J. Appl. Math. & Computing Vol. **18***(2005), No. 1 - 2, pp. 383 - 391*

FLOW SHOP SCHEDULING JOBS WITH POSITION-DEPENDENT PROCESSING TIMES

JI-BO WANG

ABSTRACT. The paper is devoted to some flow shop scheduling problems, where job processing times are defined by functions dependent on their positions in the schedule. An example is constructed to show that the classical Johnson's rule is not the optimal solution for two different models of the two-machine flow shop scheduling to minimize makespan. In order to solve the makespan minimization problem in the two-machine flow shop scheduling, we suggest Johnson's rule as a heuristic algorithm, for which the worst-case bound is calculated. We find polynomial time solutions to some special cases of the considered problems for the following optimization criteria: the weighted sum of completion times and maximum lateness. Some furthermore extensions of the problems are also shown.

AMS Mathematics Subject Classification: 90B35.
Key words and phrases: Scheduling, flow shop, learning effect, position-dependent processing time, worst-case bound.

1. Introduction

Scheduling problems have received considerable attention for many years. A common assumption in traditional scheduling is that the production time of a given product is independent of its position in the production sequence. However, in many realistic settings, because firms and employees perform a task over and over, they learn how to perform more efficiently. The production facility (a machine, a worker) improves continuously over time. As a result, the production time of a given product is shorter if it is scheduled later. This phenomenon is known as a "learning effect" in the literature.

The learning effect in scheduling may arise in a company which products similar jobs on one machine or on identical parallel machines for a number of customers. In many cases jobs will have different normal processing times due to varying quantities or slightly different components that make up the products.

Received March 18, 2004. Revised December 16, 2004.

Nevertheless, by processing one job after the other the skills of the workers continuously improve, e.g. the ability to perform setups, to deal with the operations of the machines and software or to handle raw materials, components or similar operations of jobs at a greater space faster. Although learning theory was first applied to industry more than 60 years ago (Wright [14]), it appears to have become a topic in scheduling research only in recent years.

Biskup [3] was the first to investigate the effect of learning in the framework of scheduling. He assumed a learning process that was reflected in a decrease in production time as a function of the *number of repetitions* of the production of a single item, i.e. as a function of the job position in the sequence. Biskup studied the single-machine problem of minimum total flow time, and single-machine problem of minimizing the weighted sum of completion time deviations from a common due date and the sum of job completion times. Using similar solution techniques, Mosheiov [7, 8] investigated several other single-machine problems, and minimum total flow time on identical parallel machines. Mosheiov and Sidney [9] investigated scheduling problems with *job-dependent* learning effects, i.e. the learning in the production process of some jobs to be faster than that of others. They focused on some classical single-machine objectives such as makespan and total flowtime, on a due date assignment problem and on minimizing total flowtime on unrelated parallel machines, all these problems remained polynomially solvable.

Wang and Xia [12] considered flow shop scheduling problems with a learning effect. The objective was to minimize one of the two regular performance criteria, namely, makespan and total flowtime. They gave a heuristic algorithm with worst-case bound m for each criteria, where m was the number of machines. They also found polynomial time solutions to two special cases of the problems, i.e. identical processing time on each machine and an increasing series of dominating machines. A survey on this line of the scheduling research could be found in Bachman and Janiak [1].

In this paper we study flow shop scheduling problems, where job processing times are defined by functions dependent on their positions in the schedule. The remaining part of this paper is organized as follows. In section 2 we give some general notations and assumptions. In section 3, an example is constructed to show that the classical Johnson's rule is not the optimal solution for two different models of the two-machine flow shop scheduling to minimize makespan. In order to solve the makespan minimization problem in two-machine flow shop scheduling, we suggest Johnson's rule as a heuristic algorithm, for which the worst-case bound is calculated. In section 4, We find polynomial time solutions to some special cases of the considered problems for the following optimization criteria: the weighted sum of completion times and maximum lateness. In section 5, some furthermore extensions of the problems are also shown.

2. Notations and assumptions

The flow shop scheduling consists of scheduling n jobs $N = \{J_1, J_2, \ldots, J_n\}$ on m machines $M_1, M_2, \ldots, M_m$. Job J_j consists of m operations $(O_{1,j}, O_{2,j}, \ldots, O_{m,j})$. Operation $O_{i,j}$ has to be processed on machine $M_i, i = 1, 2, \ldots, m$. Processing of operation $O_{i+1,j}$ may start only after $O_{i,j}$ having been completed and all machines process the jobs in the same order, i.e. a permutation schedule. Each job can be processed on no more than one machine at any time, while each machine can handle only one job at a time and the processing of a job may not be interrupted. All jobs are available for processing at time 0. The normal processing time of operation $O_{i,j}$ is denoted by $p_{i,j}$. The actual processing time $p_{i,j,r}$ of job J_j on machine M_i is a function dependent on its position r in a schedule. In this paper, we consider two special models of job processing time characterized by position-dependent function, namely:

$$p_{i,j,r} = p_{i,j}(a - br), i = 1, 2, \ldots, m; r, j = 1, 2, \ldots, n, \tag{1}$$

$$p_{i,j,r} = p_{i,j} r^c, i = 1, 2, \ldots, m; r, j = 1, 2, \ldots, n, \tag{2}$$

where a and b (c in model (2)) denote a constant number and a learning ratio respectively. It is assumed that a is a positive integer and b is a non-negative rational. Since job processing time is some positive value, for model (1) it is assumed that $a - (n + 1)b > 0$. It is easy to notice that the processing time of any job characterized by model (2) is always positive. Therefore, no additional assumptions are required there.

For a given schedule π, $C_{i,j} = C_{i,j}(\pi)$ represents the completion time of operation $O_{i,j}$, $C_j = C_{m,j}$ represents the completion time of job J_j, $\pi = ([1], [2], \ldots, [n])$ represents a schedule of $(1, 2, \ldots, n)$, where $[j]$ denotes the job that occupies the jth position in π, and $\gamma(\pi)$ represents a regular measure of performance. $C_{\max} = \max\{C_j | j = 1, 2, \ldots, n\}$, $\sum w_j C_j$ and $L_{\max} = \max\{C_j - d_j | j = 1, 2, \ldots, n\}$ represent makespan, weighted sum of completion times and maximum lateness of a given schedule, respectively. In the remaining part of the paper, all the problems considered will be denoted using the three-field notation schema $\alpha|\beta|\gamma$ introduced by Graham et al [5].

3. Two-machine cases

Johnson [6] proved that $F2||C_{\max}$ is optimally solved by Johnson's rule, that is,

Step 1. Construct a set of jobs $A = \{J_j \in N | p_{1,j} \leq p_{2,j}\}$;

Step 2. Schedule jobs from set A in non-decreasing order of their $p_{1,j}$'s on both machines;

Step 3. Schedule jobs from set $N - A$ in non-increasing order of their $p_{2,j}$'s on both machines.

In the following examples, we show that this policy is not optimal for the problems $F2|p_{i,j,r} = p_{i,j}(a - br)|C_{\max}$ and $F2|p_{i,j,r} = p_{i,j}r^c|C_{\max}$.

Example 1. $n = 3$, $p_{1,1} = 2$, $p_{2,1} = 3$, $p_{1,2} = 1$, $p_{2,2} = 20$, $p_{1,3} = 7$, $p_{2,3} = 5$, $a = 1$, $b = 0.1$. The schedule according to Johnson's rule is (J_2, J_1, J_3), the value $C_{\max} = 24.8$. Obviously, the optimal schedule is (J_1, J_2, J_3), the optimal value $C^*_{\max} = 24$.

Example 2. $n = 3$, $p_{1,1} = 2$, $p_{2,1} = 3$, $p_{1,2} = 1$, $p_{2,2} = 20$, $p_{1,3} = 7$, $p_{2,3} = 5$. Learning take place by the 80%-learning curve, that is $c = -0.322$ (Badiru [2]). The schedule according to Johnson's rule is (J_2, J_1, J_3), the value $C_{\max} = 26.9$. Obviously, the optimal schedule is (J_1, J_2, J_3), the optimal value $C^*_{\max} = 24.5$.

In order to solve each problem, we will use the same Johnson's rule as a heuristic algorithm for the problems $F2|p_{i,j,r} = p_{i,j}(a-br)|C_{\max}$ and $F2|p_{i,j,r} = p_{i,j}r^c|C_{\max}$. The performance of the algorithm will be verified by its worst-case bound.

Theorem 1. *Let S^* be an optimal schedule and S be an Johnson's schedule for the problem $F2|p_{i,j,r} = p_{i,j}(a - br)|C_{\max}$. Then*

$$\rho^1_J = C_{\max}(S)/C_{\max}(S^*) < (a - b)/(a - nb).$$

Proof. Let the solution of Johnson's rule for the problem $F2|p_{i,j,r} = p_{i,j}|C_{\max}$ be C. Obviously, $C_{\max}(S) < (a - b)C$, $C_{\max}(S^*) > (a - nb)C$. Hence

$$\rho^1_J = C_{\max}(S)/C_{\max}(S^*) < (a - b)/(a - nb).$$

□

Theorem 2. *Let S^* be an optimal schedule and S be an Johnson's schedule for the problem $F2|p_{i,j,r} = p_{i,j}r^c|C_{\max}$. Then $\rho^2_J = C_{\max}(S)/C_{\max}(S^*) < 1/n^c$.*

Proof. Let the solution of Johnson's rule for the problem $F2|p_{i,j,r} = p_{i,j}|C_{\max}$ be C. Obviously, $C_{\max}(S) < C$, $C_{\max}(S^*) > Cn^c$. Hence,

$$\rho^2_J = C_{\max}(S)/C_{\max}(S^*) < 1/n^c.$$

□

Both the results obtained ρ^1_J and ρ^2_J depend strongly on the parameter values and cannot be estimated by any numerical value less than infinity.

4. m-machine cases

We consider the special case of the flow shop scheduling problem with identical processing time on each machine i.e. $p_{i,j} = p_j$ (Pinedo [11]). Recall that in the classical problem $Fm|p_{i,j} = p_j|C_{\max}$, the completion time $C_{[j]}$ of $J_{[j]}$ is

$$C_{[j]}(\pi) = \sum_{k=1}^{j} p_{[k]} + (m-1)\max\Big\{p_{[1]}, p_{[2]}, \dots, p_{[j]}\Big\}, \tag{3}$$

where π denotes some schedule, $[j]$ denotes the job that occupies the jth position in π. Hence, for the models (1) and (2), the completion time $C_{[j]}$ of $J_{[j]}$ are

$$C_{[j]}(\pi) = \sum_{k=1}^{j} p_{[k]}(a-bk) + (m-1)\max\Big\{p_{[1]}(a-b), p_{[2]}(a-2b), \dots, p_{[j]}(a-jb)\Big\}, \tag{4}$$

and

$$C_{[j]}(\pi) = \sum_{k=1}^{j} p_{[k]}k^c + (m-1)\max\Big\{p_{[1]}1^c, p_{[2]}2^c, \dots, p_{[j]}j^c\Big\}. \tag{5}$$

For the three objective functions: minimizing the weighted sum of completion times, minimizing maximum lateness and minimizing the number of tardy jobs, Mosheiov [8] showed that the solutions of the classical version do not hold with learning considerations in the single machine scheduling. In the following, we show that some special cases of the problems $Fm\big|p_{i,j} = p_j(a-br)\big|\sum w_jC_j$, $Fm\big|p_{i,j} = p_jr^c\big|\sum w_jC_j$, $Fm|p_{i,j} = p_j(a-br)|L_{\max}$ and $Fm|p_{i,j} = p_jr^c|L_{\max}$ can be polynomially solved.

Theorem 3. *For the problem $Fm|p_{i,j,r} = p_j(a-br)|\sum w_jC_j$, if jobs have agreeable weights, i.e., $p_j \le p_k$ implies $w_j \ge w_k$ for all jobs J_j and J_k, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of p_j/w_j (WSPT rule).*

Proof. (by contradiction). Consider a schedule π, suppose which is not WSPT, is optimal. In this schedule there must be at least two adjacent jobs, say $J_{[i]}$ and $J_{[i+1]}$ such that $p_{[i]}/w_{[i]} > p_{[i+1]}/w_{[i+1]}$, it implies $p_{[i]} \ge p_{[i+1]}$. Schedule $\bar{\pi}$ is obtained from schedule π by interchanging jobs in the ith and in the $(i+1)$st positions of π. We have

$$\begin{aligned}
&\sum w_jC_j(\pi) - \sum w_jC_j(\bar{\pi}) \\
&= w_{[i]}\Big(p_{[i]}(a-bi) + (m-1)\max\Big\{p_{[1]}(a-b), \dots, p_{[i-1]}(a-b(i-1)), \\
&\quad p_{[i]}(a-bi)\Big\}\Big) + w_{[i+1]}\Big(p_{[i]}(a-bi) + p_{[i+1]}(a-b(i+1)) \\
&\quad +(m-1)\max\Big\{p_{[1]}(a-b), \dots, p_{[i-1]}(a-b(i-1)), p_{[i]}(a-bi),
\end{aligned}$$

$$
\begin{aligned}
&p_{[i+1]}(a-b(i+1))\Big\}\Big) - w_{[i+1]}\Big(p_{[i+1]}(a-bi) \\
&+(m-1)\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),p_{[i+1]}(a-bi)\Big\}\Big) \\
&-w_{[i]}\Big(p_{[i+1]}(a-bi)+p_{[i]}(a-b(i+1))+(m-1)\max\Big\{p_{[1]}(a-b),\ldots, \\
&p_{[i-1]}(a-b(i-1)),p_{[i+1]}(a-bi),p_{[i]}(a-b(i+1))\Big\}\Big) \\
=\ & b(p_{[i]}-p_{[i+1]})\Big(w_{[i]}+w_{[i+1]}\Big)+\Big(w_{[i+1]}p_{[i]}-w_{[i]}p_{[i+1]}\Big)(a-b(i+1)) \\
&+w_{[i]}(m-1)\Big(\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),p_{[i]}(a-bi)\Big\}- \\
&\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),p_{[i+1]}(a-bi),p_{[i]}(a-b(i+1))\Big\} \\
&\Big)+w_{[i+1]}(m-1)\Big(\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),p_{[i]}(a-bi), \\
&p_{[i+1]}(a-b(i+1))\Big\}-\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)), \\
&p_{[i+1]}(a-bi)\Big\}\Big).
\end{aligned}
$$

Since $p_{[i]} \geq p_{[i+1]}$ and $p_{[i]}/w_{[i]} > p_{[i+1]}/w_{[i+1]}$, hence

$$p_{[i]}(a-bi) \geq p_{[i+1]}(a-bi), \quad p_{[i]}(a-bi) > p_{[i]}(a-b(i+1)),$$

$$b(p_{[i]}-p_{[i+1]})\Big(w_{[i]}+w_{[i+1]}\Big)+\Big(w_{[i+1]}p_{[i]}-w_{[i]}p_{[i+1]}\Big)(a-b(i+1))>0,$$

$$
\begin{aligned}
&\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),p_{[i]}(a-bi)\Big\} \\
&\quad -\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),p_{[i+1]}(a-bi), \\
&\qquad p_{[i]}(a-b(i+1))\Big\} \geq 0
\end{aligned}
$$

$$
\begin{aligned}
&\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),\quad p_{[i]}(a-bi),\quad p_{[i+1]}(a-b(i+1))\Big\} \\
&\quad -\max\Big\{p_{[1]}(a-b),\ldots,\quad p_{[i-1]}(a-b(i-1)),\quad p_{[i+1]}(a-bi)\Big\} \geq 0.
\end{aligned}
$$

Then $\sum w_jC_j(\pi) > \sum w_jC_j(\bar{\pi})$. It follows that the weighted sum of completion times under $\bar{\pi}$ is strictly less than under π. This contradicts the optimality of π and proves the theorem. □

Theorem 4. *For the problem* $Fm|p_{i,j,r} = p_jr^c\Big|\sum w_jC_j$*, if jobs have agreeable weights, i.e.,* $p_j \leq p_k$ *implies* $w_j \geq w_k$ *for all jobs* J_j *and* J_k*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* p_j/w_j *(WSPT rule).*

Proof. It is the same as Theorem 3. □

Corollary 1. *For the problem* $Fm|p_{i,j,r},\quad p_j = p\Big|\sum w_j C_j$*,an optimal schedule can be obtained by sequencing the jobs in non-increasing order of* w_j *(WSPT rule).*

Corollary 2. *For the problem* $Fm|p_{i,j,r},\quad w_j = wp_j\Big|\sum wp_j C_j$*,an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* p_j *(SPT rule).*

Theorem 5. *For the problem* $Fm|p_{i,j,r} = p_j(a-br)|L_{\max}$*, if jobs have agreeable weights, i.e.,* $p_j \leq p_k$ *implies* $d_j \leq d_k$ *for all jobs* J_j *and* J_k*, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of* d_j *(EDD rule).*

Proof. Consider a schedule π, suppose which is not EDD, is optimal. In this schedule there must be at least two adjacent jobs, say $J_{[i]}$ and $J_{[i+1]}$, such that $d_{[i]} > d_{[i+1]}$, it implies $p_{[i]} \geq p_{[i+1]}$. Schedule $\bar{\pi}$ is obtained from schedule π by interchanging jobs in the ith and in the $(i+1)$th positions of π. From the proof of Theorem 3, under π,

$$\begin{aligned} L_{[i]}(\pi) &= p_{[1]}(a-b) + \ldots + p_{[i-1]}(a-b(i-1)) + p_{[i]}(a-bi) \\ &\quad +(m-1)\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),p_{[i]}(a-bi)\Big\} - d_{[i]} \end{aligned}$$

$$\begin{aligned} L_{[i+1]}(\pi) &= p_{[1]}(a-b) + \ldots + p_{[i-1]}(a-b(i-1)) + p_{[i]}(a-bi) \\ &\quad +p_{[i+1]}(a-b(i+1)) + (m-1)\max\Big\{p_{[1]}(a-b),\ldots, \\ &\quad p_{[i-1]}(a-b(i-1)),p_{[i]}(a-bi),p_{[i+1]}\Big(a-b(i+1)\Big)\Big\} - d_{[i+1]} \end{aligned}$$

whereas under $\bar{\pi}$ it is

$$\begin{aligned} L_{[i]}(\bar{\pi}) &= p_{[1]}(a-b) + \ldots + p_{[i-1]}\Big(a-b(i-1)\Big) + p_{[i+1]}(a-bi) \\ &\quad +(m-1)\max\Big\{p_{[1]}(a-b),\ldots,p_{[i-1]}(a-b(i-1)),p_{[i+1]}(a-bi)\Big\} \\ &\quad -d_{[i+1]} \\ L_{[i+1]}(\bar{\pi}) &= p_{[1]}(a-b) + \ldots + p_{[i-1]}(a-b(i-1)) + p_{[i+1]}(a-bi) \\ &\quad +p_{[i]}\Big(a-b(i+1)\Big) + (m-1)\max\Big\{p_{[1]}(a-b),\ldots, \\ &\quad p_{[i-1]}\Big(a-b(i-1)\Big),p_{[i+1]}(a-bi),p_{[i]}\Big(a-b(i+1)\Big)\Big\} - d_{[i]} \end{aligned}$$

Since $d_{[i]} > d_{[i+1]}$ and $p_{[i]} \geq p_{[i+1]}$, then

$$\max\Big\{L_{[i]}(\bar{\pi}), L_{[i+1]}(\bar{\pi})\Big\} \leq \max\Big\{L_{[i]}(\pi), L_{[i+1]}(\pi)\Big\}$$

Hence, interchanging the position of the jobs $J_{[i]}$ and $J_{[i+1]}$ cannot increasing the value of $L_{\max}$. A finite number of such changes transform π into EDD, showing that EDD is optimal. □

Theorem 6. *For the problem $Fm|p_{i,j,r} = p_j r^c|L_{\max}$, if jobs have agreeable weights, i.e., $p_j \leq p_k$ implies $d_j \leq d_k$ for all jobs J_j and J_k, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of d_j (EDD rule).*

Proof. It is the same as Theorem 5. □

Corollary 3. *For the problem $Fm|p_{i,j,r}, p_j = p|L_{\max}$, an optimal schedule can be obtained by sequencing the jobs in non-decreasing order of d_j (EDD rule).*

5. Extensions

Badiru [2] presented a computational survey of the various learning curve models. Thus we can extend our results to other learning curves. Similar to Biskup [3] and Mosheiov [7], we assume that the actual processing time of the operation $O_{i,j}$ if scheduled in position r, is given by

(a) Dejong's learning curve (Dejong [4]).

$$p_{i,j,r} = p_{i,j}\Big[M + (1-M)r^a\Big], i = 1, 2, \ldots, m; r, j = 1, 2, \ldots, n, \tag{6}$$

where $a \leq 0$, M is the factor of incompressibility $(0 \leq M \leq 1)$.

(b) Pegels' learning curve (Pegels [10]).

$$p_{i,j,r} = p_{i,j}\Big[\alpha a^{r-1} + \beta\Big], i = 1, 2, \ldots, m; r, j = 1, 2, \ldots, n, \tag{7}$$

where α, a and β are parameters obtained empirically.

6. Conclusions

In this paper, we consider the flow shop scheduling problems, where job processing time is defined by position-dependent function. We show that Johnson's rule does not remain optimal for the problem $F2|p_{i,j,r}|C_{\max}$. We analyze the worst-case bound of Johnson's rule for the problem $F2|p_{i,j,r}|C_{\max}$. We note that the complexity status of this problem remains an open question and solving it seems an interesting topic for future research. We also find polynomial time solutions to some special cases of the considered problems for the following optimization criteria: the weighted sum of completion times and maximum lateness.

References

1. A. Bachman and A. Janiak, *Scheduling jobs with position-dependent processing times,* Journal of the Operational Research Society **55** (2004), 257-264.
2. A. B. Badiru, *Computational survey of univariate and multivariate learning curve models,* IEEE Transactions on Engineering Management **39** (1992), 176-188.
3. D. Biskup, *Single-machine scheduling with learning considerations,* European Journal of Operational Research **115** (1999), 173-178.
4. J. R. Dejong, *The effects of increasing skill on cycle time and its consequences for time standards,* Ergonomics **1** (1957), 51-60.
5. R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, *Optimization and approximation in deterministic sequencing and scheduling: a survey,* Annals of Discrete Mathematics **5** (1979), 287-326.
6. S. M. Johnson, *Optimal two and three stage production schedules with setup times included*, Naval Research Logistics Quarterly **1** (1954), 61-68.
7. G. Mosheiov, *Scheduling problems with a learning effect,* European Journal of Operational Research **132** (2001), 687-693.
8. G. Mosheiov, *Parallel machine scheduling with a learning effect,* Journal of the Operational Research Society **52** (2001), 1165-1169.
9. G. Mosheiov and J. B. Sidney, *Scheduling with general job-dependent learning curves,* European Journal of Operational Research **147** (2003), 665-670.
10. C. C. Pegels, *On start-up or learning curves: an expanded view,* AIIE Transactions **1** (1969), 316-322.
11. M. Pinedo, *Scheduling: Theory, Algorithms and Systems,* Prentice hall, Englewood cliffs, New jersey, 1995.
12. J.-B. Wang and Z.-Q. Xia, *Flow shop scheduling with a learning effect,* Journal of the Operational Research Society (to appear).
13. J.-B. Wang and Z.-Q. Xia, *No-wait or no-idle permutation flowshop scheduling with dominating machines,* J. Appl. Math. & Computing (to appear).
14. T. P. Wright, *Factors affecting the cost of airplanes,* Journal of the Aeronautical Sciences **3** (1936), 122-128.

Ji-Bo Wang is an assistant professor at the Department of Science at Shenyang Institute of Aeronautical Engineering. He is a Ph. D student at Dalian University of Technology. Mr. Wang's research interests are the scheduling problems and operations research methods and applications.

Department of Science, Shenyang Institute of Aeronautical Engineering, Shenyang 110034, P. R. China
e-mail: wjb7575@sina.com; wangjibo75@yahoo.com.cn