batches. According to the rule which constructs schedule S, we can obtain that $i < j$ and $d_i^{'} \leq d_j^{'}$. Therefore, the only possibility should be considered is the case $p_i \leq p_j$. Let u be the finishing time of batch h. So if exchange the place of J_j with J_i in S^*, there is a feasible schedule $S^{'}$ again. Furthermore, $S^{'}$ is again optimal because $u \leq d_i^{'} \leq d_j^{'}$ where u is the completion time of the batch including J_i in S^*. Further S and $S^{'}$ coincide till $J_1, J_2, \ldots, J_i$. Now search next different job in $S^{'}$ from S. If that job cannot be found, it means that S is also an optimal batch sequence. Otherwise, check case 1 or case 2 and continue this process after a finite number of steps, an optimal schedule which coincides with S is obtained.

Note that the calculation of modified due-dates is $O(n)$ computational time and that of constructing $T_i = \{J_j | J_i \prec J_j\}$ is $O(n^2)$ computational time. By recording the value of the completion time of the last job and the value of earliest deadline in the current batch, it requires $O(n)$ time. In total, the above algorithm solves the problem in $O(n^2)$ computational time. $\square$

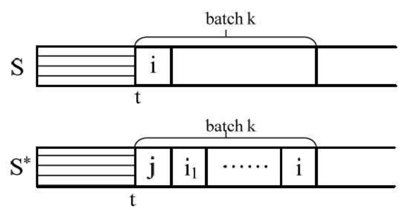

Fig. 1. Schedules for J_j and J_i in the same batch.

4. NUMERICAL EXAMPLE

Example 1 below demonstrates the use of **Algorithm 1**.

Example 1. Assume $n = 5, s = 1, b = 2$. Job set is $\{A, D, E, F, G\}$.

$$p_A = 1, p_E = 2, p_D = 3, p_G = 2, p_F = 5. d_A = 15, d_E = 7, d_D = 21, d_G = 21, d_F = 18.$$

The precedence relations among jobs are shown as Figure 3 where vertices are jobs and arcs denote precedence relations, i. e. $A \prec E, E \prec D, E \prec G, G \prec F, D \prec F$.

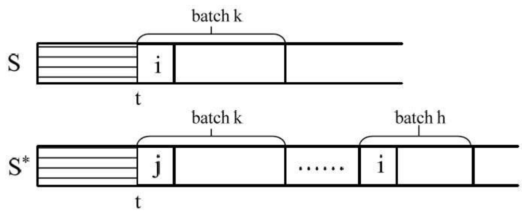

Fig. 2. Schedules for J_j and J_i in the different batches.

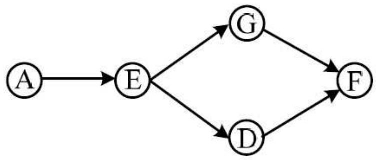

Fig. 3. Precedence relations among jobs.

From (1) we obtain: $T_A = \{E, D, G, F\}, T_E = \{D, G, F\}, T_D = \{F\}, T_G = \{F\}$.

From (2) we obtain: $d_A^{'} = 7, d_E^{'} = 7, d_D^{'} = 18, d_G^{'} = 18, d_F^{'} = 18$.

Then from (3) we obtain: $A \prec E \prec G \prec F$, $A \prec E \prec D \prec F$ and G, D are independent.

Through **Algorithm 1**, the optimal schedule is achieved as Figure 4 shows.

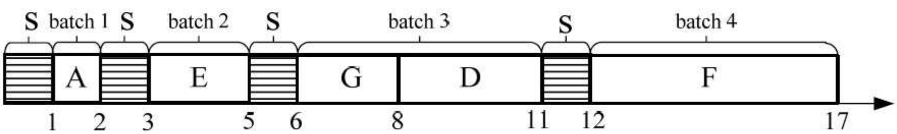

Fig. 4. Optimal schedule.

The optimal $C_{\max}$ is 17.

5. PROBLEM WITH FUZZY PRECEDENCE RELATION

In this section, the fuzzy version of a single machine batching scheduling problem is considered. An efficient algorithm is proposed to solve the bi-criteria problem, i. e. minimize the maximal completion time and maximize the minimum value of desirability of the fuzzy precedence. Then a numerical example is presented to demonstrate how the algorithm runs.

5.1. Problem formulation

There exists one machine and n jobs, $J_1, J_2, \ldots, J_n$ to be processed on this machine. Processing time P_j and due-date d_j are associated with each job J_j and they are all positive integers. Further a fuzzy precedence relation is given between every pair of two jobs. This relation is denoted with the membership function μ_{ij} for all pairs of two jobs, J_i and J_j, which denotes the degree of desirability that J_i is processed before J_j. Assume that if $0 < \mu_{ij} < 1$ then $\mu_{ji} = 1$, and in this case we allow both jobs are in the same batch. If $\mu_{ij} = 1$,$\mu_{ji} = 0$ it means that J_i must precedes J_j, that is, J_i must be scheduled in a batch before that of J_j. Both μ_{ij} and $\mu_{ji} = 1$ means J_i and J_j are independent. Let C_i denote completion time of J_i, that is, the completion time of the batch containing J_i. Further let $\pi(i)$ denote the ith job index of schedule π. Then C^{π}_{max} is defined to be maximum completion time of schedule π and $\mu^{\pi}_{\min} = \min\{\mu_{\pi(i)\pi(k)} | i, k = 1, 2, \ldots, n, i < k\}$ as the minimum value of desirability of the fuzzy precedence in π.

Under the above setting, we consider the following bi-criteria scheduling problem FP:

$$\begin{aligned} \text{FP:}\quad & \text{Minimize} && C^{\pi}_{\max} = \sum_{i=1}^{k} \sum_{j \in B^{\pi}_i} p_j + ks \\ & \text{Maximize} && \mu^{\pi}_{\min} \\ & \text{Subject to} && \sum_{i=1}^{k} |B^{\pi}_i| = n, |B^{\pi}_i| \le b,\ i = 1, \ldots, k, C_j \le d'_j, j \in B^{\pi}_i,\ i = 1, \ldots, k, \end{aligned}$$

where π is feasible batch sequence (feasible schedule), k is the batch number of schedule π and if $|B^{\pi}_i| = n_i,\ i = 1, \ldots, k$, then $B^{\pi}_1 = (\pi(1), \ldots, \pi(n_1)), \ldots, B^{\pi}_i = (\pi(n_{i-1}) + 1, \ldots, \pi(n_i)), \ldots, B^{\pi}_k = (\pi(n_{k-1}) + 1, \ldots, \pi(n))$.

Generally speaking, there may not be a schedule that optimizes both criteria, $C^{\pi}_{\max}$ and $\mu^{\pi}_{\min}$ at a time. Thus, we seek non-dominated schedules defined as below.

First define schedule vector ν^{π} as a vector consisting two elements, i. e., $C^{\pi}_{\max}$ and $\mu^{\pi}_{\min}$ in some feasible schedule π, that is, $\nu^{\pi} = (C^{\pi}_{\max}, \mu^{\pi}_{\min})$. For two vectors $\nu^1 = (\nu^1_1, \nu^1_2)$ and $\nu^2 = (\nu^2_1, \nu^2_2)$, ν^1 dominates ν^2 and denote it by $\nu^1 \le \nu^2$ when $\nu^1_1 \le \nu^2_1$, $\nu^1_2 \ge \nu^2_2$ and $\nu^1 \ne \nu^2$. If $\nu^{\pi_1} \le \nu^{\pi_2}$ for two schedules π_1 and π_2, π_1 dominates π_2. A feasible schedule π is called to be non-dominated if and only if there exists no feasible schedule π' which dominates π.

5.2. Solution procedure for FP

Sorting $0 < \mu_{ij} < 1$, let the result be $\mu^0 \triangleq 1 > \mu^1 > \mu^2 > \cdots > \mu^q > 0$ where q is the number of different μ_{ij}. The precedence relation is usually described by the precedence graph $PG(V, A)$, where V is constructed from the job vertices, i. e. $J_i,\ i = 1, 2, \ldots, n$, and A is the set of arcs (J_i, J_j) which represents $J_i \prec J_j$, i. e. J_i precedes J_j. When J_i and J_j are independent, there does not exist any arc between them. In the solution algorithm, precedence graph $PG^0(V, A^0)$ consists of vertex set V and arc set
$A^0 \triangleq \{(J_i, J_j) | \mu_{ij} = \mu^0 \text{ and } \mu_{ji} \ne \mu^0\}$.

Further, $\overline{A}^l \triangleq \{(J_i, J_j) | \mu_{ij} = \mu^0 \text{ and } \mu_{ji} = \mu^l\}, l = 1, 2, \ldots, q$, and define $PG^l(V, A^l)$ where $A^l = A^{l-1} - \overline{A}^l, l = 1, 2, \ldots, q$. Let DV be the current set of non-dominated schedule vectors and DS the current set of schedules corresponding to each vector of DV. Following is the description of the solution procedure to find non-dominated schedules of FP.

Algorithm 2 for Non-dominated Schedules of FP.

Step 0. Let $l = 0, \mu^0 = 1$ and construct $PG^0(V, A^0)$. From $PG^0(V, A^0)$, make a corresponding sequence by [8] and execute Sub-algorithm. Let corresponding optimal $C_{\max}$ be $C^0_{\max}$ and optimal schedule π^0. Set $DV \leftarrow \{C^0_{\max}, 1\}, DS \leftarrow \{\pi^0\}$, and $l \leftarrow 1$, and go to Step 1.

Step 1. From $PG^l(V, A^l)$, make a corresponding sequence by [8] and execute Sub-algorithm. Let corresponding optimal $C_{\max}$ be $C^l_{\max}$ and optimal schedule be π^l. Construct corresponding schedule vector $\nu^l = (C^l_{\max}, \mu^l_{\min})$ where $\mu^l_{\min} = \min\{\mu_{\pi^l(i)\pi^l(j)} | i, j = 1, 2, \ldots, n, i < j\}$. If ν^l is dominated by some vector of DV or already included in DV, then go to Step 2. Otherwise, set $DV \leftarrow DV \cup \{\nu^l\}$ and $DS \leftarrow DS \cup \{\pi^l\}$. Go to Step 2.

Step 2. Set $l \leftarrow l + 1$. If $l = q + 1$, terminate. Otherwise, return to Step 1.

Sub-algorithm.

Step 0. Set $j = 1, B^{\pi^l}_1 = \{J_{\pi^l(1)}\}$, and go to Step 1.

Step 1. If there exists $i < j$ such that $\mu_{\pi^l(i)\pi^l(j)} = 1, \mu_{\pi^l(j)\pi^l(i)} = 0$, then go to Step 2. Otherwise, go to Step 3.

Step 2. If the current batch $B^{\pi^l}_k$ does not include any predecessor of job $\pi^l(j)$ and all jobs can be completed till their modified due-dates by adding job $\pi^l(j)$ to $B^{\pi^l}_k$, then go to Step 3. Otherwise, go to Step 6.

Step 3. If the completion time of the current batch $B^{\pi^l}_k$ does not exceed the modified due-dates by adding job $\pi^l(j)$ to $B^{\pi^l}_k$, then go to Step 4. Otherwise, go to Step 6.

Step 4. If the number of jobs in the current batch $B^{\pi^l}_k$ does not exceed the upper bound b, then go to Step 5. Otherwise, go to Step 6.

Step 5. Set $B^{\pi^l}_k \leftarrow B^{\pi^l}_k \cup \{\pi^l(j)\}$. If $j = n$, terminate and go back to Main Algorithm. Otherwise, return to Step 2 after setting $j = j + 1$.

Step 6. Set $k = k + 1, B^{\pi^l}_k = \phi$, and return to Step 1.

Theorem 2. A schedule constructed by above **Algorithm 2** is non-dominated schedule for problem FP. The complexity of the algorithm is at most $O\{n^4\}$ computational time.

Proof. Validity is clear since each optimal schedule for a fixed precedence relation corresponding graph $PG^l(V, A^l)$ is a candidate of non-dominated batch schedules. First note that $q = O(n^2)$ and so time complexity of **Algorithm 2** is $O(n^2)\times$ complexity of sub-algorithm. Further, the sub-algorithm treats a fixed precedence relation case. Therefore the proof is similar to the proof of Theorem 1. Only difference is that the job pair (J_i, J_j) of originally fuzzy precedence relation (either $0 < \mu_{ij} < 1$ or $0 < \mu_{ji} < 1$) can be scheduled in a same batch if it is better. So total time complexity is $O(n^2) \times O(n^2) = O(n^4)$. □

5.3. Numerical example

Example 2 below demonstrates the use of **Algorithm 2**.

Example 2. Assume $n = 6, s = 1, b = 4, p_1 = 20, p_2 = 15, p_3 = 12, p_4 = 10, p_5 = 8, p_6 = 9, d_1 = 78, d_2 = 95, d_3 = 78, d_4 = 78, d_5 = 55, d_6 = 48$.

Fuzzy precedence constraints:

$\{J_1, J_2\}, \mu_{12} = 1.00, \mu_{21} = 0,$ $\{J_1, J_3\}, \mu_{13} = 1.00, \mu_{31} = 0.58,$
$\{J_1, J_4\}, \mu_{14} = 1.00, \mu_{41} = 0.63, \{J_2, J_3\}, \mu_{23} = 1.00, \mu_{32} = 0.50,$
$\{J_2, J_4\}, \mu_{24} = 1.00, \mu_{42} = 0.80, \{J_3, J_4\}, \mu_{34} = 1.00, \mu_{43} = 0.70.$

The algorithm goes as follows:

First obtain $q = 5$ and $\mu^0 \triangleq 1 > \mu^1 > \mu^2 > \mu^3 > \mu^4 > \mu^5 > 0$ where $\mu^1 = \mu_{42} = 0.80$, $\mu^2 = \mu_{43} = 0.70$, $\mu^3 = \mu_{41} = 0.63$, $\mu^4 = \mu_{31} = 0.58$, $\mu^5 = \mu_{32} = 0.50$. Also a precedence graph $PG^0(V, A^0)$ is constructed (see Figure 5(a)).

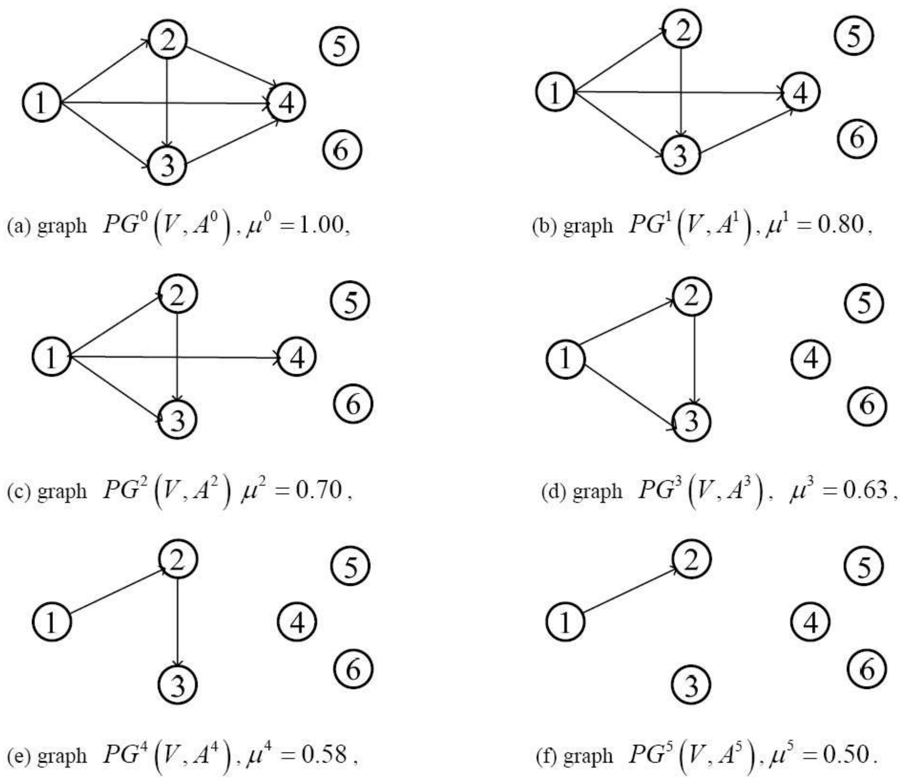

Fig. 5. Solution procedure.

From (1) we obtain: $T_1 = \{J_2, J_3, J_4\}$, $T_2 = \{J_3, J_4\}$, $T_3 = \{J_4\}$.
From (2) we obtain: $d_1' = 78, d_2' = 78, d_3' = 78, d_4' = 78, d_5' = 55, d_6' = 48$.

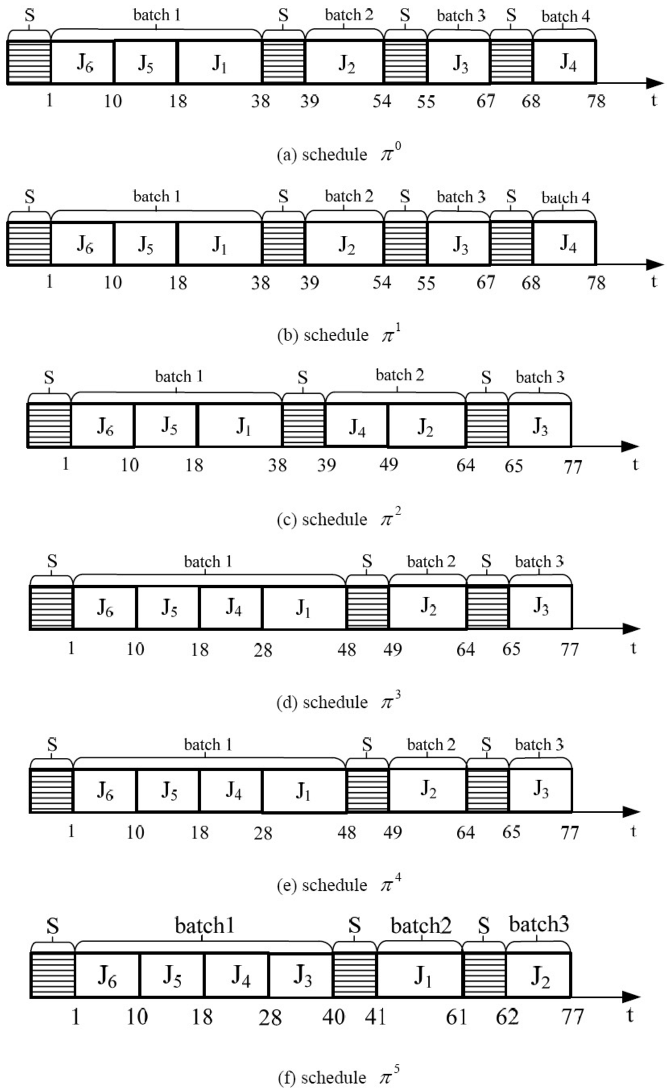

Fig. 6. Feasible schedules.

Then from (3) we obtain: $J_6 \prec J_5 \prec J_1 \prec J_2 \prec J_3 \prec J_4$ with $\mu^0_{\min} = 1.00, C^0_{\max} = 78$ and π^0 (see Figure 6(a)).

At each iteration of the algorithm, the corresponding solutions are:

If $\mu_0 = 1.00$, then π^0 (see Figure 6(a)) with $\mu^0_{\min} = 1.00, C^0_{\max} = 78$;
If $\mu_1 = 0.80$, then π^1 (see Figure 6(b)) with $\mu^1_{\min} = 0.80, C^1_{\max} = 78$;
If $\mu_2 = 0.70$, then π^2 (see Figure 6(c)) with $\mu^2_{\min} = 0.70, C^2_{\max} = 77$;
If $\mu_3 = 0.63$, then π^3 (see Figure 6(d)) with $\mu^3_{\min} = 0.63, C^3_{\max} = 77$;

If $\mu_4 = 0.58$, then π^4 (see Figure 6(e)) with $\mu^4_{\min} = 0.58, C^4_{\max} = 77$;
If $\mu_5 = 0.50$, then π^5 (see Figure 6(f)) with $\mu^5_{\min} = 0.50, C^5_{\max} = 77$.

Note that schedule π^1 is deleted because it is dominated by schedule π^0, and schedules π^3, π^4, π^5 are deleted because they are dominated by schedule π^2. The two remaining schedules constitute the set of non-dominated solutions in this example.

6. CONCLUSION

This paper has proposed an algorithm for a single machine batch scheduling problem with due-date and fuzzy precedence constraints. Modified due-date is introduced to break the precedence relations among jobs. **Algorithm 2** is based on the modified Procedure HL to solve problem. However, since sub-algorithm should not be solved from the scratch, its complexity may be improved. This problem should be extended more to the case of fuzzy due-date and fuzzy precedence. We are now attacking the extended problem.

(Received June 16, 2011)

REFERENCES

[1] P. Brucker: Scheduling Algorithm. Springer-Verlag, Berlin 2006.

[2] P. Brucker, A. Gladky, H. Hoogeveen, M. Y. Kovalyov, C. N. Potts, T. Tautenhahn, and S. V. Velde: Scheduling a batching machine. J. Scheduling *1* (1998), 31–54.

[3] T. C. E. Cheng and A. Janiak: Single machine batch scheduling with deadline and resource dependent processing times. Oper. Res. Lett. *17* (1995), 243–249.

[4] T. C. E. Cheng, A. Janiak, and M. Y. Kovalyov: Single machine batch scheduling with resource dependent setup and processing times. European J. Oper. Res. *135* (2001), 177–183.

[5] E. G. Coffman Jr., M. Yannakakis, M. J. Magazine, and C. Santos: Batch sizing and job sequencing on a single machine. Ann. Oper. Res. *26* (1990), 135–147.

[6] G. Dobson, U. S. Karmarkar, and J. L. Rummel: Batching to minimize flow times on one machine. Management Sci. *33* (1987), 784–799.

[7] D. S. Hochbaum and D. Landy: Scheduling with batching: Minimizing the weighted number of tardy jobs. Oper. Res. Lett. *16* (1994), 79–86.

[8] E. L. Lawler and J. M. Moore: A functional equation and its application to resource allocation and scheduling problems. Management Sci. *16* (1969), 77–84.

[9] G. Mosheiov and D. Oron: A single machine batch scheduling problem with bounded batch size. European J. Oper. Res. *187* (2008), 1069–1079.

[10] G. Mosheiov and D. Oron: Open-shop batch scheduling with identical jobs. European J. Oper. Res. *187* (2008), 1282–1292.

[11] G. Mosheiov, D. Oron, and Y. Ritov: Minimizing flow-time on a single machine with integer batch sizes. Oper. Res. Lett. *33* (2005), 497–501.

[12] D. Naddef and C. Santos: One-pass batching algorithms for the one-machine problem. Discrete Appl. Math. *21* (1988), 133–145.

[13] C. N. Pottsand and M. Y. Kovalyov: Scheduling with Batching: A review. European J. Oper. Res. *120* (2000), 228–249.

[14] C. N. Pottsand and L. N. Van Wasenhove: Interacting scheduling with batching and lot sizing: a review of algorithm and complexity. J. Oper. Res. Soc. *43* (1992), 395–406.

[15] C. Santos and M. Magazine: Batching in single operation manufacturing systems. Oper. Res. Lett. *4* (1985), 338–343.

[16] D. F. Shallcross: A polynomial algorithm for a one machine batching problem. Oper. Res. Lett. *11* (1992), 213–218.

[17] C. S. Sung and U. G. Joo: Batching to minimize weighted mean flow time on a single machine with batch size restrictions. Comput. and Industr. Engrg. *32* (1997), 333–340.

Xuesong Li, Department of Mathematics, Harbin Institute of Technology, No. 92, West Da-Zhi Street, Harbin, Heilongjiang, 150001. P.Ṙ. China.
e-mail: cedarbeauty@yahoo.cn

Hiroaki Ishii, Department of Mathematical Sciences, Kwansei Gakuin University, Sanda. Japan.
e-mail: ishiihiroaki@kwansei.ac.jp

Minghao Chen, Department of Mathematics, Harbin Institute of Technology, No. 92, West Da-Zhi Street, Harbin, Heilongjiang, 150001. P. R. China.
e-mail: chenmh130264@yahoo.com.cn