Improved Genetic Algorithm for Flow Shop Scheduling Problem at PT. XYZ

Widyawati[1*], **G Waliadi**[1]

[1]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Banten Jaya, Banten, Indonesia.

*widyawati@unbaja.ac.id

Abstract. Scheduling according to Takeshi Yamada was an allocation of limited resources in order to do compete activity which usually applied on the operational research field where one of the examples was machine scheduling. Machine scheduling is an activity consisting of job (represent the activity) and machines (represent the resource) where each machine can process only one activity at a time. the purpose of scheduling is to perform all of the process from beginning to end with the objective to minimize makespan time. PT. XYZ is a manufacturing company that produces component for steam turbine component products in power plants. Within a year the company obtained an average of 9-10 project orders for each type of product. Because of the large number of projects that must be done, the scheduling process is becoming one of critical process in order to achieve the fastest makespan time. One way to find the fastest makespan time is by rescheduling using the methods applied in flow shop scheduling. One algorithm that can be used is the genetic algorithm. Based on the results of the research, it was found that for the process of scheduling project "A" with six machines and five jobs is makepan time of 53 days.

1. Introduction

Scheduling according to Takeshi Yamada was an allocation of limited resources in order to do compete activity which usually applied on the operational research field where one of the examples was machine scheduling. Machine scheduling is an activity consisting of job (represent the activity) and machines (represent the resource) where each machine can process only one activity at a time. the purpose of scheduling is to perform all of the process from beginning to end with the objective to minimize makespan time [1]

PT. XYZ is a manufacturing company that produces steam turbine component products in power plants. The range of the products include CND, CC, LIC, LOC, BLR. There are six machines used in the company, namely cutting machines, grinding machines, turning, cladding, rolling, and blasting. Within a year the company obtained an average of 9-10 project orders for each type of product. Because of the large number of projects that must be done, the scheduling process is one of the critical things in order to have a schedule with the fastest makespan time. One way to find the fastest makespan time is by rescheduling using the methods applied in flow shop scheduling.

Flow Shop scheduling is a scheduling problem that takes into account the number of m different machines and n jobs. Where every job consists of m operations, and each operation requires a different machine, and each job is processed in the same processing sequence [2]. Johnson in 1954 was one of the predecessors who conducted research on the problems of flow shop. He explained the completion of how to easily solve the flow shop problem which consisted of two machines with makespan time as

the main criteria. Since then, several studies have focused on how to solve the flow shop problem of m-machine (m> 2) with the criteria of makespan time as the main benchmark.

2. Method

The following is an explanation on the method and algorithm which underline this research.

2.1. Flow Shop Scheduling

Flow Shop scheduling was an important problem which is often happen on the production management area, which is consist of *m* machines and *n* job [3], each job consist of several operation that need to process in different machine. The problem on flow shop scheduling first introduced on early works of Johnson in 1954.

The pivotal influence of his works has had some definite advantages, first by emphasizing the properties of permutation schedules; he focused flow shop research on problems of manageable size. Second, the two machine Second, analysis seemed to have captured the essence of larger problems. Thus, where Johnson's rule provided a basis for the development of a heuristic approach to larger problems (as in Gupta and CDS procedures) it was remarkably successful [4]. Each job consists of m operations that must be processed on different machines.

2.2. Genetic algorithm

Genetic algorithms was a term from biology. In term of biology, gen was a good offspring inherit from its parents (good parent produces better offspring). This concept has been adopted into the solving of Genetic algorithms (GA) [5].

Genetic algorithms in particular are applied as computer simulations where a population of abstract representations (chromosomes) of prospective (individual) solutions on a problem of optimization will develop into better solutions. Traditionally, every solution was symbolized in binary as string '0' and '1', even though it's also possible to symbolize with different encoding. Evolution start from a random individual population from one generation to another generation. In every generation, overall ability of the population are being evaluated, then several individuals from current population was picked based on stochastic of their ability then the population was modified by mutation or recombination into new population for the next iteration of the algorithm.

2.3. Pseudocode

The following pseudocode is used based on research conducted by Murata [6]:

(Initialization) Randomly generate an initial population *P*1 of *Npop* strings (i,e., *Npo* solutions).
(Selection) Select *Npop* pairs of strings from a current population according to the selection probability.
(Crossover) Apply crossover operator to each of the selected pairs in Step 2 to generate *Npop* solutions with the crossover probability *Pc*. If the crossover operator is not applied to the selected pair, one of the selected solutions remains as a new string.
(Mutation) Apply mutation operator to each of the generated *Npop* strings with the mutation probability *Pm* (we assign the mutation probability not to each bit but to each string).
(Elitist Update) Randomly remove one string from the current population and add the best string in the previous population to the current one.
(Termination) If a prespecified stopping condition is satisfied, stop this algorithm. Otherwise, return to Step 2

3. Result and discussion

3.1. Data Processing

Below the implementation of Genetic algorithm for scheduling process in PT. XYZ with a schedule described on table 1.

Table 1. Schedule Data

Project Name	5 Job Comp.	Project Start	Project Finish	6 Machine (Days)					
				Cutt	Grind	Turn	Cadd	Roll	Blast
Project A	A	23/07/15	23/11/15	1	1	1	8	7	1
	B			1	1	3	8	7	1
	C			3	1	1	8	7	1
	D			1	1	1	10	7	1
	E	4 months		1	3	1	8	7	1

3.2. Specification

Population size	:	Npop = 3
Crossover probability	:	Pc = 1.0
Crossover	:	Two-point crossover (Version I)
Mutation probability	:	Pm= 1.0
Mutation	:	Shift change mutation.

3.3. Initialisation (Coding)

Coding technique is a technique used in encoding genes from chromosome where genes are part of the chromosome. a gene usually represents a variable. Within genetic algorithm each solution coded in the form of bit string.

As an example, string “A B C D E F” has a sequence of job, where its first job that is processed is "Job A", then continue with “Job B”, and so on. If in the step of crossover and mutation we get string “A B C A D E” as the result, this string is not consider as a solution because “A” appear twice and “F” is not available on the string. By this consideration, in the step of genetic algorithm to solve flow shop scheduling problem, we will use permutation approach [6].

3.4. Fitness Function

Fitness value is a value that states whether it’s good or not the individual solution. This fitness value is used as a reference in achieving optimal value in genetic algorithms. Genetic algorithm aims to find individuals with the highest fitness value.

3.5. Generate Initial Population

Generating initial population is the process of generating a number of individuals randomly or through certain procedures. The size of the population depends on the problem to be solved and the type of genetic operator that will be implemented. After the population size is determined, then initial population generation is carried out. The conditions that must be met to show a solution must be carefully considered in the generation of each individual.

The technique in generating this initial population is to use gene permutation techniques. Figure 1 is an example illustrating the results of permutation formation from the initial population:

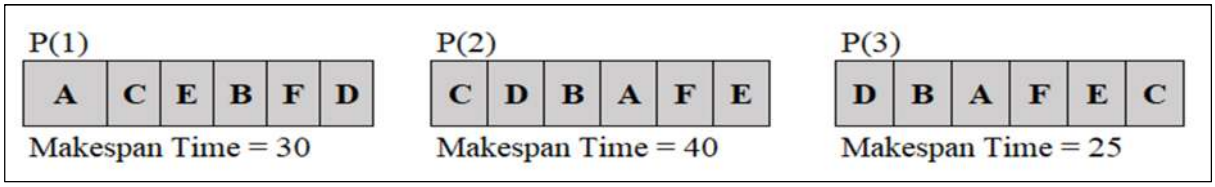

Figure 1. Permutation Formation from The Initial Population

3.6. Selection

Selection is used to select which individuals will be selected for the process of crossover and mutation. Selection is used to get a good prospective parent. "A good parent will produce good offspring." This selection stage requires the input of selected gene permutations and fitness functions.

From this selection stage is obtained by selecting the two best population choices based on the fastest / smallest makespan time from the results of the three permutations in the initial population formation stage. Figure 2 is the results of the selection process with P (3) and P (1):

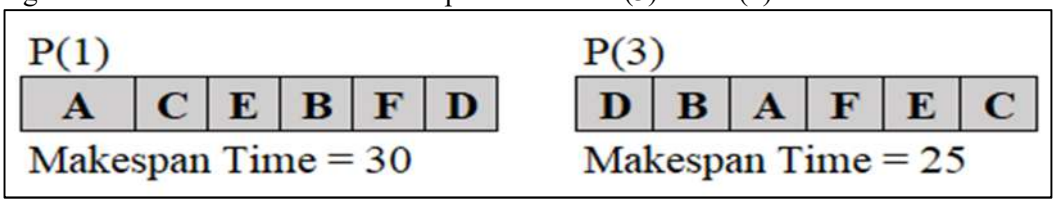

Figure 2. Selection Process

3.7. Crossover

Crossover is the stage of generating a new string (child) from two parent string. The example of crossover are One-point crossover, Two-point crossover (Version I), Two-point crossover (Version II), Two-point crossover (Version III), Position based crossover (Version I), and Position based crossover (Version II).

On this research Two-point crossover (Version I) is being used. In Two-point crossover, two points are randomly selected for dividing one parent. The jobs outside the selected two points are always inherited from one parent to the child, and the other jobs are placed in the order of their appearance in the other parent. Figure 3 is illustration of two-point crossover:

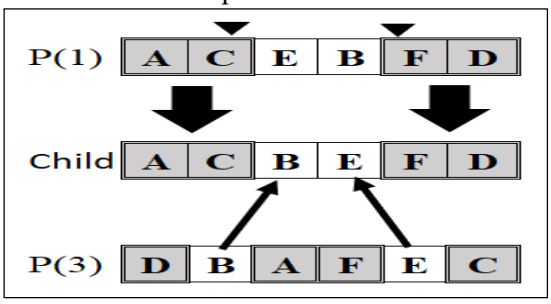

Figure 3. Illustration of Crossover

3.8. Mutation

This operator has the role of replacing genes lost from the population due to the selection process that allows the reappearance of genes that do not appear at the initialization of the population. Child chromosomes are mutated by adding a very small random value (the size of the mutation step), with a low probability.

Mutation probability (pm) is defined as the percentage of the total number of genes in the mutated population. Mutation probability control the number of new genes that will be raised for evaluation. If the chance of mutation is too small, many genes that might be useful have never been evaluated. But if the chance of this mutation is too large, then there will be too many random distractions, so the child will lose the resemblance of the parent, and also the algorithm loses the ability to learn from history. In shift change mutation, a job at one position is removed and put at another position. Then all other jobs shifted accordingly. The two positions are randomly selected.

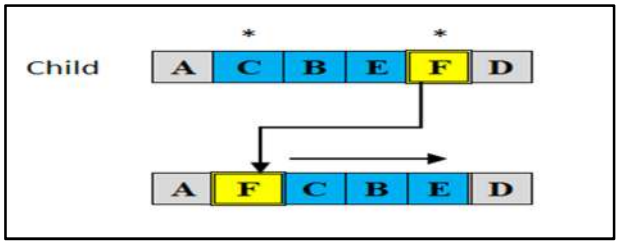

Figure 4. Illustration of Mutations

3.9. Elitist Update Strategy

We applied all the operations to *Npop* pairs of strings (parents). So, we ended up with *Npop* number of children. As a last step we randomly remove one string from the current population and add the best string in the previous population to the current one. Then we continue our processes with this newly generated population.

3.10. Termination

Total number of evaluations/generations used as a stopping condition.

3.11. Results of the Genetic Algorithm Program

The following is the result of data processing in the scheduling stage to calculate the smallest makespan time by making a program using Python as follows:

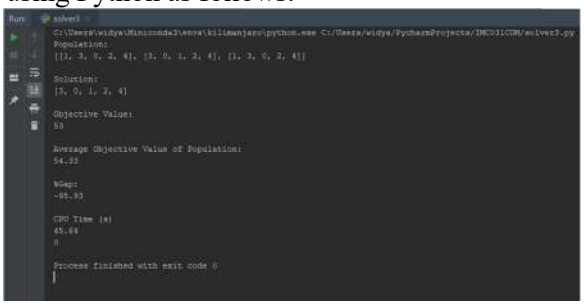

Figure 5. Result

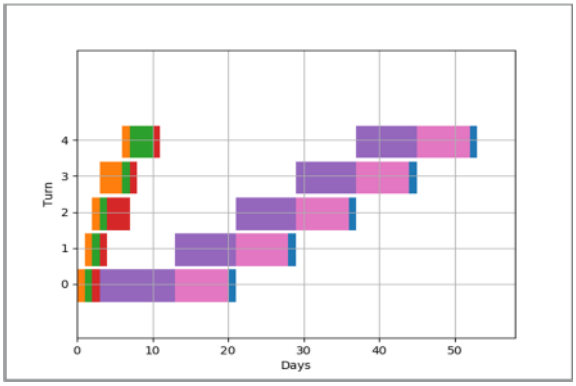

Figure 6. Result Graph

4. Conclusion

The conclusions obtained are based on processing the dataset in the application of Genetic Algorithms for Flow Shop Scheduling Problems at PT. XYZ is the value of makespan time based on computer simulation results. Based on the results of a computer simulation it was found that makespan time with 5 jobs and 6 operations was 53 days (from objective value: 53, figure 5), while the target completion

based on planning was for 4 months. For detailed results on scheduling generated using Genetic Algorithms can be seen in Figure 4. Figure 5 explains the results of the Generate Initiation Population stage. Obtained three population permutations are [[1, 3, 0, 2, 4], [3, 0, 1, 2, 4], [1, 3, 0, 2, 4]]. The result with the smallest makespan is [3, 0, 1, 2, 4] – [D, A, B, C, E], where 3 = the fourth job (component D), 0 = the first job (component A), 1 = the second job (component B), 2 = the third job (component C), 4 = the fifth job (component E). Figure 6 explains the gantchart sequence of processes with colors as follows: orange for cutting operations, green for grinding operations, red for turning operations, purple for cadding operations, pink for rolling operations, and blue for blasting operations. The following is a sequence of processes and times that can be explained through Figure 6: Cutting (D-1 day), Cutting (A-1 day), Cutting (B-1 day), Cutting (C-3 days), Cutting (E-1 day). Grinding (D-1 day), Grinding (A-1 day), Grinding (B-1 day), Grinding (C-1 day), Grinding (E-3 days). Turning (D-1 day), Turning (A-1 day), Turning (B-3 days), Turning (C-1 day), Turning (E-1 day). Next, Cadding (D-10 days), Cadding (A-8 days), Cadding (B-8 days), Cadding (C-8 days), Cadding (E-8 days). Rolling (D-7 days), Rolling (A-7 days), Rolling (B-7 days), Rolling (C-7 days), Rolling (E-7 days). Next, Blasting (D-1 day), Blasting (A-1 day), Blasting (B-1 day), Blasting (C-1 day), Blasting (E-1 day).

References

[1] T. Yamada and . R. Nakano, Job-shop scheduling, 1997.

[2] C. L. Chen, V. S. Vempati and N. Aljaber, “An Application of Genetic Algorithms for Flow Shop Problems,” *European Journal of Operational Research, 80(2),* pp. 389-396, 1995.

[3] R. Yunior César Fonseca, J. Yailen Martínez and N. Ann, “Q-Learning Algorithm Performance For M-Machine, N-Jobs Flow Shop Scheduling Problems to Minimize Makespan,” *Revista Investigacion Operacional Vol. 38, No.3,* pp. 281-290, 2017.

[4] R. H. S and . S. S, “Flowshop-Scheduling Problems With Makespan Criterion: a Review,” *International Journal of Production Research Vol. 43, No. 14,* p. 2895–2929, 15 July 2005.

[5] O. Etiler, B. Toklu, M. Atak and J. Wilson, “A Genetic Algorithm for Flow Shop Scheduling Problems,” *Journal of the Operational Research Society,* pp. 830-835, 2004.

[6] M. Tadahiko, I. Hisao and T. Hideo, “Genetic Algorithms For Flowshop Scheduling Problems,” *Computers ind. Engng Vol. 30, No. 4,* pp. 1061-107, 1996.

[7] J. A. Holland, Adaptation in Natural and Artificial Systems, University of Michigan: Ann Arbor, 1975.

[8] F. Werner, “On the Solution of Special Sequencing Problems, Ph. D. Thesis,” TU Magdeburg, 1984.

[9] M. Shouichi and Y. Seiji, “An Empirical Performance Evaluation of a Parameter-free Genetic Algorithm for Job-Shop Scheduling Problem,” *IEEE Congress on Evolutionary Computation,* pp. 3796-3803, 2007.

[10] G. Li, H. Mao and D. Zhao, “A New Genetic Algorithm in Job-shop Scheduling,” *Fifth International Conference on Natural Computation IEEE,* pp. 98-102, 2009`.

[11] M. Andresen, H. Brasel, M. Morig, J. Tusch, F. Werner and P. Willenius, “Simulated Annealing and Genetic Algorithms for Minimizing Mean Flow Time in an Open Shop,” *Mathematical and Computer Modelling 48,* p. 1279–1293, 2008.

[12] M. Nawaz, E. E. Enscore Jr. and I. Ham, “A Heuristic Algorithm for the m-Machine, n-Job Flow-shop Sequencing Problem,” *Omega The Int. JI of Mgmt Sci. Vol. 11. No 1,* pp. 91-95, 1983.