Journal of the Operations Research Society of Japan
Vol. 22, No. 3, September 1979

© 1979 The Operations Research Society of Japan

PERFORMANCE ANALYSIS OF SIX APPROXIMATION ALGORITHMS FOR THE ONE-MACHINE MAXIMUM LATENESS SCHEDULING PROBLEM WITH READY TIMES

Hiroshi Kise
Kyoto Institute of Technology
Toshihide Ibaraki
and
Hisashi Mine
Kyoto University

(Received November 24, 1978; Revised March 12, 1979)

Abstract Six approximation algorithms for the one-machine scheduling problem with ready and due times to minimize the maximum lateness are analyzed. The performance is measured by the relative deviation of approximate values to optimal ones. Best possible upper bounds on the worst case performance of all six algorithms are derived. The average performance is also examined by solving randomly generated problems; one of the six algorithms outperforms others and keeps the average relative deviation within 2%.

1. Introduction

This paper considers several approximation algorithms for a class of n-job one machine scheduling problem ($1|r_j|L_{max}$ according to the terminologies in [5]) such that (i) each job k has ready time r(k), positive processing time p(k) and due time d(k), (ii) no preemption is allowed, and (iii) the objective is to minimize the maximum lateness.

Although many exact algorithms have been proposed for this problem [2, 3, 4, 10, 13, 14, 16], they are all based on enumerative methods and their computation time seems to grow exponentially with the size of the problems. Furthermore, this problem is known to be NP-complete [9, 11]. This strongly suggests that there is no polynomially bounded algorithm. Thus, it is important to find efficient approximation algorithms with guaranteed accuracies.

In Section 2 we propose six approximation algorithms. The first two algorithms order the jobs simply according to ready times and due times, respectively. The third one chooses the better of the above two. The fourth one is proposed by Schrage [15] and the fifth one is to apply Schrage's algorithm to the transformed problem called the reverse problem. These two algorithms were used in exact branch-and-bound algorithms as a means to obtain upper bounds of an optimal schedule [10, 13]. The last one chooses the better of the fourth and the fifth. It will be shown that all the six algorithms can be executed in $O(n \log n)$ time, where n is the number of jobs.

In Section 3 $(L(\pi, Q)-L(\omega, Q))/L(\omega, Q)$, the relative deviation of approximate solutions, obtained by the above six algorithms is analyzed. Here $L(\pi, Q)$ is the maximum lateness of a schedule π for a problem instance Q and ω stands for an optimal schedule. The first main result (Theorem 3.5) is that the relative deviation is not larger than $2-(2/P)$ for any schedule and any problem instance,where P is the sum of total processing times. This bound is the best possible. The second result (Theorem 3.6) shows, however, that relative deviations for the proposed approximation algorithms are always less than 1 and the best possible upper bound for each approximation algorithm is explicitly given in term of P. The proof of Theorem 3.6 is lengthy and given in Section 4. In Section 5, we give results of computational experiments for various types of problems which are randomly generated. Comparing the approximate schedules with the exact optimal schedules obtained by a branch-and-bound algorithm, it appears that the quality of the approximate schedules is quite high (much better than that the above worst case analysis indicates). It is also seen that one of the six approximation algorithms outperforms others. The average relative deviation by the best algorithm is kept within 2% for all the tested cases.

2. Approximation Algorithms

A problem instance of our scheduling problem is specified by ready times $R=(r(1), \ldots, r(n))$, processing times $P=(p(1), \ldots, p(n))$ and due times $D=(d(1), \ldots, d(n))$. This is denoted by $Q=Q(R, P, D)$. A schedule is given by a permutation of n jobs $\pi=(\pi_1, \ldots, \pi_n)$, where π_k is the k-th job to be processed. Finishing time $f_\pi(\pi_k)$ of job π_k in schedule π is given by

(2.1) $$f_\pi(\pi_k) = \begin{cases} r(\pi_1)+p(\pi_1), \ k=1 \\ \max\{f_\pi(\pi_{k-1}), r(\pi_k)\}+p(\pi_k)=\max_{1\le i\le k}\{r(\pi_i)+ \\ \qquad \Sigma_{j=i}^{k} p(\pi_j)\}, \ k=2,3,\ldots, n, \end{cases}$$

and the maximum lateness is defined by

(2.2) $$L(\pi, Q) = \max_{1\le k\le n}\{f_\pi(\pi_k)-d(\pi_k)\}.$$

A schedule ω minimizing the maximum lateness is called optimal.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

The problem instance $\bar{Q}=Q(-D, P, -R)$ is called the reverse of $Q=Q(R, P, D)$ [10, 12], where $-R=(-r(1), \ldots, -r(n))$ and $-D=(-d(1), \ldots, -d(n))$. Furthermore, for $\pi=(\pi_1, \pi_2, \ldots, \pi_n)$, $\overleftarrow{\pi}=(\overleftarrow{\pi}_1, \overleftarrow{\pi}_2, \ldots, \overleftarrow{\pi}_n)=(\pi_n, \pi_{n-1}, \ldots, \pi_1)$ is called the reverse of π. Then, we have by (2.1) and (2.2) that

$$(2.3)\qquad \begin{aligned} L(\pi, \bar{Q}) &= \max_{1\le j\le n}\max_{1\le i\le j}\{-d(\pi_i)+\Sigma_{k=i}^{j}p(\pi_k)+r(\pi_j)\} \\ &= \max_{1\le i\le n}\max_{i\le j\le n}\{r(\overleftarrow{\pi}_i)+\Sigma_{k=i}^{j}p(\overleftarrow{\pi}_k)-d(\overleftarrow{\pi}_j)\} = L(\overleftarrow{\pi}, Q). \end{aligned}$$

Six approximation algorithms for $Q=Q(R, P, D)$ are now introduced.

Algorithm J (Jackson [8]): Order jobs in nondecreasing order of due times, breaking ties by prefering the smaller job numbers. The obtained schedule $\zeta=(\zeta_1, \ldots, \zeta_n)$ satisfies

$$(2.4)\qquad d(\zeta_1)\le d(\zeta_2)\le\ldots\le d(\zeta_n).$$

It should be noted that schedule ζ is optimal if $r(1)=r(2)=\ldots=r(n)$ [8].

Algorithm $\bar{J}$: Order jobs in nondecreasing order of ready times, breaking ties by prefering the smaller job numbers. The obtained schedule $\eta=(\eta_1, \ldots, \eta_n)$ satisfies

$$(2.5)\qquad r(\eta_1)\le r(\eta_2)\le\ldots\le r(\eta_n).$$

Clearly Algorithm $\bar{J}$ is equivalent to the following steps:

Step 1. Apply Algorithm J to $\bar{Q}$ with a modification that ties are broken by prefering the larger job numbers.

Step 2. Take the reverse of the schedule constructed in Step 1.

Algorithm mJ: Apply both algorithms J and $\bar{J}$ and select the better one, i.e., if $L(\zeta, Q)\le L(\eta, Q)$, then select schedule ζ, otherwise schedule η.

Algorithm S (Schrage [15]): Jobs are scheduled one by one from the first one: a job with the minimum due time is selected among the available jobs (i.e., those which are ready), breaking ties by prefering longer processing times (breaking ties arbitrarily, if the processing times are also the same).

Step 1. Let K be any ordered job set $(1, 2, \ldots, n)$ arranged in the nondecreasing order of ready times, i.e., $r(1)\le r(2)\le\ldots\le r(n)$, and let $i\leftarrow 0$, $K_i\leftarrow\phi$ and $f\leftarrow 0$.

Step 2. $i\leftarrow i+1$, $K_i\leftarrow K_{i-1}\cup\{k|k\in K, r(k)\le f\}$. If $K_i=\phi$, then $K_i\leftarrow$\{the first job in K\} and $f\leftarrow$its ready time. Find $\theta_i\in K_i$ such that $d(\theta_i)=\min_{k\in K_i}d(k)$ breaking ties by prefering the longer processing times. Go to Step 3, after letting $K\leftarrow K-\{\theta_i\}$, $K_i\leftarrow K_i-\{\theta_i\}$ and $f\leftarrow f+p(\theta_i)$.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Step 3. If $i=n$, halt with schedule $\theta=(\theta_1, \theta_2, \ldots, \theta_n)$. Otherwise return to Step 2.

It should be noted that $L(\theta, Q) \le L(\zeta, Q)$ always holds. Furthermore, θ is optimal if $p(k)=1$, $k=1, \ldots, n$, [7].

Algorithm $\bar{S}$: Jobs are scheduled one by one from the last one; each time a job with the maximum ready time is selected from among the remaining jobs with due times not smaller than $s(\mu_{n-k+1})$, breaking ties by prefering shorter processing times, where the last k jobs $\mu_{n-k+1}, \ldots, \mu_{n-1}, \mu_n$ are now scheduled and

$$s(\mu_k) = \begin{cases} d(\mu_n)-p(\mu_n), & k=n \\ \min\{s(\mu_{k+1}), d(\mu_k)\}-p(\mu_k), & k=1, 2, \ldots, n-1. \end{cases}$$

Formally, algorithm $\bar{S}$ is equivalent to the following steps:

Step 1. Apply Algorithm S to $\bar{Q}$.

Step 2. Take the reverse of the schedule constructed in Step 1.

The obtained schedule is denoted by $\mu=(\mu_1, \ldots, \mu_n)$.

Algorithm mS: Apply both algorithms S and $\bar{S}$ and select the better one, i.e., if $L(\theta, Q) \le L(\mu, Q)$, then select schedule θ, otherwise schedule μ.

3. Relative Deviation

To evaluate the quality of approximate solutions π of problem Q, various measures such as the absolute deviation

$$(L(\pi, Q)-L(\omega, Q)),$$

where ω denotes an optimal schedule, and the relative deviation

$$(L(\pi, Q)-L(\omega, Q))/L(\omega, Q) \tag{3.1}$$

have been customarily used. For our scheduling problems, however, these measures exhibit a shortcoming that they give different values to two problems such that one is obtained from the other by applying a simple transformation which changes neither the optimal schedule nor the approximate schedule. The following proposition describes two such transformations we have in mind.

Proposition 3.1. (1) For $Q=Q(R, P, D)$ and a positive number w, let $wQ=Q(wR, wP, wD)$ where $wR=(wr(1), \ldots, wr(n))$, $wP=(wp(1), \ldots, wp(n))$ and $wD=(wd(1), \ldots, wd(n))$. Then, Q and wQ have the same set of optimal schedules, and the same approximate schedule is obtained by each algorithm X discussed in Section 2.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

(2) For Q=Q(R, P, D) and two real numbers u and v, let Q'=Q(R', P, D') where R'=(r(1)-u, ..., r(n)-u) and D'=(d(1)-v, ..., d(n)-v). Then Q and Q' have the same set of optimal schedules, and the same approximate schedule is obtained by each algorithm X discussed in Section 2.

It should be noted that wQ and Q may have different values of the absolute deviation and that Q' and Q may have different values of the relative deviation defined by (3.1). In addition, we note that the relative deviation (3.1) cannot be defined if the optimal value is 0, that is certainly possible in our scheduling problems.

To avoid these difficulties, we propose the following modified relative deviation of a schedule π constructed by algorithm X for a problem instance Q=Q(R, P, D).

(3.2) $$A(X, Q) = (L(\pi, Q)-L(\omega, Q))/(L(\omega, Q)-R_{min}+D_{max}),$$

where $R_{min}=\min_{1\leq k\leq n} r(k)$ and $D_{max}=\max_{1\leq k\leq n} d(k)$. To justify this measure, note first that the denominator of (3.2) is always positive:

(3.3) $$L(\omega, Q)-R_{min}+D_{max}\geq P>0, \text{ where } P=\Sigma_{k=1}^{n} p(k),$$

since $f_\omega(\omega_n)\geq R_{min}+P$ by (2.1) and $L(\omega, Q)\geq f_\omega(\omega_n)-d(\omega_n)\geq f_\omega(\omega_n)-D_{max}$ by (2.2). Also it is easy to prove the following properties for all the six approximation algorithms X defined in Section 2.

$$A(X, Q) = A(X, wQ)$$
$$A(X, Q) = A(X, Q'),$$

where wQ and Q' are defined in proposition 3.1. Namely the values of A(X, Q) do not change by the transformations given in Proposition 3.1.

In view of these we may treat Q(R', P, D') instead of a given problem instance Q(R, P, D), where R' and D' are defined by $u=R_{min}$ and $v=D_{max}$ as in Proposition 3.1 (2). In other words, we may assume without loss of generality that

(3.4) $$R_{min} = D_{max} = 0.$$

In this case the relative deviation (3.2) reduces to the ordinary relative deviation.

(3.5) $$A(X, Q) = (L(\pi, Q)-L(\omega, Q))/L(\omega, Q).$$

As a result of (3.4), we assume throughout this paper that r(k), p(k) and d(k)(k=1, 2, ..., n) are all integers such that

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

(3.6) $r(k)\varepsilon Z^{+}\cup\{0\},\ p(k)\varepsilon Z^{+},\ d(k)\varepsilon Z^{-}\cup\{0\},\ k=1,\ 2,\ \ldots,\ n,$

where Z^{+} is the set of positive integers and Z^{-} is the set of negative integers. Furthermore, the set of problems Q is partitioned into sub-classes U_P such that

(3.7) $U_P = \{Q(R,\ P,\ D) \text{ subject to (3.4) and (3.6) and } \Sigma_{k=1}^{n} p(k)=p\}.$

We shall derive for each algorithm X and P an upper bound $W_X(P)$ of the worst case relative deviation, which is best possible in the sense that

$$W_X(P) = \sup_{Q\varepsilon U_P} A(X,\ Q).$$

is attained for any $P>n\geq 2$. Before proceeding to the main results (Theorems 3.5 and 3.6), some preliminary results are given as lemmas.

Lemma 3.2. For $Q=Q(R,\ P,\ D)$ and any schedule π, it holds (by (2.1) and (2.2)) that

$$L(\pi,\ Q)\leq R_{max}+P-D_{min},$$

where $R_{max}=\max_{1\leq k\leq n} r(k)$ and $D_{min}=\min_{1\leq k\leq n} d(k)$.

Lemma 3.3. For $Q=Q(R,\ P,\ D)$ and any subset K of n jobs, an optimal schedule ω satisfies

$$L(\omega,\ Q)\geq\min_{k\varepsilon K} r(k)+\Sigma_{k\varepsilon K} p(k)-\max_{k\varepsilon K} d(k).$$

In particular, for any k

$$L(\omega,\ Q)\geq r(k)+p(k)-d(k)\geq r(k)+1-d(k).$$

Lemma 3.4. $W_J(P), W_{\bar{J}}(P), W_S(P)$ and $W_{\bar{S}}(P)$ satisfy $W_J(P)=W_{\bar{J}}(P)$ and $W_S(P)=W_{\bar{S}}(P)$.

Proof. It follows from (3.7) that $Q=Q(R,\ P,\ D)\varepsilon U_P$ implies $\bar{Q}=Q(-D,\ P,\ -R)$ εU_P. Then

$$\begin{aligned} W_J(P) &= \max_{Q\varepsilon U_P} A(J,\ Q) = \max_{\bar{Q}\varepsilon U_P} A(J,\ \bar{Q}) \\ &= \max_{\bar{Q}\varepsilon U_P} (L(\overleftarrow{\eta},\ \bar{Q})-L(\overleftarrow{\omega},\ \bar{Q}))/L(\overleftarrow{\omega},\ \bar{Q}) \text{ (by definition of } \bar{J}) \\ &= \max_{Q\varepsilon U_P} (L(\eta,\ Q)-L(\omega,\ Q))/L(\omega,\ Q) \text{ (by(2.3))} \\ &= \max_{Q\varepsilon U_P} A(\bar{J},\ Q) = W_{\bar{J}}(P). \end{aligned}$$

S and $\bar{S}$ can be similarly treated.

Theorem 3.5. For any $Q=Q(R,\ P,\ D)\varepsilon U_P$ and any algorithm X (possibly different from the above six algorithms),

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$A(X, Q) \le 2-(2/P)$$

holds. This bound $W_X(P)=2-(2/P)$ is best possible for some algorithm X.

Proof. We assume $P \ge 2$ since otherwise $P=1$ and $n=1$, hence $A(X, Q)=0$. $L(\omega, Q) \ge \max_{1 \le k \le n}(r(k)+p(k)-d(k))$ follows from Lemma 3.3 and $L(\omega, Q) \ge P$ by (3.3). Hence

$$L(\omega, Q) \ge \max\{M, P\}, \tag{3.8}$$

where $M=\max\{R_{max}, -D_{min}\}+1$. If $M>P$, then $L(\omega, Q) \ge M$ by (3.8) and $L(\pi, Q) \le 2M+P-2$ by Lemma 3.2 for any schedule π. Therefore,

$$A(X, Q) = (L(\pi, Q)-L(\omega, Q))/L(\omega, Q) \le (M+P-2)/M = 1+(P-2)/M < 2-(2/P).$$

If $M \le P$, then $L(\omega, Q) \ge P$ by (3.8) and $L(\pi, Q) \le 3P-2$ by Lemma 3.2. Thus $A(X, Q) \le 2-(2/P)$. Therefore, $2-(2/P)$ is an upper bound of $A(X, Q)$.

Let $Q=Q(R, P, D)$ be such that $R=(P-1, 0, \ldots, 0)$, $P=(1, p(2), \ldots, p(n-1), 1)$ and $D=(0, \ldots, 0, 1-P)$, where $\Sigma_{k=2}^{n-1} p(k)=P-2$. Assume that algorithm X constructs $\pi=(1, 2, \ldots, n)$ with $L(\pi, Q)=3P-2$. However $\omega=(n, 2, 3, \ldots, n-1, 1)$ holds by $L(\omega, Q)=P$ and (3.3). Thus, $A(X, Q)= 2-(2/P)$, i.e., $2-(2/P)$ is best possible for this algorithm X.

Theorem 3.6. Computation times and best possible upper bounds $W_X(P)$ on relative deviations for algorithms J, $\bar{J}$, mJ, S, $\bar{S}$ and mS applied to $Q(R, P, D) \in U_P$ are given in Table 3.1.

The proof will be given in the next section. Comparing this result with Theorem 3.5, we see that the effect of using approximation algorithms defined in Section 2 is to reduce the worst case bounds of relative deviations by at least half. It should be noted, however, that the average performance of these approximation algorithms is much better than what this worst case result may suggest, as shown by computational results given in Section 5.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Table 3.1. Best Possible Upper Bounds on Relative Deviations for Approximation Algorithms

Algorithm	$W_X(P)$: Best Possible Upper Bounds on Relative Deviation	Computation Time
J, $\bar{J}$	1-(1/P)	0(nlogn)
mJ	1-(2/P)	0(nlogn)
S, $\bar{S}$	1-(3/(P+1))	0(nlogn)
mS	1-(5/(P+2))	0(nlogn)

4. Proof of Theorem 3.6.

In the following, $L(\pi, Q)$ and $A(X, Q)$ are respectively abbreviated to L_π and A_X for simplicity.

Proof of Computation Time. The computation time of J and $\bar{J}$ is dominated by the sorting time of n jobs, which takes 0(n log n) time [1]. Similarly, Algorithm mJ also requires 0(n log n) time.

Now consider Algorithm S. In Step 1, n jobs are sorted according to ready times to obtain K, which requires 0(n log n) times. To execute Step 2, jobs in K_i are sorted in an appropriate data structure such as heap, so that a minimal job in K_i with respect to the lexicographical order of due time and processing time can be selected and deleted in $0(\log |K_i|)$ ($\leq 0(\log n)$) time, and an addition of a job to K_i is also done in $0(\log |K_i|)$ time. Since n jobs are successively sent from K to K_i according to the order in K during the iterated executions of Step 2, the total time to adjust K is 0(n) and the total time to maintain K_i (i.e., addition and deletion of jobs to K_i) is 0(n log n). Other operations are obviously done in less than 0(n log n) time, and the time complexity of S is 0(n log n).

Similarly, Algorithm $\bar{S}$ and mS also require 0(n log n) time.

Proof of $W_J(P)$ (and $W_{\bar{J}}(P)$). We treat only $W_J(P)$ because the result can be extended to $W_{\bar{J}}(P)$ by Lemma 3.4. Let $\zeta=(\zeta_1, \ldots, \zeta_n)$ be the schedule constructed by Algorithm J. By definition, ζ has $i\leq j$ such that

(4.1) $$L_\zeta=r(\zeta_i)+\Sigma_{k=i}^{j}p(\zeta_k)-d(\zeta_j).$$

It follows from Lemma 3.3 and (2.4)that $L_\omega\geq r(\zeta_i)+1-d(\zeta_i)\geq r(\zeta_i)+1-d(\zeta_j)$. Thus,

(4.2) $$L_\zeta-L_\omega\leq\Sigma_{k=i}^{j}p(\zeta_k)-1\leq P-1, \text{ i.e., } A_J=(L_\zeta-L_\omega)/L_\omega\leq(P-1)/P,$$

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

since $L_\omega \geq P$ by (3.8). Finally, consider a problem instance $Q(R, P, D)$ such that $R=(P-1, 0, \ldots, 0)$, $P=(1, \ldots, 1, P-n+1)$ and $D=(0, \ldots, 0)$. Then $\zeta=(1, 2, \ldots, n)$ and $L_\zeta=2P-1$, while $\omega=(2, 3, \ldots, n, 1)$ and $L_\omega=P$. Thus, $W_J(P)=(P-1)/P$ is the best possible upper bound.

Proof of $W_{mJ}(P)$. Let $\eta=(\eta_1, \ldots, \eta_n)$ be the schedule constructed by Algorithm $\bar{J}$. It is sufficient to show that $(L_\zeta-L_\omega)/L_\omega>1-(2/P)$ implies $(L_\eta-L_\omega)/L_\omega\leq 1-(2/P)$. Thus assume

$$L_\zeta-L_\omega>(1-(2/P))L_\omega\geq P-2 \text{ (by } L_\omega\geq P),$$

i.e., $L_\zeta-L_\omega=P-1$ by (4.2). Then $\Sigma_{k=i}^{j}p(\zeta_k)=P$ holds in (4.2), i.e., $i=1$ and $j=n$. Thus, $L_\omega+P-1=L_\zeta=r(\zeta_1)+P-d(\zeta_n)$ by (4.1), while $L_\omega+P-1\geq r(\zeta_1)+1-d(\zeta_1)+P-1=r(\zeta_1)+P-d(\zeta_1)$ by Lemma 3.3. Therefore, $d(\zeta_1)\geq d(\zeta_n)$ and this reduces (2.4) to

$$d(\zeta_1) =d(\zeta_2) =\ldots =d(\zeta_n).$$

It is however easy to show that $L_\eta=L_\omega$ (i.e., $(L_\eta-L_\omega)<(1-(2/P))L_\omega$) holds in this case. Thus $W_{mJ}(P)=(1-2/P)$ is an upper bound. Finally, consider a problem instance $Q(R, P, D)$ such that $n=3$, $R=(P-1, 0, 0)$, $P=(1, P-2, 1)$ and $D=(0, 0, 1-P)$. Then $\zeta=(3, 1, 2)$, $\eta=(2, 3, 1)$ and $L_\zeta=L_\eta=2P-2$, while $\omega=(3, 2, 1)$ and $L_\omega=P$. Thus $A_{mJ}=1-(2/P)$ holds and $W_{mJ}(P)=1-(2/P)$ is best possible.

Proof of $W_S(P)$ (and $W_{\bar{S}}(P)$). Let $\theta=(\theta_1, \ldots, \theta_n)$ be the schedule constructed by Algorithm S, and let

$$L_\theta = f_\theta(\theta_j)-d(\theta_j) = r(\theta_i)+\Sigma_{k=i}^{j}p(\theta_k)-d(\theta_j), \tag{4.3}$$

where j is the maximum index satisfying (4.3) and i is the minimum one satisfying (4.3) for the j. We may assume $L_\theta>L_\omega$. In this case $i<j$ (since $i=j$ implies $L_\theta=L_\omega$ by Lemma 3.3) and $f_\omega(\theta_j)<f_\theta(\theta_j)$ holds since $f_\omega(\theta_j)-d(\theta_j)\leq L_\omega<L_\theta=f_\theta(\theta_j)-d(\theta_j)$. Thus, there are some jobs in $\{\theta_i, \theta_{i+1}, \ldots, \theta_j\}$ which precede job θ_j in θ but follow in ω. Now let θ_g be the job in $\{\theta_i, \ldots, \theta_j\}$ which is processed last in ω. It is known that Algorithm S minimizes the schedule length (the time when all jobs finish) [10, 12]. Thus, $f_\theta(\theta_j)$ is the minimum schedule length for the job set $\{\theta_i, \theta_{i+1}, \ldots, \theta_j\}$. Therefore, we have $f_\omega(\theta_g)\geq f_\theta(\theta_j)$, and hence

$$d(\theta_g)>d(\theta_j) \tag{4.4}$$

by $f_\omega(\theta_g)-d(\theta_g)\leq L_\omega<L_\theta=f_\theta(\theta_j)-d(\theta_j)$. θ_g can precede θ_j in θ in spite of (4.4) only if θ_j is not available when job θ_{g-1} finishes, i.e.,

$$0\leq r(\theta_i)\leq f_\theta(\theta_{g-1})<r(\theta_j). \tag{4.5}$$

Thus, $L_\omega\geq r(\theta_j)+p(\theta_j)-d(\theta_j)\geq f_\theta(\theta_{g-1})+1+p(\theta_j)-d(\theta_j)$ by Lemma 3.3, and hence

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

(4.6) $L_\theta-L_\omega\le[f_\theta(\theta_{g-1})+\Sigma_{k=g}^{j}p(\theta_k)-d(\theta_j)]-[f_\theta(\theta_{g-1})+1+p(\theta_j)-d(\theta_j)]$
$=\Sigma_{k=g}^{j-1}p(\theta_k)-1\le P-p(\theta_j)-1\le P-2.$

It is also known [12, 13] that

$$L_\omega \ge \begin{cases} f_\theta(\theta_j)-D_{max} & \text{if } d(\theta_j)=D_{max} \\ f_\theta(\theta_j)-D_{max}+1 & \text{otherwise.} \end{cases}$$

Thus, by (4.4) we have

(4.7) $L_\omega\ge f_\theta(\theta_j)+1.$

Furthermore, $d(\theta_j)\ge r(\theta_j)+1-L_\omega\ge 2-L_\omega$ by Lemma 3.3 and $r(\theta_j)\ge 1$ (see (4.5)). Therefore,

(4.8) $L_\theta-L_\omega=f_\theta(\theta_j)-d(\theta_j)-L_\omega\le(L_\omega-1)-(2-L_\omega)-L_\omega=L_\omega-3.$

Now, if $L_\omega=P$, then $A_S\le(P-3)/P<1-(3/(P+1))$ by (4.8). If $L_\omega\ge P+1$, $A_S\le(P-2)/(P+1)=1-(3/(P+1))$ by (4.6). Therefore, $W_S(P)=1-(3/(P+1))$ is an upper bound.

Finally consider a problem instance Q(R, P, D) such that n=2, R=(0, 1), P=(P-1, 1) and D=(0, 1-P). Then θ=(1, 2) and $L_\theta=2P-1$, while ω=(2, 1) and $L_\omega=P+1$. Thus, $A_S=(P-2)/(P+1)$, i.e., $W_S(P)=1-(3/(P+1))$ is best possible.

Proof of $W_{mS}(P)$. We show that two assumptions

(4.9) $A_S=(L_\theta-L_\omega)/L_\omega>(P-3)/(P+2)$ $(=W_{mS}(P))$

(4.10) $A_{\bar{S}}=(L_\mu-L_\omega)/L_\omega>(P-3)/(P+2)$ $(=W_{mS}(P))$

can not hold true simultaneously, where θ and μ are schedules constructed by algorithms S and $\bar{S}$, respectively. We may assume that

(4.11) $n\ge 3$, $P>n$,

since clearly $L_\omega=\min\{L_\theta, L_\mu\}$ if $n\le 2$, and $L_\theta=L_\omega$ if P=n (i.e., p(k)=1, k=1, ..., n) as shown in [7]. Let L_θ be given by (4.3). Then there is h such that

(4.12) $h=\min\{k\,|\,f_\theta(\theta_{k-1})\ge r(\theta_j)\}.$

Obviously h satisfies $g<h\le i$ (see (4.5)) and

(4.13) $0\le r(\theta_k)\le f_\theta(\theta_{h-2})<r(\theta_j)$, k=1, ..., h-1,

(4.14) $d(\theta_k)\le d(\theta_j)$, k=h, ..., j (by $f_\theta(\theta_k)\ge r(\theta_j)$ and Algorithm S).

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Furthermore, the start time of θ_g in θ, $\max\{r(\theta_g), f_\theta(\theta_{g-1})\}$, is earlier than $r(\theta_k)$, k=h, ..., j, since θ_g is scheduled before θ_k in Algorithm S and $d(\theta_g)>d(\theta_k)$ holds by (4.4) and (4.14). Hence it holds by Lemma 3.3 and (4.14) that

$$0\le\max\{r(\theta_g), f_\theta(\theta_{g-1})\}<r(\theta_k)\le L_\omega-1+d(\theta_k)\le L_\omega-1+d(\theta_j),\ k=h,\dots,j. \tag{4.15}$$

It follows from (4.3), (4.7) and (4.9) that

$$-d(\theta_j)=L_\theta-f_\theta(\theta_j)>((P-3)/(P+2))L_\omega+L_\omega-L_\omega+1=((P-3)/(P+2))L_\omega+1. \tag{4.16}$$

Now recall that schedule μ is constructed by applying Algirithm S to Q(-D, P, -R) and taking its reverse schedule. Hence, let $\overleftarrow{\mu}=(\overleftarrow{\mu}_1, \dots, \overleftarrow{\mu}_n)=(\mu_n, \dots, \mu_1)$ and L_μ given by

$$L_\mu=-d(\overleftarrow{\mu}_\ell)+\Sigma_{k=m}^{\ell}p(\overleftarrow{\mu}_k)+r(\overleftarrow{\mu}_m)\ \text{(see (2.3))}. \tag{4.17}$$

$r(\overleftarrow{\mu}_m)$ in $\overleftarrow{\mu}$ plays the same role as $-d(\theta_j)$ in θ, and $-d(\overleftarrow{\mu}_m)$ in $\overleftarrow{\mu}$ plays the same role as $r(\theta_j)$ in θ. Therefore, it follows from (4.13) and (4.16) that

$$r(\overleftarrow{\mu}_m)>((P-3)/(P+2))L_\omega+1,\ -d(\overleftarrow{\mu}_m)>0. \tag{4.18}$$

We will show in the following that (4.18) leads to a contradiction, by considering three cases A), B) and C) according to values of L_ω.

A) $L_\omega\ge P+2$. It follows from (4.6) and (4.9) that $P-2\ge\Sigma_{k=g}^{j-1}p(\theta_k)-1\ge L_\theta-L_\omega>((P-3)/(P+2))L_\omega\ge P-3$, i.e., $L_\theta-L_\omega=P-2$ and $\Sigma_{k=g}^{j-1}p(\theta_k)=P-1$. The latter implies that i=g=1, j=n and $f_\theta(\theta_j)=P$ by (4.3) and (3.4) (i.e., $r(\theta_1)=0$). Hence it holds by (4.3), (4.14), (4.18) and lemma 3.3 that

$$r(\overleftarrow{\mu}_m)>1=L_\omega-1+P-L_\theta=L_\omega-1+d(\theta_j)\ge L_\omega-1+d(\theta_k)\ge r(\theta_k),\ k=h, \dots,n.$$

Therefore, $\overleftarrow{\mu}_m=\theta_k$ for some $1\le k\le h-1$. But this is a contradiction to (4.13).

Now we may assume that $L_\omega\le P+1$. In this case first assume that $r(\theta_i)=1$ holds in (4.3). Then $r(\theta_i)\le r(\theta_1)+p(\theta_1)$ (since $p(\theta_1)\ge 1$) and hence i=1 in (4.3) (recall that i is the minimum one satisfying (4.3)), a contradiction to $r(\theta_1)=0$. Next assume that $r(\theta_1)\ge 2$. Then $-d(\theta_j)\le L_\omega-1-r(\theta_j)\le L_\omega-r(\theta_i)-2\le L_\omega-4$ by Lemma 3.3 and (4.13), which contradicts (4.16) under the assumption $L_\omega\le P+1$. Therefore we have $r(\theta_i)=0$, i.e.,

$$f_\theta(\theta_j)=\Sigma_{k=1}^{j}p(\theta_k)\ \text{(i.e., i=1 holds in (4.3))}. \tag{4.19}$$

Furthermore,

$$L_\theta-L_\omega\le P-3, \tag{4.20}$$

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

since otherwise $L_\theta - L_\omega = P-2$ by (4.6), which is a contradiction as proved in the same manner as the case of $L_\omega \ge P+2$. Now consider the next cases.

B) $L_\omega = P+1$. Since $P-3 \ge L_\theta - L_\omega > ((P-3)/(P+2))L_\omega > P-4$ (by (4.20), (4.9) and $L_\omega = P+1$)

(4.21) $L_\theta - L_\omega = P-3$, i.e., $L_\theta = 2P-2$.

Furthermore, $f_\theta(\theta_j) = L_\theta + d(\theta_j)$ (by (4.3)) $\ge L_\theta + r(\theta_j) + 1 - L_\omega$ (by Lemma 3.3) $\ge P-1$ (by (4.5) and (4.21)). Thus, the following two cases are possible by (4.19) and (4.21).

B1) $L_\omega = P+1$, $f_\theta(\theta_j) = P-1$, and $d(\theta_j) = f_\theta(\theta_j) - L_\theta = 1-P$. Then,

(4.22) $r(\theta_k) = 1$ (by (4.15)), $p(\theta_k) = 1$ and $d(\theta_k) = 1-P$, $k = h, \ldots, j$ (by (4.14) and Lemma 3.3),

which leads to h=2 by (4.12). Thus, by Lemma 3.3

$$P+1 = L_\omega > \min_{h \le k \le j} r(\theta_k) + \Sigma_{k=h}^{j} p(\theta_k) - \max_{h \le k \le j} d(\theta_k) = P + \Sigma_{k=h}^{j} p(\theta_k),$$

and hence j=h=2. On the other hand, j=n-1 holds since $\Sigma_{k=1}^{j} p(\theta_k) = f_\theta(\theta_j) = P-1$. Therefore,

(4.23) $j = h = 2$, $n = 3$, $p(\theta_n) = P - \Sigma_{k=1}^{j} p(\theta_k) = 1$.

Now, we have $\overleftarrow{\mu}_m = \theta_3$, since $r(\overleftarrow{\mu}_m) \ge P-2 \ge 2$ by (4.18) and (4.11), and hence, $r(\overleftarrow{\mu}_m) > 1 = r(\theta_j) = r(\theta_2) > r(\theta_1) = 0$ by (4.22), (4.23) and (4.13). Furthermore, we have $d(\theta_1) = 0$ since $d(\theta_2) = d(\theta_j) < 0$ by (4.22), $d(\theta_3) = d(\overleftarrow{\mu}_m) < 0$ by (4.18) and $D_{max} = 0$. Therefore, only the next case is possible;

$r(\theta_1) = 0$, $p(\theta_1) = P - p(\theta_2) - p(\theta_3) = P-2$ $d(\theta_1) = 0$,
(by (4.22) and (4.23)),

(4.24) $r(\theta_2) = 1$, $p(\theta_2) = 1$, $d(\theta_2) = 1-P$,

$r(\theta_3) = r(\overleftarrow{\mu}_m) \ge 2$, $p(\theta_3) = 1$ (by (4.23)), $2-P \le d(\theta_3) = d(\overleftarrow{\mu}_m) < 0$
(by (4.18) and Lemma 3.3).

Applying Algorithm $\bar{S}$ to (4.24), we have $\overleftarrow{\mu}_1 = \theta_1$, $\overleftarrow{\mu}_2 = \theta_3$ $(= \overleftarrow{\mu}_m)$, $\overleftarrow{\mu}_3 = \theta_2$ and $L_\mu = -d(\overleftarrow{\mu}_1) + p(\overleftarrow{\mu}_1) + p(\overleftarrow{\mu}_2) + r(\overleftarrow{\mu}_2) = r(\overleftarrow{\mu}_m) + P - 1$ (see (4.17) and (4.24)). Thus, $r(\overleftarrow{\mu}_m) = L_\mu - P + 1 > ((P-3)/(P+2))L_\omega + L_\omega - P + 1 = P - 2 + (5/(P+2))$ by (4.10) and $r(\overleftarrow{\mu}_m) \le L_\omega - 1 + d(\overleftarrow{\mu}_m) \le P-1$ by Lemma 3.3 and (4.18). Therefore,

(4.25) $r(\theta_3) = r(\overleftarrow{\mu}_m) = P-1$, $d(\theta_3) = d(\overleftarrow{\mu}_m) = -1$ (by Lemma 3.3).

It then follows from (4.24) and (4.25) that $\omega = (\theta_2, \theta_1, \theta_3)$ and $L_\omega = P+2$, which contradicts $L_\omega = P+1$.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

<u>B2) $L_\omega=P+1$, $f_\theta(\theta_j)=P$, $d(\theta_j)=f_\theta(\theta_j)-L_\theta=2-P$.</u> $f_\theta(\theta_j)=P$ implies j=n by (4.19). Also $r(\theta_k)\le L_\omega-1+d(\theta_k)\le 2$, k=h, ..., j (by (4.15)) and $r(\theta_k)<r(\theta_j)$, k=1, 2, ..., h-1 (by (4.13)). Then P=4 holds by (4.11) since otherwise $r(\overleftarrow{\mu}_m)>2$ follows from (4.18). This means

(4.26) $\quad P=4,\ n=3$ (by (4.11)), $r(\overleftarrow{\mu}_m)=2$.

By (4.12) and (4.26) we have h=2 or 3. $r(\overleftarrow{\mu}_m)=2$ then implies

(4.27) $\quad \{\overleftarrow{\mu}_m\}\varepsilon\{\theta_h, \dots, \theta_j\}\subseteq\{\theta_2, \theta_3\}$.

In this case, it holds by Lemma 3.3, (4.26) and (4.14) that $5=L_\omega\ge r(\overleftarrow{\mu}_m)+p(\overleftarrow{\mu}_m)-d(\overleftarrow{\mu}_m)\ge r(\overleftarrow{\mu}_m)+p(\overleftarrow{\mu}_m)-d(\theta_j)$. The last term is not greater than 5 only if

(4.28) $\quad r(\overleftarrow{\mu}_m)=2,\ p(\overleftarrow{\mu}_m)=1,\ d(\overleftarrow{\mu}_m)=-2$.

First consider the case of h=2 and ν such that $\{\nu, \overleftarrow{\mu}_m\}=\{\theta_2, \theta_3\}$ (see (4.27)). Then, $\min\{r(\nu), r(\overleftarrow{\mu}_m)\}+p(\nu)\le L_\omega-p(\overleftarrow{\mu}_m)+\max\{d(\nu), d(\overleftarrow{\mu}_m)\}=2$ by Lemma 3.3, (4.28) and $d(\nu)\le d(\theta_j)=2-P=-2$ (see (4.14)). Therefore, $r(\nu)+p(\nu)\le 2$ by (4.26). $r(\nu)\ge 1$ (by (4.15)) then implies

(4.29) $\quad r(\nu)=p(\nu)=1,\ p(\theta_1)=P-p(\nu)-p(\overleftarrow{\mu}_m)=2,\ d(\nu)<-2$.

It follows from (4.26), (4.28) and (4.29) that only the next case is possible when h=2;

$$r(\theta_1)=0,\ p(\theta_1)=2,\ d(\theta_1)=0\ \text{(by } d(\nu)<0,\ d(\overleftarrow{\mu}_m)<0 \text{ and } D_{max}=0)$$

$$r(\nu)=1,\quad p(\nu)=1,\quad d(\nu)\le d(\theta_j)=-2$$

$$r(\overleftarrow{\mu}_m)=2,\ p(\overleftarrow{\mu}_m)=1,\ d(\overleftarrow{\mu}_m)=-2.$$

Thus, $\overleftarrow{\mu}=(\theta_1, \overleftarrow{\mu}_m, \nu)$ and $L_\mu=L_{\overleftarrow{\mu}}=5=L_\omega$, which contradicts (4.10).

Next consider another case, i.e., h=3, then

(4.30) $\quad \overleftarrow{\mu}_m=\theta_j=\theta_3$ (by (4.27)).

It follows from (4.13) and (4.28) that $p(\theta_1)\le f_\theta(\theta_1)=f_\theta(\theta_{h-2})<r(\theta_j)=r(\overleftarrow{\mu}_m)=2$, which means that

(4.31) $\quad p(\theta_1)=1,\ p(\theta_3)=p(\overleftarrow{\mu}_m)=1$ (by (4.28)), $p(\theta_2)=P-p(\theta_1)-p(\theta_3)=2$.

Thus, $p(\theta_1)<p(\theta_2)$, and hence

(4.32) $\quad 0=r(\theta_1)<r(\theta_2)\quad$ or $\quad d(\theta_1)<d(\theta_2)=0$,

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

since otherwise Algorithm S does not select θ_1 as the first job. Now assume that $\overleftarrow{\mu}_1=\theta_1$. Then $p(\overleftarrow{\mu}_1)=p(\theta_1)<p(\theta_2)$ and hence

(4.33) $$0=d(\theta_1)=d(\overleftarrow{\mu}_1)>d(\theta_2) \quad \text{or} \quad 0=r(\theta_1)=r(\overleftarrow{\mu}_1)>r(\theta_2),$$

since otherwise Algorithm $\bar{S}$ does not specify θ_1 as the last job. The latter case of (4.33), however, is not possible by $R_{min}=0$. Thus, it follows from (4.32) and (4.33) that $0=r(\theta_1)<r(\theta_2)$ and $0=d(\theta_1)>d(\theta_2)$. In this case, $L_\omega\geq P+2$ can be easily shown from (4.28) and (4.31), a contradiction to $L_\omega=P+1$. Finally, since $\overleftarrow{\mu}_1\neq\overleftarrow{\mu}_m=\theta_3$, there remains the case of $\overleftarrow{\mu}_1=\theta_2$. This implies that $d(\theta_2)=d(\overleftarrow{\mu}_1)=D_{max}=0$. Then Algorithm $\bar{S}$ selects $\overleftarrow{\mu}_2=\overleftarrow{\mu}_m=\theta_3$, since $f_{\overleftarrow{\mu}}(\overleftarrow{\mu}_1)=p(\overleftarrow{\mu}_1)=p(\theta_2)=2$ (see (4.31))$=-d(\overleftarrow{\mu}_m)$ (see (4.28)) (i.e., θ_3 is ready) and $0=-r(\theta_1)>-r(\theta_3)=-2$ (see (4.28)). Therefore, we have $\overleftarrow{\mu}=(\theta_2, \theta_3, \theta_1)$ and $L_\mu=5=P+1=L_\omega$, which contradicts (4.10).

C) $L_\omega=P$. It follows from (4.18) that $r(\overleftarrow{\mu}_m)>((P-3)/(P+2))P+1$, while $r(\theta_k)\leq L_\omega-1+d(\theta_j)<3-(10/(P+2))$, $k=h, \ldots, j$ by (4.15) and (4.16). Thus $r(\overleftarrow{\mu}_m)>r(\theta_k)$, $k=h, \ldots, j$ by $P\geq 4$ of (4.11), i.e.,

(4.34) $$\overleftarrow{\mu}_m\neq\theta_1, \ldots, \theta_j \quad \text{(note that (4.13))}.$$

By (2.2), (2.1) and (3.4) $P=L_\omega\geq f_\omega(\omega_n)-d(\omega_n)\geq r(\omega_1)+P-d(\omega_n)\geq P$, which implies

(4.35) $$r(\omega_1)=d(\omega_n)=0.$$

Thus, by $r(\theta_k)>0$, $d(\theta_k)\leq -2$, $k=h, \ldots, j$ (see (4.14), (4.15) and (4.16))

(4.36) $$\{\omega_1, \omega_n\}\cap\{\theta_h, \ldots, \theta_j\}=\phi.$$

Finally $r(\overleftarrow{\mu}_m)>0$ and $d(\overleftarrow{\mu}_m)<0$ (see (4.18)), and hence

(4.37) $$\overleftarrow{\mu}_m\neq\omega_1, \omega_n.$$

Now the following two cases are possible for h, noting that $h\geq 2$.

C1) $L_\omega=P$, $h=2$. First note that $j\leq n-2$, since otherwise $|\{\omega_1, \omega_n\}\cup\{\theta_h, \ldots, \theta_j\}|=n$ by $h=2$ and (4.36) and hence $\overleftarrow{\mu}_m\neq\theta_1, \ldots, \theta_n$ by (4.34) and (4.37), a contradiction. On the other hand, $L_\theta\geq 2P-4$ by (4.9) and $d(\theta_j)\geq r(\theta_j)+1-L_\omega\geq 2-P$ by Lemma 3.3 and (4.13). Thus, $P-2\geq P-p(\theta_{n-1})-p(\theta_n)\geq\Sigma_{k=1}^{j}p(\theta_k)=L_\theta+d(\theta_j)$ (by (4.19))$\geq P-2$, and hence $j\geq n-2$. Therefore,

(4.38) $$j=n-2,\ f_\theta(\theta_j)=\Sigma_{k=1}^{j}p(\theta_k)=P-2,\ d(\theta_j)=2-P,\ p(\theta_{n-1})=p(\theta_n)=1,$$
$$L_\theta=2P-4.$$

Hence $1\leq r(\theta_k)\leq L_\omega-1+d(\theta_j)=1$, $k=h, \ldots, j$ by (4.15), and

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

(4.39) $r(\theta_k)=1,\ p(\theta_k)=1,\ d(\theta_k)=2-P,\ k=h, \ldots, j,$

by (4.14) and Lemma 3.3. This implies $P=L_\omega \geq \min_{h\leq k\leq j} r(\theta_k)+\Sigma_{k=h}^{j} p(\theta_k)-\max_{h\leq k\leq j} d(\theta_k)=P+(j-h)\geq P$ by Lemma 3.3. Thus,

(4.40) $j=h=2,\ n=j+2=4,$

by h=2 and (4.38). Note that θ_g is not the first job in ω by definition, while $g<h=2$ by (4.12), i.e., θ_g is the first job in θ. Thus, $\theta_1=\theta_g\neq\omega_1$ and hence $\omega_1=\theta_3$ or θ_4 by (4.36) and (4.40). Namely

(4.41) $\{\overleftarrow{\mu}_m,\ \omega_1\}=\{\theta_3,\ \theta_4\}$ (by 4.34) and (4.37)), $\omega_4=\theta_1$ (by (4.36) and h=2).

Furthermore, $\overleftarrow{\mu}_m$ precedes ω_4 in ω by (4.37) and $p(\omega_4)=P-\Sigma_{k=2}^{4} p(\theta_k)=P-3$ by (4.38), (4.39) and (4.40). Then $P=L_\omega\geq r(\overleftarrow{\mu}_m)+p(\overleftarrow{\mu}_m)+p(\omega_4)-d(\omega_4)=r(\overleftarrow{\mu}_m)+P-2$ by (4.35), (4.38), (4.41). Thus $r(\overleftarrow{\mu}_m)\leq 2$, a contradiction to (4.18) under the assumption $P\geq 5$ (see (4.11)).

C2) $L_\omega=P,\ h\geq 3$. Note first that $d(\theta_j)\leq 3-P$ by (4.16) and hence $f_\theta(\theta_2)=\Sigma_{k=1}^{2} p(\theta_k)$ (by (4.19)) $\geq 2\geq L_\omega-p(\theta_j)+d(\theta_j)\geq r(\theta_j)$ (by Lemma 3.3) $>f_\theta(\theta_{h-2})$ (by (4.13)) $\geq p(\theta_1)\geq 1$. Thus,

(4.42) $h=3,\ r(\theta_j)=2,\ p(\theta_j)=1,\ d(\theta_j)=3-P,\ p(\theta_1)=1,$

(4.43) $P\geq 9$ (by $P-3>((P-3)/(P+2))P+1$ derived from (4.16)).

It holds by (4.19), (4.3), (4.42) and $L_\theta\geq 2P-4$ (by (4.9)) that $\Sigma_{k=1}^{j} p(\theta_k)=f_\theta(\theta_j)=L_\theta+d(\theta_j)\geq P-1$, and hence $j\geq n-1$. On the other hand, $\Sigma_{k=1}^{j} p(\theta_k)=f_\theta(\theta_j)\leq L_\omega-1=P-1$ by (4.7), and hence $j\leq n-1$. Therefore,

(4.44) $j=n-1,\ f_\theta(\theta_j)=P-1,\ \overleftarrow{\mu}_m=\theta_n$ (by $j=n-1$ and (4.34)).

In addition $\{\omega_1,\ \omega_n\}=\{\theta_1,\ \theta_2\}$ by h=3, (4.44), (4.36) and (4.37). Here, it should be noted that θ_k, k=h, ..., j-1 precedes θ_j in ω since otherwise $f_\omega(\theta_\ell)\geq f_\theta(\theta_j)$ holds for the last θ_ℓ $(k\leq\ell\leq j-1)$ in ω (for the same reason as applied to θ_g prior to (4.4)), and $d(\theta_\ell)\leq d(\theta_j)$ (by (4.14)) implies $L_\omega\geq L_\theta$, a contradiction. If $\omega_1=\theta_2$ and $\omega_n=\theta_1$, then $P=L_\omega\geq f_\theta(\theta_j)-p(\theta_1)-d(\theta_j)=2P-5$ (i.e., $P\leq 5$) by (4.42) and (4.44), which contradicts (4.43). On the other hand, if $\omega_1=\theta_1$ and $\omega_n=\theta_2$, we have similarly $P\geq f_\theta(\theta_j)-p(\theta_2)-d(\theta_j)=2P-4-p(\theta_2)$. Thus, $p(\theta_2)\geq P-4$. However, $P=L_\omega\geq r(\overleftarrow{\mu}_m)+p(\overleftarrow{\mu}_m)+p(\omega_n)-d(\omega_n)\geq P-2+p(\omega_n)$, by (4.18) and (4.35). Thus, $P-4\leq p(\theta_2)=p(\omega_n)\leq 2$, which contradicts (4.43).

Now that $W_{mS}(P)=(P-3)/(P+2)$ is proven to be an upper bound on A_{mS}, we show that this is best possible. Consider a problem instance Q(R, P, D) such that n=3, R=(0, 1, P), P=(P-2, 1, 1) and D=(0, -P, -1), then $\theta=(1, 2, 3)$, $\mu=(2, 3, 1)$ and $L_\theta=L_\mu=2P-1$, while $\omega=(2, 1, 3)$ and $L_\omega=P+2$.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Thus, $\min\{A_S, A_{\bar{S}}\}=(P-3)/(P+2)$, i.e., $W_{mS}(P)=(P-3)/(P+2)$ is best possible.

5. Numerical Experiments

Some numerical experiments were carried out to see how the proposed approximation algorithms behave on the average [6]. The average performance seems to be more important in practice than the worst case performance discussed so far.

For each test problem with n jobs, 3n integers r(i), p(i), d(i) are generated from uniform distributions over the ranges $[0, R_{max}]$, $[1, P_{max}]$ and $[D_{min}, 0]$ respectively, where n, R_{max}, P_{max} and D_{min} are given integers. This method of problem generation is adopted after [3], [12]. 100 problems are tested for each type of problems. Exact optimal schedules ω are obtained by a branch-and-bound algorithm, which is a modification of [14], to compute relative deviations. When the branch-and-bound algorithm fails to obtain an optimal schedule due to the lack of computer time, the best lower bound available by then is employed instead. (Thus the reported relative deviations are slightly biased in the pessimistic direction.) Algorithms are coded in FORTRAN VI and run on FACOM M-190 at Data Processing Center, Kyoto University, Kyoto.

The results obtained are shown in Fig. 5.1 and 5.2. Fig. 5.1 shows how the ratio of R_{max} to $|D_{min}|$ affects the average relative deviations of the proposed six approximation algorithms, since this seems to be a crucial parameter in determinind the quality of approximate schedules in view of that ready times r(k) and due times d(k) play quite different roles in all approximation algorithms.
Here n=20, $P_{max}=25$ and $R_{max}+|D_{min}|=1000$ are employed. Fig. 5.1 shows that the average relative deviations of Algorithms J and $\bar{J}$ have opposite trend; the relative deviation of J decreases rapidly with the increase of $|D_{min}|$, while the relative deviation of $\bar{J}$ decreases rapidly with the increase of R_{max}. This trend also holds between average relative deviations of Algorithms S and $\bar{S}$. On the other hand, average relative deviations of Algorithms mJ and mS stay almost constant for the wide range of $R_{max}/|D_{min}|$. Finally, it should be noted that mS outperforms other algorithms; the average relative deviation of mS is always kept within 0.2%, which is usually sufficient for practical purposes.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Fig. 5.1 The effect of the ratio of R_{max} to $|D_{min}|$ ($N=20$, $P_{max}=25$ and $R_{max}+|D_{min}|=1000$ are used).

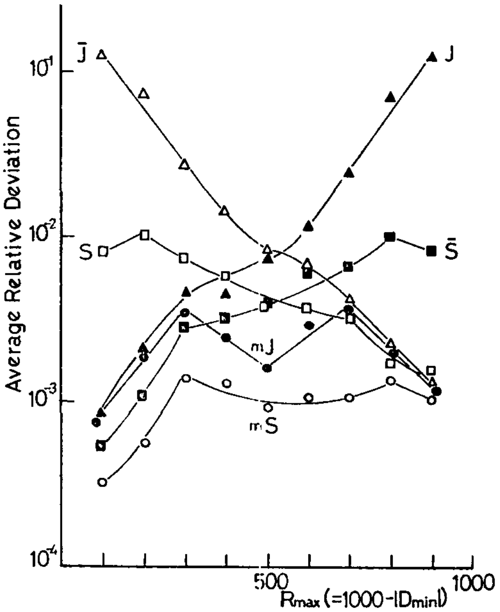
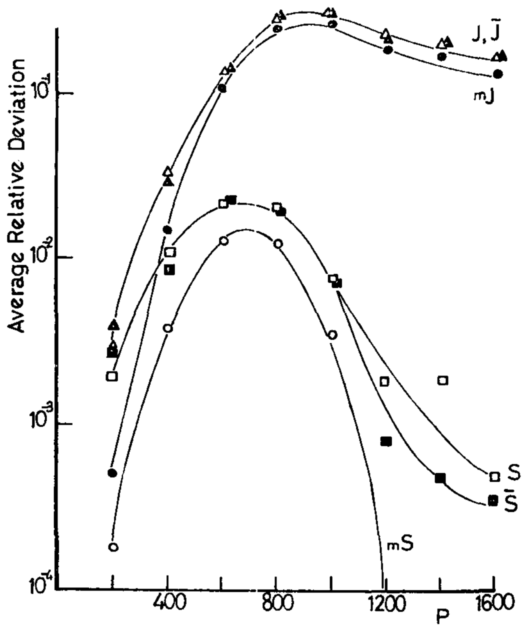


Fig. 5.2 The effect of the sum of processing times P (R_{max} and $|D_{min}|$ are fixed to 500).



Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Fig. 5.2 indicates how the sum of processing times P affects the average relative deviations of six approximation algorithms, where n=20 and R_{max}= $|D_{min}|$=500 are used. Fig. 5.2 shows that each of the approximation algorithms has the worst performance roughly around P=1.0R_{max}∿2.0R_{max}. This is consistent with the theoretical results that all problem instances with the worst relative deviations constructed in proofs of previous theorems have R_{max} and $|D_{min}|$ which are equal or nearly equal to P. To see more carefully the performance of approximation algorithms in this difficult situation, another experiment is performed with P=1.4R_{max}, $R_{max}=|D_{min}|$, R_{max}=100, 200,...,1200. However no significant dependency on the size of R_{max} is observed and hence the result is not cited here. This suggests that the quality of approximate schedules is not affected by the magnitude of P, R_{max}, D_{min}, but rather determined by their ratios. It should be noted that Algorithm mS keeps its average relative deviation within 2% in all the tested cases.

We conclude that Algorithm mS is efficient and gives good approximate solutions for the one machine scheduling problem discussed here. It can be recommended for practical uses.

Acknowledgment

The authors wish to thank Professor T. Hasegawa of Kyoto University for his comments, and Mr. O. Hasegawa for carrying out part of the experiment reported in Section 5. They are also indebted to Miss T. Kanazawa for her excellent typing.

References

[1] Aho, A., Hopcroft, J. and Ullman, J.: The Design and Analysis of Computer Algorithms. *Addison-Wesley Company*, 1974.

[2] Baker, K. and Su, Z. S.: Sequencing with Due-Date and Early Start Times to Minimize Maximum Tardiness. *Naval Research Logistics Quarterly*, Vol. 21 (1976), 171-176.

[3] Bratley, B., Florian, M. and Robillard, P.: On Sequencing with Earliest Starts and Due-Date with Application to Computing Bounds for the (n/m/G/F_{max}) Problem. *Naval Research Logistics Quarterly*, Vol. 20 (1973), 57-67.

[4] Dessouky, M. and Margenthalar, C.: The One Machine Sequencing Problem with Early Starts and Due-Dates. *AIIE Transaction*, Vol. 4 (1972), 214-222.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

[5] 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. *Proceedings of Do 77*, 1977.

[6] Hasegawa, O., Ibaraki, T., Hasegawa, T. and Kise, H.: Approximation Methods for the One Machine Scheduling Problem with Ready and Due Times. Presented at the Meeting of Operations Research Society of Japan, May, 1978.

[7] Horn, W. A.: Some Simple Scheduling Algorithms. *Naval Research Logistics Quarterly*, Vol. 21 (1974), 177-185.

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

[9] Kise, H., Ibaraki, T. and Mine, H.: n-job One-Machine Scheduling Problems with Ready and Due Times. *Working Paper*, Department of Applied Mathematics and Physics, Kyoto University, 1975.

[10] Lageweg, B., Lenstra, J. and Rinnooy Kan, A. H. G.: Minimizing Maximum Lateness on One Machine : Computational Experience and Some Applications. *Statistica Neerlandica*, Vol. 30 (1976), 25-40.

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

[12] Lenstra, J.: Sequencing by Enumerative Methods. *Mathematical Centre Tract 69*, Mathematical Centrum, Amsterdam, 1977.

[13] McMahon, G. and Florian, M.: On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness. *Operations Research*, Vol. 23 (1975), 475-482.

[14] Mine, H., Ibaraki, T. and Kise, H.: An Algorithm for a Scheduling Problem. *Keiei-Kagaku* (the Official Journal of Operations Research of Japan), Vol. 18 (1974), 23-37.

[15] Schrage, L.: Obtaining Optimal Solutions to Resource Constrained Network Scheduling Problems, Unpublished Manuscript, 1971.

[16] Walters, J.: A New Algorithm for Scheduling the Single Machine to Minimize Maximum Lateness. *Working Paper No. 6*, Ball State University, 1976.

Hiroshi KISE: Department of Mechanical Engineering, Faculty of Engineering, Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto 606, Japan.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.