Journal of the Operations Research
Society of Japan
Vol. 23, No. 2, June 1980

ANALYSIS FOR MINIMIZING WEIGHTED MEAN FLOW-TIME IN FLOW-SHOP SCHEDULING

Shigeji Miyazaki
and
Noriyuki Nishiyama
University of Osaka Prefecture

(Received May 11, 1979; Revised February 7, 1980)

Abstract This paper deals with the problem of minimizing the weighted mean flow-time in n/m flow-shop scheduling where no passing is allowed. Analysis, through the adjacent pairwise interchange method, leads to a condition for determining the precedence relation between adjacent jobs. The condition consists of inequalities, the number of which equals the square of the number of machines. An algorithm based on these inequalities is proposed to obtain the optimal or near optimal solution. The numerical examples show that the algorighm can produce a solution which has an average approximation ratio of 91.4 percent over 160 problems. The three factors: the number of jobs, the number of machines and the range of weights do not affect the approximation ratio of the tested problems. The computational time required to obtain a solution through the proposed algorithm is proportional to (the number of jobs) x (the number of machines)^2. As a result, the CPU time needed to solve a seven job and six machine problem through TOSBAC 5600/120 is 0.25 sec.

1. Introduction

There have been many theoretical studies on flow-shop scheduling [1 ∿ 5, 8 ∿ 15, etc.]. The performance measures considered in these papers are mainly concentrated on maximal flow-time. In the previous paper [8], we investigated the minimization of mean flow-time in n/m flow-shop scheduling by means of adjacent pairwise interchange method. The paper presented sufficient conditions to decide the precedence relations between adjacent pairwise jobs. On the basis of the conditions, a computational algorithm was proposed for an optimal or near optimal solution.

The model studied in the paper [8], however, takes no account of job importance. In many situations, the jobs do not have equal importance. For instance, the earlier due-dates are, and the higher inventory costs are, the jobs should be regarded as more important objects for scheduling. This paper introduces the weighting factor w_i to each job (the larger w_i, the higher priority of job) and deals with the problem of minimization of weighted mean flow-time. A computational algorithm is presented for an optimal or near optimal solution on the basis of adjacent pairwise approach. The efficiency of the algorithm is verified by means of numerical experiment.

2. Flow-Shop Model

2.1 Definition and Notation of the Model

The discussed model can be stated as follows:

1) Let n be the number of jobs to be processed, and ith job in the arbitrary sequence S is denoted by J_i where $i=1,2,\ldots,n$. All these jobs are available for processing at time zero.

2) The manufacturing system consists of m different machines which are numbered according to the order of production stage. Let M_j be the jth machine in the system where $j=1,2,\ldots,m$. Every machine is continuously available. A machine can process only one job at a time.

3) Every job is completed through the same production stage that is $M_1 \to M_2 \to, \ldots, \to M_m$.

4) Let $p_{i,j}$ denote the processing time of J_i on M_j. Setup times for operations are sequence-independent and are included in processing times. Handling times are assumed to be so limited that they can be neglected.

5) Let $F_j(i)$ denote the partial flow-time of J_i counted from the starting time of first job J_1 on M_1 to the completion time of J_i on M_j, referring to Fig. 1. In paticular, $F_m(i)$ is called as flow-time of J_i.

6) The same job sequence occurs on each machine; in other words, no passing is allowed in the shop.

7) Each job is assigned weight w_i according to its importance.

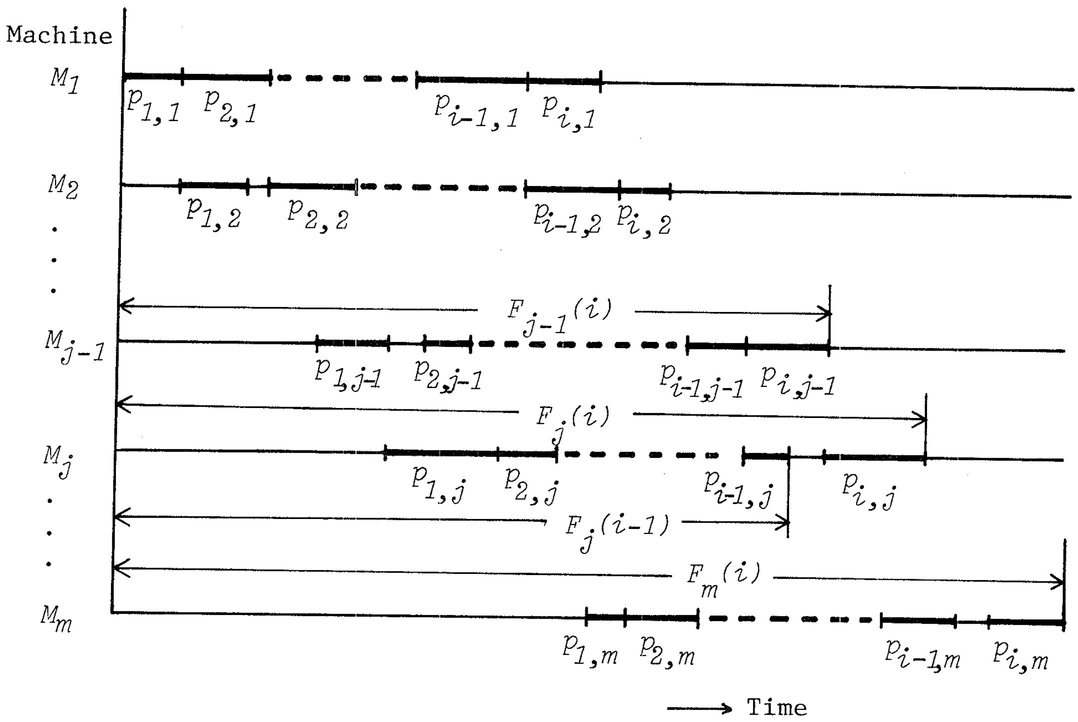

Fig. 1. Definition of $F_j(i)$.

2.2 Performance Measure

The performance measure studied is weighted mean flow-time defined by:

(1.1) $\bar{F}_w = \{\Sigma_{i=1}^{n} W_i F_m(i)\}/n.$

This measure can be redefined as:

(1.2) $\bar{F}_w = \{\Sigma_{i=1}^{n} W_i F_m(i)\}/\Sigma_{i=1}^{n} W_i.$

Each definition produces the same solution, since the denominators of (1.1) and (1.2) are sequence-independent. From (1.1) we have:

(1.3) $$n\bar{F}_w = \Sigma_{i=1}^{n} W_i F_m(i),$$

where $n\bar{F}_w$ expresses the total weighted flow-time. $n\bar{F}_w$ shall be used in place of $\bar{F}_w$ in the further analysis.

3. Analysis

In the sequence S, let s be a subsequence consisting of the first $q-1$ jobs, that is, $J_1, J_2, \dots, J_{q-1}$, and in succession to s, J_q and J_{q+1} (these two jobs are called adjacent two jobs hereafter) are assumed to be processed in the order $J_q\ J_{q+1}$. Now consider the sequence S' in which J_q and J_{q+1} are pairwise interchanged and are processed in the order $J_{q+1}\ J_q$. The sequence is the same for the first $q-1$ jobs and the last $(n-q-1)$ jobs under either S or S' as illustrated in Fig. 2.

$$S = J_1,\ J_2, \dots,\ J_{q-1},\ J_q,\ J_{q+1}, \dots,\ J_n$$

$$S' = \underbrace{J_1,\ J_2, \dots,\ J_{q-1}}_{\text{partial sequence } s},\ J_{q+1},\ J_q, \dots,\ J_n$$

Fig. 2. The Relationship between Sequence S and S'

In order to distinguish the notation of partial flow-time under S from S', let $F_j(q)$, $F_j(q,q+1)$, and $F_j(i)$ $(i=q+2,q+3,\dots,n)$ denote the partial flow-time of J_q, J_{q+1}, and J_i $(i=q+2,q+3,\dots,n)$ under S in turn, and let $F'_j(q+1)$, $F'_j(q+1, q)$, and $F'_j(i)$ $(i=q+2,q+3,\dots,n)$ denote the partial flow-time of J_{q+1}, J_q, and J_i $(i=q+2,q+3,\dots,n)$ under S' in turn, moreover let $\bar{F}'_w$ be the weighted mean flow-time under S'. Then the total weighted flow-times under S and S' are expressed by:

(3.1) $$n\bar{F}_w = \Sigma_{i=1}^{q-1} W_i F_m(i) + W_q F_m(q) + W_{q+1} F_m(q,q+1) + \Sigma_{i=q+2}^{n} W_i F_m(i),$$

and

(3.2) $$n\bar{F}'_w = \Sigma_{i=1}^{q-1} W_i F_m(i) + W_{q+1} F'_m(q+1) + W_q F'_m(q+1,q) + \Sigma_{i=q+2}^{n} W_i F'_m(i).$$

Eliminating the common terms between (3.1) and (3.2) from the each equation, and denoting the remaining, $<n\bar{F}_w>$ and $<n\bar{F}'_w>$, respectively, we have:

(3.3) $$<n\bar{F}_w> = W_q F_m(q) + W_{q+1} F_m(q,q+1) + \Sigma_{i=q+2}^{n} W_i F_m(i),$$

and

(3.4) $$<n\bar{F}'_w> = W_{q+1} F'_m(q+1) + W_q F'_m(q+1,q) + \Sigma_{i=q+2}^{n} W_i F'_m(i).$$

If

(3.5) $$<n\bar{F}_w> \leqq <n\bar{F}'_w>$$

that is:

(3.6) $$n\bar{F}_w \leqq n\bar{F}'_w$$

holds, J_{q+1} cannot directly precede J_q in the optimal sequence. Therefore, we shall investigate the sufficient conditions, which have transitive property of job ordering, for satisfying (3.5) independently of the two jobs' position, as shown in the following:

Comparing each term of (3.3) with the corresponding term of (3.4), we have:

(3.7) $$W_q F_m(q) \leqq W_{q+1} F'_m(q+1),$$

(3.8) $$W_{q+1} F_m(q,q+1) \leqq W_q F'_m(q+1,q),$$

and

(3.9) $$\Sigma_{i=q+2}^{n} W_i F_m(i) \leqq \Sigma_{i=q+2}^{n} W_i F'_m(i),$$

which are to be sufficient conditions for (3.5).

There exist next recurrence relations on partial flow-time $F_j(i)$, referring to Fig. 1.

(3.10) $$F_j(i) = \max\{F_{j-1}(i),\ F_j(i-1)\} + P_{i,j},$$
$$(i = 1, 2, \ldots, n\ ;\ j = 1, 2, \ldots, m),$$

where $F_0(i) \equiv 0$, $F_j(0) \equiv 0$.

Working out the recurrence relations (3.10), we have:

(3.11) $$F_j(i) = \max_{r=1\sim j}\{F_{j-r+1}(i-1) + \Sigma_{t=1}^{r} P_{i,j-t+1}\}.$$

Substituted into (3.7), (3.11) gives

(3.12) $$W_q \max_{r=1\sim m}\{F_{m-r+1}(q-1) + \Sigma_{t=1}^{r} P_{q,m-t+1}\}$$
$$\leqq W_{q+1} \max_{r=1\sim m}\{F_{m-r+1}(q-1) + \Sigma_{t=1}^{r} P_{q+1,m-t+1}\}.$$

The comparison between the respectively corresponding terms of (3.12) gives the following sufficient conditions of (3.7):

(3.13) $$W_q \leqq W_{q+1},$$

and

(3.14) $$W_q\, \Sigma_{t=1}^{r} P_{q,m-t+1} \leqq W_{q+1}\, \Sigma_{t=1}^{r} P_{q+1,m-t+1},\quad (r = 1, 2, \ldots, m).$$

Now the partial flow-time $F_j(i,i+1)$ is given as similar to (3.10),

(3.15) $$F_j(i,i+1) = \max\{F_{j-1}(i,i+1),\ F_j(i)\} + P_{i+1,j},$$
$$(i = 1, 2, \ldots, n-1\ ;\ j = 1, 2, \ldots, m),$$

where $F_0(i,i+1) \equiv 0$, $F_j(0) \equiv 0$.

Working out the recurrence relation (3.15), we have:

(3.16) $$F_j(i,i+1) = \max_{r=1\sim j}\ \max_{t=1\sim r}\{F_{j-r+1}(i-1) + \Sigma_{k=j-t+1}^{j} P_{i+1,k}$$
$$+ \Sigma_{k=j-r+1}^{j-t+1} P_{i,k}\}.$$

Substituted into (3.8), (3.16) gives

(3.17) $$W_{q+1} \max_{r=1\sim m}\ \max_{t=1\sim r}\{F_{m-r+1}(q-1) + \Sigma_{j=m-t+1}^{m} P_{q+1,j} + \Sigma_{j=m-r+1}^{m-t+1} P_{q,j}\}$$
$$\leqq W_q \max_{r=1\sim m}\ \max_{t=1\sim r}\{F_{m-r+1}(q-1) + \Sigma_{j=m-t+1}^{m} P_{q,j} + \Sigma_{j=m-r+1}^{m-t+1} P_{q+1,j}\}.$$

The comparison between the respectively corresponding terms of (3.17) gives:

(3.18) $W_q \geqq W_{q+1}$,

(3.19) $W_q \sum_{j=m-t+1}^{m} P_{q,j} \geqq W_{q+1} \sum_{j=m-t+1}^{m} P_{q+1,j}$,

$(t = 1, 2, \ldots, m)$,

and

(3.20) $(\sum_{j=m-r+1}^{m-t+1} P_{q,j})/W_q \leqq (\sum_{j=m-r+1}^{m-t+1} P_{q+1,j})/W_{q+1}$,

$(r = 1, 2, \ldots, m \; ; \; t = 1, 2, \ldots, r)$.

If

(3.21) $F_m(i) \leqq F'_m(i)$, $(i = q+2, q+3, \ldots, n)$

hold, (3.9) should be satisfied. Moreover, Yueh [15] shows that

(3.22) $\min (P_{q,u}, P_{q+1,v}) \leqq \min (P_{q+1,u}, P_{q,v})$, $(1 \leqq u < v \leqq m)$

is the sufficient condition of (3.21) that is (3.9).

The discussion above has led the sufficient conditions of (3.7), (3.8) and (3.9) individually. Since all of these sufficient conditions have transitive property, the temporary sequence can be induced from each sufficient condition. In the case that all of these temporary sequences are equal to each other, the sequence is the optimal solution for this problem. According to the following algorithm an optimal solution can be produced in this case. In the usual cases in which all of the temporary sequences do not coincide with one another, a suboptimal solution can be obtained through the same algorithm.

Considering that (3.5) is composed of the sum of (3.7), (3.8), and (3.9), we make, in the algorithm, a solution by the procedure that calculates the sum of the ordinal numbers according to the temporary sequences. Between two inequalities (3.13) and (3.18) the expressions of the both sides are identical but only the sign of inequality is opposite. Such is the case between inequalities (3.14) and (3.19) too. The sum of the ordinal numbers derived from four inequalities: (3.13), (3.14), (3.18), and (3.19) becomes always equal to each other job. Therefore, we can eliminate these four inequalities in the following algorithm from the beginning.

4. Algorithm

The algorithm will be explained by solving an example problem listed in Table 1.

Step 1. Decide the $m(m+1)/2$ kinds of temporary sequences which can be led from (3.20) as follows: calculate the value $(\sum_{j=m-r+1}^{m-t+1} P_{i,j})/w_i$ of all jobs for each combination of $r(=1,2,..,m)$ and $t(=1,2,...,r)$, as tabulated in Table 2. Make the temporary sequences in accordance with the non-decreasing order of each row value in Table 2. Assign an integer to each job according to its order, as shown in Table 3. In case more than two jobs have the same value in a row, assign the same integers to them.

Step 2. Make the temporary sequence in which all jobs satisfy (3.22) for each combination of u and v, using Johnson's Algorithm [5]. Assign an integer to each job as similar to Step 1. The results of this is indicated in Table 4. This step produces $m(m-1)/2$ kinds of temporary sequences.

Step 3. Calculate the sum of integers assigned to each job in the Step 1 and 2 as Table 5. Arrange each job in the nondecreasing order of the total integers. Break a tie by placing jobs with lower original numbers first.

The solution for this example becomes J_1-J_4-J_2-J_3.

Table 1. Four Job Three Machine Problem.

Job		J_1	J_2	J_3	J_4
Processing times	M_1	2	5	3	6
	M_2	3	8	6	3
	M_3	2	5	4	7
Weight		6	5	3	7

Table 2. The Value of $(\sum_{j=m-r+1}^{m-t+1} P_{i,j})/w_i$.

r	t	Job J_1	J_2	J_3	J_4
1	1	2/6	5/5	4/3	7/7
2	1	5/6	13/5	10/3	10/7
	2	3/6	8/5	6/3	3/7
3	1	7/6	18/5	13/3	16/7
	2	5/6	13/5	9/3	9/7
	3	2/6	5/5	3/3	6/7

Table 3. Ordinal Numbers by Step 1.

r	t	Job J_1	J_2	J_3	J_4
1	1	1	2	4	2
2	1	1	3	4	2
	2	2	3	4	1
3	1	1	3	4	2
	2	1	3	4	2
	3	1	3	3	2

Table 4. Ordinal Numbers by Step 2.

u	v	Job J_1	J_2	J_3	J_4
1	2	1	3	2	4
	3	1	3	2	4
2	3	4	2	3	1

Table 5. Sum of Ordinal Numbers.

Job	J_1	J_2	J_3	J_4
Sum of ordinal numbers	13	25	30	20

5. Efficiency of the Algorithm

5.1 Approximation Ratio

The definition of the approximation ratio, to evaluate the quality of the solution, used in this paper, is different from that often used in previous papers [3, 9, etc.]. Previously, the approximation ratio was simply defined by:

$$\eta_1 = 100 \times (o/a), \quad (\%) \tag{5.1}$$

where o and a are the values of performance measures of the optimal and obtained solutions, respectively. This ratio, however, does not take into consideration the existing range of possible solutions. Consequently, it has the following shortcomings: Suppose that there exist two flow-shop scheduling problems I and II of which possible solutions are distributed as shown in Fig. 3. If the obtained solutions a_I and a_{II} for each problem have a equal value of performance measure, the approximation ratio defined by η_1 indicates the same percentage. The quality of a_{II}, however, is practically higher than a_I, as the existing range of possible solutions for problem II is wider than that for problem I. Moreover, it is a shortcoming of η_1 that it indicates a percentage grater than zero even if the obtained solution coincides with the worst possible solution.

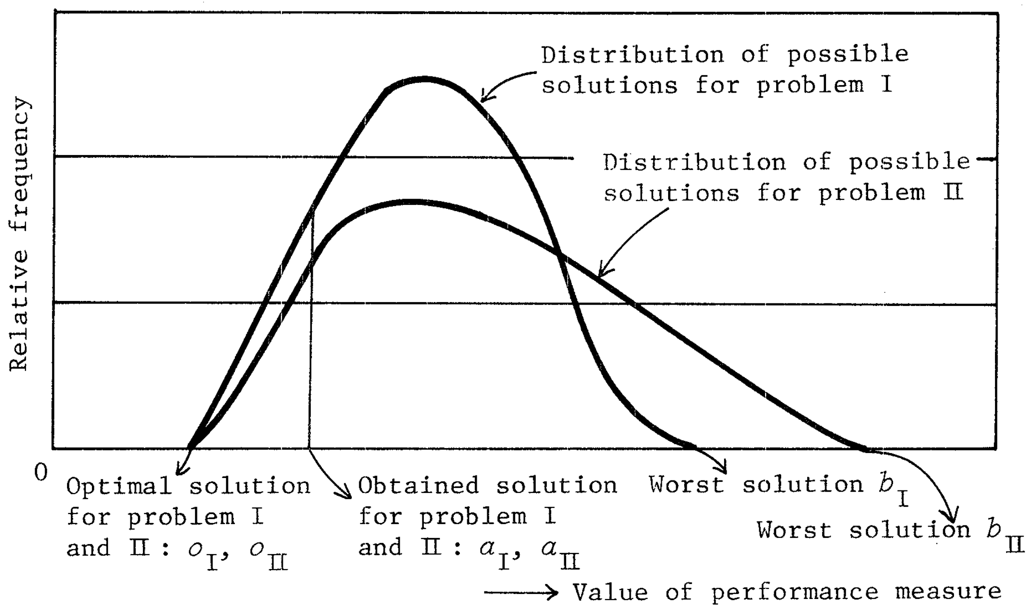

Fig. 3. Distribution of Possible Solutions for Problem I and II .

We defined the approximation ratio as:

(5.2) $\eta_2 = 100 \times (b-a)/(b-o) \quad (\%)$

where b is the value of worst possible solution [8]. This ratio contains the optimal and the worst value of performance measure so as to reach 0% in case the obtained solution coincides with the worst one, and 100% in case the obtained solution coincides with the optimal one.

5.2 Computational Experience

In order to verify the efficiency of the algorithm, the example problems composed of 4 ∿ 7 jobs and 4 ∿ 6 machines are solved through the proposed algorithm and the solutions are evaluated by approximation ratio η_2. Table 6 shows the results of this evaluation together with η_1 for references. The processing times in the example problems are distributed uniformly between 1 and 99, and the weights assigned to jobs are distributed uniformly between 1 and 10 or 1 and 40. The optimal and worst solutions to calculate η_2 were obtained from complete enumeration method.

Table 6. Results of the Experiment

Problems					Approximation ratio			
					η_1		η_2	
n	m	p_i	w_i	Problem numbers	Mean	Range	Mean	Range
4	4	1 ∿ 99	1 ∿ 10	20	96.5	100.0 ∿ 75.8	88.4	100.0 ∿ 31.3
			1 ∿ 40	20	96.6	100.0 ∿ 81.8	91.1	100.0 ∿ 37.0
5	5	1 ∿ 99	1 ∿ 10	20	97.1	100.0 ∿ 87.8	92.6	100.0 ∿ 64.9
			1 ∿ 40	20	96.0	100.0 ∿ 84.7	89.9	100.0 ∿ 55.4
6	6	1 ∿ 99	1 ∿ 10	20	96.9	100.0 ∿ 91.1	93.6	100.0 ∿ 81.2
			1 ∿ 40	20	95.3	100.0 ∿ 86.9	90.5	100.0 ∿ 75.8
7	6	1 ∿ 99	1 ∿ 10	20	94.8	100.0 ∿ 88.8	91.0	100.0 ∿ 73.3
			1 ∿ 40	20	96.4	100.0 ∿ 89.9	93.9	100.0 ∿ 76.4
Grand average					96.2		91.4	

The results of Table 6 indicate that the average approximation ratio η_2 is 91.4% and that none of the three factors, the number of jobs, the number of machines, and the range of weights affect the average approximation ratio for the tested problems. The minimal approximation ratio becomes larger as the number of jobs and the number of machines increase.

Table 7. Mean Computational Times (CPU Time, sec.)

Problem n	m	Proposed method	Complete enumeration
4	4	0.09	0.20
5	5	0.14	0.80
6	6	0.21	2.50
7	6	0.25	16.0

Table 7 shows the mean computational times of TOSBAC-5600/120 required to obtain a solution through the proposed algorithm and complete enumeration, respectively. Little time variation occurs in solving the problem which has the same job number and machine number, through both methods. The structure of the algorithm should lead the computational times to be proportional to (the number of jobs) $\times$ (the number of machines)^2.

The number of machines, which affects the computational time quadratically, has an upper limitation practically, for it coincides with the number of operations needed to complete a job in a flow-shop. Data from an actual machining shop indicate that over ninety-five percent of jobs are produced through the operation stages less than 11 [7]. Although the number of jobs becomes considerably large in practical shop, the computational time of the algorithm goes up just linearly with the number of jobs. The discussion above shows that the algorithm is much effective than general purpose optimizing techniques such as B. & B. method [14] or D.P. [6] from the viewpoint of computational times.

The required memory capacity should never become a major limitation in executing the algorithm, as it needs only 68K for solving a 1000 job and 25 machine problem.

6. Conclusion

In this paper, we dealt with the minimization of weighted mean flow-time problem in *n/m* flow-shop scheduling. A computational algorithm was proposed through an adjacent pairwise approach. In order to evaluate the quality of the solution, we used the new approximation ratio η_2 derived from the discussion of the previous approximation ratio η_1. The algorithm produces the solution which has 91.4% of average approximation ratio η_2. The computational time for the algorithm is proportional to (the number of jobs) × (the number of machines)^2. As a result, the CPU time needed to solve a seven job and six machine problem through TOSBAC 5600/120 is below 0.3 sec. Memory capacity should never be the major restriction on solving practical problems.

References

[1] Furukawa, M., Kakazu, Y. and Okino, N.: A Practical Method of Solving Flow Shop Scheduling Problems. *Journal of the Japan Society of Precision Engineering*, (in Japanese), Vol. 43, No. 2(1977), 156-161.

[2] Gupta, J. N. D.: Optimal Flowshop Scheduling with Due Dates and Penalty Costs. *Journal of the Operations Research Society of Japan*, Vol. 14, No. 1(1971), 35-46.

[3] Gupta, J. N. D.: Heuristic Algorithms for Multistage Flowshop Scheduling Problem. *AIIE Transactions*, Vol. 4, No. 1(1972), 11-18.

[4] Ignall, E. and Schrage, L.: Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems. *Operations Research*, Vol. 13, No. 3(1965), 400-412.

[5] 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.

[6] Lawler, E. L.: On Scheduling Problems with Deferal Costs. *Management Science*, Vol. 11, No. 2(1964), 280-288.

[7] Miyazaki, S. and Hashimoto, F.: Applications of Probabilistic Dispatching Method to the Dynamic Scheduling Problem. *Journal of Japan Industrial Management Association*, (in Japanese), Vol. 28, No. 1(1977), 84-90.

[8] Miyazaki, S., Nishiyama, N., and Hashimoto, F.: An Adjacent Pairwise Approach to the Mean Flow-Time Scheduling Problems. *Journal of the Operations Research Society of Japan*, Vol. 21, No. 2(1978), 287-301.

[9] Nabeshima, I.: The Order of n Items Processed on m Machines [III]. *Journal of the Operations Research Society of Japan*, Vol. 16, No. 3(1973), 163-185.

[10] Nabeshima, I.: *Scheduling Theory* (in Japanese). Morikita-Shuppan Co., Tokyo, 1974.

[11] Panwalkar, S. S. and Khan, A. W.: An Ordered Flow-Shop Sequencing Problem with Mean Completion Time Criterion. *International Journal of Production Research*, Vol. 14, No. 5 (1976), 631-635.

[12] Smith, M. L., Panwalkar, S. S. and Dudek, R. A.: Flowshop Sequencing Problem with Ordered Processing Time Matrices. *Management Science*, Vol. 21, No. 5(1975), 544-549.

[13] Szwarc, W.: Optimal Elimination Methods in the $m \times n$ Flow-Shop Scheduling Problem. *Operations Research*, Vol. 21, No. 6(1973), 1250-1259.

[14] Uskup, E. and Smith, S. B.: A Branch-and-Bound Algorithm for Two-Stage Production-Sequencing Problems. *Operations Research*, Vol. 23, No. 1 (1975), 118-136.

[15] Yueh Ming-I: On the n Job, m Machine Sequencing Problem of Flow-Shop. *OR'75: Proceedings of the Seventh International Conference on Operational Research (ed. K. B. Haley).* North-Holland Publishing Company, Amsterdam, 1976, 179-200.

Sigeji MIYAZAKI: Department of Industrial Engineering, College of Engineering, University of Osaka Prefecture, Mozu-Umemachi, Sakai, Osaka, 591 Japan.

アブストラクト

フロー・ショップ・スケジューリングにおける重みつき平均滞留時間小化問題の解析

大阪府立大学 宮崎茂次
西山徳幸

本論文では，ジョブ数，機械台数が任意で追抜禁止のフロー・ショップ・スケジューリングにおいて，重みつき平均滞留時間最小化問題を取り扱っている。一般に，スケジューリングの対象となるジョブには，納期までの余裕時間や，仕掛在庫コストなどの大小によって，重要視されるものとそれほど重要視されないものとがある。そこで，各ジョブの重要度に応じた「重み」を付与したモデルを設定し，重みの大きなジョブには高い優先度を与えるという重みつき平均滞留時間を評価尺度に取り上げる。

隣接2ジョブ交換法による解析で，隣接ジョブの先行関係を決定するための次のような不等式を導いた。

$$\left(\sum_{j=m-r+1}^{m-t+1} P_{q,j}\right)/W_q \leqq \left(\sum_{j=m-r+1}^{m-t+1} P_{q+1,j}\right)/W_{q+1},$$
$$(r=1,2,\cdots\cdots,m;\ t=1,2,\cdots\cdots,r),$$

$$\min(P_{q,u}, P_{q+1,v}) \leqq \min(P_{q+1,u}, P_{q,v}),$$
$$(1 \leqq u < v \leqq m)$$

ジョブに関する推移性を満足するこれらの不等式に基づいて，重みつき平均滞留時間最小化のための近似アルゴリズムが提案されている。

ジョブ数を4～7，機械台数を4～6に設定し，各ジョブの重みを1～10，1～40の一様乱数で与えた例題を160種類作成して，アルゴリズムの有効性を検証した。その結果，提案アルゴリズムで平均91.4％の近似率をもつ解を得ることができた。例題のジョブ数，機械台数および重みの範囲は，近似率に影響を与えなかった。なお，求めた解が最適解にどの程度近いかを表すための近似率は，従来のものの問題点を指摘し，これに代わる新しい近似率を使用した。

提案アルゴリズムで解を得るために必要な計算時間は，(ジョブ数)×(機械台数)^{2}に比例し，たとえば7ジョブ，6機械問題をTOSBAC－5600／120で解くのに0.25秒要した。また記憶容量は，実用的規模のスケジューリング問題を解く際の主要な制約とはならないことなどが判明した。