A Neighborhood Limitation Method for Job-Shop Scheduling Based on Simulated Annealing*

Kenta Teramoto†‡, Eiji Morinaga§, Hidefumi Wakamatsu† and Eiji Arai†

Because of non-deterministic polynomial time hardness of job-shop scheduling problem (JSP), approximate optimization based on meta-heuristics has been actively discussed. Considering position of planners in production sites, it is desirable to develop a method in which their know-how is respected. An approach for meeting this requirement is to set the schedule generated by a planner as the initial solution and then gradually improve the solution by repeating a search in its neighborhood so that he/she can follow and thoroughly examine the improved solution. For this reason, this research is focused on scheduling using simulated annealing (SA). Because SA has a disadvantage that good solutions cannot be obtained efficiently if the initial solution has not been given appropriately, methods for solving this problem have been proposed for JSPs aiming at minimizing makespan. In high-mix low-volume manufacturing, it is also important to minimize production lead time to reduce work-in-process inventory. This research takes up production lead time defined as the time between the starting and the finishing times of a job considering strong constraint on places for putting work-in-process in production of large equipment, and deals with development of an efficient method using SA for JSPs aiming at minimizing the average value of the production lead times. Two methods of neighborhood limitation in SA for reducing the evaluation value were developed by focusing on waiting time of operations. It was proven that using one of the proposed methods in SA with appropriate probabilities is effective to JSPs of a certain size by numerical examples.

1. Introduction

Diversified consumers’ needs have caused transition from low-mix high-volume manufacturing to high-mix low-volume manufacturing. As a result of this transition, job-shop scheduling problem (JSP) has been an important topic of discussion[1]. This problem is known as a non-deterministic polynomial time (NP) hard problem, and it is required to develop methods for finding an appropriate schedule for real-world problems in a short time. Application of approximate optimization by meta-heuristics to JSP has been expected to be an approach for satisfying this requirement[2–5].

In actual production, there is a person who is in charge of planning and deciding a production schedule. The planner is held responsible for the schedule. For this reason, a planner often generates and decides a schedule based on his/her own know-how. Considering such position of planners, it is desirable to develop a method in which the know-how of a planner is respected. An approach for meeting this requirement is to set the schedule generated by a planner as the initial solution and then gradually improve the solution by repeating a search in its neighborhood so that he/she can follow and thoroughly examine the improved solution. For this reason, this research is focused on scheduling using simulated annealing (SA). Because SA has a disadvantage that good solutions cannot be obtained efficiently if the initial solution has not been given appropriately, methods for solving this problem have been proposed for JSPs aiming at minimizing makespan. In high-mix low-volume manufacturing, it is also important to minimize production lead time to reduce work-in-process inventory. This research takes up production lead time defined as the time between the starting and the finishing times of a job considering strong constraint on places for putting work-in-process in production of large equipment, and deals with development of an efficient method using SA for JSPs aiming at minimizing the average value of the production lead times. Two methods of neighborhood limitation in SA for reducing the evaluation value were developed by focusing on waiting time of operations. It was proven that using one of the proposed methods in SA with appropriate probabilities is effective to JSPs of a certain size by numerical examples.

* Manuscript Received Date: October 3, 2018

† Graduate School of Engineering, Osaka University; 2-1 Yamadaoka, Suita city, Osaka 565-0871, JAPAN
‡ Currently, YANMAR Co., Ltd.
§ Osaka Prefecture University; 1-1 Gakuen-cho, Nakaku, Sakai, Osaka 599-8531, JAPAN

Key Words: production scheduling, simulated annealing, neighbor solutions, job-shop.
sider not only makespan but also other indices such as total weighted tardiness, weighted sum of tardy jobs, and so on[13–15]. It is necessary to develop efficient search methods for those indices. This research takes average production lead time into consideration, which is the average value of the time required for production of each product. This paper describes an improved scheduling method using SA with a process for performing the search in an appropriate direction or for avoiding the search in a wrong direction for efficiently reaching the optimal solution of JSP aiming at minimizing average production lead time.

2. Job-Shop Scheduling Using SA

2.1 Considered problem

The following JSP is considered in this paper. J jobs are processed with M machines. Job \( j \) needs \( O_j \) operations, and operation \( o \in \{1, \ldots, O_j\} \) is processed with a predetermined machine \( m^{j_o} \) and the time required for the operation has been given. Setup time, transportation time and restriction of place for putting a semi-finished product are ignored.

Many of existing job-shop scheduling methods aim to minimize makespan. However, in actual production, it is also important to shorten production lead time and reduce work-in-process (WIP) inventory in the factory. This paper deals with minimizing the average lead time \( ALT \) defined by (1), where \( ST_j \) and \( ET_{jO_j} \) stand for the starting time and the finishing time of job \( j \), respectively.

\[
ALT := \frac{1}{J} \sum_{j=1}^{J} (ET_{jO_j} - ST_j) \tag{1}
\]

The term production lead time is used for different purposes. It is usually used in the sense of flow time, which is defined as the time between the release time and the finishing time \([16,17]\), and job-shop scheduling considering average flow time has been discussed by various research groups[18–22]. It is sometimes defined as the time between the release time and the due date, and a release time determination method using backward and forward scheduling based on dispatching rules was proposed[23]. In production of large equipment such as turbine generator, propulsion device, and so on, constraint on place for putting WIPs is serious due to their size and it is therefore desirable to finish the production as soon as possible once after the production starts. For this reason, unlike the conventional researches, this research considers the time between the starting time and the finishing time as given in equation (1).

The constraint on place for putting WIPs can be rephrased into buffer capacity constraint as a more general term, which has been also discussed by various research groups based on mathematical programming and graph theory[24–27] and heuristics[28–30]. In particular, minimization based on the index (1) is essentially equivalent to minimization of waiting time, which has been performed by means of heuristic rules[31] and mathematical programming[32], since it is assumed that processing times are known in advance and do not change in this research. However, JSP with buffer capacity constraint or minimization of waiting time by SA has not been discussed.

2.2 Simulated annealing and its application to job-shop scheduling

SA imitates the annealing process with a computer in which heated material is cooled slowly so that stable material structure without defects is obtained. The basic algorithm is described as follows, and its flowchart is shown in Fig.1:

1. Decide an initial solution \( x_S \), an initial temperature \( T_S \), a final temperature \( T_F \), a cooling rate \( R(<1) \), and a number of repetitions at each temperature \( C \). Initialize the current solution \( x = x_S \), the temperature parameter \( T = T_S \) and the temporary solution \( x^* = x \).
2. Initialize the counter \( k = 1 \).
3. Select randomly a solution \( y \) in the neighborhood \( N(x) \), and calculate \( \Delta = f(y) - f(x) \). The probability of updating the current solution is set to 1 if \( \Delta \leq 0 \); to \( e^{(-\Delta)/T} \) otherwise. Then update the current solution \( x = y \) with the probability. Furthermore, if \( f(y) < f(x^*) \), update the temporary solution \( x^* = y \).
4. Increment \( k = k + 1 \). If \( k < C \), go to Step 2. Go to Step 5 otherwise.
5. Update \( T = RT \). If \( T > T_F \), go to Step 2. Otherwise, accept the temporary solution \( x^* \) as the final solution and stop.

In order to apply SA to production scheduling, it is necessary to define a representation method of a schedule which corresponds to a solution in the algorithm. A representative method is permutation representation. In this method, a one-dimensional array of \( \sum_{j=1}^{J} O_j \) elements is defined and numbers between 1 and \( J \) are assigned to the elements for \( O_j \) times. The value of an element represents an operation number of the job the ID number of which is equal to the value. The operation number is determined by the number of appearances of the value in the elements located prior to the element (Fig.2). In the array, a prior element has higher priority than posterior elements. By assigning the operation represented by each element to the earliest idle time of the machine in the order of the priority, it is possible to generate a schedule and evaluate average production lead time.

Based on this permutation representation, the neighborhood is often defined as the set of solutions generated by selecting two elements and exchanging them (Fig.3). In SA, the search proceeds by randomly selecting a solution included in this neighborhood. However, with such a random selection, there is a high possibility of transitioning to an inappropriate neighbor solution, and it is possible that optimization cannot be performed sufficiently or efficiently.
3. Proposed Method

This section describes methods for limitation of the neighborhood so that it becomes easier to reduce average production lead time than considering all solutions in the neighborhood. Implementing an operation which limits the neighborhood appropriately in SA reduces transition to an improper neighbor solution and is expected to be a method that can efficiently perform optimization (Fig. 4).

3.1 Method 1

For the limitation of the neighborhood, this method is focused on the operation with the longest waiting time to start. In the schedule generated from the current solution, waiting time of each operation is evaluated, and the operation with the longest waiting time is selected. It is expected that shortening the waiting time for that operation (the third operation of job 2 in the example shown in Fig. 5) is effective for shortening the average lead time. Therefore, one of the operations within this waiting time is randomly selected as the candidate A to be exchanged for generating a neighbor solution.

As the other candidate B to be exchanged, the final operation of the job having the operation with the longest waiting time is selected. There are two reasons for adopting this selection. The first one is that, when an operation is chosen as the candidate A and the element indicating the operation is selected in the array, it is possible to surely move this operation out of the longest waiting time by inserting the number of the job having the longest waiting time into the selected element. The other one is that, in the per-
mutation representation, it is expected that the lead time of a job becomes shorter as the numbers of the job are placed as close to each other as possible.

By selecting the above two elements as candidates A and B to be exchanged and swapping them on the permutation representation, the operation A placed within the longest waiting time moves out of it, the longest waiting time becomes shorter, and therefore it is expected that the solution is updated in the direction so that the average lead time becomes shorter.

3.2 Method 2

Method 1 intends to find the operation with the longest waiting time and shorten it. Method 2 is focused on an operation which would be much effective for reduction of waiting time of another operation by moving itself.

For the limitation of the neighborhood, the operation the processing time of which is longest is chosen from operations included in waiting time of an operation. By moving this operation out of the waiting time, the starting time of the operation that was made to wait by that operation is put greatly ahead, and it is expected that the lead time of this job is shortened and the average lead time is also reduced. For this reason, this operation is selected as the candidate A to be exchanged for generating a neighbor solution. As the candidate B to be exchanged, the final operation of the job which is made to wait by the selected operation for the same reason as Method 1.

4. Numerical Examples

4.1 Verification with industrial size problems

The proposed method was applied to industrial size examples of \( M = 50, J = 89 \) and \( \max_j O_j = 38 \) for verification of its effectiveness. Five problems were generated by determining the values of processing times from a predetermined range based on the uniform distribution. In order to check the existence of initial solution dependence, we prepared initial solution 1, 2, 3, 4, and 5 with random. The normal SA and the proposed methods were applied to each of the five problems for 10 times by changing the seed for random numbers, and the average values of the average lead time of each problem were compared. The final temperature \( T_F \), the cooling rate \( R \) and the number of repetitions at each temperature \( C \) were set to 1, 0.9995, and 1, respectively. Optimizations were performed using a generic workstation (CPU: Intel Core i5-6803.60GHz, 4.00GB RAM).

The two proposed methods of neighborhood solution generation were utilized in SA with the probability of 0%, 10%, 20%, ..., 100%, respectively. The initial temperature \( T_S \) was set to 1000. Figs. 10, 11 show the result of Method 1 and Method 2. The horizontal axis shows the ratio of using the method in generation of the neighbor solution \( P \) and the vertical axis shows the average lead time. There are no significant differences among the results of any initial solutions. As for Method 2, the limitation did not have a positive effect but rather a negative effect.
Fig. 7 Flowchart of SA including the proposed neighborhood limitation methods ("rand[0,1]" stands for a real number generated from the closed interval [0,1] randomly.)

Find an operation that has the longest waiting time, and define the operation and the waiting time as $O_{MN}$ and $LWT_{MN}$

Store operations in the $LWT_{MN}$ to the array of candidate operations

Select an operation randomly from the array and define it as A

Define the last operation of M as B

Swap A with B

Output the nearest neighbor solution

Fig. 8 Flowchart of Method 1

Choose the longest operation ($O_{MN}$) which is in waiting time $WT_{ij}(i=1,2,...,I)(j=1,2,...,J)$, and define the operation as A

Define the last operation of M as B

Swap A with B

Output the nearest neighbor solution

Fig. 9 Flowchart of Method 2
on finding a good solution for almost all the values of $P$. This would be because using this method imposes too strong constraints, since the limited neighborhood includes only one candidate to be chosen. As for Method 1, better solutions could be found than the normal SA ($P = 0$) and the evaluation value becomes better as $P$ increases except 100%. As described in 3.3, the improved SA forces generation of the next neighbor solution without limitation when the last neighbor solution has not been accepted as the current solution, since the limitation may not be effective in such a case. Due to this intentional avoidance of the neighborhood limitation, the actual percentage of using Method 1 differs from the specified value of $P$. Table 1 shows the average value of actual number and percentage of the normal SA (column A), Method 1 (column B) and the intentional avoidance (column C). Although the actual percentage is quite different from the specified value in the range of 20% and 90%, it can be confirmed that the neighborhood limitation by Method 1 is effective for finding a better solution. On the other hand, the evaluation value was quite worse when $P = 100$, where the actual percentage was 96.8% and near the specified value, though a good solution was sometimes found even in such a case. This implies that using this method almost always and prohibiting random search almost every time results in imposing too much limitation.

Table 1 The average value of actual number and percentage of the normal SA (column A), Method 1 (column B) and the intentional avoidance (column C)

| $P$ | A   |   | B   |   | C   |   |
|-----|-----|---|-----|---|-----|---|
|     | #   | % | #   | % | #   | % |
| 0   | 13813 | 100 | 0   | 0 | 0   | 0 |
| 10  | 11538 | 83.5 | 1301 | 9.4 | 794 | 5.7 |
| 20  | 9616  | 69.6 | 2371 | 17.2 | 1826 | 13.2 |
| 30  | 7962  | 57.6 | 3411 | 24.7 | 2240 | 16.2 |
| 40  | 6438  | 46.6 | 4309 | 31.2 | 3066 | 22.2 |
| 50  | 4995  | 36.2 | 4947 | 35.8 | 3871 | 28 |
| 60  | 3878  | 28.1 | 5627 | 40.7 | 4308 | 31.2 |
| 70  | 2754  | 19.9 | 6478 | 46.9 | 4581 | 33.2 |
| 80  | 1768  | 12.8 | 6876 | 49.8 | 5169 | 37.4 |
| 90  | 928   | 6.7  | 7366 | 53.3 | 5519 | 40 |
| 100 | 0     | 0    | 13377 | 96.8 | 436 | 3.2 |

In order to investigate effectiveness of the intentional avoidance of neighborhood limitation, the improved SA with Method 1 without this operation was also applied to the same examples. The result is shown in Fig. 12. It seems that the proposed limitation is most effective when $P$ is in or around the range of 40% and 60%. This accords with the result of Method 1 with the intentional avoidance shown in Fig. 10 and Table 1. Considering these results, it seems that the intentional avoidance does not have a positive effect or a negative effect.
Fig. 11 The average value of the average lead time obtained by Method 2 ($T_S = 1000$)

Fig. 12 The average value of the average lead time obtained by Method 1 without the intentional avoidance of the neighborhood limitation ($T_S = 1000$)
As for Method 1, effect of initial temperature was also examined, which is an important parameter of SA, by changing $T_S$ to 10000, 500 and 100, respectively. Figs. 13–15 show results of those temperatures. These figures show results for only one of the five problems due to limitations of space, but similar results were obtained for the rest of the problems. The method is more effective when initial temperature is set to a higher value. On the other hand, when the method is used with a low initial temperature, the method is not more effective than the normal SA. These results imply that Method 1 is effective for avoiding updating of a solution in a wrong direction rather than for forcing a solution to be updated in an appropriate direction.

4.2 Verification with problems of different size

Effectiveness of the proposed method was also investigated from the point of view of problem size. Five problems of $M = 15$, $J = 30$ and $\max_j O_j = 15$
were generated by determining the values of processing times from a predetermined range based on the uniform distribution. The improved SA using Method 1 with the intentional avoidance of neighborhood limitation was applied to them and optimizations were performed under the same condition. Figs. 16–19 show the results of one of the five problems where the initial temperature was set to 10000, 1000, 500 and 100, respectively. Similar results were obtained for the rest of the five problems. There is almost no difference among the obtained values in the range of 0% and 90%. This would be because that the solution space is small and random search without neighborhood limitation has a sufficient ability.

Effect of the proposed method was also investigated with a larger size example of $M = 50$, $J = 130$ and $\max_j \omega_j = 60$, though multiple optimizations under various conditions could not be performed due to large computational load. Fig. 20 shows the result obtained with an initial solution and the initial temperature of 1000. Positive effects by the proposed method were not found either. The reason may be that the solution space is quite large and the improved SA is no longer effective with the initial temperature. Positive effects might be found if the optimization is performed with higher initial temperature, though it requires higher computational power and longer computation time.

5. Conclusions

This paper has described a scheduling method using SA for JSP aiming at minimizing the average value of production lead time which was defined as the time between the starting and the finishing times of a job considering constraint on places for putting WIPs. Two methods for limiting the neighborhood of a solution in SA have been proposed by focusing on the operation having the longest waiting time and the operation which is the cause of the waiting time of another operation and has a long processing time. It has been confirmed that, for JSP of a certain size, applying one of those two methods with an appropriate probability in SA makes it possible to obtain a better solution than that obtained by the normal SA because the method is effective for avoiding updating of a solution in a wrong direction.

The effectiveness of the proposed method for larger size JSPs was not shown in this paper. However, the method may be effective to those problems if optimizations are performed with sufficiently high initial temperature. It is desirable to develop a suitable computational environment and make this point clear. In addition, the verifications were performed with only JSPs uniquely generated based on an actual production in industry. It is also desirable to perform solid verifications using benchmark problems proposed by other research groups.

In order to advance the proposed method to a practical one, it is necessary to consider various evaluation indices such as makespan, total weighted tardiness, earliness, and so on. It is also necessary to take requirements and constraints in actual production sites into consideration such as overtime and worker scheduling. Moreover, it is expected that the method of generating neighbor solutions aiming at shortening the average production lead time dealt with this research can be applied not only to SA but also to other meta-heuristics. In addition, evaluation and discussion from the point of view of respecting know-how of planners should be performed. These issues will be discussed in future works.

References

[1] R. Meller: A review of job shop scheduling; *Operational Research Quarterly*, Vol. 17, No. 2, pp. 161–171 (1996)
[2] B. Çalış and S. Bulkan: A research survey: Review of AI solution strategies of job shop scheduling problem; *Journal of Intelligent Manufacturing*, Vol. 26, No. 5, pp. 961-973 (2015)
[3] R. J. M. Vaessens, E. H. L. Aarts and J. K. Lenstra: Job shop scheduling by local search; *INFORM Journal on Computing*, Vol. 8, No. 3, pp. 302–317 (1996)
[4] R. Cheng, M. Gen and Y. Tsujimura: A tutorial
survey of job-shop scheduling problems using genetic algorithms—I. Representation; Computers & Industrial Engineering, Vol. 30, No. 4, pp. 983–997 (1996)

[5] R. Cheng, M. Gen and Y. Tsujimura: A tutorial survey of job-shop scheduling problems using genetic algorithms, Part II: Hybrid genetic search strategies; Computers & Industrial Engineering, Vol. 36, No. 2, pp. 343–364 (1999)

[6] S. Kirkpatrick, C. D. Gelatt and M. P. Vecch: Optimization by simulated annealing; Science, Vol. 220, No. 4598, pp. 671–680 (1983)

[7] P. M. van Laarhoven, E. H. L. Aarts and J. K. Lenstra: Job shop scheduling by simulated annealing; Operations Research, Vol. 40, No. 1, pp. 113–125 (1992)

[8] S. Ponnambalam, N. Jawahar and P. Aravindan: A simulated annealing algorithm for job shop scheduling; Production Planning & Control, Vol. 10, No. 8, pp. 767–777 (1999)

[9] T. Yamada, B. E. Rosen and R. Nakano: Critical block simulated annealing for job shop scheduling; Transactions of IEE, Vol. 114-C, No. 4, pp. 476–482 (1994) (In Japanese)

[10] T. Yamada and R. Nakano: Job-shop scheduling by simulated annealing combined with deterministic local search; Meta-Heuristics (I. H. Osman and J. P. Kelly, Eds.), Springer, pp. 237–248 (1996)

[11] K. Steinhöfel, A. Albrecht and C. K. Wong: Two simulated annealing-based heuristics for the job shop scheduling problem; European Journal of Operational Research, Vol. 118, Issue 3, pp. 524–548 (1999)

[12] T. Satake, K. Morikawa, K. Takashashi and N. Nakamura: Simulated annealing approach for minimizing the makespan of the general job-shop; International Journal of Production Economics, Vols. 60–61, pp. 515–522 (1999)

[13] C. Bierwirth and J. Kuhpfahl: Extended GRASP for the job shop scheduling problem with total weighted tardiness objective; European Journal of Operational Research, Vol. 261, pp. 835–848 (2017)

[14] R. Zhang and C. Wu: A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective; Computers & Operations Research, Vol. 38, Issue 5, pp. 854–867 (2011)

[15] Y. Mati, S. Dauzere-Peres and C. Lahhou: A general approach for optimizing regular criteria in the job-shop scheduling problem; European Journal of Operational Research, Vol. 212, pp. 33–42 (2011)

[16] F. Karaesmen, G. Liberopoulos and Y. Dallery: Production/inventory control with advance demand information; Stochastic Modeling and Optimization of Manufacturing Systems and Supply Chains (G. Shanthikumar, D. D. Yao and W. H. M. Zijm, Eds.), Springer, pp. 243–270 (2003)

[17] S. M. Meekov and C. B. Yan: Production lead time problem: Formulation and solution for Bernoulli Serial Lines; IFAC Proceedings Volumes, Vol. 46, pp. 676–681 (2013)

[18] W. Kubiak, S. X. C. Lou and Y. Wang: Mean flow time minimization in reentrant job shops with a hub; Operations Research, Vol. 44, pp. 764–776 (1996)

[19] O. Holthaus and C. Rajendran: Efficient dispatching rules for scheduling in a job shop; International Journal of Production Economics, Vol. 48, pp. 87–105 (1997)

[20] V. K. Ganesan and A. I. Sivakumar: Scheduling in static jobshops for minimizing mean flowtime subject to minimum total deviation of job completion times; International Journal of Production Economics, Vol. 103, pp. 633–647 (2006)

[21] V. Sels, N. Gheyseren and M. Vanhoucke: A comparison of priority rules for the job shop scheduling problem under different flow-time and tardiness-related objective functions; International Journal of Production Research, Vol. 50, pp. 4255–4270 (2012)

[22] K. C. Udaiyakumar and M. Chandrasekaran: Optimization of multi objective job shop scheduling problems using firefly algorithm; Applied Mechanics and Materials, Vol. 591, pp. 157–162 (2014)

[23] S. Hirano and T. Eguchi: A release time determination method to meet due-dates and minimize lead-times in job shops; Proceedings of 2010 International Symposium on Flexible Automation, IPSF-2099 (2010)

[24] H. Tamaki and Y. Nishikawa: Modeling of job-shop scheduling problem with in-process buffer capacity; Transactions of the Society of Instrument and Control Engineers, Vol. 31, pp. 933–940 (1995) (In Japanese)

[25] H. Tamaki and Y. Nishikawa: Solution of the job-shop scheduling problem with consideration of in-process buffer capacity; Transactions of the Society of Instrument and Control Engineers, Vol. 31, pp. 1193–1201 (1995) (In Japanese)

[26] P. Brucker, S. Heitmann, J. Hurink and T. Nieberg: Job-shop scheduling with limited capacity buffers; OR Spectrum, Vol. 28, pp. 151–176 (2006)

[27] R. Hino, Y. Kodama, N. Suzuki and E. Shamoto: Job shop scheduling focusing on role of buffers (2nd Report, Feasibility judgement by utilizing graph); Transactions of the Japan Society of Mechanical Engineers, C., Vol. 73, pp. 3084–3091 (2007) (In Japanese)

[28] R. Hino, T. Kusumi, J. K. Yoo and Y. Shimizu: Job shop scheduling focusing on role of buffer; JSME International Journal Series C, Vol. 49, pp. 950–956 (2006)

[29] A. Witt and S. Voß: Job shop scheduling with buffer constraints and Jobs consuming variable buffer space; Advanced Manufacturing and Sustainable Logistics. IHNS 2010 (W. Dangelmaier, A. Blecken, R. Delius, S. Klöpfer, Eds.), Springer, pp. 295–307 (2010)

[30] S. Q. Liu, E. Koza, M. Masoud, Y. Zhang and F.T.S. Chan: Job shop scheduling with a combination of four buffering constraints; International Journal of Production Research, Vol. 56, pp. 3274–3293 (2018)

[31] C. Chu and M. C. Portmann: Job-shop scheduling to minimize total waiting time; Applied Stochastic Models and Data Analysis, Vol. 9, pp. 177–185 (1993)

[32] R. Hino: Job shop scheduling focusing on role of buffers (3rd Report, Optimization by mixed-integer programming); Transactions of the Japan Society of Mechanical Engineers, C., Vol. 74, pp. 1669–1675 (2008) (In Japanese)
Kenta Teramoto

Kenta Teramoto received his B.Eng. and M.Eng degrees from Osaka University, Japan, in 2017 and 2019, respectively. He then joined YANMAR Co., Ltd.

Eiji Morinaga (Member)

Eiji Morinaga received his B.Eng., M.Eng. and Ph.D. degrees in Mechanical Engineering from Osaka University, Japan, in 2000, 2002 and 2005, respectively. He worked as a Designated Researcher at Center for Advanced Science and Innovation from 2005 to 2007 and as an Assistant Professor at Division of Materials and Manufacturing Science from 2007 to 2020, Osaka University. He then joined Osaka Prefecture University as an Associate Professor. His research interests include system design and integration in product design and manufacturing. He is a member of the Japan Society of Mechanical Engineers (JSME), the Japan Society for Precision Engineering (JSPE), the Japan Society for Manufacturing Engineering (JSME), the Japan Society for Precision Engineering (JSPE), the Japan Society for Manufacturing Engineers (JSME) and the Textile Machinery Society of Japan.

Hidefumi Wakamatsu (Member)

Hidefumi Wakamatsu received his B.Eng., M.Eng. and Ph.D. degrees from Osaka University, Japan, in 1993, 1994 and 2001, respectively. From 1995 to 2006, he worked as a Research Associate, Osaka University. He then became an Associate Professor at Division of Materials and Manufacturing Science, Osaka University. His research interests include handling of flexible object. He is a member of the Robotics Society of Japan (RSJ), the Japan Society of Mechanical Engineers (JSME), the Japan Society for Precision Engineering (JSPE), the Japan Welding Society (JWS) and the Textile Machinery Society of Japan.

Eiji Arai (Member)

Eiji Arai received his B.Eng., M.Eng. and Dr.Eng. degrees from The University of Tokyo, Japan, in 1975, 1977 and 1980, respectively. He worked as a Research Associate at Kobe University from 1980 to 1984, an Associate Professor at Shizuoka University from 1984 to 1992, and an Associate Professor at Tokyo Metropolitan University from 1992 to 1995, respectively. He then joined Osaka University where he worked as a Professor of Department of Materials and Manufacturing Science from 1995 to 2018 and he is a Professor Emeritus currently. His research interests include intelligent CAD/CAM for mechanical design.