BATCH SCHEDULING PROBLEM WITH FUZZY DUE-DATE AND PRECEDENCE CONSTRAINT

HIROAKI ISHII and XUESONG LI
School of Science and Technology\(^1\)
Kwansei Gakuin University
Hyogo, 669-1337, Japan
ishihiroaki@Kwansei.ac.jp

Department of Mathematics, Harbin Institute of Technology\(^2\)
Harbin, 150001, China
cedarbeauty@yahoo.cn

ABSTRACT. A single-machine batch scheduling problem is investigated. Each job has a positive processing time and fuzzy due-date which means flexible completion time. 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 precedence constraints. Our aim is to find an optimal batch sequence minimizing the maximal completion time and maximizing the minimum value of desirability of the fuzzy due-date. But usually there exists no batch sequence optimizing both objectives at a time and so we seek some non-dominated batch sequences after the definition of non-dominated batch sequence. An efficient algorithm is presented which is based on an iterative Procedure HL proposed by Chen et.al (European Journal of Operational Research. 135 (2001)) to find some non-dominated batch sequences.

Keywords: Single Machine Batch Scheduling; Fuzzy Due-Date; Precedence Relation; Non-Dominated Batch Sequence

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. Jobs in the same batch have a same completion time, that is, the completion time of the batch is the completion time of the final job in the batch. There are two types of batching problems, denoted by p-batching problem and s-batching problem (Potts and Van Wasenhove 1992). For p-batching problem (s-batching problem) the length of a batch is equal to the maximum (sum) of processing times of all jobs in the batch.

Till now, there exist many researches on a batch problem, also see survey papers (Brucker, 2006)(Cheng and Janiak, 1995). This paper treats one model of s-batch problem which considers a single machine batch scheduling problem with fuzzy due-date and ordinary precedence constraints. Section 2 formulates our problem, that is, the fuzzy version of a single machine batching scheduling problem, i.e. the fuzzy due-date. Section 3 proposes an efficient solution procedure based on the Procedure HL(Sung and Joo, 1997) to partition the job sequence into batches maximizing the minimum value of desirability of the
Section 4 summarizes results in this paper and discusses further research problems.

2. Problem Formulation. There are \(n\) simultaneously available jobs \(\{J_1, J_2, \cdots, J_n\}\) 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 fuzzy due-date \(D_i\). Fuzzy due-date \(D_i\) is a flexible due-date described by the satisfaction degree with respect to the completion time of job \(J_i\), which is denoted by the membership function as follows:

\[
\mu_i(C_i) = \begin{cases} 
1, & \text{if } C_i \leq d_i \\
1 - \frac{C_i - d_i}{e_i - d_i}, & \text{if } d_i \leq C_i \leq e_i \\
0, & \text{if } C_i \geq e_i 
\end{cases}
\]

where \(C_i\) is the completion time of job \(J_i\), \(0 < d_i < e_i, i = 1, 2, \cdots, n\).

A setup time \(s\) is incurred when starting a new batch. Setup times are assumed to be identical for all batches. All batch sizes cannot exceed a common upper bound \(b\). Jobs have to be subject to precedence constraints. A precedence relation \(J_i \prec J_j\) implies that job \(J_i\) must be completed before job \(J_j\) starts to be processed. This means that \(J_i\) must be processed in a batch before that of \(J_j\). The problem is to find the optimal solution consisting of a batch number and allocation of jobs to batches maximizing the minimum satisfaction degree of fuzzy due-date under a common limited batch size and a precedence relation.

Under the above setting, we consider the following single machine scheduling problem:

\[
P: \text{Maximize } \min \{\mu_i(C_i) \mid i = 1, 2, \cdots, n\} \\
\text{subject to } \sum_{i=1}^{u} |B_i| = n_i |B| \leq b, i = 1, \cdots, u
\]

where \(|B_i|\) denotes the number of jobs in batch \(|B_i|\) and the batch number \(u\) is also decision variable.

3. Solution Procedure. Let

\[
T_i = \{J_i \mid J_i \prec J_j\}
\]

be a job set consisting of jobs that \(J_i\) precedes. If we denote \(\alpha = \min \{\mu_i(C_i) \mid i = 1, 2, \cdots, n\}\), then

\[
\mu_i(C_i) \geq \alpha \Rightarrow C_i \leq d_i + (1 - \alpha)(e_i - d_i) \Delta D_i(\alpha), i = 1, 2, \cdots, n
\]

This means \(D_i(\alpha)\) is a due-date of job \(J_i, i = 1, 2, \cdots, n\).

Lawler and Moore [9] has 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-dates \(D_i(\alpha)\) defined as below without precedence relation.
If all processing times are positive, then a sequence ordered as non-decreasing modified due dates is compatible with the precedence constraints. If jobs are indexed in the non-decreasing order of modified due-dates for a fixed $\alpha$ such that

$$D_1(\alpha) \leq D_2(\alpha) \leq \cdots \leq D_n(\alpha)$$

where $p_i \leq p_{i+1}$ if $D_i(\alpha) = D_{i+1}(\alpha), (i = 1, \ldots, n-1)$ and in this indexing if $J_i < J_j$, then $i < j$. Since order of modified due date depend on the value $\alpha$, first we calculate critical values $\alpha$ such that $D_i(\alpha) = D_j(\alpha)$ for each pair $i, j, 1 \leq i \leq j \leq n$, let these values be $\alpha_{ij} = \frac{e_i - e_j}{f_i - f_j}$ when $f_i \Delta e_i - d_i \neq f_j \Delta e_j - d_j$ and be undefined when $f_i = f_j$. Sorting defined $\alpha_{ij}$ such that $0 < \alpha_{ij} < 1$, let the result be

$$\alpha^0 \Delta 0 < \alpha^1 < \cdots < \alpha^q < \alpha^{q+1} \Delta 1$$

where $q$ is the different number of them. This means that in each interval $\alpha \in (\alpha^l, \alpha^{l+1}), l = 0, 1, \ldots, q$, order of $D_i(\alpha), i = 1, 2, \ldots, n$ is uniquely determined.

Now we define $I(\alpha), 2(\alpha), \ldots, n(\alpha)$ describing the order of $D_i(\alpha), i = 1, 2, \ldots, n$, that is, $D_{I(\alpha)}(\alpha) \leq D_{2(\alpha)}(\alpha) \leq \cdots \leq D_{n(\alpha)}(\alpha)$ since its order depends on $\alpha$. Now we define sub-problem $P(\alpha)$ as a batch scheduling to find a feasible batch sequence satisfying deadline $D_{I(\alpha)}, D_{2(\alpha)}, \ldots, D_{n(\alpha)}$ and the precedence relation, that is,

$$C_j \leq D_j(\alpha), j = 1, 2, \ldots, n, \sum_{i=1}^{n} |B_i| = n, |B_i| \leq b, i = 1, \ldots, u$$

W can check the feasibility of $P(\alpha)$ using Algorithm 1 below.

**Algorithm 1:**

**Step 0:** set $j = I(\alpha), k = 1, B_i = \{J_{I(\alpha)}\}$, Go to Step 1.

**Step 1:** If the current batch $B_k$ does not include any predecessor of job $j(\alpha)$ and all jobs can be completed till their modified due dates by adding job $j(\alpha)$ 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 $j(\alpha)$ 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 $B_k \leftarrow B_k \cup \{j(\alpha)\}$. If $j = n$, terminate. Otherwise return to Step 1 after setting $j = j + 1$.

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

**Theorem 3.1.** Algorithm 1 checks the feasibility of sub-problem $j(\alpha)$ in at most $O(n^2)$ computational time.

**Proof:** In order to the convenience sake of proof, let $l(\alpha) = 1, 2(\alpha) = 2, \ldots, n(\alpha) = n$ without loss of generality by changing the job index and denote $D_i(\alpha), i = 1, 2, \ldots, n$ with $D_i$ simply. Further let $S$ be a schedule constructed by the algorithm (feasible or not) and $S'$ a
feasible schedule. Assume that both schedules coincide until $J_1, J_2, \ldots, J_{i-1}$. Then we have two situations as shown in Figure 3.1 and Figure 3.2. We have $D'_i \leq D'_j$ which follows from the fact that in $S$ job $j$ is scheduled after job $i$.

**Case 1.** $J_j$ and $J_i$ are in the same batch. Let $i_1, i_2, \ldots, i_k$ be all jobs scheduled in $S'$ between $J_j$ and $J_i$. Furthermore, we assume that these jobs are ordered according to starting time. If we exchange $i$ with $j$, we again have a feasible schedule $S'$ also because $D'_i \leq D'_j$. $S$ and $S'$ coincide till $J_1, J_2, \ldots, J_i$. Now we find next different job in $S'$ from $S$. If we cannot find that job, it means that $S$ is also an optimal batch sequence. Otherwise, we check case 1 or case 2 as below and continue. Continuing this process after a finite number of steps, we obtain an optimal schedule which coincides with $S$.

**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 we should consider is the case $p_i \leq p_j$. Let $u$ be the finishing time of batch $h$. So if we exchange the place of $J_j$ within $S'$, we again have a feasible schedule $S'$ 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 we find next different job in $S'$ from $S$. If we cannot find that job, it means that $S$ is also feasible batch sequence. Otherwise, we check case 1 or case 2 and continue this process after a finite number of steps, we have a feasible schedule which coincides with $S$.

Note that the calculation of modified due-dates is $O(n)$ computational time and construct $T = \{J_i | J_i \prec J_j\}$ is $O(n^2)$. 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.

Q. E. D.

**FIGURE 3.1.** Schedules for $J_j$ and $J_i$ in the same batch

**FIGURE 3.2.** Schedules for $J_j$ and $J_i$ in different batch
In order to solve $P$, we first check the feasibility of $P(1)$ and $P(0)$.

In case $P(1)$ is feasible, the optimal value of $P$ is 1 and an optimal batch sequence is feasible batch sequence of $P(1)$. If $P(0)$ is infeasible, there exists no batch sequence of $P$.

Otherwise, that is, if $P(1)$ is infeasible and $P(0)$ is feasible, then using a binary search, we find the maximum value among

$$
\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{q-1}, \alpha_q
$$

such that corresponding sub-problems are feasible. Let the maximum value be $\alpha^*$. Then we check the feasibility of $P(\alpha^*)$. If $P(\alpha^*)$ is feasible, an optimal value of $P$ exists in the interval $[\alpha^*, 1]$ (in case $t = q$, the optimal value exists in the interval $[\alpha^*, 1]$). If $P(\alpha^*)$ is infeasible, then the optimal value exists in the interval $[\alpha^*, \alpha^* + 1]$. Let the resulting interval be $[\alpha_L, \alpha_U]$. Further the feasible batch sequence of $P(\alpha_L)$ be as follows:

$$
B_1 = \{J_{\pi(1)}, J_{\pi(2)}, \ldots, J_{\pi(j_1)}\}, \quad B_2 = \{J_{\pi(j_1+1)}, J_{\pi(j_1+2)}, \ldots, J_{\pi(j_1+\theta)}\}, \ldots, \\
B_{k'} = \{J_{\pi(j_{1-1}+1)}, J_{\pi(j_{1-1}+2)}, \ldots, J_{\pi(n)}\}
$$

where $\pi(j)$ denote the job index processed in the $j$-th order and $k^*$ the batch number.

**Theorem 3.2.** An optimal batch sequence of $P$ is same as $P(\alpha_L)$ (rigorously same as $P(\alpha_L + 0)$).

**Proof:** First note that corresponding processing order of sub-problem $P(\alpha)$ is same as $P(\alpha_L)$ for any $\alpha \in (\alpha_L, \alpha_U)$ if using algorithm 1 since the order of $D_i(\alpha), i = 1, 2, \ldots, n$ is not changed. Thus we should show the optimal contents of batches are not changed as $P(\alpha_L)$. We assume that the first different batch between feasible batch sequence of $P(\alpha_L)$ and an optimal batch sequence of $P$ is the $k$-th batch. This means that $k$-th batch of the latter batch sequence includes at least one more job than that of the former batch sequence. But this is impossible since Algorithm 1 decides that job $J_{\pi(i_{k+1})}$ should be included in the latter batch than $k$ due to a limitation of the modified due-date of job $J_{\pi(i_{k+1})}$ and increase $\alpha$ from $\alpha_L$ makes the modified due-date of job $J_{\pi(i_{k+1})}$ smaller. From this argument, an optimal batch sequence of $P$ is same as $P(\alpha_L)$ (rigorously same as $P(\alpha_L + 0)$). Q. E. D.

From Theorem 3.2, the optimal value $\alpha^*$ of $P$ is calculated as follows:

$$
\alpha^* = \min\{\{\alpha(1), \alpha(2), \ldots, \alpha(k^*)\}
$$

where $\alpha(k)$ is determined by $\alpha$ such that

$$
ks + \sum_{j=1}^{j_k} p_{\pi(j)} = d_{\pi(j_{k-1}+1)} + (1 - \alpha)(e_{\pi(j_{k-1}+1)} - d_{\pi(j_{k-1}+1)}),
$$

that is,

$$
\alpha(k) = \frac{e_{\pi(j_{k-1}+1)} - (ks + \sum_{j=1}^{j_k} p_{\pi(j)})}{e_{\pi(j_{k-1}+1)} - d_{\pi(j_{k-1}+1)}}, \quad k = 1, 2, \ldots, k^*.
$$

**Theorem 3.3.** Our solution procedure solves our problem $P$ in at most $O(n^2 \log n)$ computational time.

**Proof:** Validity of our solution procedure is clear and so we show its complexity.
Calculation of $\alpha_j - O(n^2)$ and so that of $\alpha^n, \alpha^1, \ldots, \alpha^n, \alpha^{n+1} - O(n^2 \log n)$. Calculation of the modified due-dates $D'_i(\alpha) - O(n^2)$. Each execution of algorithm 1 takes $O(n^2)$. Computational time. Since we must check the feasibility of at most $O(\log n)$ sub-problems in order to find the existence interval of the optimal value for $P$, totally $O(n^2) \times O(\log n) = O(n^2 \log n)$ computational time is needed to find the interval $[\alpha_L, \alpha_U]$. Calculation of the optimal value $\alpha^*$ takes $O(n)$ and optimal batch sequence is constructed in at most $O(n^2)$. Finally totally at most $(n^2 \log n)$ computational time is needed.

4. Conclusions. This paper has proposed a solution procedure for a single machine batch scheduling problem with fuzzy due-date and ordinary precedence constraints. Modified due-date is introduced to break the precedence relations among jobs. The Algorithm 1 is based on the modified Procedure HL to solve problem. This problem should be extended more to the case of fuzzy due-date and fuzzy precedence. We are now attacking the extended problem though this problem is a bi-criteria problem. Further multiple processor batch scheduling problems should be investigated in order to cope with actual applications though it becomes a tedious problem.

REFERENCES


