

A METHOD FOR FLEXIBLE JOB-SHOP SCHEDULING USING GENETIC ALGORITHM

E. Morinaga¹, Y. Sakaguchi^{1,2}, H. Wakamatsu¹ and E. Arai¹

¹Division of Materials and Manufacturing Science, Osaka University, Japan.

²YANMAR Corporation, Ltd.

Corresponding Author's Email: ¹morinaga@mapse.eng.osaka-u.ac.jp

Article History: Received 11 August 2017; Revised 28 November 2017;
Accepted 6 December 2017

ABSTRACT: This paper focused on solving a flexible job-shop scheduling problem. Because this problem is known as NP-hard, methods using meta-heuristics especially genetic algorithm (GA) have been actively proposed. Although it is possible to obtain solutions of large problems in a reasonable time by those methods, the quality of the solutions decreases as the scale of the problem increases. Hence, taking advantage of knowledge included in heuristic dispatching rules in the optimization by GA was proposed, and its effectiveness was proven. However, in this method, the two kinds of selection required in flexible job-shop production, machine selection and job selection, were carried out sequentially. Because this may result in insufficient search of the solution space, this paper provided a method using GA in which those two selections were performed at once. The method was applied to an example and it was shown that better solutions could be obtained.

KEYWORDS: *Flexible Job-shop Scheduling; Genetic Algorithms; Heuristic Rules*

1.0 INTRODUCTION

In order to provide an artifact, where data have been created in the product design phase, as a product with high competitiveness, it is necessary to optimally design, properly manage and efficiently operate a manufacturing system. Production scheduling is a key issue for an efficient operation of manufacturing systems, and has been discussed for a long time [1-4]. Numerous studies have been performed in various approaches such as mathematical programming [5], heuristic dispatching methods [6], artificial intelligence [7-9] and graphs [10].

Diversified customers' needs has driven transition from the low-mix high-volume manufacturing to the high-mix low-volume manufacturing. This transition increases the importance of job-shop scheduling, and various methods have been proposed. In many of

them, it is assumed that each operation can be executed by only one specific machine. However, in actual manufacturing, there are often multiple machines which can perform an operation, that is, it is necessary to take flexible job-shop production into account. Because flexible job-shop scheduling problem (FJSP) which is known as NP-hard, optimization by metaheuristic algorithms has been discussed. In particular, scheduling methods using genetic algorithm (GA) have been actively proposed [11-15]. Although it is possible to obtain solutions of large problems in a reasonable time by those methods, the quality of the solutions decreases as the scale of the problem increases.

For this reason, incorporation of GA and heuristic dispatching rules is proposed [16]. In FJSP, two kinds of selection need to be performed, one is for machine selection and the other is for job selection. In machine selection, it is determined by which machine each operation is executed. In job selection, it is determined which operation in the waiting queue of a machine is performed when the machine has become available. In that method, two kinds of chromosomes are introduced for those two selections. The random key coding method is applied and each gene is given as a value which indicates priority for those selections. Those values are multiplied by priority value given by a heuristic rule, and the selections are carried out based on those values. By this incorporation mechanism, it is possible to take advantage of optimization by both GA and heuristic knowledge, and it is proven that a better solution for minimizing mean job tardiness for FJSP can be obtained by the method. However, in this method, the machine selection is first carried out and then the job selection is performed. This two-step selections may result in insufficient search of the solution space. This paper provided a flexible job-shop scheduling (FJS) method using GA which performed the two selections at once.

2.0 PROPOSED METHOD

This study focused on flexible job-shop production with J jobs and M machines. Job $j \in J = \{1, \dots, J\}$ needed O_j operations. Operation $o \in O_j = \{1, \dots, O_j\}$ of job j was executed by a machine m_{jo} , which belonged to the set of machines that could deal with the operation $M_{jo} \subset M = \{1, \dots, M\}$. Each machine could execute only one operation at a time. It is assumed that the buffer size of a machine was infinite and time for transportation could be ignored. The objective of the scheduling was the minimization of make span $MSPN$

$$MSPN := \max_j ET_{jO_j} \quad (1)$$

where ET_{j_o} stands for the completion time of operation o of job j .

In the scheduling method using GA and heuristic rules [16], each gene is given a real value between 0 and 1, indicating priority for machine or job selection. This coding method is suitable for incorporating heuristic rules and taking advantage of knowledge included in the rules. For this reason, the same coding method was adopted in the proposed method. A chromosome was composed of loci, $\sum_{j=1}^J \sum_{o=1}^{O_j} \text{card} M_{j_o}$, where “ $\text{card} M_{j_o}$ ” stands for the number of the elements of M_{j_o} . To each locus, a value between 0 and 1, a job number, an operation number and a machine number are assigned. Figure 1 shows an example of $J=3$, $O_1=O_2=O_3=3$, $M=4$, $M_{11}=M_{21}=M_{22}=M_{33}=\{3\}$, $M_{12}=\{1,4\}$, $M_{13}=M_{32}=\{2\}$, $M_{23}=\{1\}$ and $M_{31}=\{1,2,4\}$. In this case, 12 loci compose a chromosome. When an individual was given a chromosome described above, it was necessary to carry out decoding and generate a corresponding schedule in order to evaluate the individual.

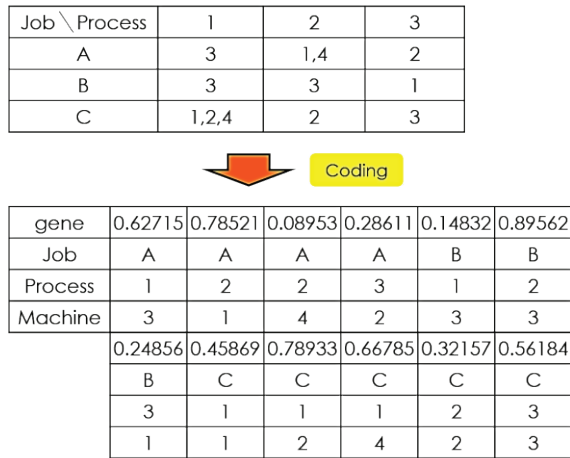


Figure 1: Coding for a chromosome

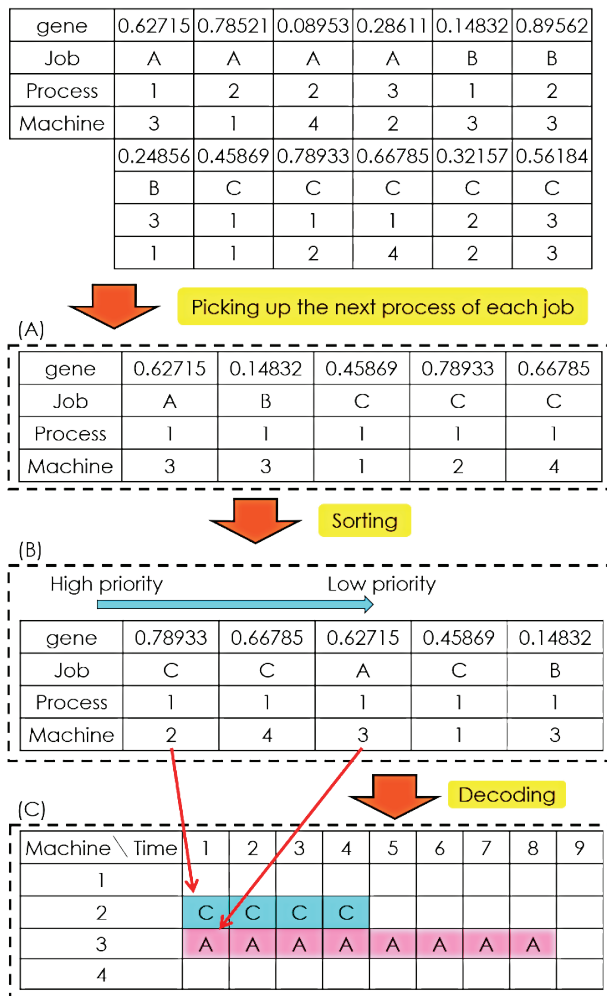


Figure 2: Decoding for a chromosome

For an example shown in Figure 1, this decoding was performed as follows; The decoding started at time 0. At this time, the first operation of each job could be taken into consideration, and their corresponding loci were picked-up ((A) in Figure 2). Next, those loci were sorted by the value of the genes ((B) in Figure 2). They were then assigned to the corresponding position of the Gantt chart from the loci of the highest priority ((C) in Figure 2). In this assignment process, unfeasible loci were ignored even if they had high priority. As a result, only the first and third loci were assigned to the Gantt chart for time 0, and then the time was updated to 5 when machine 2 could accept another operation. This process was continued until all operations had been assigned to the chart, and then the make span was evaluated.

Note that, as well as the existing method [16], it is possible to incorporate heuristic rules with the proposed method just by multiplying the value of a gene by the value obtained by the rule. For example, the SPT (shortest processing time) rule can be incorporated just by multiplying the value of a gene by the reciprocal of the processing time.

Figure 3 shows the flowchart of the proposed method. After generating initial population, decoding and evaluation of each individual were performed. Unless the termination condition was satisfied, genetic operation was executed. Selection was performed based on the elitist selection scheme as well as the existing method. The 20% of the best individuals were copied to the next generation. The rest of the next generation were generated by crossover of the 20% of the best individuals in the current generation. In the crossover process, a gene in an offspring was generated by copying the gene in a parent individual of a pair with the possibility of 70% and the gene in the other individual of the pair with the possibility of 30%. The mutation process was performed by randomly changing the value of a gene to a value between 0 and 1 with the possibility of 5%.

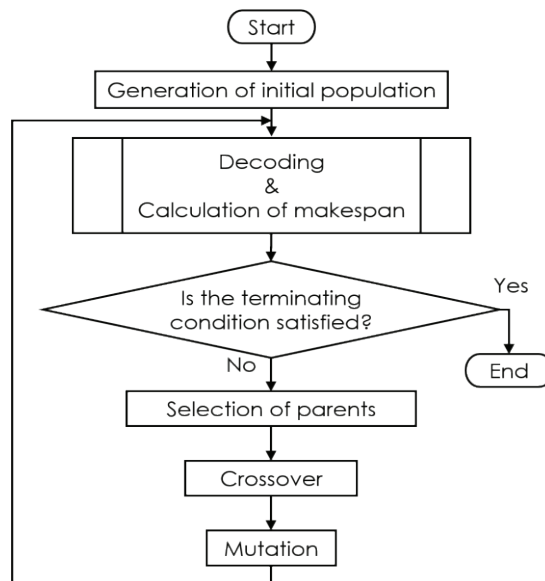


Figure 3: Flowchart of the proposed method

3.0 CASE STUDY

The proposed method was applied to an example of $J=89$, $\max_j O_j=43$ and $M=51$, the size of which was as large as that of an actual manufacturing. The population size was 100. The termination condition was given as the maximum number of alternation of generations, and its value was 100. The existing method (1) with no rule, (2) with LPT (Longest Processing Time) and $(WINQ+RPT+PT)\times PT$ rules, the proposed method (3) with no rule, (4) LPT rule and (5) SPT rule were applied, respectively. Figure 4 shows the result. The results of (1) and (3) showed that executing both machine selection and job selection at once enabled finding a better solution because it was possible to search the solution space more sufficiently.

Although it is necessary to sufficiently study other rules and examples because the existing method (2) was proposed for minimization of mean job tardiness, the results of (2) and (3) implied that the execution of the two selections at once could be more effective in finding a better solution than incorporating heuristic knowledge. This is also implied by those of (3), (4) and (5), though it is necessary to study and apply other rules which are suitable for this example.

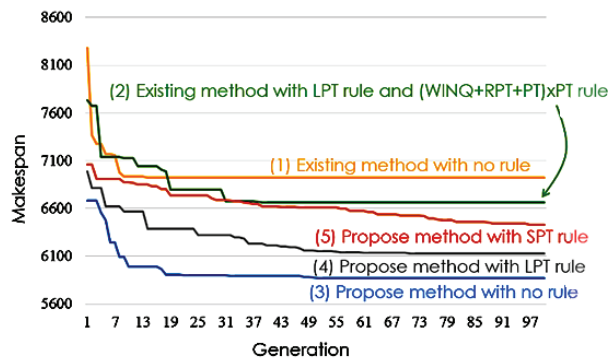


Figure 4: Result of makespan versus alternation number of generations

4.0 CONCLUSION

In this paper, an FJS method using GA has been proposed. Unlike the existing method, the machine selection and the job selection are performed at once in the proposed method. This feature makes it possible to search in larger area of the solution space and therefore, it is expected that a better solution can be obtained. The proposed method is applied to an example and it is implied that this hypothesis is true and

this method is effective. Detailed study and discussion using various examples to show effectiveness of the method will be performed in future works.

ACKNOWLEDGMENTS

We thank Prof. Toru Eguchi, Hiroshima University, Japan, for his advice.

REFERENCES

- [1] S.C. Graves, "A review of production scheduling", *Operations Research*, vol. 29, no. 4, pp. 646-675, 1981.
- [2] T.C.E. Cheng and M.C. Gupta, "Survey of scheduling research involving due date determination decisions", *European Journal of Operational Research*, vol. 38, no. 2, pp. 156-166, 1989.
- [3] D. Lei, "Multi-objective production scheduling: A survey", *International Journal of Advanced Manufacturing Technology*, vol. 43, no. 9-10, pp. 926-938, 2009.
- [4] J. Branke, S. Nguyen, C.W. Pickardt and M. Zhang, "Automated design of production scheduling heuristics: A review", *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 110-124, 2016.
- [5] C.H. Pan, "A study of integer programming formulations for scheduling problems", *International Journal of Systems Science*, vol. 28, pp. 33-41, 1997.
- [6] S.S. Panwalker and W. Iskander, "A survey of scheduling rules", *Operations Research*, vol. 25, pp. 45-61, 1977.
- [7] D.E. Akyol and D.M. Bayhan, "A review on evolution of production scheduling with neural networks", *Computers and Industrial Engineering*, vol. 53, pp. 95-122, 2007.
- [8] K.S. Metaxiotis, D. Askounis and J. Psarras, "Expert systems in production planning and scheduling: a state-of-the-art survey", *Journal of Intelligent Manufacturing*, vol. 13, no. 4, pp. 252-260, 2002.
- [9] S.J. Noronha and V.V.S. Sarma, "Knowledge-based approaches for scheduling problems: a survey", *IEEE Transactions on Knowledge and Data Engineering*, vol. 3, pp. 160-171, 1991.
- [10] G. Tuncel and M. Bayhan, "Applications of petri nets in production scheduling: a review", *International Journal of Manufacturing Technology*, vol. 34, pp. 762-773, 2007

- [11] F. Pezzella, G. Morganti and G. A. Ciaschetti, "Genetic algorithm for the flexible job-shop scheduling problem", *Computers & Operations Research*, vol. 35, pp. 3202-3212, 2008.
- [12] L. De Giovanni and F. Pezzella, "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem", *European Journal of Operational Research*, vol. 200, pp. 395-408, 2010.
- [13] B.A. Norman and J.C. Bean, "A genetic algorithm methodology for complex scheduling problems", *Naval Research Logistics*, vol. 46, pp. 199-211, 1999.
- [14] G. Zhang, L. Gao and Y. Shi, "An effective genetic algorithm for the flexible job-shop scheduling problem", *Expert Systems with Applications*, vol. 38, pp. 3563-3573, 2011.
- [15] N. Morad and A. Zalzal, "Genetic algorithms in integrated process planning and scheduling", *Journal of Intelligent Manufacturing*, vol. 10, pp. 169-179, 1999.
- [16] P. Kaweegitbundit and T. Eguchi, "Flexible job shop scheduling using genetic algorithm and heuristic rules", *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 10, no. 1, pp. 1-18, 2015.