A study on *m*-machines *n*-jobs flow shop scheduling problems

Yudai Kitagawa (Faculty of Production Systems Engineering and Sciences, Komatsu University, 18111020@komatsu-u.ac.jp, Japan)
Shinichi Funase (Faculty of Production Systems Engineering and Sciences, Komatsu University, shinichi.funase@komatsu-u.ac.jp, Japan)
Toshihiko Shimauchi (Faculty of Intercultural Communication, Komatsu University, toshihiko.shimauchi@komatsu-u.ac.jp, Japan)
Haruhiko Kimura (Faculty of Production Systems Engineering and Sciences, Komatsu University, haruhiko.kimura@komatsu-u.ac.jp, Japan)

Abstract

This paper deals with one of the most difficult sequencing problems, *m*-machines *n*-jobs flow shop scheduling problems. Conventional methods adopted in previous studies include the Monte Carlo method by sampling or the simulation method in which the target or evaluation standard is determined and some machining priority rules are used. In this paper, we employed the Monte Carlo method. Sampled "job execution sequence" is arranged in ascending order of the total elapsed time, and the features obtained from the top "job execution sequence" are reported. In addition, we investigated the relationship between the operating time of each machine on each job and optimum solutions. Specifically, a simple setting with same operating time for each machine was used as a baseline, and then cases where the operating times were changed stepwise were considered to show these changes lead to a proportional rise in difficulty in finding an optimum solution. These results can significantly contribute to a more efficient determination of optimum or quasi-optimum solutions.

Key words

flow-shop scheduling problem, *m*-machines *n*-jobs problem, job execution sequence, total elapsed time, Gantt chart

1. Introduction

The production system can be categorized into two groups: one is to minimize the total cost by finding the optimum number of machines such as a meeting problem, and the other is to minimize the total elapsed time until all jobs are completed by finding the optimum machining (job) sequence. The former addresses problems related to the waiting time of both arriving customers (jobs) and the windows (machines), and the waiting time and the number of waiting lots are stochastically studied by increasing or decreasing the number of machines. However, companies are required to make the best use of the machines' capabilities without changing the number of machines as much as possible. To meet these operational requirements, the latter aims to minimize the total elapsed time by fixing some machines to the current state and managing the processing sequence of arriving lots and waiting lots. These are problems of determining the machining (job) sequence so as to increase the operating time of each machine or minimize the related cost. This is called the sequencing theory (Ibaraki et al., 2011; Mohammed, 2016; Sakuraba, 2010). The issues dealt with in this paper fall into the latter category of sequencing problems.

The sequencing problem is a common problem at manufacturing sites, especially in machine assembly factories and semi-process factories. In the "*m*-machine *n*-job" sequencing problem, there are *n* types of jobs executed by using m-machines. The aim is to find the order of job execution of each machine to minimize the total elapsed time when either (1) the job order can be changed for each machine, or (2) the machine usage order of each job is the same. The former case is called the job-shop scheduling problem (Hino, 2017), and the latter is called the flow-shop scheduling problem, where various models are proposed (Imaizumi, 2000).

In this paper, we focus on the flow shop scheduling problem whose solutions can be divided into following two categories:

- Optimum solution based on the branch-and-bound algorithm
- Approximate solution method

There are many studies on optimum solutions, many of which are introduced in Baker (1975) and Cariler and Rebai (1996). Among them, an efficient algorithm called Johnson's rule (Johnson, 1954; Johnson's rule, 2021) that finds the optimum solution (execution order of job to minimize the total elapsed time) of 2-machines *n*-jobs type is famous. be. As for 3-machines *n*-jobs problems, Johnson's rule can be used only when certain conditions are satisfied, in which case the optimum solution can be efficiently obtained (Moriya, 1973). For other *m*-machine *n*-job problems, NP-hard occurs and there is no algorithm that efficiently finds the optimum solution (Ibaraki, 1994). Therefore, to find the optimum solution for these problems, the complete enumeration method where all the solutions are calculated to find the best one is applied, which requires a huge amount of time.

In this research, the number of possible "job execution sequence" is *n!* and the optimum solution is the job execution sequence that minimizes the total elapsed time. The total elapsed time can be calculated using the Gantt chart

(2021) once the job execution sequence is determined. The time complexity required for creating the chart is $O\ (n \cdot m)$. Therefore, the time complexity required to find the optimum solution by the complete enumeration method is $O\ (n! \cdot N \cdot m)$. In the approximate solution method, there are a Monte Carlo method by sampling and a simulation method in which a target or an evaluation standard is set and some machining priority rules are used (Moriya, 1973).

The purpose of this paper is to find the optimum or semi-optimum solution in a short time within the permissible range. The method used is to determine the number of samples from the permissible time range, arrange the "job execution sequence" sampled using the Monte Carlo method in ascending order of total elapsed time, and select the upper ranked "job execution sequence". In addition, by summarizing the characteristics of the experimental results obtained from the upper "job execution sequence", new findings for obtaining the optimum or quasi-optimum solution in the permissible time range are shown. Furthermore, a simple setting with same operating time for each machine is used as a baseline, and then cases where the operating times are changed stepwise are considered to show these changes lead to a proportional rise in difficulty in finding an optimum solution.

2. Proposed method

The problem addressed in this paper is the m-machines n-jobs problem, assuming the sequence of job for each machine is same. The Monte Carlo method by sampling is used. The "job execution sequence" are sampled and arranged in ascending order of the total elapsed time to obtain features of the upper "job execution sequence". Samplings are conducted so that the "job execution sequence" of the top 5 %, 10 %, 15 %, · · ·, 30 %, arranged in ascending order based on the total elapsed time, are included with a 95 % probability.

2.1 Outline of proposed method

The Monte Carlo method by importance sampling is applied. The "job execution sequence" is randomly selected and repeated several times to obtain the "job execution sequence" of t routes. The "job execution sequence" of the top r routes ($r < t$), arranged in ascending order of the total elapsed time, is extracted.

The method of randomly selecting the "job execution sequence" is as follows:

As an initial value, 1, 2, · · ·, n are input respectively into $A(1)$, $A(2)$, · · ·, $A(n)$. The result goes into $B(1)$, $B(2)$, · · ·, $B(n)$. INT (X) is a function that truncates the decimal point of X. RND (1) is a real random number between 0 and 1 (0 < RND (1) < 1).

```
For i = 1 to n
  A(i) = i
Next i
For i = n to 1 step – 1
  t = INT (RND (1) × i) + 1
  B(i) = A(t)
  A(t) = A(i)
Next i
```

2.2 Example 1

The number of execution sequence of n-jobs is $n!$ routes. How many samples (k) are required to ensure that the execution sequence of a certain job is within the top P (× 100) % with a probability of α (× 100) % or higher, when the sequences are arranged in ascending order of the total elapsed time until n-jobs completion?

Since the probability that one sample falls within P (× 100) % is P, the probability that one sample does not fall within P is $1 - P$. The probability that the number of samples k does not fall within P (× 100) % is $(1 - P)^k$ and the probability that at least one of k falls within P is $1 - (1 - P)^k$. Since this probability should be set to α or more,

$$
\begin{aligned}
&1 - (1 - P)^k \geqq \alpha \\
&1 - \alpha \geqq (1 - P)^k \\
&\log (1 - \alpha) \geqq k \log (1 - P) \\
&k \geqq \log (1 - \alpha) \div \log (1 - P)
\end{aligned}
\tag{1}
$$

For example, to find k with a 95 % probability of being in the top 10 %, $\alpha = 0.95$ and $P = 0.1$ are input to Equation (1) to yield 28.3. Hence the necessary number of samples is 29 or more.

3. Experiments

3.1 Experiment 1

The accuracy of the uniform random numbers used in this paper was confirmed by the following experiments.

3.1.1 Experiment 1 (a)

As a first step, 1000 sets of the execution sequence of 10 jobs J_1, J_2, · · ·, J_{10} were generated by using the uniform random numbers and the average rank of each job J_i was calculated.

3.1.2 Results

The results in Table 1 show that the sequence of job execution is different in each set. A_i, the theoretical value of the average ranking of each job J_i, can be calculated as follows:

$$
\begin{aligned}
A_i &= \Sigma_i / 10 \\
&= (1 + 2 + \cdots + 10) \div 10 \\
&= 5.5
\end{aligned}
$$

The value of A_i and the actual average ranking of each job

Table 1: Execution sequence of each job

No.	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
1	8	2	6	7	3	10	5	9	1	4
2	6	1	3	2	7	9	5	4	8	10
3	6	7	1	8	10	5	4	9	2	3
4	1	8	9	2	5	10	3	7	4	6
5	1	2	7	5	9	10	8	6	4	3
6	3	1	10	5	7	8	6	4	9	2
7	3	7	8	2	9	6	1	4	10	5
8	2	4	7	5	10	1	8	9	3	6
9	9	8	5	3	7	4	10	6	2	1
10	9	2	4	10	1	5	6	3	8	7
11	6	2	3	8	10	9	4	1	5	7
12	3	4	8	9	1	6	2	5	7	10
13	5	7	10	1	4	9	3	6	8	2
14	1	6	8	10	5	4	7	3	2	9
15	10	3	1	7	9	5	2	6	4	8
16	8	6	1	2	4	3	9	10	5	7
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
991	9	1	5	4	8	7	10	2	3	6
992	2	9	4	3	7	10	8	1	5	6
993	2	3	7	1	8	5		9	6	10
994	8	10	9	5	1	2	3	4	7	6
995	4	5	6	3	7	8	10	2	1	9
996	1	4	2	8	9	6	5	7	10	3
997	1	4	8	7	2	9	5	10	3	6
998	8	3	4	7	2	10	6	5	9	1
999	9	8	2	7	4	6	5	10	1	3
1000	1	3	7	5	6	4	9	8	2	10
Total	5470	5523	5527	5394	5563	5509	5471	5484	5437	5622
Average	5.47	5.52	5.53	5.39	5.56	5.51	5.47	5.48	5.44	5.62

$J_i = 5.62$, as shown in Table 1, are almost identical.

3.1.3 Experiment 1 (b)

In the next step, the frequency of the sequence of execution of each job in Table 1 was obtained to check even dispersions of the sequence of execution of each job.

3.1.4 Results

The results in Table 2 show that the sequence of execution of each job is dispersed around 100 times, with infrequent outlying cases such as 73 and 127 times.

The results of Experiments 1(a) and (b) suggest the accuracy of the uniform random numbers used in this paper.

3.2 Experiment 2

2-machines 10-jobs problem was addressed in the experiment. Samplings containing with a 95 % probability the "job execution sequence" of the top 5 %, 10 %, 15 %, · · ·, 30 %, arranged in ascending order based on the total elapsed time were used. The operating time of each machine on each job was set to two digits (uniform random numbers) and the same data was used repeatedly in each experiment session.

3.2.1 Results

Table 3 shows the operating time of each machine on each job. From Equation (1), 59 samples are required to ensure that the "job execution sequence" in the top 5 % is included in the sample with a probability of 95 %. Table 4 partially shows the "job execution sequence" when the total elapsed time is arranged in ascending order. Table 5 shows the main results of Experiment 2.

A calculation of the optimum execution sequence of job

Table 2: Frequency of execution sequence of each job

Order	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	Total
1	104	98	102	104	93	108	91	100	102	98	1000
2	101	106	96	94	91	91	108	121	99	93	1000
3	102	97	95	91	103	107	107	103	98	97	1000
4	97	104	118	101	91	102	112	93	84	98	1000
5	102	102	94	107	82	88	109	106	102	108	1000
6	96	107	95	107	105	99	91	86	98	116	1000
7	97	90	109	100	107	106	98	112	96	85	1000
8	107	99	88	103	112	98	91	103	96	103	1000
9	112	104	92	95	113	91	92	103	98	100	1000
10	82	93	111	98	103	110	101	73	127	102	1000
Total	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	---

Table 3: Operating hours of each machine (in two digits)

Job number	Machine A	Machine B
J_1	63	59
J_2	68	48
J_3	11	41
J_4	45	80
J_5	75	65
J_6	13	62
J_7	50	43
J_8	26	81
J_9	23	89
J_{10}	48	69

Table 4: Sample data containing top 5 % "job execution sequence" with 95 % probability

Rank	Machine A	Time
1	$J_3, J_6, J_2, J_7, J_4, J_5, J_8, J_{10}, J_9, J_1$	648
2	$J_3, J_6, J_5, J_1, J_4, J_8, J_{10}, J_2, J_7, J_9$	648
3	$J_6, J_4, J_{10}, J_5, J_9, J_3, J_2, J_1, J_8, J_7$	650
4	$J_6, J_9, J_{10}, J_2, J_3, J_4, J_1, J_5, J_8, J_7$	650
5	$J_6, J_9, J_3, J_4, J_2, J_1, J_5, J_{10}, J_7, J_8$	650
6	$J_6, J_1, J_3, J_8, J_7, J_2, J_9, J_4, J_{10}, J_5$	651
⋮	⋮	⋮
59	$J_2, J_7, J_1, J_3, J_{10}, J_6, J_4, J_5, J_9, J_8$	727

by Johnson's rule is $J_3, J_6, J_9, J_8, J_4, J_{10}, J_5, J_1, J_2, J_7$, with the minimum total elapsed time of 648. Two cases in Table 4 have this minimum time, showing two optimum solutions exist in the experiment. The possible "job execution sequence" is 10! = 3,628,800 cases since the number of the job is 10. Johnson's rule and Table 4 shows 2 out of 59 cases are optimum solutions. Hence, proportional calculation of these findings yields 123,010 cases of optimum solutions out of possible cases of 3,628,800.

$3{,}628{,}800 : x = 59 : 2$. So $x = 3{,}628{,}800 \times 2 \div 59 = 123{,}010$.

Table 5 shows the optimum solutions were included in the samples of the top 5 %, 20 %, and 30 % cases. This dispersion can be interpreted that the optimum solutions are included

Table 5: The result of sample data with 2-digit machine operating time and same date in 2-machines 10-jobs problem (probability = 95 %)

Top	5 %	10 %	15 %	20 %	25 %	30 %
Number of samples	59	29	19	14	11	9
Time range			11-89			
Optimum TET			648			
Number of Optimum TET	2	0	0	2	0	1
TET (minimum)	648	650	650	648	650	648
TET (average)	684	682	679	672	681	676
TET (maximum)	727	720	719	716	712	712

Notes: TET = Total Elapsed Time. Underlined number signifies the optimum total elapsed time.

in every combination with sufficiently high probability. Of course, since the number of samples are shown to decrease from top 5 % to 30 %, the probability for the optimum solutions to be included becomes smaller accordingly.

3.3 Experiment 3

To show that the results in Table 5 are not due to the specificity of the data in Table 3, the experiment was conducted in which the operating time of each machine on each job was set to two digits (uniform random numbers) and the data was changed each time.

3.3.1 Results

Table 6 shows the experimental results. In all cases, the sample data contains an optimum solution or a quasi-optimum solution (a value close to the optimum solution). This result is compatible with the result of Experiment 2 shown in Table 5. Therefore, the specificity of the data does not determine the results of the Experiments 2 and 3.

3.4 Experiment 4

In Experiments 2 and 3, the operating time of each machine on each job was set to 2 digits. In order to investigate the effect of this setting on the results, another experiment was conducted with the operating time being changed from 2 digits to 3 digits or less (uniform random number).

Table 7: Operating hours of each machine (in three digit)

Job number	Machine A	Machine B
J_1	73	427
J_2	281	771
J_3	705	375
J_4	328	109
J_5	125	150
J_6	381	184
J_7	165	258
J_8	734	20
J_9	869	557
J_{10}	76	69

3.4.1 Results

Table 7 shows the operating time of each machine actually used in the top 5 %. As shown in the table, most are 3 digits with few 2 digits. This is logical since the two-digit time width (10 to 99) is 1/10 of the three-digit time width (100 to 999).

Table 8 shows the experimental results. In Table 6, the theo-

Table 6: The result of sample data with 2-digit machine operating time and different date in 2-machines 10-jobs problem (probability = 95 %)

Top	5 %	10 %	15 %	20 %	25 %	30 %
Number of samples	59	29	19	14	11	9
Time range	15-95	11-96	15-96	12-82	12-99	11-97
Optimum TET	730	510	569	553	601	658
Number of Optimum TET	8	1	1	0	0	1
TET (minimum)	<u>730</u>	<u>510</u>	<u>569</u>	554	630	<u>658</u>
TET (average)	769	571	655	611	662	684
TET (maximum)	854	667	778	674	705	716

Notes: TET = Total Elapsed Time. Underlined number signifies the optimum total elapsed time.

Table 8: The result of sample data with 2- or 3-digit machine operating time and different date in 2-machines 10-jobs problem (probability = 95 %)

Top	5 %	10 %	15 %	20 %	25 %	30 %
Number of samples	59	29	19	14	11	9
Time range	20-869	134-965	52-994	18-951	17-998	14-990
Optimum TET	3757	6203	7039	5400	7091	5737
Number of Optimum TET	3	2	0	1	0	1
TET (minimum)	<u>3757</u>	<u>6203</u>	7084	<u>5400</u>	7215	<u>5737</u>
TET (average)	4314	6778	7829	6058	7678	6250
TET (maximum)	5163	7553	8545	6744	7995	6862

Notes: TET = Total Elapsed Time. Underlined number signifies the optimum total elapsed time.

retical maximum width of the operating time of each machine on each job is 10 to 99, whereas it is 0 to 999 in Table 8. In other words, the width becomes 11 times.

The average value of the optimum total elapsed time in Tables 6 and 8 are 603.5 and 5871.1, respectively. The difference is 9.7 times.

Although the theoretical and actual differences do not match exactly, considering the difference in data and its variation, they can be considered essentially identical.

3.5 Experiment 5

Another experiment was conducted to investigate the variation of the minimum total elapsed time and the optimum value among Experiments 2, 3 and 4. Specifically, the deviations between the minimum total elapsed time and the optimum value of Experiments 2, 3 and 4 were obtained.

3.5.1 Results

Table 9 shows the experimental results. Experiment 2 had the smallest deviation value. The deviation values or the top 5 %, 10 %, · · ·, 30 % were almost the same, which is considered to be because the same data was used each set. In the top 25 % of Experiment 3, the deviation value was 29, but the relative error was 4.6 %. In Experiment 4, the top 15 % had a deviation value of 45 and a relative error of 0.6 %. In the top 25 %, the deviation value is 124 and the relative error is as small as 1.7 %. These results show that in most cases, the optimum values or quasi-optimum values were obtained in all the experiments.

Table 9: Difference between the minimum total elapsed time of Experiments 2, 3 and 4 and the optimum value (optimum value, number of appearance)

Top	Experiments		
	2	3	4
5 %	0 (648,2)	0 (730,8)	0 (3757,3)
10 %	2 (648,0)	0 (510,1)	0 (6203,2)
15 %	2 (648,0)	0 (569,1)	45 (7039,0)
20 %	0 (648,2)	1 (553,0)	0 (5400,1)
25 %	2 (648,0)	29 (630,0)	124 (7091,0)
30 %	0 (648,1)	0 (658,1)	0 (5737,1)

Comparison of the frequencies of the optimum values among three experiments show that the result of Experiment 3 is better than those of the other experiments. This shows that the smaller the width (maximum value-minimum value) of the operating time of each machine on each job, the higher the frequency of the optimum value, and the easier it is to find the optimum solution.

3.6 Experiment 6

An experiment was conducted to compare the 3-machines *n*-jobs problem, where no algorithm for obtaining the optimum total elapsed time exists, and the 2-machines *n*-jobs problem, where the algorithm (Johnson's rule) exists. Because of the lack of the algorithm, in 3-machines *n*-jobs problem, 100 % inspection is necessary to find the optimum total elapsed time (that is, all "job execution sequence" must be tried). For simplifying calculations, *n* was set to 5. Still, 120 patters need to be inspected.

Samplings containing with a 95 % probability the "job execution sequence" of the top 5 %, 10 %, 15 %, · · ·, 30 %, arranged in ascending order based on the total elapsed time were used. The operating time of each machine on each job was set to two digits (uniform random numbers) and the data was changed and updated in each experiment session.

3.6.1 Results

Tables 10 and 11 respectively show the results of sample data of 2-machines 5-jobs problems and those of the 3-machines 5-jobs problem. A comparison of the two tables shows that the optimum total elapsed time does not change, but the maximum value of the total elapsed time of the sample data is larger for the 3-machines 5-jobs problem, and the average value is slightly larger for 3-machine 5-jobs problem.

The examination of the total elapsed times of the 2-machines 5-jobs problem and the 3-machines 5-jobs shows that the former (1) was more likely to overlap, and (2) has less

Table 10: Results of sample data for 2-machines 5-jobs problem

Top	5 %	10 %	15 %	20 %	25 %	30 %
Number of samples	59	29	19	14	11	9
Time range	12-90	18-99	12-93	25-94	25-99	21-97
Optimum TET	238	379	357	319	398	351
Number of Optimum TET	3	0	3	0	0	1
TET (minimum)	<u>238</u>	384	<u>357</u>	325	417	<u>351</u>
TET (average)	276	405	371	357	453	378
TET (maximum)	314	438	390	396	503	407

Notes: TET = Total Elapsed Time. Underlined number signifies the optimum total elapsed time.

Table 11: Results of sample data for 3-machines 5-jobs problem

Top	5 %	10 %	15 %	20 %	25 %	30 %
Number of samples	59	29	19	14	11	9
Time range	16-93	15-89	11-96	17-95	14-97	13-93
Optimum TET	373	335	370	395	420	396
Number of Optimum TET	3	1	0	1	1	1
TET (minimum)	<u>373</u>	<u>335</u>	403	<u>395</u>	<u>420</u>	<u>396</u>
TET (average)	446	390	456	441	485	477
TET (maximum)	512	457	540	490	579	559

Notes: TET = Total Elapsed Time. Underlined number signifies the optimum total elapsed time.

number of patters of total elapsed time than the latter.

Table 12 shows the deviation between the minimum total elapsed time and the optimum value of the 2-machines 5-jobs problem and the 3-machine 5-jobs problem. The frequency of the optimum values does not change in both problems. The deviation between the minimum total elapsed time and the optimum value can be considered to be identical since the sums of deviation for each problem are almost the same.

4. Variations from the simplest model

In this chapter, we will investigate the relationship between the operating time of each machine on each job and optimum solutions. Specifically, a simple setting with same operating time for each machine is used as a baseline, and then cases where the operating times are changed stepwise are considered to show these changes lead to a proportional rise in difficulty in finding an optimum solution.

Table 12: Difference between the minimum total elapsed time and the optimum value in 2-machines 5-jobs and 3-machines 5-jobs problem (optimum value, number of appearance)

Top	2-machines 5-jobs	3-machines 5-jobs
5 %	0 (238,3)	0 (373,3)
10 %	5 (379,0)	0 (335,1)
15 %	0 (357,3)	33 (370,0)
20 %	6 (319,0)	0 (395,1)
25 %	19 (398,0)	0 (420,1)
30 %	0 (351,1)	0 (396,1)

Baseline: In the *m*-machines *n*-jobs problem, it is assumed that the operating time of each machine of each job is the same, which is represented by t^*. Figure 1 shows the Gantt chart for calculating the total elapsed time in this setting. The total elapsed time, $(m - 1 + n)\, t^*$, is the optimum value under which the optimum (minimum) "job execution sequence" can be decided discretionary.

When only the operating time of the machine M_2 for the job J_3 is changed to $t^* + a$ $(a > 0)$ and the others remain to be t^*, as shown in Table 13, the optimum total elapsed time of the *m*-machines *n*-jobs problem is $(m - 1 + n)\, t^* + a$, according to the Gantt chart in Figure 2.

However, the increment value to the reference time t^* is not added to the total elapsed time each time it occurs. For example, as shown in Figure 3, the operating time of the machine M_2 on job J_i is $t^* + a$, the operating time of the machine M_1 on job $J_i + 1$ is $t^* + b$ $(b > a)$, and the operating time of other jobs by each machine is t^*, the total elapsed time is $(m - 1 + n)\, t^* + b$ with no effect of a.

Figure 4 shows a Gantt chart when $t^* + a$, the operating time of the machine M_2 of the job J_3 in Table 12, is set to $t^* - a$

Table 13: 3-machines 5-jobs hours

Job number	M_1	M_2	M_3
J_1	t^*	t^*	t^*
J_2	t^*	t^*	t^*
J_3	t^*	$t^* + a$	t^*
J_4	t^*	t^*	t^*
J_5	t^*	t^*	t^*

Execution sequence: J_1, J_2, J_3, J_4, J_5

M_1	J_1	J_2	J_3	J_4	J_5			
M_2		J_1	J_2	J_3	J_4	J_5		
M_3			J_1	J_2	J_3	J_4	J_5	

0 t^* $2t^*$ $3t^*$ $4t^*$ $5t^*$ $6t^*$ $7t^*$

Figure 1: Gantt chart with identical operating time for every machine

Execution sequence: J_1, J_2, J_3, J_4, J_5

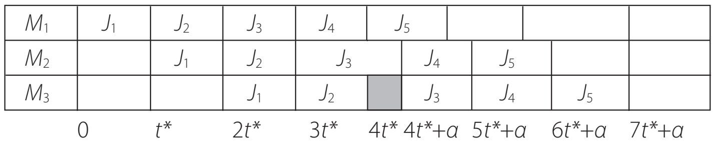

Figure 2: Gantt chart of Table 13

Note: Gray cell signifies hours without machine operation.

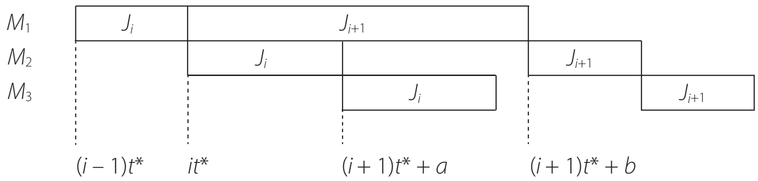

Figure 3: Partial Gantt chart

Execution sequence: J_1, J_2, J_3, J_4, J_5

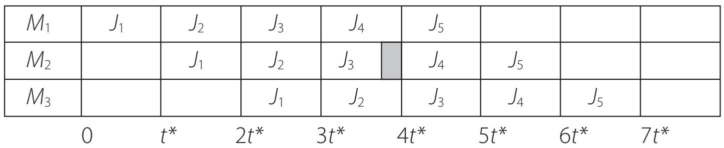

Figure 4: Gantt chart with $t^* + a$ changed to $t^* - a$ in Table 13

Note: Gray cell signifies hours without machine operation.

($t^* - a \geqq 0$). In this setting, the total elapsed time is $(m - 1 + n)t^*$, which is the same as the result of the Gantt chart in Figure 1. However, if this $t^* - a$ is transferred to the operating time of the machine M_1 on job J_1, the total elapsed time becomes $(m - 1 + n)t^* - a$, which is smaller by a than the result of the Gantt chart in Figure 1.

From above, there are many cases where the operating time of each machine on each job is close to each other, and when the number of digits is the same, the total elapsed time is quasi-optimum or the optimum solutions in all "job execution sequence".

If, on the other hand, the variation in the operating time of each machine on each job becomes large, the number of cases which include quasi-optimum or the optimum solutions will decrease. In cases where the operating times are almost identical, such as t^* in Figures 1 to 4, the optimum solution can be obtained in polynomial time. However, as the variance in the operating times expands, it becomes more complicated and difficult to find the optimum solution.

5. Conclusion

The results of this paper are summarized below.

From Table 5, which shows the results of Experiment 2 with 2-digit operating time and same data processing in 2-machines 10-jobs problem, it has been shown that the optimum solutions were included in the samples of the top 5 %, 20 %, and 30 % cases. This dispersion can be interpreted that the optimum solutions are included in every combination with sufficiently high probability. Of course, since the number of samples decreases from top 5 % to 30 %, the probability becomes smaller accordingly.

Experiment 3 has shown that the result summarized in Table 5 is not due to the specificity of the data in Table 3 (operating time of each machine). The results of Experiment 4 have shown that the width of the data (maximum-minimum) is almost proportional to the optimum total elapsed time.

The results of Experiment 5 have shown that in most cases, the optimum or quasi-optimum values were obtained. Comparisons of the frequencies of the optimum values among different settings has shown that a smaller width of the operating time (maximum-minimum) of each machine on each job leads to more frequent appearance of the optimum solution, which signifies this setting is easier in finding the optimum solution.

In Experiment 6, the 2-machines n-jobs problem and the 3-machines n-jobs problem were compared. The results have shown (1) the optimum total elapsed times for the both problems were similar; (2) the maximum value of the total elapsed time for the sample data is clearly larger in the latter problem; (3) the frequency of the optimum value and the deviation between the minimum total elapsed time and the optimum value were similar in both cases; (4) when the total elapsed times were examined, the former was more likely to overlap than the latter, and the types of total elapsed time were less than the latter. The factor for the last findings can be there are many cases similar to those illustrated in Figures 2 and 3 in 3-machines n-jobs problem.

In the cases when the operating time of each machine on each job is close to each other or the number of digits is the same, it is considered that there are many cases where every "job execution sequence" contains many optimum or quasi-optimum solutions. On the contrary, if the variation in the operating time of each machine on each job becomes large, the number of cases where the optimum or quasi-optimum solutions contained in the "job execution sequence" will decrease.

These results show that the optimum solution or quasi-optimum solution of the flow shop scheduling problem can be obtained in a time within the allowable range by the method proposed in this paper. This method can be highly effective in meeting the practical needs of lines with many time constraints. However, since the method uses probabilities, the expected time saving effect cannot always be obtained. We will tackle this issue in a future research.

References

Baker, K. R. (1975). A comparative study of flow shop algorithms. *Operations Research*, Vol. 23, No.1, 62-73.

Carlier, J. and Rebai, I. (1996). Two branch and bound algorithms for the permutation flow shop problem. *European Journal of Operational Research*, Vol. 90, Issue 2, 238-251.

Gantt chart (2021). Wikipedia. Accessed on January 10, 2021. https://en.wikipedia.org/wiki/Gantt_chart.

Hino, R. (2017). Mathematical optimization models for job-shop scheduling problem. *Systems, Control and Information*, Vol. 6, No. 1, 14-19. (in Japanese)

Ibaraki, T. (1994). A new approach to scheduling problems: (1) Scheduling problems and algorithm complexities. *Journal of the Operations Research Society of Japan*, Vol. 39, No. 10, 541-546. (in Japanese)

Ibaraki, T., Imamichi, T., Koga, Y., Nagamochi, H., Nonobe, K., and Yagiura, M. (2011). Efficient branch-and-bound algorithms for weighted MAX-2-SAT. *Mathematical Programming*, Vol. 127, pp.297-343.

Imaizumi, J. (2000). Flow shop scheduling problem and its variants: models and their classification. *Studies of Business Management*, Vol. 23, 259-274. (in Japanese)

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, Vol. 1, Issue 1, 61-68.

Johnson's rule (2021). Wikipedia. Accessed on January 3, 2021. https://en.wikipedia.org/wiki/Johnson%27s_rule.

Mohammed, G. T. (2016). New algorithm for *n*-jobs on *m*-machine flow shop scheduling problems. *Applied and Computational Mathematics*, Vol. 5, No. 1, 1-9.

Moriya, E. (1973). *Operations research*. Ohmsha. (in Japanese)

Sakuraba, C. S. (2010). Efficient local search algorithms for the linear sequencing problem. *International Transactions in Operational Research*, Vol. 17, 711-737.

(Received: July 27, 2021; Accepted: August 21, 2021)