Journal of the Operations Research
Society of Japan
Vol. 26, No. 3, September 1983

SOME BOUNDS ON APPROXIMATION ALGORITHMS FOR $n/m/I/L_{MAX}$ AND $n/2/F/L_{MAX}$ SCHEDULING PROBLEMS

Teruo Masuda
University of Osaka Prefecture

Hiroaki Ishii
Osaka University

Toshio Nishida
Osaka University

(Received August 2, 1982; Final May 30, 1983)

Abstract In this paper, we analyze approximation algorithms for two types of scheduling problems. The first is the n jobs scheduling problem with due dates on m identical machines to minimize the maximum lateness. For this problem $n/m/I/L_{max}$, we propose two approximation algorithms and derive their worst case bounds. The second is the $2 \times n$ flow shop scheduling problem with due dates to minimize the maximum lateness. For this problem $n/2/F/L_{max}$, we first give a solvable case in the sense that the optimal schedule can be easily found. Then we again propose an approximation algorithm for general $n/2/F/L_{max}$ and derive its worst case bound.

1. Introduction

We consider two scheduling problems whose objective is to minimize maximum lateness. The first is a following parallel machine scheduling problem; (i) a set of n independent jobs $J=(J_1,J_2,\cdots,J_n)$ is to be processed on a set of m identical machines $M=(M_1,M_2,\cdots,M_m)$, (ii) each job J_i has a processing time $t_i \geq 0$ and due date $d_i \geq 0$, (iii) preemption is not allowed and (iv) the objective is to minimize the maximum lateness.

The second is the following two-machine flow shop scheduling problem; (i) a set of n independent jobs $J=(J_1,J_2,\cdots,J_n)$ is to be processed on two machines A and B, (ii) each job J_i has two processing times a_i and $b_i \geq 0$ corresponding to A and B respectively, and a due date $d_i \geq 0$, (iii) the processing order of each job J_i is first on A and next on B, and (iv) the objective is to minimize the maximum lateness.

Further it is assumed in both problems that no machine can simultaneously process two or more jobs and no job can be processed simultaneously on more than one machine. Hereafter, according to Lenstra et al. [6], these problems are compactly denoted by n/m/I/L_{max} and n/2/F/L_{max}, respectively.

For the maximum lateness problem on a single machine, Jackson [2] has obtained an exact algorithm which finds an optimum schedule in a polynomial time of problem size. Furthermore, Lawler [5] has obtained $O(n^2)$ exact algorithm for the related problem with arbitrary nondecreasing cost function and general precedence constraints where n is the number of jobs. While, both problems in this paper are known to be *NP*-complete (Lenstra et al. [6]). That is, they are among notoriously intractable problems and so there do not and perhaps will not exist any polynomially bounded exact algorithms for them. For these reasons, practioners are willing to accept good feasible solution. Indeed, for *NP*-complete problems, many approximation algorithms and their bouds for the worst cases are derived. With respect to scheduling problems with due dates, however, very few worst case bounds have been obtained. (See Graham et al. [1] for details.) Kise et al. [4] have developed effective approximation algorithms and showed their worst case bounds for the maximum lateness problem on a single machine.

In general, to evaluate the effectiveness of approximation algorithm, various measures such as the absolute deviation $\omega-\omega'(\Pi)$ and the relative deviation $(\omega-\omega'(\Pi))/\omega$ has been customarily used so far, where ω denotes the value of optimal schedule and $\omega'(\Pi)$ the value of approximate schedule generated by the approximation algorithm Π. As pointed out by Kise et al. [4], however, above measures exhibit a shortcoming that they give different values between two equivalent problems, where equivalence means that one is obtained by applying a simple transformation to the other, and the optimal and the approximate schedule are the same in both problems. This pathology urges us to employ the following modified relative deviation

$$\frac{\omega-\omega'(\Pi)}{\omega+d_{max}}$$

proposed by Kise et al. [4] as the effective measure of approximation algorithm Π, where $d_{max}=\max\{d_i \mid i=1,2,\cdots,n\}$.

In the sequel, first we propose two approximation algorithms for n/m/I/L_{max} and derive their worst case bounds. Next for n/2/F/L_{max}, we give the solvable case in the sense that the optimal schedule can be easily found. Then we propose an approximation algorithm for general n/2/F/L_{max} and again derive its worst case bound.

2. Approximation Algorithms for $n/m/I/L_{max}$ and Their Bounds

In this chapter, we consider the scheduling problem $n/m/I/L_{max}$. This problem is characterized by the processing time vector $T=(t_1,\cdots,t_n)$ and the due date vector $D=(d_1,d_2,\cdots,d_n)$. Here without any loss of generality, we assume that $d_1\leq\cdots\leq d_n(=d_{max})$. Our first approximation algorithm *EDD* (*Earliest Due Date*) is a list scheduling and the second algorithm *LPT* (*Longest Processing Time*) is its refinement. A *list scheduling* produces a schedule of jobs based on a list as follows: When one of machines becomes available, first unexecuted job on the list is assigned to be processed on it.

For the maximum completion time scheduling problem with n independent jobs and m identical machines, ($n/m/I/C_{max}$ according to Lenstra et al. [6] and the objective is only different from $n/m/I/L_{max}$), Graham [1] obtained following results.

Theorem (Graham [1]). For $n/m/I/C_{max}$, let ω' be the maximum completion time of any list scheduling and ω^* that of optimal scheduling. Then

$$\frac{\omega'}{\omega^*}\leq 2-\frac{1}{m}$$

holds. Further for the list such that the jobs are ordered in nondecreasing order of processing times,

$$\frac{\omega'}{\omega^*}\leq \frac{4}{3}-\frac{1}{3m}$$

holds.

For the job set J, the maximum lateness of schedule constructed by some algorithm Π (approximation or exact) is defined by

$$L(J;\Pi)=\max_{1\leq i\leq n}\{C_i(\Pi)-d_i\},$$

where $C_i(\Pi)$ is the completion time of job J_i in a schedule constructed by algorithm Π. Especially, hereafter, $L(J;EDD)$, $L(J;LPT)$ and $L(J;\Pi^*)$ are used to denote their maximum latenesses by *EDD*, *LPT* and a certain optimum algorithm Π^* respectively.

Now we propose the approximation algorithm *EDD*.

Algorithm EDD: Assign jobs on machines in an order $J_1,J_2,\cdots,J_n$.

In order to analyze the worst case bound of this algorithm *EDD*, following lemma is needed.

Lemma 2.1. For any m and certain number K, $\bar{J}=(\bar{J}_1,\cdots,\bar{J}_n)$ is the minimal job set among ones for which modified relative deviation of *EDD* exceeds

K, i.e.,

$$\frac{L(\bar{J};EDD)-L(\bar{J};\Pi^*)}{L(\bar{J};\Pi^*)+d_n} > K$$

holds. Then $\bar{J}_n$ determines the maximum lateness of *EDD*, $L(\bar{J};EDD)$.

Proof: We prove this lemma by contradiction. We assume that job $\bar{J}_i$, $i<n$, determines the maximum lateness of algorithm *EDD*. Let $J'=(\bar{J}_1,\cdots,\bar{J}_i)$ be the subset of $\bar{J}$ obtained by eliminating jobs $\bar{J}_{i+1},\cdots,\bar{J}_n$. Clearly

$$L(\bar{J};EDD)=L(J';EDD),$$
$$L(\bar{J};\Pi^*)\geq L(J';\Pi^*)$$

and

$$d_n=\max_{1\leq j\leq n} d_j \geq \max_{1\leq j\leq i} d_j=d_i$$

hold. These imply

$$K< \frac{L(\bar{J};EDD)-L(\bar{J};\Pi^*)}{L(\bar{J};\Pi^*)+d_n} \leq \frac{L(J';EDD)-L(J';\Pi^*)}{L(J';\Pi^*)+d_i}$$

That is, we have a smaller job set J'. This contradicts the minimality of job set $\bar{J}$.

Thus job $\bar{J}_n$ determines the maximum lateness of algorithm *EDD*. □

Fully utilizing Lemma 2.1, we can derive a worst case bound of algorithm *EDD*.

Theorem 2.1. For any job set J,

$$\frac{L(J;EDD)-L(J;\Pi^*)}{L(J;\Pi^*)+d_n} \leq 1-\frac{1}{m}$$

holds. Moreover this bound is best possible.

Proof: Since the first half of our proof will be a proof by contradiction, it is necessary to develop relationship only for the smallest n for which the theorem may be violated. Thus we assume that J defines a minimal job set for which the theorem does not hold.

Now by Lemma 2.1, job J_n determines the maximum lateness of algorithm *EDD*, i.e.,

$$L(J;EDD)=C_n(EDD)-d_n, \tag{2.1}$$

where $C_n(EDD)$ is the completion time of job J_n in a schedule constructed by algorithm *EDD*. Since algorithm *EDD* is the list scheduling, Graham's theorem shows

$$(2.2)\quad \frac{C_n(EDD)}{C(\bar{\Pi}^*)} \le 2 - \frac{1}{m},$$

where $\bar{\Pi}^*$ is a certain exact algorithm minimizing maximum completion time and $C(\bar{\Pi}^*)$ is its maximum completion time.

$$(2.2)'\quad C_n(EDD) \le (2 - \frac{1}{m})C(\bar{\Pi}^*)$$

holds from (2.2). Substituting (2.2)' into (2.1),

$$(2.3)\quad L(J;EDD) \le (2 - \frac{1}{m})C(\bar{\Pi}^*) - d_n$$

results.

While we have

$$(2.4)\quad L(J;\Pi^*) \ge C(\bar{\Pi}^*) - d_n.$$

Hence (2.3) and (2.4) imply

$$(2.5)\quad L(J;EDD) - L(J;\Pi^*) \le (2 - \frac{1}{m})C(\bar{\Pi}^*) - d_n - (C(\bar{\Pi}^*) - d_n)$$
$$= (1 - \frac{1}{m})C(\bar{\Pi}^*).$$

Since $d_{\max}=d_n$, (2.4) and (2.5) together show

$$\frac{L(J;EDD) - L(J;\Pi^*)}{L(J;\Pi^*) + d_{\max}} \le \frac{(1 - \frac{1}{m})C(\bar{\Pi}^*)}{C(\bar{\Pi}^*)} = 1 - \frac{1}{m}.$$

This contradicts our assumption. Thus we have the desired worst case bound.

To see that this bound is best possible, consider three cases depending on m (mod4). (This example is almost the same as the example of Graham's theorem.)

Case 1. When $m=2r$, let the processing times and due dates be given by

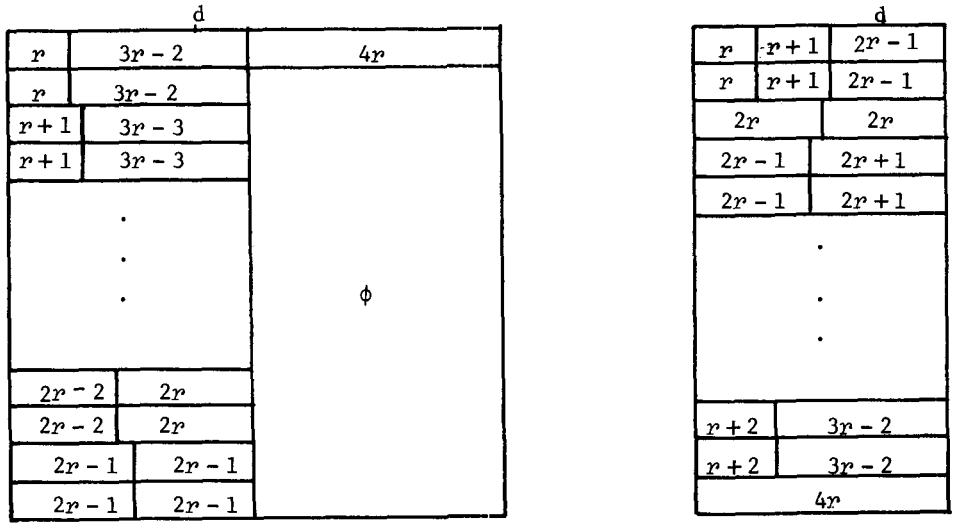
$$t_{2i-1} = t_{2i} = \begin{cases} r + (i-1) & \text{for } 1 \le i \le r, \\ 4r - (i+1) & \text{for } r+1 \le i \le 2r \end{cases}$$

and $t_{4r+1}=4r$, and $d_i=$d(=const.) for $1 \le i \le n=2m+1$. Since all the due dates are equal, we may assume that the i-th job assigned by algorithm *EDD*, $1 \le i \le 2m+1$, is job J_i. Then we obtain the schedule shown in Fig.2.1a. Since the optimal schedule by some exact algorithm Π^* becomes as shown in Fig.2.1b, and $L(J;\Pi^*)$ is $2m$-d, we have

$$\frac{L(J;EDD) - L(J;\Pi^*)}{L(J;\Pi^*) + d_{\max}} = 1 - \frac{1}{2r} = 1 - \frac{1}{m}.$$

Case 2. When $m=4r+1$, let the processing times and the due dates be given by

$$t_i = \begin{cases} 2r+2\lfloor \frac{i}{4} \rfloor - 1 & \text{for } 1 \le i \le 4r \\ 4r & \text{for } i = 4r+1 \\ 8r-2\lfloor \frac{i-1}{4} \rfloor + 1 & \text{for } 4r+2 \le i \le 8r+1 \\ 4r & \text{for } i = 8r+2 \\ 8r+2 & \text{for } i = 8r+3 \end{cases}$$



(a) approximate schedule (b) optimal schedule

Fig. 2.1 An example giving the tight bound of Theorem 2.1 in case $m=2r$.

and d_i=d for $1\leq i\leq n=2m+1$. Similarly to Case 1, we obtain $L(J;EDD)=16r+2-d$ and $L(J;\Pi^*)=8r+2-d$. Thus we have

$$\frac{L(J;EDD)-L(J;\Pi^*)}{L(J;\Pi^*)+d_{max}}=\frac{8r}{8r+2}=1-\frac{1}{m}.$$

Case 3. When $m=4r+3$, let the processing times and the due dates be given by

$$t_i=\begin{cases} r+[\frac{i}{4}] & \text{for } 1\leq i\leq 4 \\ 2r+1 & \text{for } i=4r+1,\ 4r+2,\ 4r+3,\ 8r+4,\ 8r+5,\ 8r+6 \\ 4r+2-[\frac{i}{4}] & \text{for } 4r+4\leq i\leq 8r+3 \\ 4r+3 & \text{for } i=8r+7, \end{cases}$$

and d_i=d for $1\leq i\leq n=2m+1$. Again similarly to Case 1 and Case 2, we have $L(J;EDD)=8r+5-d$ and $L(J;\Pi^*)=4r+3-d$. Hence we prove

$$\frac{L(J;EDD)-L(J;\Pi^*)}{L(J;\Pi^*)+d_{max}}=\frac{4r+2}{4r+3}=1-\frac{1}{m}.$$

This completes the proof of Theorem 2.1. □

Above worst case examples show that when the number of distinct due dates is small, this algorithm *EDD* is not so effective. In such a case, the maximum lateness may be greatly influenced on the maximum completion time rather than the due date. Now we propose another algorithm *LPT* which is more effective in such a situation. The algorithm *LPT* is a hybrid algorithm which consists of *LPT* rule and *EDD* rule.

Algorithm LPT :

Step 1. Assign the jobs to each machine according to the list such that the jobs are ordered in nondecreasing order of the processing times. (*LPT rule part*)

Step 2. On each machine, reorder the assigned jobs according to nondecreasing order the due dates. (*EDD rule part*)

Next Theorem 2.2 gives a worst case bound of algorithm *LPT*. But probably algorithm *LPT* has a better worst case bound than that of Theorem 2.2 in most cases.

Theorem 2.2. Let $L(J;LPT)$ and $L(J;\Pi^*)$ be the maximum lateness of schedule constructed by the algorithm *LPT* and some exact algorithm Π^* for the job set J respectively. Then

$$\frac{L(J;LPT)-L(J;\Pi^*)}{L(J;\Pi^*)+d_n} \le \min\left\{\frac{4}{3}-\frac{1}{3m}-\frac{mt_{\min}}{P},\ \frac{1}{3}-\frac{1}{3m}-\frac{m(d_n-d_1)}{P}\right\}$$

holds, where $t_{\min}=\min_{1\le i\le n} t_i$ and $P=\sum_{i=1}^{n} t_i$.

Proof: Let $\bar{\Pi}^*$ be any exact algorithm minimizing the maximum completion time for job set J and $C(\bar{\Pi}^*)$ the maximum completion time of $\bar{\Pi}^*$.

It is clear that

(2.6) $L(J;\Pi^*) \ge C(\bar{\Pi}^*) - d_n$

holds. Also we have

(2.7) $L(J;LPT) \le C(LPT) - d_1$,

where $C(LPT)$ is the maximum completion time of schedule constructed by algorithm *LPT*. From (2.6) and (2.7), we have

$$(2.8)\quad \frac{L(J;LPT)-L(J;\Pi^*)}{L(J;\Pi^*)+d_n} \le \frac{C(LPT)-C(\bar{\Pi}^*)}{C(\bar{\Pi}^*)} + \frac{d_n-d_1}{C(\bar{\Pi}^*)}.$$

Since

$$(2.9)\quad \frac{C(LPT)}{C(\bar{\Pi}^*)} \le \frac{4}{3}-\frac{1}{3m}$$

by Graham's theorem and $C(\bar{\Pi}^*)\ge P/m$, it holds that

$$(2.10)\quad \frac{L(J;LPT)-L(J;\Pi^*)}{L(J;\Pi^*)+d_n} \le \frac{1}{3}-\frac{1}{3m}+\frac{m}{P}(d_n-d_1).$$

Further, let $L(J;LPT)=C_k-d_k$, where C_k is the completion time of job J_k in the schedule obtained by algorithm *LPT*. It is clear that

(2.11) $L(J;\Pi^*) \ge t_{\min} - d_k$.

Since $C_k \leq C(LPT)$, (2.9) and (2.11) imply that

$$L(J;LPT) - L(J;\Pi^*) \leq C_k - d_k - (t_{\min} - d_k) \leq C(LPT) - t_{\min}$$
$$\leq (\frac{4}{3} - \frac{1}{3m})C(\bar{\Pi}^*) - t_{\min}.$$

From (2.6), we obtain

$$\frac{L(J;LPT) - L(J;\Pi^*)}{L(J;\Pi^*) + d_{\max}} \leq \frac{(\frac{4}{3} - \frac{1}{3m})C(\bar{\Pi}^*) - t_{\min}}{C(\bar{\Pi}^*)} = \frac{4}{3} - \frac{1}{3m} - \frac{t_{\min}}{C(\bar{\Pi}^*)}$$
$$\leq \frac{4}{3} - \frac{1}{3m} - \frac{t_{\min}}{P}.$$

Thus we prove Theorem 2.2. □

3. n/2/F/L_{max} Problem

In this section, we consider the n/2/F/L_{max} scheduling problem. A set of n independent jobs $J=(J_1,\cdots,J_n)$ is to be processed on two machines A and B. Each job J_i has the processing times a_i and b_i on machines A and B respectively and the due date d_i. Again we assume that $d_1 \leq d_2 \leq \cdots \leq d_n$. Each job J_i is to be processed on machine A and next on machine B after the processing on machine A. The objective is again to the maximum lateness.

3.1. Solvable case for n/2/F/L_{max}

General n/2/F/L_{max} scheduling problem is *NP*-complete. Hence we first consider the solvable case in the sense that an optimal schedule can be easily found. We assume that for $1 \leq i,j \leq n$,

(C) $d_i \leq d_j \longleftrightarrow \min(a_i,b_j) \leq \min(a_j,b_i)$.

EDD rule: EDD rule schedules jobs according to nonincreasing due dates, i.e., in the order $J_1,J_2,\cdots,J_n$.

Theorem 3.1. If (C) holds, EDD rule constructs an optimal schedule for n/2/F/L_{max}.

Proof: The completion time of job J_i scheduled by EDD rule, C_i, is given as follows;

$$C_i = \max_{1 \leq u \leq i} \{\sum_{j=1}^{u} a_j + \sum_{j=u}^{i} b_j\} = \max\{C_{i-1}, \sum_{j=1}^{i} a_j\} + b_i.$$

(See Johnson [3].) Then the lateness of job J_i, L_i, becomes as follows;

$$L_i = C_i - d_i = \max\{C_{i-1}, \sum_{j=1}^{i} a_j\} + b_i - d_i.$$

Similarly $L_{i+1} = C_{i+1} - d_{i+1} = \max\{C_i, \sum_{j=1}^{i+1} a_j\} + b_{i+1} - d_{i+1}$

$$= \max\{\max(C_{i-1}, \textstyle\sum_{j=1}^{i} a_j) + b_i, \sum_{j=1}^{i+1} a_j\} + b_{i+1} - d_{i+1}$$

$$= \max\{C_{i-1} + b_i, \textstyle\sum_{j=1}^{i} a_j + b_i, \sum_{j=1}^{i+1} a_j\} + b_{i+1} - d_{i+1}$$

Let L_i' and L_{i+1}' be the lateness of i-th and $(i+1)$st jobs in the schedule obtained by interchanging jobs J_i and J_{i+1}. (In the resulting schedule, i-th job is job J_{i+1} and $(i+1)$st job is job J_i.) That is, L_i' is the lateness of job J_{i+1} in the resulting schedule and L_{i+1}' that of job J_i. Thus we have

$$L_i' = \max\{C_{i-1}, \textstyle\sum_{j=1}^{i-1} a_j + a_i\} + b_{i+1} - d_{i+1},$$

$$L_{i+1}' = \max\{C_{i-1} + b_{i+1}, \textstyle\sum_{j=1}^{i-1} a_j + a_{i+1} + b_{i+1}, \sum_{j=1}^{i+1} a_j\} + b_i - d_i.$$

First we show that

$$\max(L_i, L_{i+1}) \le \max(L_i', L_{i+1}').$$

Since $d_i \le d_{i+1}$ and $\min(a_i, b_{i+1}) \le \min(a_{i+1}, b_i)$, we have $L_i \le L_{i+1}'$ and $L_i' \le L_{i+1}$. That is, we shall prove $L_{i+1} \le L_{i+1}'$.

Case (i) $a_i \le b_{i+1}$

Note that inequalities $a_i \le a_{i+1}$ and $a_i \le b_i$, also hold in this case.

Subcase (i-a) $L_{i+1} = C_{i-1} + b_i + b_{i+1} - d_{i+1}$

From $d_i \le d_{i+1}$, we have

$$L_{i+1} = C_{i-1} + b_i + b_{i+1} - d_{i+1} \le C_{i-1} + b_i + b_{i+1} - d_i \le L_{i+1}'.$$

(i-b) $L_{i+1} = \sum_{j=1}^{i} a_j + b_i + b_{i+1} - d_{i+1}$

Since $d_i \le d_{i+1}$ and $a_i \le a_{i+1}$, we prove

$$L_{i+1} = \textstyle\sum_{j=1}^{i} a_j + b_i + b_{i+1} - d_{i+1} \le \sum_{j=1}^{i-1} a_j + a_{i+1} + b_i + b_{i+1} - d_i \le L_{i+1}'.$$

(i-c) $L_{i+1} = \sum_{j=1}^{i+1} a_j + b_{i+1} - d_{i+1}$

By $d_i \le d_{i+1}$ and $a_i \le b_i$, we have

$$L_{i+1} = \textstyle\sum_{j=1}^{i+1} a_j + b_{i+1} - d_{i+1} \le \sum_{j=1}^{i-1} a_j + a_{i+1} + b_i + b_{i+1} - d_i \le L_{i+1}'.$$

Hence if $a_i \le b_{i+1}$, then we have $L_{i+1} \le L_{i+1}'$.

Case (ii) $b_{i+1} < a_i$

Note that $b_{i+1} \le a_{i+1}$ and $b_{i+1} \le b_i$ also hold in this case. We can prove $L_{i+1} < L_{i+1}'$ by the similar manner to *Case* (i).

Therefore if $\min(a_i, b_{i+1}) \le \min(a_{i+1}, b_i)$ and $d_i \le d_{i+1}$, then $\max(L_i, L_{i+1}) \le \max(L_i', L_{i+1}')$.

Let C_k' and L_k' be the completion time and lateness of job J_k in the schedule obtained by interchanging jobs J_i and J_{i+1}. For $k<i$, it is clear that $C_k'=C_k$ and $L_k'=L_k$.

Since for $k>i+1$, $C_k'\geq C_k$ holds by virtue of Johnson's rule, we have $L_k'\geq L_k$.

Thus since (C) holds among all jobs and the relation (C) is transitive, we prove the theorem by repeating pairwise interchanging of adjacent jobs. □

Since this problem is *NP*-complete, it seems likely that an efficient algorithm does not and will not exist for this problem. Hence enumerative type methods such as branch-and-bound ones may be the only available ones for obtaining optimal solution.

One may suspect that we can decrease the number of enumerations by applying Theorem 3.1 to a number of job pairs for some of which the relation (C) holds. The following example shows the case that the conjecture fails.

Example 3.1. Let $J=(J_1,J_2,J_3)$,

$a_1=2$, $b_1=5$, $d_1=55$,

$a_2=10$, $b_2=1$, $d_2=50$,

$a_3=4$, $b_3=100$, and $d_3=60$.

In this example, though $\min(a_1,b_3)\leq\min(a_3,b_1)$ and $d_1\leq d_3$, the optimal schedule is given in Fig.3.1. The maximum lateness in the optimal schedule is $L^*_{max}=44$.

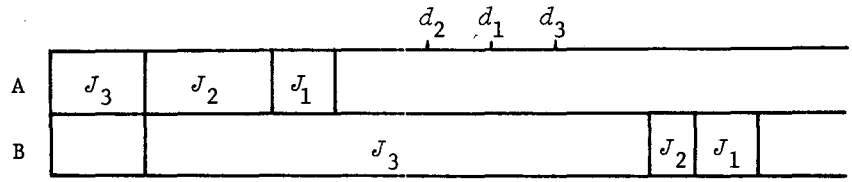


Fig. 3.1 An optimal schedule of Example 3.1.

3.2. Bound on approximation algorithm for n/2/F/L_{max}

In subsection 3.1, we showed the solvable case n/2/F/L_{max}. Unfortunately general one is *NP*-complete. Therefore in this section, we give an approximation algorithm and show how it behaves on the worst case. We call the algorithm for the problem n/2/F/L_{max} based on *EDD* rule algorithm *FEDD*, which assigns the jobs according to *EDD* rule. We first prove Lemma 3.1 giving the bound of the maximum completion time when a set of jobs is scheduled by algorithm *FEDD*.

Lemma 3.1. Let C' be the maximum completion time of schedule constructed

by algorithm *FEDD* and C^* that of schedule constructed by Johnson's rule[†]. (See Johnson [3].) Then we have

$$\frac{C'}{C^*} \leq 2 .$$

Proof: It is clear that

$$C^* \geq \max(\sum_{i=1}^{n} a_i, \sum_{i=1}^{n} b_i).$$

Also it follows that

$$C' \leq \sum_{i=1}^{n} (a_i + b_i) \leq 2\max(\sum_{i=1}^{n} a_i, \sum_{i=1}^{n} b_i) \leq 2C^*.$$

Thus we prove

$$\frac{C'}{C^*} \leq 2. \quad \square$$

By using this lemma, we obtain the bound on algorithm *FEDD*.

Theorem 3.2. Let $L'_{\max}$ and $L^*_{\max}$ be the maximum lateness of schedule constructed by applying algorithm *FEDD* and any optimal algorithm for the problem $n/2/F/L_{\max}$ respectively. Then we have

$$\frac{L'_{\max} - L^*_{\max}}{L^*_{\max} + d_n} \leq 1.$$

Further this bound is asymptotically best possible.

Proof: Similarly to $n/m/I/L_{\max}$, we may consider only the case $L'_{\max}=C'-d_n$, where C' is the same as defined in Lemma 3.1. Let $L'_{\max}=C'-d_n$. It is clear that

$$L^*_{\max} \geq C^* - d_n.$$

Thus

$$\frac{L'_{\max} - L^*_{\max}}{L^*_{\max} + d_n} \leq \frac{C' - d_n - (C^* - d_n)}{C^*} = \frac{C'}{C^*} - 1,$$

Since $C'/C^* \leq 2$ by Lemma 3.1, we prove

$$\frac{L'_{\max} - L^*_{\max}}{L^*_{\max} + d_n} \leq 1.$$

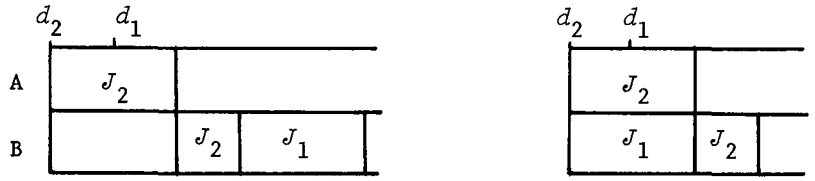
The following example shows that this bound is asymptotically best possible.

Let $a_1=0$, $b_1=K$, $d_1=\varepsilon(>0)$, $a_2=K$, $b_2=\varepsilon$, and $d_2=0$, where K is arbitrary positive constant number. For this instance, the approximate and the optimal

[†]Note that the optimal schedule of $n/2/F/C_{\max}$, which is in turn to seek a schedule minimizing the maximum completion time on 2-machine flow shop, is obtained by applying Johnson's rule.

schedule are given in Fig.3.2 (a) and (b) respectively. Then we have $L'_{max} = 2K+\varepsilon-\varepsilon=2K$ and $L^*_{max}=K+\varepsilon-0=K+\varepsilon$. Therefore we prove

$$\frac{L'_{max} - L^*_{max}}{L^*_{max} + d_n} = \frac{K - \varepsilon}{K + 2\varepsilon} \longrightarrow 1 \quad (\varepsilon \to 0)$$



(a) approximate schedule (b) optimal schedule

Tig. 3.2 An asymptotically tight example

This example completes the proof of the theorem. □

Acknowledgement

We are thankful to the anonymous referees for their helpful comments and suggestions.

References

[1] Graham, R. L., Lawler, E. L., Lenstra, J. K. and Rinnooy Kan, A. H. G.: Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics*, Vol.5 (1979), 287-326.

[2] Jackson, J. R.: Scheduling a Production Line to Minimize Maximum Tardiness. *Research Report 43, Management Science Research Project*, University of California, Los Angels, 1955.

[3] Johnson, S. M.: Optimal Two- and Three-Stage Production Schedules with Setup Times Included. *Naval Research Logistics Quarterly*, Vol.1, No.1 (1954), 61-68.

[4] Kise, H., Ibaraki, T. and Mine, H.: Performance Analysis of Six Approximation Algorithms for the One-Machine Maximum Lateness Scheduling Problem with Ready Times. *Journal of the Operations Research Society of Japan*, Vol.22, No.3 (1979), 205-224.

[5] Lawler, E. L.: Optimal Sequencing of a Single Machine Subject to Precedence Constraints. *Management Science*, Vol.19, No.5 (1973), 544-546.

[6] Lenstra, J. K., Rinnooy Kan, A. H. G. and Brucker, P.: Complexity of Machine Scheduling Problems. *Annals of Discrete Mathematics*, Vol.1 (1977), 343-362.

Teruo MASUDA: Department of Mathematics,
College of Integrated Arts and Sciences,
University of Osaka Prefecture, Sakai,
Osaka, 591, Japan.