Job-shop Scheduling Incorporating Dynamic and Flexible Facility Layout Planning

Ryota Kamoshida
Intelligent Information Research Department, Hitachi, Ltd., Tokyo, Japan
Email: ryota.kamoshida.vw@hitachi.com

***Abstract*— A method for solving the job-shop scheduling problem (JSSP) integrated with dynamic and flexible facility layout planning (FLP) is proposed. The FLP is formulated as a strip packing problem under certain conditions and incorporated into the Giffler and Thompson (GT) algorithm, which is widely used to solve the JSSP. The makespan of the schedule is minimized by a tabu search algorithm. The performance of the proposed method was evaluated on benchmark datasets for FLP-integrated JSSP (FLPIJSSP), which were made anew for the FLPIJSSP. The evaluation results show that the proposed method can solve the FLPIJSSP efficiently in a reasonable time while outperforming conventional methods.**

***Index Terms*—job-shop scheduling, facility layout planning, strip packing, metaheuristic, tabu search**

I. INTRODUCTION

The job-shop scheduling problem (JSSP) is an important problem that affects the efficiency and productivity of manufacturing systems. In general JSSP, machines used in the operations of each job are allocated to predetermined locations during certain scheduling periods. However, in actual factories, certain operations can be executed by portable machines, which are arbitrarily allocated to vacant locations in the factory. The area of vacant locations for portable machines is often not large enough to hold all portable machines at the same time. The portable machines are thus allocated to the area while they are used in operations and stored in depots after the operations are completed. This requires the Facility Layout Planning (FLP), which determines when and where to use portable machines during a scheduling period.

Combining the JSSP and the FLP has been studied extensively [1]-[7]. In these previous studies, however, machine locations are fixed during a scheduling period.

In this study, a problem called "facility-layout-planning-integrated job-shop scheduling problem (FLPIJSSP)," is focused on. In the FLPIJSSP, facility locations change during a scheduling period (i.e., some operations are executed by portable machines in vacant locations only during processing). The objective of the FLPIJSSP is to minimize the makespan (i.e., maximum time to complete all jobs) of the schedule. To solve the FLPIJSSP, this paper proposes a new method for solving the JSSP and the FLP concurrently. In this paper, the FLP is formulated as a strip packing problem under certain conditions and incorporated into the Giffler and Thompson (GT) algorithm [8], which is widely used for the JSSP.

This paper is an extended version of a paper presented at the 5th International Conference on Industrial Engineering and Applications (ICIEA 2018) [9]. In the previous paper, the number of vacant locations was limited to one and transportation costs were ignored. In this paper, the case of multiple vacant locations is evaluated, and transportation costs are considered. Moreover, another conventional method for comparing the effectiveness of the proposed method is added, and the computational cost of the optimization process is discussed. In summary, this paper demonstrates the applicability of the proposed method to more realistic cases.

This paper aims to answer the following research questions.

RQ1 Can makespan be reduced by the proposed method?

RQ2 What affects the effectiveness of the proposed method?

RQ3 Is the computational cost of the proposed method reasonable in realistic cases?

The contributions of this paper are as follows:

- An approach to solve the FLPIJSSP, in which facility locations change during a scheduling period, is presented;
- Positive evaluation results demonstrating the effectiveness of the proposed approach are presented;
- The low computational cost of the proposed method is validated.

The remainder of this paper is organized as follows. Section II presents related work and a motivating example. Section III introduces the proposed method for solving the FLPIJSSP. Section IV describes the experimental setup, presents the results of the experiment, and discusses the implications of the results. Finally, Section V presents the conclusions of this study and addressed future work.

II. BACKGROUND

A. Job-shop Scheduling Problem

The job-shop scheduling problem (JSSP) is a common scheduling problem, in which multiple jobs $J_j(1 \leq j \leq$

Manuscript received December 1, 2018; revised June 10, 2019.

doi: 10.18178/jiii.7.1.12-17

n) are processed on multiple machines $M_r (1 \leq r \leq m)$. Each job consists of a technological sequence of operations, and each operation must be processed on a specific machine. An operation of job J_j processed on machine M_r is described as O_{jr}. The operation O_{jr} must be exclusively executed by machine M_r for its processing time p_{jr}. A schedule is defined as a set of completion times for each operation $\{c_{jr}\}$. One of the most common objectives of the JSSP is to minimize makespan (i.e., maximum time to complete all jobs). To solve the JSSP, metaheuristic search algorithms using local search methods, such as tabu search [10], are often used [11], and the critical block (CB, i.e., a sequence of adjacent critical operations on the same machine) neighborhood has been defined [12], [13].

B. Facility Layout Planning

The facility layout planning (FLP), which is known to be an NP-hard problem, involves finding the most-efficient facility locations on the factory floor to optimize objective functions such as transportation cost.

The FLP has been solved as an assignment problem such as the Quadratic Assignment Problem (QAP) [14]. In general QAP, N facilities are assigned to N locations to minimize the total assignment cost, given as

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} f_{ij} d_{ij} \tag{1}$$

where f_{ij} is the required flow between facilities i and j, and d_{ij} is the distance between locations i and j.

Although the QAP in the FLP can be solved efficiently by a metaheuristic approach such as a genetic algorithm, even in a large number of facilities, layouts that cannot be represented exist, since location candidates for facilities are predetermined and fixed. Furthermore, the objective of the FLP in this paper is to minimize the makespan.

In this study, several operations are executed by portable machines that are arbitrarily allocated to vacant locations in the factory while only being used in those operations. Therefore, a flexible and dynamic FLP method is required.

In this paper, the FLP is formulated as a strip packing problem to handle the dynamic and flexible FLP. The proposed method is described in detail in Section III.

C. Motivating Example

A motivating example that demonstrates the importance of concurrent optimization of the JSSP and the dynamic and flexible FLP is presented. A job-shop problem that has three jobs and six machines (3 × 6 job-shop problem) is described in Table I. In the table, the first operation of job 1 uses machine 1, and its processing time is one. A layout in which machines 1, 3, and 5 are used at predetermined locations is shown in Fig. 1 (a). The numbers in the layout correspond to the respective machine numbers. Operations that are executed by portable machines are shown in Fig. 1 (b). The two numbers in the squares are job number and machine number, e.g., "1-2" means the operation of job 1 uses machine 2. The areas occupied by the operations are defined by machines and jobs, and they are predetermined in this study. The operations need to be located at the appropriate position in the vacant location. If the sum of the areas occupied by the operations that are executed by portable machines is larger than the area of vacant locations, those operations cannot all be located in the vacant locations at the same time. Therefore, it is necessary to determine the order in which each operation uses a vacant location in the layout.

TABLE I. 3 × 3 JOB-SHOP PROBLEM

Job	Operation routing (processing time)		
1	1 (1)	2 (3)	5 (1)
2	1 (1)	3 (2)	4 (2)
3	5 (1)	6 (3)	3 (2)

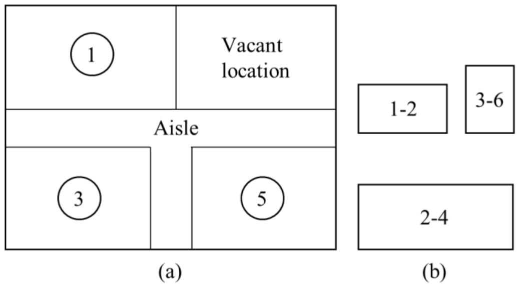


Figure 1. Layout with vacant location.

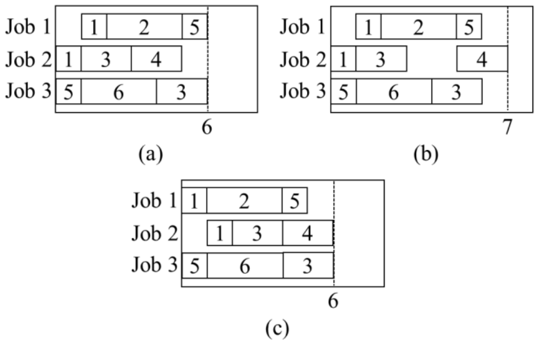


Figure 2. Examples of Gantt chart.

Fig. 2 shows how the FLP affects the result of the JSSP. The result of scheduling without considering the FLP, which is represented as a Gantt chart, is shown in Fig. 2 (a). The horizontal axis represents time. In this case, the makespan is six. However, when the FLP problem is considered, operations 2, 4, and 6 cannot be processed at the same time. Hence, the schedule is modified as shown in Fig. 2 (b). The makespan increases from six to seven. An optimal solution, where the makespan is six, is shown in Fig. 2(c). As shown above, sequential optimization of the JSSP and the FLP may not be able to minimize the makespan. Thus, this paper focuses on concurrent optimization of the JSSP and the FLP.

D. Concurrent Optimization

Several studies on solving the JSSP and the FLP simultaneously have been reported. Ripon, Glette, Hovin, and Torresen [1], [2] presented a genetic algorithm for solving the integrated the JSSP and the FLP considering multiple objectives and Pareto-optimality. Dastpak, Poormoaied, and Naderpoor [3] assumed that each job has several fuzzy parameters, such as processing time, and proposed a genetic algorithm and simulated annealing approaches for solving a fuzzy scheduling and location problem in a job-shop environment. Ranjbar and Razavi [4] developed a hybrid metaheuristic approach based on the scatter search algorithm to concurrently make the layout and scheduling decisions in a job-shop environment. These previous studies formulated the FLP as a QAP in which the number of facilities was equal to the number of location candidates for facilities. However, as described in the motivating example, the FLPIJSSP requires more flexible layout planning.

Arkat, Farahani, Hosseini, and Ahmadizar [5], [6] and Ebrahimi, Kia, and Komijan [7] solved the layout problem more flexibly than a QAP concurrently with a scheduling problem in cellular manufacturing systems. In [5], [6], machines were taken as squares with a unit dimension, and the machines were configured in a multi-row grid-like layout creating more location candidates than machines. In [7], squared machines could be allocated to arbitrary positions in the squared cell.

In all these previous studies, machine locations were fixed during the scheduling period. However, in the case of the FLPIJSSP, locations of portable machines change during a scheduling period, so dynamic layout planning is required.

III. Methodology

A. Assumptions

The proposed method can be described on the basis of three assumptions:

1. Areas of vacant locations and operations that are executed by portable machines are rectangles with predetermined dimensions,
2. The operation requiring the maximum dimensions can be located in the vacant location with the minimum dimensions,
3. Areas of operations that are executed by portable machines are adjacent to an aisle.

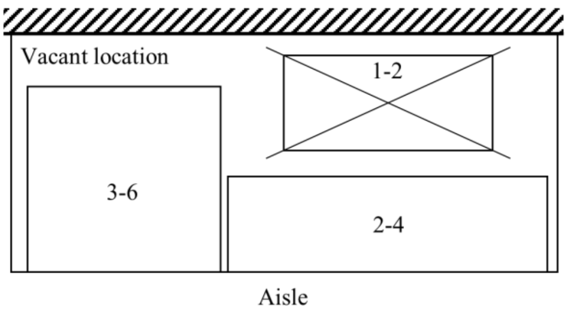


Figure 3. Schematic view to describe assumption 3.

Assumption 3, schematically illustrated in Fig. 3, is reasonable because if an operation that is executed by a portable machine is located at the secluded area of a vacant location (like operation 1-2 in Fig. 3), it may be difficult to change locations after the operation finishes due to interference by other operations.

B. Facility Layout Planning as Strip Packing Problem

To solve the FLPIJSSP, it is necessary to determine when and where operations that are executed by portable machines are processed. In this paper, the FLP is formulated as a strip packing problem, in which a rectangular strip with width W and infinite height is given. As described in assumption 3, since operations that are executed by portable machines are adjacent to an aisle, the way of use of only the width of the vacant location, not its depth, is considered. If the infinite height of the strip is regarded as the time axis and an operation is regarded as a rectangle with width w and height t that is equal to the process time of the operation, the FLP can be formulated as a strip packing problem.

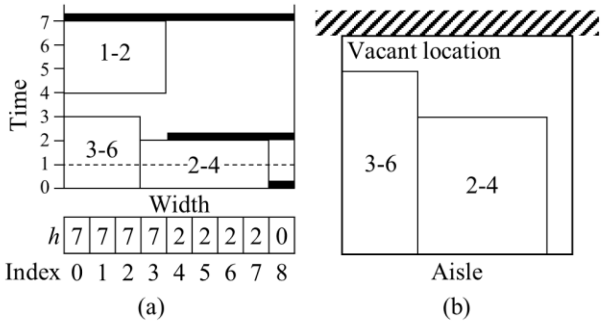


Figure 4. Example of best fit algorithm.

The proposed approach for solving the strip packing problem is based on the Best Fit algorithm (BF) [15]. The BF uses a linear array $\boldsymbol{h}$ that has a number of elements equal to the strip width W, where W is an integer. Each element of the array holds the total height of the packing corresponding to the coordinate of the width of the vacant location. An example of strip packing using BF is given in Fig. 4 (a). Operation 3-6 uses the width of the vacant location from coordinate 0 to 2 and from time 0 to 3. The vertical axis represents time, not height. Therefore, there might be a gap between the operations in the vertical axis (i.e., the gap between operation 3-6 and operation 1-2 in Fig. 4 (a)). Using a linear array makes it easier to find the coordinate of the available spaces in the horizontal axis (bold line in Fig. 4 (a)). The actual layout of a vacant location at time 1 (dashed line in Fig. 4 (a)) is represented in Fig. 4 (b). The overall structure of the FLP algorithm is described as follows.

Algorithm 1

Step 1: Make a linear array $\boldsymbol{h}$ with length equal to the width of the vacant location and initialize $\boldsymbol{h}$ to zero.

Step 2: Obtain an operation O_{jr}.

Step 3: If d_{jr} (depth of O_{jr}) < w_{jr} (width of O_{jr}) and w_{jr} < d_{vl} (depth of vacant location), rotate the dimensions of O_{jr} so that $w_{jr} < d_{jr}$.

Step 4: Obtain the earliest starting time of O_{jr}, $ES(O_{jr})$.

Step 5: Find the lowest available space such that:

- time of the space $\geq ES(O_{jr})$,
- width of the space $\geq w_{jr}$.

Step 6: Place O_{jr} in the leftmost part of the space.
Step 7: Raise $\boldsymbol{h}$ to appropriately reflect the skyline.
Step 8: Set the height of the space to $ES(O_{jr})$.
Step 9: Set the height of O_{jr} to the earliest completion time of O_{jr}, $EC(O_{jr})$.

C. Modified Job-shop Scheduling

To solve the JSSP and the FLP simultaneously, a modified Giffler and Thompson (GT) algorithm is applied to make an active (i.e., feasible) schedule. The GT algorithm is given as follows.

Algorithm 2
Step 1: Let D be a set of all the earliest operations in a technological sequence.
Step 2: Let O_{jr} be an operation with minimum earliest completion time in D. When there are multiple candidates for O_{jr}, choose one arbitrarily.
Step 3: Let C be a set of all operations that use machine M_r.
Step 4: Remove operations O such that $ES(O) \geq EC(O_{jr})$ from C.
Step 5: Select an operation $O^* \in C$.
Step 6: Set $D = (D - \{O^*\}) \cup \mathrm{SUC}(O^*)$; where $\mathrm{SUC}(O)$ is the next operation to O.
Step 7: If $D \neq \emptyset$, go to Step 2; otherwise, stop.

D. Metaheuristic Optimization

Tabu search [10] is applied to optimize the FLPIJSSP. Tabu search is a widely used metaheuristic optimization algorithm and a local search algorithm. Local search algorithms including tabu search need to define and calculate neighbor solutions. The active critical block (CB) neighborhood proposed by Yamada and Nakano [16] is adopted here. An example of generation of active CB neighborhoods is shown in Fig. 5. Let S be an active schedule and $B_{k,h,M}$ be a critical block of S on a machine M, where the front and the rear operations of $B_{k,h,M}$ are the k-th and h-th operations on M, respectively. Let $O_{p,M}$ be an operation in $B_{k,h,M}$ that is the p-th operation on M. An active CB neighbor schedule $S_{M,p,k}$ (or $S_{M,p,h}$) is generated by modifying S such that $O_{p,M}$ is moved to the position as close as possible to front position k (or rear position h) of $B_{k,h,M}$. The new active CB neighborhood $AN^C(S)$ is now defined as a set of all $S_{M,p,k}$'s and $S_{M,p,h}$'s over all critical blocks as

$$AN^C(S) = \bigcup_{B_{k,h,m}} \left\{S' \in \left\{S_{M,p,k}\right\}_{k<p<h} \cup \left\{S_{M,p,h}\right\}_{k<p<h}, S' \neq S\right\}. \quad (2)$$

The overall structure of the algorithm is described below.

Algorithm 3
Step 1: Let S be an active schedule.
Step 2: Set $S_b = S$.
Step 3: Obtain a set of active CB neighborhoods $AN^C(S)$.
Step 4: Choose the best solution S' such that S' does not violate the tabu conditions.
Step 5: If S' is better than S_b, $S_b = S'$.
Step 6: Update the tabu list.
Step 7: Set $S = S'$.
Step 8: If a stopping condition is met, stop; otherwise, go to Step 3.

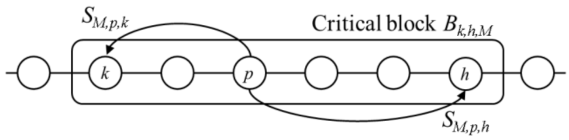


Figure 5. $S_{M,p,k}$ and $S_{M,p,h}$ generation.

IV. EVALUATION

A. Benchmark Datasets

OR-Library [17] is one of the most commonly used benchmark datasets for the JSSP and the FLP. However, no benchmark datasets for the FLPIJSSP have been published. Benchmark datasets were thus made for the FLPIJSSP by modifying the datasets for the JSSP in OR-Library. Datasets la01, la06, and la16 were chosen, which were 10×5, 15×5, and 10×10 JSSP, respectively. Varying parameters for making datasets are listed in Table II. r_p was a ratio of portable machines, and a set of portable machines $\{M_p\}$ was selected at random from a dataset so that the number of portable machines $|M_p|$ was $r_p \times |M|$, where $|M|$ was the number of machines in the dataset. The r_p was selected from the set {0.25, 0.5, 0.75, 1.0}. The width w and height (depth) h of the set of operations that is executed by a portable machine $\{O_{jp}\}$ were set at random, where w and h are integer values in the range of one to ten. The number of vacant locations N_{v} was selected from the set {1, 2, 3}. The width of a vacant location that was selected from the set {10, 20, 30} was given as W. The depth of a vacant location was set to ten, which guaranteed that the largest portable machine was able to be located in the vacant location. A transportation cost from operation O_{jr} to operation $O_{jr'}$ was taken as $C_{jrr'}$ which for each operation was set as

$$C_{jrr'} = C_{ave} + \begin{cases} 0 & if\ C_{ave} = 0 \\ randi(-10,10) & else \end{cases} \quad (3)$$

where C_{ave} is an average transportation cost selected from the set {0, 10, 30, 50}, and $randi(a, b)$ represents a uniformly selected random integer with minimum value a and maximum value b.

Furthermore, a scaling parameter scl was selected from the set {1, 5, 10}. The computational cost of the optimization process depends on the spatial resolution of the area of vacant locations and operations. Thus, to evaluate the effect of the spatial resolution on the computational cost, w, h, and W were scaled by multiplying them by scl. Finally, 1296 datasets were made.

TABLE II. VARYING PARAMETERS

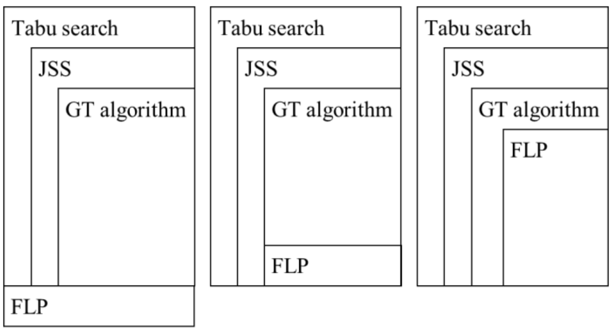
Parameter	Setting
Datasets in OR-Library	la01, la06, la16
Ratio of portable machine r_p	0.25, 0.5, 0.75, 1.0
Number of vacant locations N_v	1, 2, 3
Width of vacant locations W	10, 20, 30
Average transportation cost C_{ave}	0, 10, 30, 50
Scaling parameter scl	1, 5, 10

B. *Settings*

The number of tabu moves was set to seven. Iteration of the tabu search was stopped when 1000 iterations were performed without improving the makespan. These parameters were selected by performing informal searches.

C. *Results and Discussion*

In order to evaluate the effectiveness of the proposed method, the proposed method was compared with two conventional methods, namely, sequential optimization and quasi-concurrent optimization. In sequential optimization, FLPs were solved only once sequentially after the JSSP had been solved using the tabu search. On the other hand, in the quasi-concurrent optimization, FLPs were solved after generating an active schedule by the GT algorithm used in the JSSP. The difference between optimization methods is described schematically in Fig. 6.



(a) Sequential (b) Quasi-concurrent (c) Proposed

Figure 6. Difference between optimization methods.

TABLE III. EVALUATION RESULTS

Dataset	Makespan		
	Sequential	Quasi-concurrent	Proposed
la01	902	849	**769**
la06	1223	1164	**1035**
la16	1518	1441	**1281**

The evaluation results (i.e., makespan values) are listed in Table III. For each dataset, the values of makespan are averaged. The relationship between the narrowness of the vacant locations and the makespan determined by each method is shown in Fig. 7, where narrowness of the vacant locations is defined as

$$Narrowness = \frac{r_p}{N_v \cdot (W/10)} \tag{4}$$

Change in makespan when average transportation cost C_{ave} changes is represented in Fig. 8. Average execution times of the three optimization methods are compared in Fig. 9. As for the machine specification, the CPU was an Intel(R) Core i7-7700 (3.60GHz) combined with 16 GB RAM.

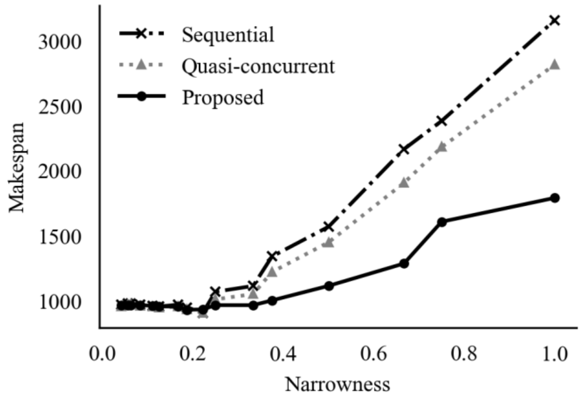


Figure 7. Relationship between narrowness of vacant location and makespan

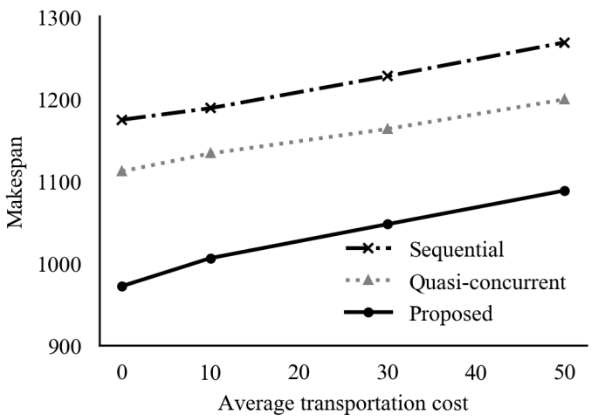


Figure 8. Relationship between average transportation cost and makespan.

The research questions are answered as follows:

RQ1: The results listed in Table III demonstrate that the proposed method outperforms the conventional methods in the case of all datasets.

RQ2: Fig. 7 shows that the narrower the area of vacant locations, the more efficiently the proposed method solves the FLPIJSSP. When the value of narrowness is small (i.e., number of portable machines is small, and area of the vacant locations is large), the facility layout does not need to be optimized, so all the makespan values are almost the same. On the other hand, Fig. 8 shows that transportation costs do not affect the effectiveness of the proposed method.

RQ3: Fig. 9 shows that the computational cost of the proposed method, in contrast to the conventional methods, increases exponentially when the scaling parameter increases. The validity of spatial resolution and computational cost are discussed hereafter. It is assumed that a maximum width of vacant locations is 10 meters. Then, the spatial resolution is equivalent to 0.1 meters when the scaling parameter is 10. Under this condition, the proposed method takes a few minutes to solve the FLPIJSSP. The author confirmed that spatial resolution of a 0.1 meters and execution time of a few minutes are reasonable for daily operations in actual factories by informally interviewing a manager of a factory.

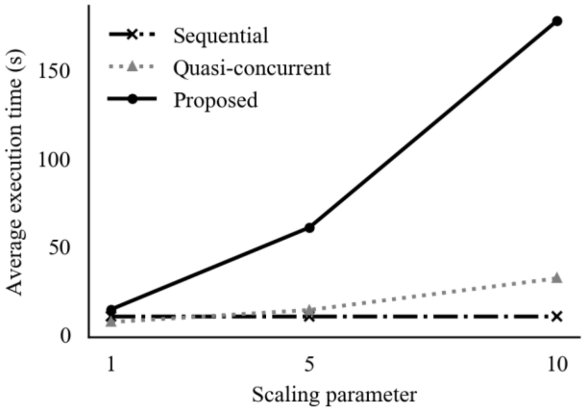


Figure 9. Comparison of average execution times.

V. Conclusion

A method for solving a facility-layout-planning-integrated job-shop scheduling problem (FLPIJSSP) was proposed. The FLP was formulated as a strip packing problem under certain conditions and incorporated into the Giffler and Thompson (GT) algorithm. The benchmark datasets were made anew for the FLPIJSSP to evaluate the effectiveness of the proposed method. Evaluation results show that the proposed method can reduce makespan compared with the other two conventional methods. The narrower the area of vacant locations, the more efficiently the proposed method solves the FLPIJSSP. Furthermore, it was confirmed that the proposed method can solve the FLPIJSSP in reasonable time.

It is hoped that the outcome of the present study would be of some worth to improve the efficiency and productivity of manufacturing systems. For future work, the author will collect real data and evaluate the effectiveness of the proposed method in actual factories.

References

[1] K. S. N. Ripon, K. Glette, M. Hovin, and J. Torresen, “Job shop scheduling with transportation delays and layout planning in manufacturing systems: A multi-objective evolutionary approach,” in *Proc. Autonomous and Intelligent Systems*, Aveiro, 2012, pp. 209-219.

[2] K. S. N. Ripon and J. Torresen, “Integrated job shop scheduling and layout planning: A hybrid evolutionary method for optimizing multiple objectives,” *Evolving Systems*, vol. 5, no. 2, pp. 121-132, Jun. 2014.

[3] M. Dastpak, S. Poormoaied, and M. Naderpoor, “Location and job shop scheduling problem with minimizing maximum completion time goal in fuzzy environment simultaneously through ranking fuzzy numbers and α-cut fuzzy,” in *Proc. Industrial Engineering and Operations Management*, Istanbul, 2012, pp. 10-19.

[4] M. Ranjbar and M. N. Razavi, “A hybrid metaheuristic for concurrent layout and scheduling problem in a job shop environment,” *Advanced Manufacturing Technology*, vol. 62, no. 9, pp. 1249-1260, Oct. 2012.

[5] J. Arkat, M. H. Farahani, and L. Hosseini, “Integrating cell formation with cellular layout and operations scheduling,” *Advanced Manufacturing Technology*, vol. 61, no. 5, pp. 637-647, Jul. 2012.

[6] J. Arkat, M. H. Farahani, and F. Ahmadizar, “Multi-objective genetic algorithm for cell formation problem considering cellular layout and operations scheduling,” *Computer Integrated Manufacturing*, vol. 25, no. 7, pp. 625-635, Mar. 2012.

[7] A. Ebrahimi, R. Kia, and A. R. Komijan, “Solving a mathematical model integrating unequal-area facilities layout and part scheduling in a cellular manufacturing system by a genetic algorithm,” *SpringerPlus*, vol. 5, no. 1, p. 1254, Aug. 2016.

[8] B. Giffler and G. L. Thompson, “Algorithms for solving production-scheduling problems,” *Operations Research*, vol. 8, no. 4, pp. 487-503, Aug. 1960.

[9] R. Kamoshida, “Concurrent optimization of job shop scheduling and dynamic and flexible facility layout planning,” in *Proc. Industrial Engineering and Applications*, Singapore, 2018, pp. 289-293.

[10] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Computers and Operations Research*, vol. 13, no. 5, pp. 533-549, May 1986.

[11] B. Çaliş and S. Bulkan, “A research survey: Review of AI solution strategies of job shop scheduling problem,” *Intelligent Manufacturing*, vol. 26, no. 5, pp. 961-973, Oct. 2015.

[12] P. Brucker, B. Jurisch, and B. Sievers, “A branch and bound algorithm for the job-shop scheduling problem,” *Discrete Applied Mathematics*, vol. 49, no. 1, pp. 107-127, Mar. 1994.

[13] M. Dell’Amico and M. Trubian, “Applying tabu search to the job-shop scheduling problem,” *Annals of Operations Research*, vol. 41, no. 3, pp. 231-252, Sep. 1993.

[14] S. P. Singh and R. R. K. Sharma, “A review of different approaches to the facility layout problems,” *Advanced Manufacturing Technology*, vol. 30, no. 5, pp. 425-433, Sep. 2006.

[15] E. K. Burke, G. Kendall, and G. Whitwell, “A new placement heuristic for the orthogonal stock-cutting problem,” *Operations Research*, vol. 52, no. 4, pp. 655-671, Aug. 2004.

[16] T. Yamada and R. Nakano, “A genetic algorithm with multi-step crossover for job-shop scheduling problems,” in *Proc. Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sheffield, 1995, pp. 146-151.

[17] J. E. Beasley, “OR-Library: Distributing test problems by electronic mail,” *Operational Research Society*, vol. 41, no. 11, pp. 1069-1072, Nov. 1990.

Ryota Kamoshida obtained a master’s degree in engineering from Waseda University, Japan. He is now working as a senior researcher at the Center for Technology Innovation - Digital Technology, Hitachi, Ltd. Research & Development Group. His research interests include Manufacturing Systems, Operations Research, Combinatorial Optimization and Machine Learning.