International Business Management 10 (19): 4708-4712, 2016
ISSN: 1993-5250

A Knowledge-Based Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem

Muhammad Ridwan Andi Purnomo
Department of Industrial Engineering, Universitas Islam Indonesia, Yogyakarta, Indonesia

Abstract: This study presents application of an improved Genetic Algorithm (GA) for solving Flexible Job Shop Scheduling Problem (FJSP). Flexible job Shop Production System (FJPS) is the extension of classical job shop production system. In the FJPS, a job has fixed operations sequence and every operation could be processed by one of machines in a Work Station (WS). The processing time could be different if the job is processed by different machine in same WS. FJPS are commonly found in furniture or semi-conductor industries. In term of scheduling, problem in FJSP is distribution of jobs and their schedule in every machine. Such problem is a hard combinatorial problem and one of the algorithm that could be used to solve the problem is GA. However, based on preliminary study, a conventional GA could not perform effective searching process when being used to solve FJSP. In this study, a conventional GA would be improved by using a knowledge-based system which extracted from a FJPS. Further, the improved GA is called as Knowledge-Based GA (KB-GA). A case study shows that the proposed KB-GA could conduct effective searching process and has superior performance compared to a conventional GA.

Key words: Genetic algorithm, knowledge-based genetic algorithm, flexible job shop, scheduling, optimisation

INTRODUCTION

In order-based manufacturing environment which will start production activities only after orders were came, a flexible production shop floor is required since the product variation would be high with low to medium volume for each product variant. The suitable production facilities layout for that manufacturing environment is job shop layout which groups the production machines into several WS based on their manufacturing process type. Consequently, if parts to form a product have different process sequence, then the material flow would be complicated and causes longer manufacturing lead time. Therefore, productivity of the system would be relatively low or medium.

To increase productivity of a production system with job shop layout, one of the effort is adding other machines into the WS so that a WS could process several parts at a time. Such manufacturing system is called flexible job Shop Production System (FJPS). In an FJPS, when a part is decided to be processed by different machine in a WS its processing time could be different. In term of scheduling, there are 2 questions to be answered, 1st is what to be processed at a time and the second is when to process a material. In order to answer the first question, from the machine side, there should be a decision regarding distribution of jobs to every machine. At the same time, the second question must be answered by determining sequence of the jobs to be processed in every machine. Flexible job shop scheduling problem (FJSP) becomes more complicated compared to classic job shop scheduling problem, because in FJSP, all of the decision variables must be determined at the same time (Gao *et al.*, 2015a) and unfortunately, there is still no formal algorithm to solve it.

Several studies about complexity of FJSP have been investigated by previous researchers. Abdelmaguid (2015) has studied about separable sequence-dependent setup times of jobs in an FJSP. In that study, sequence of the jobs in machines affects to setup time and finally affects the makespan directly. Birgin *et al.* (2015) has investigated an FJSP with sequencing flexibility. In that study, a precedence operations could be given to a job and it must be considered when doing job sequencing. Gao *et al.* (2015b) has investigated insertion of new jobs in FJPS. Insertion of new jobs into existing FJPS schedule is very complicated problem. It is because for the existing jobs, their scheduling could be considered as normal FJSP, while the scheduling for new jobs must be considered as an FJSP with different jobs starting time and machines ready time. Mokhtari and Dadgar (2015) have investigated an FJSP with time-varying machine failure rate. In such study, preventive maintenance for machines would make jobs have routing alternatives and it makes the FJSP became more complicated. Another study that consider uncertainty in FJPS has been carried out by Palacios *et al*. (2015). In that study, the uncertainty factor is jobs duration. A fuzzy number has been applied to model the jobs duration and enables the case could be solved. Besides studies on complexity of FJPS, from the method to find the solution, there are several previous studies on FJSP that use meta-heuristics algorithm. For example, Tabu Search (TS) (Brandimarte, 1993), Genetic Algorithm (GA) (Gao et al., 2008), Ant Colony Optimisation (ACO) (Xing *et al*., 2010) and Particle Swarm Optimisation (PSO) (Zhang *et al*., 2009).

This study focuses on GA as the optimisation algorthm in finding the solution for an FJSP. Generally, it has realised that improvements are still required to increase performance of a conventional GA, especially when it faces a complicated problem such as FJSP. In conventional GA, solution exploration is carried out by random searching process. Even though that would be an advantage of GA so that it could explore solution domain well but some time, the random searching process could cause the GA has an ineffective searching process. It leads to long time in solution finding and poor solution quality.

This study elaborates the use of a knowledge-based system to avoid ineffective random searching process in conventional GA, so that it could find solution faster. The knowledge-based system is developed by extracting knowledge from an FJSP and it would be guidance for the GA to do effective random searching process. The improved GA is called Knowledge-Based GA (KB-GA). To show superiority of the proposed KB-GA in solving FJSP, a comparison study with conventional GA is also conducted.

MATERIALS AND METHODS

Flexible job shop knowledge-extraction: In this study, there are two basic knowledge of FJSP to be extracted. The first knowledge is maximum completion time of job in a WS could be minimised when every machine in a WS are working with same duration (same work load). Such knowledge could be explained by using following equations. Let say there are J machines in a WS, hence:

$$\text{Max}\left(\frac{\sum_{j}^{J}{}_{=1}t_j}{J}\right) < \text{Max}(C_j, \forall j, j=1,2,......,J) \tag{1}$$

Where:
j = job index

The second knowledge is in an FJSP with non-identical machines, to minimise makespan, a job must be assigned to a machine that could process the job with shortest processing time. Following equations explain such knowledge.

$$C_{j,m,k} = S_{j,m,k} + t_{j,m,k} \tag{2}$$

$$C_{j,n,k+1} = \text{Max}(C_{j,m,k}; C_{j-1,m,k+1)} \tag{3}$$

$$C_{j,n,k+1} = S_{j,n,k+1} + t_{j,n,k+1} \tag{4}$$

Where:
C = Completion time
S = Starting time
t = Processing time
m, n = Machine index
k = Stage index

Equation 2 above, if is longer, then would be higher and could be higher. Hence, probability to get longer makespan would be higher. Extracted knowledge then would be arranged in the form of IF-THEN rules and called as knowledge-based system. The knowledge-based system would control the GA in assigning the jobs to the machines when GA produces a new solution in initialisation stage, crossover and mutation operation.

GA modelling

Chromosome representation: In this study, assignment and sequence-based encoding is used to generate the chromosomes. Once a chromosome is generated, then assignment of jobs to the machines would be carried out according to the sequence of the jobs in the chromosome. Figure 1 shows example of the chromosome when there are 5 jobs and 2 WS in a FJSP.

If there are 3 machines in WS 1 and 2 machines in WS 2, then in WS 1, job 1, 3, 2 will be processed in machine 1, 2 and 3 respectively. Job 5 and 4 will be assigned sequentially to a machine with shortest work duration. In WS 2, machine 1 would process job 3 and job

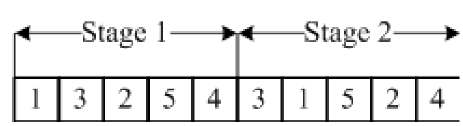

Fig. 1: Example of a chromosome

5 while machine 2 would process job 1 and job 2. Job 4 would be processed by a machine with shortest work duration. Such jobs assignment technique is carried out based on the 1st knowledge in the knowledge-based system.

Fitness function: By nature, GA is "blind", it doesn't know the problem to be solved. One and only one information used by GA as guidance to go to the right direction' is fitness value. Therefore, fitness function must be determined thoroughly. In relation with the problem, fitness function must be derived based on objective function of the problem. In this study, the objective function is to minimise makespan, hence, the fitness function must be derived from makespan function.

The nature of GA is keeping stronger chromosomes and improve it during evolution process until last generation. The stronger chromosomes are related to higher fitness value and it against the objective function. A conversion function is required to convert the objective function to be fitness function. In this study, inverse function is used for that purpose. Eq. 1 shows the fitness function of the chromosome.

$$f_r = \frac{1}{\text{makespan}} \tag{5}$$

$$\text{makespan} = \text{Max}(C_{j,k}\ j = 1, 2, 3, \ldots, J) \tag{6}$$

Where:
- f = Fitness function
- r = Chromosome index
- K = Number of WS
- J = Number of job

Crossover mechanism: In FJSP, because there are two decision variables, hence, it is not pure combinatorial problem. However, in this study, the chromosome is generated based on combinatorial problem so that crossover operators for combinatorial problem could be applied. Partially Mapped Crossover (PMX) which is one of the crossover that widely applied for combinatorial problem, is proposed in this study. Gen and Cheng (1997) for the detail mechanism of such crossover.

Mutation mechanism: A simple mutation for combinatorial problem, which is insertion, is proposed. Such mutation would select a locus from a chromosome randomly and insert it in random position. In order to avoid ineffective searching process, then the standard insertion mutation is modified by applying the second knowledge in the knowledge-based system. If a selected locus is inserted in a position that make machines in a WS are working with unbalance work load, then the mutation process would be repeated. Figure 2 explains the mechanism of that mutation operation.

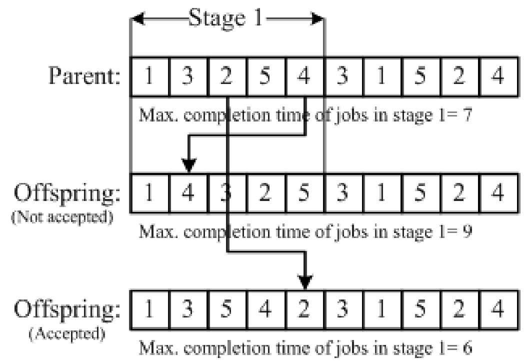

Fig. 2: Knowledge-based insertion mutation mechanism

Figure 2 even though minimising maximum completion of jobs in a WS is just local optimum, however, it is potential to minimise makespan.

RESULTS AND DISCUSSION

A case study: To test the proposed KB-GA, a case study is presented. In the production shop floor, there are 3 WS as shown in Fig. 3a. In a production cycle, there are 7 parts to be produced and production routing of each part is depicted in Fig. 3b. Table 1 shows the processing time of each part in each machine in every WS. The KB-GA has been ran for 1000 generations, with crossover rate is 0.2 and mutation rate is 0.1 and it provides a solution for the FJSP. Gantt chart of the solution is shown in Fig. 4 bellow.

Based on Fig. 4 it could be analysed visually that work load of machines in a WS is almost same. Table 2 shows detail work load analysis of machines in every WS based on standard deviation value. It could be said fairly that the proposed KB-GA is able to balance the work of all machines in every WS. To show superiority of the proposed KB-GA compared to conventional GA, there are two parameters that has been used which are solution quality based on makespan and speed to find solution based on number of generation required by GA to find the solution. Both GA have been ran 10 times for 500 generations and Table 3 shows the performance comparison between them.

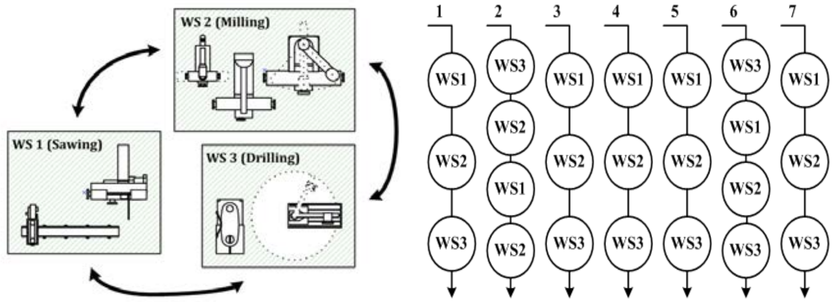

Fig. 3: a) Layout of the flexible job shop system; b) Operations process chart of the parts

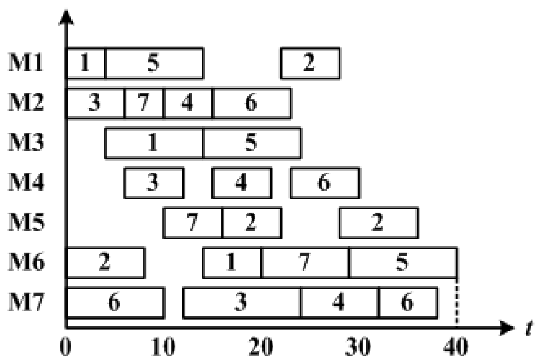

Fig. 4: Gantt chart of the solution

Table 1: Processing time of parts in machines

Job	WS 1 (Sawing)		WS 2 (Milling)		WS 3 (Drilling)		
	M1	M2	M3	M4	M5	M6	M7
1	4	5	10	9	10	6	5
2	6	8	7 (8)	6 (7)	6 (8)	8	10
3	6	6	6	6	6	11	12
4	5	5	4	5	5	8	8
5	10	10	10	9	9	11	11
6	7	8	7	7	7	10 (6)	10 (5)
7	3	4	6	6	6	9	8

( ) is the second process in that WS

Table 2: Work load of every machines

WS	Machine	Work load (total time)	SD
Sawing	M1	20	2.12
	M2	23	
Milling	M3	20	1.15
	M4	18	
	M5	20	
Drilling	M6	34	0.70
	M7	35	

With refer to Table 3 it could be seen that KB-GA could find solution with better makespan than conventional GA. Speed to find the solution of KB-GA is also relatively faster than conventional GA. There are 2 experiments, which are experiment no 2 and 9 that show conventional GA could find solution faster than KB-GA. However, in both experiments, conventional GA found solution with worse makespan compared to KB-GA.

Table 3. Performance comparison between KB-GA and conventional GA

Experiment	KB-GA		Conventional GA	
	Makespan	Convergence	Makespan	Convergence
1	40	121	45	235
2	40	100	42	95
3	40	154	40	397
4	42	125	48	350
5	40	195	46	257
6	40	100	45	232
7	40	94	49	376
8	40	127	50	321
9	40	132	44	125
10	40	145	42	339

CONCLUSION

Based on explanations above, it could be concluded, even though FJSP has 2 decisions variable but it could be modelled as combinatorial optimisation problem. GA could be one of algorithm to find the solution for FJSP. When the problem is complicated, in order to increase possibility of the GA in finding better solution, an improvement for the GA is still required. In this study, knowledge-based system could be the improvement tool for the GA and it is proven that the improved GA could find better solution when being used to solve an FJSP.

REFERENCES

Abdelmaguid, T.F., 2015. A neighborhood search function for flexible job shop scheduling with separable sequence-dependent setup times. Applied Math. Comput., 260: 188-203.

Birgin, E.G., J.E. Ferreira and D.P. Ronconi, 2015. List scheduling and beam search methods for the flexible job shop scheduling problem with sequencing flexibility. Eur. J. Operat. Res., 247: 421-440.

Brandimarte, P., 1993. Routing and scheduling in a flexible job shop by taboo search. Ann. Operat. Res., 41: 157-183.

Gao, J., L. Sun and M. Gen, 2008. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Comput. Operat. Res., 35: 2892-2907.

Gao, K.Z., P.N. Suganthan, M.F. Tasgetiren, Q.K. Pan and Q.Q. Sun, 2015a. Effective ensembles of heuristics for scheduling flexible job shop problem with new job insertion. Comput. Ind. Eng., 90: 107-117.

Gao, K.Z., P.N. Suganthan, T.J. Chua, C.S. Chong, T.X. Cai and Q.K. Pan, 2015b. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion. Expert Syst. Applic., 42: 7652-7663.

Gen, M. and R. Cheng, 1997. Genetic Algorithms and Engineering Design. John Wiley and Sons, New York, USA., ISBN-13: 9780471127413, Pages: 411.

Mokhtari, H. and M. Dadgar, 2015. Scheduling optimization of a stochastic flexible job-shop system with time-varying machine failure rate. Comput. Operat. Res., 61: 31-45.

Palacios, J.J., I. Gonzalez-Rodriguez, C.R. Vela and J. Puente, 2015. Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop. Fuzzy Sets Syst., 278: 81-97.

Xing, L.N., Y.W. Chen, P. Wang, Q.S. Zhao and J. Xiong, 2010. A knowledge-based ant colony optimization for flexible job shop scheduling problems. Applied Soft Comput., 10: 888-896.

Zhang, G., X. Shao, P. Li and L. Gao, 2009. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Comput. Indus. Eng., 56: 1309-1318.