JIRAE, Vol. 1, No. 1, September 2016, 25-32
e-ISSN 2407-7259
DOI: 10.9744/JIRAE.1.1.25-32

Application of Genetic Algorithm on Spare Part Automotive Body Scheduling

Rendy[a], Anastasia Lidya Maukar[b], Arthur Silitonga[c]

President University, Jl. Ki Hajar Dewantara, Jababeka Education Park, Cikarang, Indonesia
[a]rendy.yuan@gmail.com, [b]almaukar@gmail.com, [c]arthur@president.ac.id

Abstract. Job shop scheduling is one of the complex problems in the manufacturing industry, such as an automotive body manufacturer. This manufacturing company, located in Cikarang, Indonesia, deals with huge customer demand that leads to difficulties in production scheduling. This will cause a delay in some jobs and product deliveries. The current system is using the semi-active scheduling approach and requires 637 minutes for performing 6 jobs with 5 machines. The genetic algorithm (GA) model is proposed as an alternative solution to solve this problem. The GA parameter is set as follow: The population size expected is 30 with maximum generation can be produced in amount of 50. The crossover rate, mutation, and preservation are set to 0.3, 0.1, and 0.1, respectively. After 50 generations are obtained, the optimum solution is shown in generation 6 with a makespan of 597 minutes. Thus, the genetic algorithm model is effectively reducing the makespan of the job-shop scheduling problem by 10% compared to the current method applied at the company.

Keywords: Genetic Algorithm, Job-shop, Makespan, Proposed, Scheduling.

1. Introduction

In doing business, a manufacturing company has several targets that need to be achieved. Among the targets are the following categories: maximizing production output, fulfilling customer's satisfaction in terms of product quality and delivery time, and minimizing the cost [1]. In simple terms, the targets pertain to output, quality, time, and cost. In order to achieve those targets, good scheduling is needed for the production process.

Scheduling is a decision making tool for the company to deal with customer demand and production process. Furthermore, scheduling is the process of organizing, selecting, and determining the time by considering the resources to do the activity or operation. Scheduling is set by considering the situation and condition of the company [1]. Based on process flow, there are two types of scheduling which are flow shop scheduling and job shop scheduling. Most automotive or spare part companies are using the job shop scheduling type.

Job shop is defined as a flow of production process with each routing being unique for all jobs. There are several constraints on jobs and machines such as [2]: a job does not produce on the same machine twice, there is no precedent constraint among operations of different jobs, operations are fixed, each machine is only processing one job at a time, and release time or due dates are specified.

Makespan is the time needed to process all jobs starting from the first process until the end of the process. Good scheduling and sequence of jobs will decrease the makespan. Job Shop Scheduling problems are categorized into combinatorial optimization problems and commonly solved using meta-heuristic methods [3]. Those meta-heuristic methods are Ant Colony System, Artificial Immune Systems, Consultant Guided Search, Genetic Algorithm, etc. Based on the research of others [4] in solving a certain job shop scheduling problem, Genetic Algorithm was chosen among the meta-heuristic methods to improve the performance of makespan used in the scheduling problem.

The aim of this research is to propose a new method, i.e. the Genetic Algorithm in creating the job shop schedule in the making of spare parts for the automotive body's process for minimizing makespan. The significance of the proposed model is done by comparing the Genetic Algorithm Scheduling to the current scheduling plan in term of makespan minimization. This case study is taken from an automotive body manufacturer, which is located in Cikarang, with selected jobs and machines. Due to the request of the manufacturer, the company can only be referred to as "the manufacturing company", "the company", or "automotive body manufacturer" to maintain its confidentiality.

The manufacturing company produces spare parts for the automotive bodies, such as SPRT Comp (J1), BU Arm L (J2), BU Arm R (J3), IGP (J4), Stiffener (J5), and Leg RD (J6). As one of the spare part suppliers, the company has to meet customer demand. With several types of customers and quantity of demand, the mentioned company has to cover the orders with the available resources. Based on initial observation, the company cannot meet the demand on time.

Each job goes through several different processes. The machines and processing times are varying and become the obstacle in delivering the product. Due to the lack in scheduling system, to finish 6 jobs with 5 machines requires 11 hours' production process and it causes a delay in some jobs and deliveries. Thus, the selected method is applied to solve the Job Shop problem at the company. The effective approach could optimize the production process while increasing the productivity of the machine and minimizing the makespan. Recently, Genetic Algorithm is well-known as an optimization technique to solve complex problems and has been successfully applied in the Industrial Engineering arena [2].

Many types of problems can be solved using this algorithm including vehicle routing, facility layout, trans-

porttation, as well as scheduling and sequencing. Based on the GA process, the algorithm obtains the optimum result from all the possibilities of the population. The genetic algorithm is being proposed to enhance the performance of the current job shop scheduling method used in the company, so the makespan of the job shop scheduling can be minimized.

Genetic Algorithm is defined as a search technique based on natural selection and natural genetics [2]. The Genetic Algorithm starts with initial random solutions called *population*. This population can be done randomly or using algorithm. Each individual in the population represents a solution to the problem, called a *chromosome*. Later, this chromosome *evolves* through several iterations, called *generations*. In the generations, the chromosomes are *evaluated* using some measures of *fitness*. *Offspring* is the next generation or new chromosome, which is obtained by *crossover* (mating) or *mutation* (modifying). A new generation is obtained by selecting the parents and offspring and rejecting others, so the population is kept constant. After several generations, the algorithms gather to the best chromosome, which represent the optimum solution.

The six components in Genetic Algorithm are: Coding Technique, Initialization Procedure, Evaluation Function, Selection, Genetic Operator, and Parameters Determination. Mostly, initialization procedures are done randomly. The selection procedure used is the roulette wheel. The genetic operator used is a one-point crossover and job-based mutation. The Breeder's GA (BGA) is used as proposed by Mühlenbein. For building a genetic algorithm, an appropriate representation of solutions together with problem-specific genetic operations should be obtained. This aims to produce feasible schedules on generation of all chromosomes [2].

All of the six components are implemented in a certain application created using MATLAB Software. Therefore, data from the company are processed using our application, and they are compared at the end to the results of the existing job shop scheduling used at the company.

2. Research Methods

In the Genetic Algorithm, the two types of operations are Genetic Operations and Evolution Operations. Genetic operations consist of crossover and mutation. It creates the new offspring at each generation. While the Evolutions operations (selection) follows the process of Darwinian evolution to create populations from generation to generation [5]. Genetic algorithms have several methods in searching procedure [6]. These methods are as follows:

- Genetic algorithm works with a coding of solution set, not the solutions themselves.
- Genetic algorithm searches from a population of solutions, not a single solution.
- Genetic algorithm uses payoff information (fitness function), not derivatives or other auxiliary knowledge.
- Genetic algorithm uses probabilistic transition rules, not deterministic rules.

The method used for this research is job shop scheduling using the Genetic Algorithm approach. The semi-active scheduling is done as a requirement of the manufacturing company. A schedule is semi-active if no *left-shift* is evident and the operation sequence is not changed. A dispatching rule that is made a priority to be applied is the First Come First Serve (FCFS) rule. This rule makes the operation which enters the station first a priority.

The priority-ruled-based genetic algorithm is proposed by the following process [7]: A chromosome is encoded as a sequence of dispatching rules for job assignment and the schedule is constructed with priority dispatching heuristics. Genetic Algorithm is used to evolve these chromosomes to produce a better sequence of dispatching rules.

For an *n*-job and *m*-machine problem, a chromosome is set by a string of $n \times m$ entry ($p_1, p_2 \ldots p_{nm}$). An entry p_i represents one rule of priority dispatching rules [2]. The entry in i^{th} position says that a conflict in the i^{th} iteration should be resolved using priority rule p_i. The following notation is used to generate genetic algorithm for the priority rule:

PS_t = a partial schedule containing t scheduled operations
S_t = the set of schedulable operations at stage t, corresponding to a given PS_t
$Ó_i$ = the earliest time at which operation $i \in S_t$ could be started
$Ø_i$ = the earliest time at which operation $i \in S_t$ could be completed
C_t = the set of conflicting operations in iteration t

The procedures to deduce a schedule from given chromosome ($p_1, p_2 \ldots p_{nm}$) are as follows:

Step 1: Let $t=1$, and begin with PS_t as the null partial schedule. Initially, S_t includes all operations with no predecessors.

Step 2: Determine $Ø_t^* = \min i \in S_t \{Ø_i\}$ and the **machine** m^* on which $Ø_t^*$ could be realized. If more than one machine exists, the random choice is obtained.

Step 3: For each operation $i \in S_t$ that **requires machine** m^* and for which $Ó_i < Ø_i^*$, calculate a priority index according to the specific rule. Find the operation with the smallest index and add this operation to PS_t as early as possible, thus creating only one partial schedule, PS_{t+1}, for the next stage.

Step 4: For the new partial schedule PS_{t+1}, created in step 3, update the data set as follows:
   a. Remove operation i from S_t
   b. Form S_{t+1} by adding the direct successor of operation i to S_t
   c. Increment t by one

Step 5: Return to step 2 for the PS_{t+1} created in step 3 and step 4 until a completed schedule is generated.

The population size expected is 30. Table 1 shows the initial population of the proposed model. The population is done randomly with identical order. This means there is no similar job order in the population. In addition, the makespan of each chromosome is computed and the fitness value is shown. Based on Table 1, the maximum and minimum makespan are 550 (Individual 30) and 368 (Individual 17). This means the best fitness for initial population is Individual 17.

Table 1. Initial Chromosome and Objective Function

Individual	Chromosome						Cmax	Fitness Value	Pb	Pbc
1	4	1	6	5	3	2	432	0.0023148	0.036503	0.036503
2	4	2	6	1	3	5	510	0.0019608	0.030921	0.067424
3	2	1	6	3	4	5	516	0.0019380	0.030561	0.097985
4	4	3	5	6	2	1	534	0.0018727	0.029531	0.127520
5	5	2	4	3	6	1	514	0.0019455	0.030680	0.158200
6	2	6	1	4	3	5	506	0.0019763	0.031165	0.189360
7	4	6	3	1	2	5	478	0.0020921	0.032991	0.222350
8	5	6	4	1	3	2	422	0.0023697	0.037368	0.259720
9	5	2	6	3	4	1	514	0.0019455	0.030680	0.290400
10	4	1	5	2	3	6	384	0.0026042	0.041066	0.331470
11	5	4	2	6	3	1	534	0.0018727	0.029531	0.361000
12	6	4	2	5	3	1	486	0.0020576	0.032448	0.393440
13	4	3	1	6	5	2	532	0.0018797	0.029642	0.423090
14	3	6	5	2	1	4	480	0.0020833	0.032853	0.455940
15	6	3	1	2	5	4	482	0.0020747	0.032717	0.488660
16	2	3	6	5	4	1	412	0.0024272	0.038275	0.526930
17	6	5	1	2	3	4	368	0.0027174	0.042852	0.569780
18	2	5	4	1	3	6	474	0.0021097	0.033269	0.603050
19	4	2	3	6	5	1	416	0.0024038	0.037907	0.640960
20	3	6	1	2	4	5	492	0.0020325	0.032052	0.673010
21	3	6	1	5	4	2	504	0.0019841	0.031289	0.704300
22	4	2	1	6	3	5	534	0.0018727	0.029531	0.733830
23	2	4	6	5	1	3	452	0.0022124	0.034888	0.768720
24	5	4	1	6	2	3	450	0.0022222	0.035043	0.803760
25	2	3	4	5	1	6	384	0.0026042	0.041066	0.844830
26	4	6	2	1	3	5	478	0.0020921	0.032991	0.877820
27	5	3	4	2	6	1	514	0.0019455	0.030680	0.908500
28	6	3	4	5	2	1	500	0.0020000	0.031539	0.940040
29	2	6	5	4	1	3	504	0.0019841	0.031289	0.971330
30	5	4	2	1	6	3	550	0.0018182	0.028672	1.000000
Total								0.0634140		

Figure 1 shows the flowchart of Genetic Algorithm in the job shop scheduling problem. The initialization of population is random. The objective function is determined to minimize makespan and the fitness function is based on objective function. The selection process is done by using the roulette-wheel selection, thus, the candidates of parents are selected. The operator of genetics is using crossover (one-point crossover) and mutation (order changing mutation). Then, the preservation process is done after the mutation process. The result of the preservation process will be the population of the following generation. The routing of the algorithm is repeated until the optimization (makespan GA < current) and the stopping criteria (number of generation = maximum generation) are achieved. The flowchart shows the procedure in doing genetic algorithm until the model is created.

3. Result and Discussion

3.1 Current Model

The sample data is taken from Line A of an automotive body manufacturer. This research focuses on Line A, which manufactures piping products. The production order, schedule delivery, and MRP are based on 26th August, 2015. The delivery time of finished goods is always one day after the production order is released, so the due date term is not used. This research assumes there is no machine breakdown. Also, there is no machine that is specifically used for certain jobs or products. All machines can be used by all products based on its process routing. In addition, the moving time from one machine to another is neglected; it is assumed to be zero. This research is applied to 6 jobs and 5 machines. The operation sequences and processing times are shown in Table 2 and Table 3 respectively. The priority scheduling is based on the first come first serve rule.

Table 2. Operation Sequence

Job	Operation				
	1	2	3	4	5
Job 1	M1	M2	M5	M3	M4
Job 2	M5	M1	M3	M2	M4
Job 3	M5	M1	M3	M2	M4
Job 4	M1	M3	M2	M4	M5

Table 3. Processing Time

Job	Machine [seconds]				
	1	2	3	4	5
Job 1	1,600	1,600	1,600	1,600	1,600
Job 2	2,800	2,800	2,800	4,000	6,000
Job 3	2,800	2,800	2,800	4,000	6,000
Job 4	2,000	1,200	1,600	1,600	1,600
Job 5	1,200	1,000	1,600	1,600	2,000
Job 6	1,000	800	1,400	800	1,600

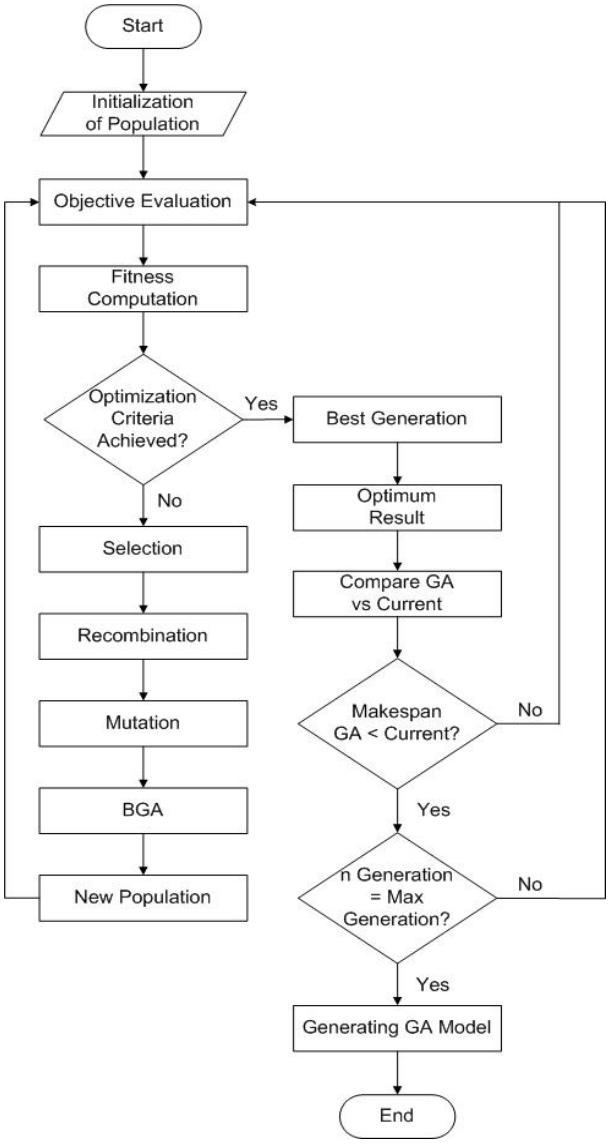

Figure 1. Genetic Algorithm Flowchart

Proposed Model using GA

The current data of job shop scheduling is calculated using the genetic algorithm model. The objective of the genetic algorithm model is to find the best job order sequence which gives the minimum makespan. The processing time is the multiplication result of cycle time and customer demand (order quantity). Since each operation is done by particular machines with different processing times, the problem is called a Non-deterministic Polynomial-time hard (NP-hard) problem [8].

The parameter of the current model is being set as recommended by De Jong (1975) [9]. The population size expected is 30 and the maximum generation can be produced in amount of 50. The crossover rate, mutation, and preservation are 0.3, 0.1, and 0.1, respectively. The makespan and fitness value of each chromosome on the first generation can be obtained. The maximum and minimum makespan are 550 (Individual 30) and 368 (Individual 17). This means the best fitness of initial population is from Individual 17. The fitness value is calculated using Eq. 1.

$$F_n = \frac{1}{Makespan} \quad (1)$$

The selection process is performed using the roulette-wheel method. The Roullete Wheel Selection is chosen because it gives the best fitness value compared to the other selection methods, such as: Random Selection and Elitist Selection [10]. The probability of fitness (*Pb*) is obtained by dividing the fitness value with total fitness value. The total fitness value is 0.0634140. The probability of Individual 1 selected is 0.036503 or 3.65%. The probability cumulative (*Pbc*) is the cumulative score of probability of fitness (*Pb*). Then, a random number (*Rs*) is generated. The value range of *Rs* is between 0 and 1. This random number is compared to the probability of fitness value cumulative (*Pbc*). The individual or chromosome n is selected if $Rs_n < Pbc_{n,\, n+1,\, n+2\, \ldots,\, n+k}$. The new chromosome *U* is obtained by replacing the initial population with the candidates of parents based on the selection process (roulette-wheel).

The random number of a crossover process (*Rc*) is generated and compared with the crossover rate (*pc*). Based on parameter, the *pc* value is 0.3 which means the individual that has *Rc* value less than 0.3 will be selected as the Parent of the Crossover. There are eight individuals with an *Rc* value less than *pc*, which are: Individual 1, Individual 4, Individual 5, Individual 8, Individual 13, Individual 18, Individual 19, and Individual 26. Another random number is generated to determine the crossing point (*Rcp*). The new chromosome is obtained after the crossover process named a V. The example of the crossover process is shown in Figure 2.

Rcp[1] = 2

Parent 1	5	1	4	3	2
Parent 2	4	5	1	2	3
Offspring 1	5	1	4	2	3

Rcp[2] = 2

Parent 1	4	5	1	2	3
Parent 2	5	1	4	3	2
Offspring 2	4	5	1	3	2

Figure 2. The Example of Crossover Process

The result of the crossover becomes the target of the mutation process. The random number of *Rm* is generated as the parameter to select the candidate from chromosome V. If *Rm* is less than the mutation rate (*pm* = 0.1), then the chromosome will be selected. Individual 2, Individual 10, Individual 19, and Individual 22 are the mutation candidates. Two random numbers (*Rmp*) are generated to determine the mutation point. For instance, Individual 2 has *Rmp* value 1 and 4 which means gene number 1 and gene number 4 in chromosome 2 are switched.

The Breeder Genetic Algorithms or BGA is conducted to select the chromosome that will be replaced by the best fitness among chromosome W. The best fitness among chromosome W is Individual number 13 (2-3-4-6-1-5) with the fitness value of 0.0027778. The random number of the preservation process (*Rb*) is generated and the preservation rate (*kb*) is used as the parameter. If the preservation process *Rb* is less than *kb* or 0.1, then the chromosome will be replaced by Individual 13 (2-3-4-6-1-5). Individual 2, Individual 21, and Individual 22 are selected and being replaced by Individual 13. The result of the BGA process is symbolized by X. The chromosome among X is the last chromosome for this first generation. An evaluation is

needed to determine the best fitness and the worst fitness for the first generation. The best fitness is Individual 2, Individual 13, Individual 21, and Individual 23. One of best individuals is chosen; and then Individual 2 will be the optimum solution for the first generation. Furthermore, the chromosome X is the parent for the second generation and so on. The example of the chromosome on the first generation is shown in Appendix A.

The genetic algorithm process will be repeated from the selection to the BGA after the first generation is obtained. The Maxgen Parameter is used as the stopping criteria. The proposed model sets the maximum generation to 50; this means there will be 50 times looping on the algorithm. It shows the best fitness is 0.0028 starting from the first generation. The worst fitness is fluctuating from 0.0018 to 0.0020. The extreme worst value is shown in generation 19 which is 0.0024.

3.3 Discussion

Based on the genetic algorithm results, the job sequence with the lowest makespan is 2-3-4-6-1-5 on Generation 6. This job sequence gives the best fitness score of 0.0028409 or the makespan of 597 minutes. The Gantt Chart of the proposed model using genetic algorithm can be seen in Figure 3.

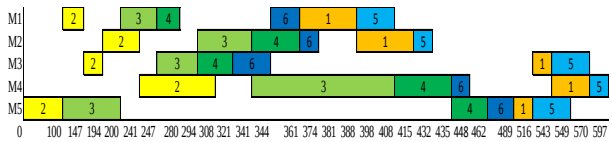

Figure 3. The Gantt Chart of Our Proposed Method

The current system and the proposed model are compared based on the makespan. The current model requires 637 minutes while the proposed model requires only 597 minutes. This means the makespan can be reduced by 10%. In addition, the lateness of the jobs can be reduced. The current system took 11 hours to produce 6 products, whereas the proposed model only takes 10 hours. It means the production process is faster by 1 hour and will impact the production delivery. It proves the genetic algorithm model is better in terms of computation of the job shop scheduling problem and the output (makespan).

The fitness value of each generation is shown in Figure 4. The graph shows the constant value for the first 6 generations and it increases on Generation 7 until the maximum generation (50). Because of the BGA process, the maximum fitness is obtained faster. The minimum fitness obtained for each generation shows the fluctuate score. In this research, the average value of each generation is neglected. The 50 generations can be seen in Table 4.

The verification and validity test are used to check whether or not the logic function is correct and gives an optimum solution. The validity test is done by using T-test (Paired Difference Test). The makespan of random job order is calculated using current system and software computation, and then these computations are compared. The verification is completed by checking the operation routing of scheduling. The selected job order is checked; if there is no overlapping job order, the model is assumed to be correct.

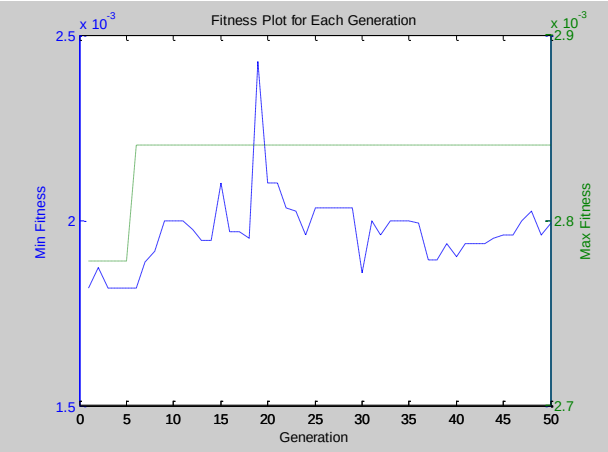

Figure 4. Fitness Plot of Each Generation

Table 5 shows the makespan result of the current system and numerical computation software after running randomly 10 times. The statistic test is required to validate this system. The hypothesis of the statistic test is:

H_0 : $\mu_D = 0$; there is no difference between these models

H_1 : $\mu_D \neq 0$; there is at least one difference between these models

Paired T-Test is being used to test the validity of the model. The p-value of the result is very high (*), which means the null hypothesis should be accepted. To conclude, there is no difference on the current system and the numerical computation software model. The result of the statistical software of the current system and the software model is shown in Appendix B.

The first run is selected to be verified; the job order is 2-4-5-1-3 ($Cmax$ = 105). The makepsan computation is done by Matlab software and shown in Appendix C. It shows that there is no overlapping of the job and the machine. For instance, operation 1 on job 1 (Machine 3) starts at time 54 after operation 2 on job 5 (Machine 3) finishes at time 53.

4. Conclusion

Since authors were requested by an automotive body manufacturer to find and propose a scientific method which may enhance the performance of its existing job shop scheduling method, we proposed the application of genetic algorithm for minimizing the makespan of the current job shop scheduling method used at the company.

The genetic algorithm model is successfully done and the objectives of the research can be obtained, i.e. to propose a certain new job shop scheduling method to the company, and to prove by using the application of the company's real data that the proposed method is able to minimize the makespan of the current job shop scheduling method. The problem of the current system is that scheduling 6 jobs on 5 machines are done manually. Using semi-active scheduling and the first come first serve priority rule, the current system requires 637 minutes. Meanwhile, the proposed model of genetic algorithm requires only 597 minutes. This means the makespan can be reduced by 10%. It proves the genetic algorithm model is better in terms of output result (makespan).

Table 4. Summary of Each Generation

Gen	Chromosome	Best F	Worst F	Gen	Chromosome	Best F	Worst F
1	2-3-4-6-1-5	0.0028	0.0018	26	2-3-4-6-5-1	0.0028	0.0020
2	2-3-4-6-1-5	0.0028	0.0019	27	2-3-4-6-5-1	0.0028	0.0020
3	2-3-4-6-5-1	0.0028	0.0018	28	2-3-4-6-5-1	0.0028	0.0020
4	2-3-4-6-1-5	0.0028	0.0018	29	2-3-4-6-5-1	0.0028	0.0020
5	2-3-4-6-1-5	0.0028	0.0018	30	2-3-4-6-5-1	0.0028	0.0019
6	2-3-4-6-1-5	0.0028	0.0018	31	2-3-4-6-5-1	0.0028	0.0020
7	2-3-4-6-1-5	0.0028	0.0019	32	2-3-4-6-5-1	0.0028	0.0020
8	2-3-4-6-1-5	0.0028	0.0019	33	2-3-4-6-5-1	0.0028	0.0020
9	2-3-4-6-1-5	0.0028	0.0020	34	2-3-4-6-5-1	0.0028	0.0020
10	2-3-4-6-1-5	0.0028	0.0020	35	2-3-4-6-5-1	0.0028	0.0020
11	2-3-4-6-1-5	0.0028	0.0020	36	2-3-4-6-5-1	0.0028	0.0020
12	2-3-4-6-5-1	0.0028	0.0020	37	2-3-4-6-5-1	0.0028	0.0020
13	2-3-4-6-5-1	0.0028	0.0019	38	2-3-4-6-5-1	0.0028	0.0019
14	2-3-4-6-5-1	0.0028	0.0019	39	2-3-4-6-5-1	0.0028	0.0019
15	2-3-4-6-5-1	0.0028	0.0021	40	2-3-4-6-5-1	0.0028	0.0019
16	2-3-4-6-5-1	0.0028	0.0020	41	2-3-4-6-5-1	0.0028	0.0019
17	2-3-4-6-5-1	0.0028	0.0020	42	2-3-4-6-5-1	0.0028	0.0019
18	2-3-4-6-5-1	0.0028	0.0020	43	2-3-4-6-5-1	0.0028	0.0019
19	2-3-4-6-5-1	0.0028	0.0024	44	2-3-4-6-5-1	0.0028	0.0020
20	2-3-4-6-5-1	0.0028	0.0021	45	2-3-4-6-5-1	0.0028	0.0020
21	2-3-4-6-5-1	0.0028	0.0021	46	2-3-4-6-5-1	0.0028	0.0020
22	2-3-4-6-5-1	0.0028	0.0020	47	2-3-4-6-5-1	0.0028	0.0020
23	2-3-4-6-5-1	0.0028	0.0020	48	2-3-4-6-5-1	0.0028	0.0021
24	2-3-4-6-5-1	0.0028	0.0020	49	2-3-4-6-5-1	0.0028	0.0020
25	2-3-4-6-5-1	0.0028	0.0020	50	2-3-4-6-5-1	0.0028	0.0020

Table 5. Statistic Data for Validity Test

Run	Job Order	Makespan in t unit (Cmax)		Difference (Di = x-y)
		Current System (x)	Software Model (y)	
1	2-4-5-1-3	105	105	0
2	3-1-5-4-2	114	114	0
3	3-5-2-4-1	107	107	0
4	2-1-5-4-3	117	117	0
5	3-4-5-1-2	94	94	0
6	1-2-5-4-3	105	105	0
7	1-5-3-4-2	100	100	0
8	4-1-2-5-3	93	93	0
9	3-5-2-1-4	101	101	0
10	2-4-3-1-5	118	118	0

Further research can be done for finding the best combination of GA parameter in order to improve optimally the performance of the current job shop scheduling method viewed from its makespan. The experimental design can be employed to measure the effects of parameters to the model output (makespan). In addition, the factors that influencing time consumption in computing the scheduling can be obtained.

References

1. Morton, T. and Pentico, D., *Heuristic Scheduling Systems*, John Wiley Interscience, New York, 1993.
2. Gen, M. and Cheng, R., *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.
3. Maghfiroh, M.F.N., Darmawan, A., and Yu, V.F., Genetic Algorithm for Job Shop Scheduling Problem: A Case Study, *International Journal of Innovation, Management and Technology*, 4(1), 2013, pp. 137-139.
4. Omar, M., Baharun, A., and Hasan, Y.A., A Job-Shop Scheduling Problem (JSSP) Using Genetic Algorithm (GA), *Proc. of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications*, Universiti Sains Malaysia, Penang, 2006.
5. Holland, J.H., *Adaptation in Natural and Artificial Systems*, second edition, MIT Press, Cambridge, 1992.
6. Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, 1989.
7. Dorndorf, U. and Pesch, E., Evolution Based Learning in a Job Shop Scheduling Environment, *Computers and Operations Research*, 22(1), 1995, pp. 25-44.

8. Mesghouni, K., Hammadi, S., and Borne, P., Evolutionary Algorithm for Job-Shop Scheduling, *International Journal of Applied Mathematics and Computer Science*, 14(1), 2004, pp. 91-103.
9. Patil, V.P. and Pawar, D.D., The Optimal Crossover or Mutation Rates in Genetic Algorithm: A Review, *International Journal of Applied Engineering and Technology*, 5(3), 2015, pp. 38-41.
10. Kapoor, M. and Wadhwa, V., Optimization of DE Jong's Function Using Genetic Algorithm Approach, *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, 1(1), 2012, pp. 14-17.

Appendix A: The Example of Chromosomes on the First Generation

Indv	Initial	U	V	W	X
1	416532	216345	215463	215463	215463
2	426135	254136	254136	154236	**234615**
3	216345	234516	234516	234516	234516
4	435621	542631	542631	542631	542631
5	524361	541623	541623	541623	541623
6	261435	421635	421635	421635	421635
7	463125	254136	254136	254136	254136
8	564132	642531	623541	623541	623541
9	526341	431652	431652	431652	431652
10	415236	365214	365214	362514	362514
11	542631	431652	431652	431652	431652
12	642531	463125	463125	463125	463125
13	431652	236541	234615	234615	234615
14	365214	426135	426135	426135	426135
15	631254	651234	651234	651234	651234
16	236541	426135	426135	426135	426135
17	651234	462135	462135	462135	462135
18	254136	462135	462135	462135	462135
19	423651	542631	542631	541632	541632
20	361245	542163	542163	542163	542163
21	361542	265413	265413	265413	234615
22	421635	542163	542163	542136	542136
23	246513	524361	524361	524361	234615
24	541623	416532	416532	416532	416532
25	234516	564132	564132	564132	564132
26	462135	542631	542631	542631	542631
27	534261	564132	564132	564132	564132
28	634521	246513	246513	246513	246513
29	265413	564132	564132	564132	564132
30	542163	365214	365214	365214	365214

Appendix B: Paired T-Test of Model Validity

Paired T-Test and CI: Current System (x); Software Model (y)
Paired T for Current System (x) - Software Model (y)

	N	Mean	StDev	SE Mean
Current System (x)	10	105.4	8.83	2.79
Software Model (y)	10	105.4	8.83	2.79
Difference	10	0	0	0

95% CI for mean difference: (0.000000; 0.000000)
T-Test of mean difference = 0 (vs ≠ 0): T-Value = * P-Value = *
* NOTE * All values in column are identical.

Appendix C: Makespan Computation on Matlab for Verification

Job 2	Job 4	Job 5	Job 1	Job 3
job=2,mc:2=6	job=4,mc:4=4	job=5,mc:5=5	job=1,mc:3=2	job=3,mc:1=7
M=2,1=2	M=4,21=4	M=5,42=5	**M=3,54=1**	**M=1,65=3**
M=2,2=2	M=4,22=4	M=5,43=5	M=3,55=1	M=1,66=3
M=2,3=2	M=4,23=4	M=5,44=5		M=1,67=3
M=2,4=2	M=4,24=4	M=5,45=5		M=1,68=3
M=2,5=2		M=5,46=5		M=1,69=3
M=2,6=2				M=1,70=3
				M=1,71=3
job=2,mc:3=5	job=4,mc:3=5	job=5,mc:3=7	job=1,mc:1=8	job=3,mc:5=8
M=3,7=2	M=3,25=4	M=3,47=5	**M=1,57=1**	M=5,82=3
M=3,8=2	M=3,26=4	M=3,48=5	M=1,58=1	M=5,83=3
M=3,9=2	M=3,27=4	M=3,49=5	M=1,59=1	M=5,84=3
M=3,10=2	M=3,28=4	M=3,50=5	M=1,60=1	M=5,85=3
M=3,11=2	M=3,29=4	M=3,51=5	M=1,61=1	M=5,86=3
		M=3,52=5	M=1,62=1	M=5,87=3
		M=3,53=5	M=1,63=1	M=5,88=3
			M=1,64=1	M=5,89=3
job=2,mc:5=4	job=4,mc:2=5	job=5,mc:1=3	job=1,mc:2=4	job=3,mc:4=4
M=5,12=2	M=2,30=4	M=1,54=5	M=2,65=1	M=4,90=3
M=5,13=2	M=2,31=4	M=1,55=5	M=2,66=1	M=4,91=3
M=5,14=2	M=2,32=4	**M=1,56=5**	M=2,67=1	M=4,92=3
M=5,15=2	M=2,33=4		M=2,68=1	M=4,93=3
	M=2,34=4			
job=2,mc:1=3	job=4,mc:1=4	job=5,mc:2=6	job=1,mc:4=6	job=3,mc:3=9
M=1,16=2	M=1,35=4	M=2,57=5	M=4,69=1	M=3,94=3
M=1,17=2	M=1,36=4	M=2,58=5	M=4,70=1	M=3,95=3
M=1,18=2	M=1,37=4	M=2,59=5	M=4,71=1	M=3,96=3
	M=1,38=4	M=2,60=5	M=4,72=1	M=3,97=3
		M=2,61=5	M=4,73=1	M=3,98=3
		M=2,62=5	M=4,74=1	M=3,99=3
				M=3,100=3
				M=3,101=3
				M=3,102=3
job=2,mc:4=2	job=4,mc:5=3	job=5,mc:4=4	job=1,mc:5=7	job=3,mc:2=3
	M=5,39=4	M=4,63=5	M=5,75=1	M=2,103=3
M=4,19=2	M=5,40=4	M=4,64=5	M=5,76=1	M=2,104=3
M=4,20=2	M=5,41=4	M=4,65=5	M=5,77=1	M=2,105=3
		M=4,66=5	M=5,78=1	
			M=5,79=1	
			M=5,80=1	
			M=5,81=1	