Journal of Scheduling **8:** 537–542, 2005.

A NOTE ON THE SCHEDULING WITH TWO FAMILIES OF JOBS*

J. J. YUAN†, W. P. SHANG AND Q. FENG

Department of Mathematics, Zhengzhou University, Zhengzhou, Henan 450052, China

ABSTRACT

Baker and Smith [J. Scheduling, 6, 7–16, 2003] introduced a new model of scheduling in which there are two or more distinct families of jobs pursuing different objectives. Their contributions include two polynomial-time dynamic programming recursions, respectively, for the single machine scheduling with two families of jobs to minimize a positive combination of total weighted completion time, or maximum lateness, of the first family of jobs and maximum lateness of the second family of jobs. Unfortunately, these dynamic programming recursions are incorrect. In this paper, we solve the same problems by an $O(n_1 n_2(n_1 + n_2))$ time algorithm.

1. PROBLEM FORMULATION AND DISCUSSIONS

The following scheduling problem was studied by Baker and Smith (2003). We are given two families of jobs $\mathcal{J}^{(1)} = \{J_1^{(1)}, J_2^{(1)}, \ldots, J_{n_1}^{(1)}\}$ and $\mathcal{J}^{(2)} = \{J_1^{(2)}, J_2^{(2)}, \ldots, J_{n_2}^{(2)}\}$ to be processed in a single machine. Let $x \in \{1, 2\}$ be given. The processing time of a job $J_i^{(x)} \in \mathcal{J}^{(x)}$ is denoted by $p_i^{(x)}$. Suppose that each job $J_i^{(x)} \in \mathcal{J}^{(x)}$ has a due date $d_i^{(x)}$ and a nonnegative weight $w_i^{(x)}$. For a given schedule π for the jobs in $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}$, we use $C_i^{(x)}(\pi)$ to denote the completion time of a job $J_i^{(x)} \in \mathcal{J}^{(x)}$. The lateness of a job $J_i^{(x)} \in \mathcal{J}^{(x)}$ under π is denoted by $L_i^{(x)}(\pi)$. The maximum lateness of the jobs in $\mathcal{J}^{(x)}$ under π is denoted by $L_{\max}^{(x)}(\pi)$. The maximum completion time of the jobs in $\mathcal{J}^{(x)}$ under π is denoted by $C_{\max}^{(x)}(\pi)$. $f^{(x)}(\pi)$ is used to denote the objective of the jobs in $\mathcal{J}^{(x)}$ under π. In the paper, we assume

$$f^{(x)} \in \left\{ C_{\max}^{(x)}, L_{\max}^{(x)}, \sum w_i^{(x)} C_i^{(x)} \right\}.$$

Let θ be a given positive number. For a given schedule π, we define

$$f(\pi) = f^{(1)}(\pi) + \theta f^{(2)}(\pi).$$

The objective of the considered problem is to find a schedule π for the jobs in $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}$ such that $f(\pi)$ is as small as possible. We will denote this problem by $1\|f^{(1)} + \theta f^{(2)}$.

The following three properties are established in [1].

Property 1. If, for some $x \in \{1, 2\}$, $f^{(x)} = C_{\max}^{(x)}$, then there is an optimal schedule for the jobs in $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}$ such that all jobs belonging to $\mathcal{J}^{(x)}$ are processed consecutively.

*Project supported by NNSFC (Grant 10371112) and NSFHN (Grant 0411011200)
†Corresponding author: Jinjiang Yuan. E-mail: yuanjj@zzu.edu.cn

Property 2. If, for some $x \in \{1, 2\}$, $f^{(x)} = L_{\max}^{(x)}$, *then there is an optimal schedule for the jobs in* $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}$ *such that all jobs belonging to* $\mathcal{J}^{(x)}$ *are processed in nondecreasing order of* $d_i^{(x)}$, *that is, in EDD order.*

Property 3. If, for some $x \in \{1, 2\}$, $f^{(x)} = \Sigma w_i^{(x)} C_i^{(x)}$, *and* $p_i^{(x)} \le p_j^{(x)}$ *whenever* $p_i^{(x)}/w_i^{(x)} \le p_j^{(x)}/w_j^{(x)}$, *then there is an optimal schedule for the jobs in* $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}$ *such that all jobs belonging to* $\mathcal{J}^{(x)}$ *are processed in nondecreasing order of* $p_i^{(x)}/w_i^{(x)}$, *that is, in SWPT order.*

The properties 1 and 2 can be easily verified. But, Property 3 is incorrect. Consider the following instance of $1 \| f^{(1)} + \theta f^{(2)}$ with $f^{(1)} = \Sigma w_i^{(1)} C_i^{(1)}$ and $f^{(2)} = L_{\max}^{(2)}$:

$$
\begin{aligned}
&n_1 = 2, \quad n_2 = 1 \quad \text{and} \quad \theta = 1000;\\
&p_1^{(1)} = 10, \quad p_2^{(1)} = 9, \quad p_1^{(2)} = 10;\\
&w_1^{(1)} = 11, \quad w_2^{(1)} = 10, \quad \text{and} \quad d_1^{(2)} = 20.
\end{aligned}
$$

One can easily verify that the only optimal schedule is $(J_1^{(1)}, J_1^{(2)}, J_2^{(1)})$. Since $p_1^{(1)}/w_1^{(1)} > p_2^{(1)}/w_2^{(1)}$ and $p_1^{(1)} > p_2^{(1)}$, this instance is indeed a counterexample for Property 3.

The following is a weakened version of Property 3.

Property 3. If, for some $x \in \{1, 2\}$, $f^{(x)} = \sum w_i^{(x)} C_i^{(x)}$, *and* $p_i^{(x)} \le p_j^{(x)}$ *whenever* $w_i^{(x)} > w_j^{(x)}$, *then there is an optimal schedule for the jobs in* $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}$ *such that all jobs belonging to* $\mathcal{J}^{(x)}$ *are processed in nondecreasing order of* $p_i^{(x)}$, *that is, in SPT order.*

Proof. Suppose that there is an optimal schedule for the problem $1 \| f^{(1)} + \theta f^{(2)}$ such that jobs belonging to $\mathcal{J}^{(x)}$ are not processed in SPT order. Then there must be two jobs $J_i^{(x)}, J_j^{(x)} \in \mathcal{J}^{(x)}$ such that $p_i^{(x)} < p_j^{(x)}$ but $J_j^{(x)}$ is processed before $J_i^{(x)}$. By exchanging the positions of $J_i^{(x)}$ and $J_j^{(x)}$ in the schedule, we obtained a new schedule such that the completion time of every job other than $J_i^{(x)}$ and $J_j^{(x)}$ is not increased. By noting the fact that $p_i^{(x)} < p_j^{(x)}$ and $w_i^{(x)} \ge w_j^{(x)} \ge 0$, the new schedule is still optimal. Continuing this procedure, we eventually obtain an optimal schedule with the required property. ■

As a consequence of Property 3*, we have

Property 4. If, for some $x \in \{1, 2\}$, $f^{(x)} = \sum C_i^{(x)}$, *then there is an optimal schedule for the jobs in* $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}$ *such that all jobs belonging to* $\mathcal{J}^{(x)}$ *are processed in nondecreasing order of* $p_i^{(x)}$, *that is, in SPT order.*

A remark on $1 \| L_{\max}^{(1)} + \theta L_{\max}^{(2)}$: The following dynamic programming recursion for $1 \| L_{\max}^{(1)} + \theta L_{\max}^{(2)}$ is established in Baker and Smith (2003). Sorting the jobs in $\mathcal{J}^{(1)}$ and the jobs in $\mathcal{J}^{(2)}$ by EDD rule, respectively. Let (u, v) represent the partial schedule consisting of the first u jobs from $\mathcal{J}^{(1)}$ and the first v jobs from $\mathcal{J}^{(2)}$. Write

$$\mathcal{J}(u, v) = \left\{J_1^{(1)}, \ldots, J_u^{(1)}, J_1^{(2)}, \ldots, J_v^{(2)}\right\}.$$

Let $F(u, v)$ denote the minimum performance measure for the jobs in the set $\mathcal{J}(u, v)$ corresponding to (u, v). Define $L^{(x)}(u, v)$ to be the maximum lateness of the jobs in $\mathcal{J}^{(x)} \cap \mathcal{J}(u, v)$, $x = 1, 2$. Then the dynamic programming recursion is:

$$F(i, j) = \min \begin{cases} F(i-1, j) + \max\left\{t - d_i^{(1)}, L^{(1)}(i-1, j)\right\} - L^{(1)}(i-1, j), \\ F(i, j-1) + \theta\left(\max\left\{t - d_j^{(2)}, L^{(2)}(i, j-1)\right\} - L^{(2)}(i, j-1)\right), \end{cases}$$

where $t = \sum_{1 \le i \le u} p_i^{(1)} + \sum_{1 \le i \le v} p_i^{(2)}$. Furthermore, for each of these computations, if we set $F(i, j)$ according to the first term, we define

$$L^{(1)}(i, j) = \max\left\{t - d_i^{(1)}, L^{(1)}(i-1, j)\right\} \quad \text{and} \quad L^{(2)}(i, j) = L^{(2)}(i-1, j),$$

and otherwise, we define

$$L^{(1)}(i, j) = L^{(1)}(i, j-1) \quad \text{and} \quad L^{(2)}(i, j) = \max\left\{t - d_j^{(2)}, L^{(2)}(i, j-1)\right\}.$$

The initial condition can be naturally defined (see [1]). The optimal value of the scheduling problem will be calculated by $F(n_1, n_2)$.

Unfortunately, the above dynamic programming recursion for $1 \| L_{\max}^{(1)} + \theta L_{\max}^{(2)}$ is incorrect. In fact, if the value of $F(i, j)$ is minimized by the x-th term, it is possible to shift some processed jobs to $\mathcal{J}^{(x)}$ right and some processed jobs to $\mathcal{J}^{(3-x)}$ left so that the objective value is further reduced. For example, consider the following instance of $1 \| L_{\max}^{(1)} + \theta L_{\max}^{(2)}$:

$$n_1 = 1, \quad n_2 = 2 \quad \text{and} \quad \theta = 1;$$
$$p_1^{(1)} = 20, \quad p_1^{(2)} = 5, \quad p_2^{(2)} = 30;$$
$$d_1^{(1)} = 0, \quad d_1^{(2)} = 0, \quad d_2^{(20)} = 2.$$

The above dynamic programming recursion returns $F(1, 2) = 78$ corresponding to the schedule $(J_1^{(2)}, J_1^{(1)}, J_2^{(2)})$. But, the objective value under the schedule $(J_1^{(1)}, J_1^{(2)}, J_2^{(2)})$ is $73 < F(1, 2)$.

A remark on $1 \| \sum C_i^{(1)} + \theta L_{\max}^{(2)}$: The following dynamic programming recursion for $1 \| \sum C_i^{(1)} + \theta L_{\max}^{(2)}$ is established in [1]. Sorting the jobs in $\mathcal{J}^{(1)}$ by SPT rule and the jobs in $\mathcal{J}^{(2)}$ by EDD rule. Let (u, v) represent the partial schedule consisting of the first u jobs from $\mathcal{J}^{(1)}$ and the first u jobs from $\mathcal{J}^{(2)}$. Write

$$\mathcal{J}(u, v) = \left\{J_1^{(1)}, \ldots, J_u^{(1)}, J_1^{(2)}, \ldots, J_v^{(2)}\right\}.$$

Let $F(u, v)$ denote the minimum performance measure for the jobs in the set $\mathcal{J}(u, v)$ corresponding to (u,v). Define $L^{(2)}(u, v)$ to be the maximum lateness of the jobs in $\mathcal{J}^{(2)} \cap \mathcal{J}(u, v) = \{J_1^{(2)}, \ldots, J_v^{(2)}\}$. Then the dynamic programming recursion is:

$$F(u, v) = \min\{F(u-1, v) + t, F(u, v-1) + \theta\left(\max\left\{t - d_v^{(2)}, L(u, v-1)\right\} - L(u, v-1)\right)\},$$

where $t = \sum_{1\le i\le u} p_i^{(1)} + \sum_{1\le i\le v} p_i^{(2)}$. Furthermore, $L(u, v) = \max\{t - d_v^{(2)}, L(u, v-1)\}$ if $F(u, v)$ is minimized by the second term, and $L(u, v) = L(u-1, v)\}$ otherwise. The initial condition is $F(0, 0) = 0$. The optimal value of the scheduling problem will be calculated by $F(n_1, n_2)$.

Unfortunately, the above dynamic programming recursion for $1\|\sum C_i + \theta L_{\max}^{(2)}$ is incorrect. In fact, if the value of $F(u, v)$ is minimized by the second term, it is possible to shift some jobs $J_i \in \mathcal{J}^{(1)}(i \le u)$ to left and some jobs $J_j^{(2)} \in \mathcal{J}^{(2)}(j \le v-1)$ to right so that the objective value is further reduced. For example, consider the following instance of $1\|\sum C_i + \theta L_{\max}^{(2)}$;

$$n_1 = 1, \quad n_2 = 2 \quad \text{and} \quad \theta = 1;$$
$$p_1^{(1)} = 20, \quad p_1^{(2)} = 5, \quad p_2^{(2)} = 30;$$
$$d_1^{(2)} = 0, \quad d_2^{(2)} = 2.$$

The above dynamic programming recursion returns $F(1, 2) = 78$ corresponding to the schedule $(J_1^{(2)}, J_1^{(1)}, J_2^{(2)})$. But, the objective value under the schedule $(J_1^{(1)}, J_1^{(2)}, J_2^{(2)})$ is $73 < F(1, 2)$.

In the next section, we will give a polynomial-time algorithm for the above two problems.

2. ALGORITHMS

Consider the general problem $1\|f^{(1)} + \theta f^{(2)}$. If, for some $x \in \{1, 2\}$, $f^{(x)} = L_{\max}^{(x)}$, then, by Property 2, there is an optimal schedule for the jobs in $\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}$ such that all jobs belonging to $\mathcal{J}^{(x)}$ are processed in EDD order. An easy observation is that, for two jobs $J_i^{(x)}$ and $J_j^{(x)}$ in $\mathcal{J}^{(x)}$ with $d_i^{(x)} = d_j^{(x)}$, there must be an optimal schedule π for $1\|f^{(1)} + \theta f^{(2)}$ such that $J_i^{(x)}$ and $J_j^{(x)}$ are processed consecutively. Such two jobs $J_i^{(x)}$ and $J_j^{(x)}$ can be merged into a big job $J_{ij}^{(x)}$ with processing time $p_i^{(x)} + p_j^{(x)}$ and due date $d_i^{(x)}$. Hence, we suppose in the following that the jobs in $\mathcal{J}^{(x)}$ have distinct due dates when $f^{(x)} = L_{\max}^{(x)}$.

We say a schedule π for $1\|L_{\max}^{(1)} + \theta_{\max}^{(2)}$ is regular if π sequence jobs in $\mathcal{J}^{(x)}$ in EDD order for any $x \in \{1, 2\}$. By property 2, there is an optimal schedule for the problem $1\|L_{\max}^{(1)} + \theta_{\max}^{(2)}$ such that π is regular.

We say a schedule π for $1\|\sum C_i^{(1)} + \theta L_{\max}^{(2)}$ is regular if π sequences jobs in $\mathcal{J}^{(1)}$ in SPT order and the jobs in $\mathcal{J}^{(2)}$ in EDD order. By property 2 and 4, there is an optimal schedule for the problem $1\|\sum C_i + \theta L_{\max}^{(2)}$ such that π is regular.

In the following we consider the problem $1\|f^{(1)} + \theta L_{\max}^{(2)}$, where $f^{(1)} \in \{L_{\max}^{(1)}, \sum C_i^{(1)}\}$. We re-label the jobs in $\mathcal{J}^{(2)}$ such that $d_1^{(2)} < d_2^{(2)} < \cdots < d_{n_2}^{(2)}$. Furthermore, if $f^{(1)} = L_{\max}^{(1)}$, we re-label the jobs in $\mathcal{J}^{(1)}$ such that $d_1^{(1)} < d_2^{(2)} < \cdots < d_{n_1}^{(1)}$, and if $f^{(1)} = \sum C_i^{(1)}$, we re-label the jobs in $\mathcal{J}^{(1)}$ such that $p_1^{(1)} \le p_2^{(1)} \le \cdots \le p_{n_1}^{(1)}$. The remaining question is how to interleave the two sequences optimally.

Let $\pi_1 = (J_1^{(1)}, J_2^{(1)}, \ldots, J_{n_1}^{(1)}, J_1^{(2)}, J_2^{(2)}, \ldots, J_{n_2}^{(2)})$ and $\pi_2 = (J_1^{(2)}, J_2^{(2)}, \ldots, J_{n_2}^{(2)}, J_1^{(1)}, J_2^{(1)}, \ldots, J_{n_1}^{(1)})$. Define $UB = L_{\max}^{(2)}(\pi_1)$ and $LB = L_{\max}^{(2)}(\pi_2)$. Then, for any regular schedule π for $1\|f^{(1)} + \theta L_{\max}^{(2)}$, we must have

$$LB \le L_{\max}^{(2)}(\pi) \le UB.$$

For $0 \le u \le n_i$ and $1 \le v \le n_2$, write

$$t(u, v) = \sum_{t \le i \le u} p_i^{(1)} + \sum_{1 \le i \le v} p_i^{(2)}$$

and define

$$y(u, v) = t(u, v) - d_v^{(2)}.$$

By noting that, in any regular schedule π for $1\|f^{(1)} + \theta L_{\max}^{(2)}$, the completion time of each job $J_v^{(2)}$ must be of the form $t(u, v)$ for some u with $0 \le u \le n_1$, we must have

$$L_{\max}^{(2)}(\pi) \in \{y(u, v) : 0 \le u \le n_1, 1 \le v \le n_2\}.$$

For each $y \in \{y(u, v) : 0 \le u \le n_1, 1 \le v \le n_2\}$ with $LB \le y \le UB$, we consider the problem $1\|f^{(1)} + \theta L_{\max}^{(2)}$ under the restriction that $L_{\max}^{(2)} = y$. The restricted version will be denoted by $1|L_{\max}^{(2)} = y|f^{(1)}$. There may be some y such that the problem $1 \mid L_{\max}^{(2)} = y \mid f^{(1)}$ is infeasible. Hence, we prefer to consider the relaxed version $1 \mid L_{\max}^{(2)} \le y \mid f^{(1)}$.

In an optimal regular schedule for $1 \mid L_{\max}^{(2)} \le y \mid f^{(1)}$, suppose (by the regular property) that the set of the first $u + v$ jobs is $\{J^{(1)}, J_2^{(1)}, \ldots, J_u^{(1)}, J_2^{(2)}, \ldots, J_v^{(2)}\}$, where $1 \le u \le n_1$ and $1 \le v \le n_2$. If $t(u, v) - d_v^{(2)} > y$, then the $u + v$-th job under a certain optimal regular schedule π is the job $J_u^{(1)}$. If $t(u, v) - d_v^{(2)} \le y$, then the $u + v$-th job under a certain optimal regular schedule π is $J_v^{(2)}$. Consequently, the problem $1 \mid L_{\max}^{(2)} \le y|f^{(1)}$ can be solved by the following linear-time algorithm.

Linear-time algorithm for $1|L_{\max}^{(2)} \le y|f^{(1)}$

Step 1. Set $u := n_1$, $v := n_2$ and

$$F := \begin{cases} -\infty & \text{if } f^{(1)} = L_{\max}^{(1)} \\ 0, & \text{if } f^{(1)} = \sum C_i^{(1)}. \end{cases}$$

Step 2. If $u = 0$, then define $\pi(i) = J_i^{(2)}$, $1 \le i \le v$, and stop. Otherwise, turn to Step 4.
Step 3. If $v = 0$, then define $\pi(i) = J_i^{(1)}$, $1 \le i \le u$, set

$$F := \begin{cases} \max\left\{F, \max\left\{t(i, 0) - d_i^{(1)} : 1 \le i \le u\right\}\right\}, & \text{if } f^{(1)} = L_{\max}^{(1)}, \\ F + \sum\limits_{1 \le i \le u} t(i, 0), & \text{if } f^{(1)} = \sum C_i^{(1)} \end{cases}$$

and stop. Otherwise, turn to Step 4.
Step 4. If $t(u, v) - d_v^{(2)} > y$, then define $\pi(u + v) = J_u^{(1)}$ and set

$$F := \begin{cases} \max\left\{F, \left\{t(u, v) - d_u^{(1)}\right\}, & \text{if } f^{(1)} = L_{\max}^{(1)}, \\ F + t(u, v), & \text{if } f^{(1)} = \sum C_i^{(1)} \end{cases}$$

and $u := u - 1$; return to Step 2.

If $t(u, v) - d_v^{(2)} \le y$, then define $\pi(u + v) = J_v^{(2)}$ and set $v := v - 1$; return to Step 3.

Denote by F_y the F-value returned by the above algorithm for a given y. Our final observation is that the optimal objective value of the problem $1\|f^{(1)} + \theta L_{\max}^{(2)}$ must be

$$\min\{F_y + \theta_y : LB \le y \le UB, y \in \{y(u, v) : 0 \le u \le n_1, 1 \le v \le n_2\}\}.$$

Since the complexity of the algorithm for $1|L_{\max}^{(2)} \le y|f^{(1)}$ is $O(n_1 + n_2)$ and we have at most $n_1 n_2$ choices for y, we conclude that the problem $1\|f^{(1)} + \theta L_{\max}^{(2)}$ can be solved in $O(n_1 n_2 (n_1 + n_2))$ time.

REFERENCES

Baker, K. R. and J. C. Smith, "A multiple-criterion model for machine scheduling," *Journal of Scheduling*, **6**, 7–16, (2003).