Knowledge Engineering and Data Science (KEDS)
Vol 3, No 2, December 2020, pp. 60–66

pISSN 2597-4602
eISSN 2597-4637

Efficient Scheduling of Plantation Company Workers using Genetic Algorithm

Wayan Firdaus Mahmudy [1], *, Andreas Pardede [2], Agus Wahyu Widodo [3], Muh Arif Rahman [4]

Faculty of Computer Science, Brawijaya University
Jl. Veteran no. 8, Malang 65145, Indonesia

*[1] wayanfm@ub.ac.id *; [2] andreas.pgpard@gmail.com; [3] a_wahyu_w@ub.ac.id; [4] m_arif@ub.ac.id*
** corresponding author*

ARTICLE INFO

Article history:
Received 23 August 2020
Revised 21 October 2020
Accepted 02 November 2020
Published online 31 December 2020

Keywords:
Scheduling
Genetic Algorithm
Crossover
Mutation
Best parameter

ABSTRACT

Workers at large plantation companies have various activities. These activities include caring for plants, regularly applying fertilizers according to schedule, and crop harvesting activities. The density of worker activities must be balanced with efficient and fair work scheduling. A good schedule will minimize worker dissatisfaction while also maintaining their physical health. This study aims to optimize workers' schedules using a genetic algorithm. An efficient chromosome representation is designed to produce a good schedule in a reasonable amount of time. The mutation method is used in combination with reciprocal mutation and exchange mutation, while the type of crossover used is one cut point, and the selection method is elitism selection. A set of computational experiments is carried out to determine the best parameters' value of the genetic algorithm. The final result is a better 30 days worker schedule compare to the previous schedule that was produced manually.

I. Introduction

Scheduling is an activity carried out by allocating certain resources intended to perform a job or task related to time. Scheduling is a part of the industry's decision-making process to allocate existing data or resources to be utilized more optimally [1]. Good scheduling is required by a company that works in the field of plantation and garden management. The company has a number of tasks such as planning all the care of plants and green areas, providing all fertilizer and plant material needs to ensure all plants are growing well and productive, harvesting crops, and maintaining the stock of fertilizer. Plantation activities are carried out by plantation workers whose schedules are determined by the company. The density of worker activities must be balanced with efficient and fair work scheduling. A good schedule will minimize worker dissatisfaction and work stress while maintaining physical health [2][3]. The schedule should consider a fair allocation of working time in holidays, ensuring all tasks are assigned to a worker, and even the number of working days of each worker every month.

Scheduling that involves complex constraints is a challenging task. Several methods have been proposed in the literature. Meta-heuristic algorithms are often applied as is have the capability to deal with complex constraints. Examples of the algorithms that have been applied for the complex scheduling are ant colony optimization algorithm [4][5], simulated annealing [6][7], tabu search [8], particle swarm optimization [9][10], variable neighborhood search [11][12][13], and genetic algorithm [14][15][16].

The genetic algorithm is a class of evolutionary algorithm which is the most widely used for optimization [17]. Several studies have reported the genetic algorithm's robustness to deal with a complex problem, including scheduling [18][19][20]. This study aims to optimize the plantation company worker's schedules using a genetic algorithm. A proper chromosome representation for the genetic algorithm is key to efficiently explore a large search space of the problem [21]. Thus, an

https://doi.org/10.17977/um018v3i22020p60-66

efficient chromosome representation is designed to produce a good schedule in a reasonable amount of time.

This study considers a scheduling problem in a company that has a large garden and plantation area. There is a total plantation area of ± 50 ha and plantation management of ± 200 ha. 18 workers assign to 3 work shifts every day. Scheduling used in this study is to process guidelines in terms of work time and vacation time of plantation workers. In the process of scheduling, the components used are workers, days, and shifts. The worker shift schedule has the goal of providing a daily schedule divided into thirty-day shifts for each worker. In this study, thirteen employees are assigned into 3 shifts of working time. The morning shift is started at seven o'clock in the morning until three o'clock in the afternoon, then continued with the afternoon shift that is started at three o'clock in the afternoon until eleven o'clock at night. The night shift is started at eleven at night o'clock and is ended at seven o'clock in the morning.

II. Method

The current schedule is made manually. Some workers feel dissatisfied as they have more working time in national holiday than other workers. Other problems are raised when a worker is assigned working time in 2 nights shifts continuously as it will impact worker's physical health.

A genetic algorithm is an algorithm that provides an alternative to the traditional search techniques by adapting the mechanisms found in the genetic world. The genetic algorithm has been successfully applied for complex scheduling [18][19][20]. There are several stages in implementing the genetic algorithm, including determining chromosome structure as solution representation, crossover, and mutation to produce new solutions and selection to pass the chosen solution for the next iterations [22].

Determining chromosome representation is a crucial step in implementing the genetic algorithm. The chromosome representation used is an integer representation. Chromosomes are made based on the division of the number of plants, the number of workers, and also shift workers. Each chromosome will contain numbers from the number of workers that will be entered into each existing shift. An example of part of a chromosome for 2 working days is presented in Figure 1.

There are three shifts: morning, afternoon, and night. The division of workers will then be divided based on working days and work shifts set by the company. In Figure 1, some numbers refer to worker numbers. The number of workers in this calculation is 13. Then, the numbers from 1 to 13 will be assigned to 9 cells (each work shift requires three workers). Then do as much as 30 days / 1 month each worker must not work on the same day and time, and there must also be a balance in the distribution of employee scheduling systems. Therefore, it needs good scheduling in which each employee's composition must be on par with the others.

Penalty calculation is used to find the fitness value of each possible solution [23]. The assessment is used to measure how good the chromosome is for worker scheduling. The number of violations or penalties that appear on the chromosome is required to calculate the fitness value. For each violation, the penalty value will increase 1. The list of penalties that have been determined is presented in Table 1.

	Day 1									**Day 2**								
	Morning			**Afternoon**			**Night**			**Morning**			**Afternoon**			**Night**		
Chromosomes' genes	1	4	2	3	1	11	12	6	5	6	7	9	9	1	2	3	8	10

Fig. 1. Example of a chromosome

Table 1. List of penalties

Penalty	**Mean**
P1	One worker is registered on the same day/shift
P2	Workers have more vacation days than specified
P3	Workers who have shifted the night will work again in the morning shift the next day

How to calculate fitness on the representation of chromosomes is done by first calculating the number of penalties on each chromosome. For example, there are 9, 12, and 4 violations, according to P1, P2, and P3, respectively. Thus, the total penalty is calculated in equation (1):

$$total_{penalty} = penaty\ 1 + penalty\ 2 + penalty\ 3 \tag{1}$$

$$= 9 + 12 + 4 = 25$$

The fitness is calculated according to the total penalty as shown in equation (2):

$$fitness = \frac{100}{100+25} = 0.8 \tag{2}$$

Initial population generation is required for the initial steps in carrying out the counting process. The population is formed from randomized data, with a predetermined number of populations. The amount of the population will be equalized for each generation.

The crossover needed in the genetic process is to get offspring or new chromosomes by choosing two parents. One cut point crossover is a process of crossing between 2 chromosomes or random individuals, which cut the two chromosomes and then combine them to the other cut results.

The mutation process is used to get offspring by using 1 parent. Exchange mutation is used. The method works by randomly selecting two positions (exchange point) and then exchanging the two positions' values.

After doing a fitness calculation for all chromosomes resulting from the crossover and mutation process, it is also necessary to select which chromosomes will be passed to the next generation. Elitsm selection is implemented by gathering both parent and offspring into a pool, then the highest value of chromosomes will be selected

III. Results and Discussions

A. Population Size Testing

Testing on this stage is carried out to measure the effect of population size (number of chromosomes in the population) on the fitness values. The test will use the crossover rate with a value of 0.4 and a mutation rate with a value of 0.6. The results of the test are presented in Table 2 and Figure 2.

Figure 2 shows that in the low population size, the fitness value is lower than the others. The greater the population's size, the fitness value will also tend to be better, but at some point, the increase in population size does not provide a significant increase in fitness value. In the 140th population and 200th population, fitness value achieves the best value compared to other populations. Then the optimal solution in testing the population is a population of 200. Therefore, the next test population will be used 200 for further testing. In the 200th population, the fitness value achieved is 0.6944.

Table 2. Result of population size testing

Population Size	Fitness Average
20	0.66225
40	0.66666
60	0.68493
80	0.67567
100	0.68027
120	0.67567
140	0.69444
160	0.68027
180	0.68965
200	0.69444
220	0.68027
240	0.67114

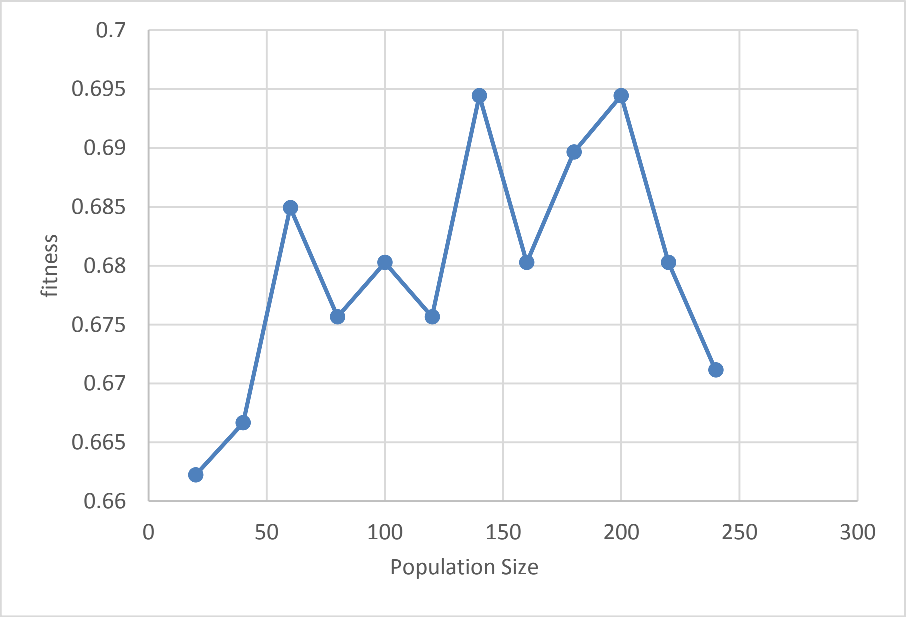

Fig. 2. Result of population size testing

B. Number of Generations Testing

Testing the number of generations aims to find the optimum generation known by looking at the best average fitness results. In this stage, the population size used is 200, and the values of the crossover rate and mutation rate are 0.4 and 0.6, respectively.

As with the effect of population size, the greater the number of generations, the fitness value will also tend to be better, but at a certain point, the addition of the number of generations does not provide a significant increase in fitness. Figure 3 shows that from generation 0 to generation 140, there has been a significant increase in the average fitness value. In the 172nd generation, the average value of fitness has not increased at all. In the 172nd generation, the average fitness value obtained was 0.71942. Therefore, the 172nd generation is the best.

C. Crossover and Mutation Rate Testing

Tests of crossover rate and mutation rate are made to determine the optimal value to reach the best fitness value. The use of crossovers and mutations is to combine and produce offspring for the next generations from the previous explanation. In this test, the value of the rate is used, starting from 0 to 1 with an interval of 0.1. From these results, it can be seen which combination is the best result in this optimization. The result is presented in Table 3.

The crossover and mutation rate testing is started from the value mutation rate of 1 and a crossover rate of 0. The first test's value produces a value of 0.609, which is still relatively low compared with the other results. The highest fitness average is found in the mutation rate of 0.4 and a crossover rate of 0.6. Therefore, the values will be used for the next stage.

D. Scheduling Result

Testing at this stage is used to get the best schedule using the best parameters that have been obtained in the previous tests. In the last test, the best population was the 200th population, with 172 generations using a crossover rate and mutation rate of 0.6 and 0.4, respectively. The change of fitness values during generations is shown in Figure 4.

In the generations of 172, the genetic algorithm obtained a solution with a smaller number of penalties. In this case, the fitness value obtained is 0.7246, and the schedule is shown in Table 4. In the worker schedule for 30 days, there are 3 division numbers indicating each shift. The division starts from the morning, afternoon, and night shifts. The worker's ID will fill each shift.

Table 3. Result of crossover dan mutation rate testing

No	Mutation rate	Crossover rate	Fitness Average
1	1	0	0.6092052235648612
2	0.9	0.1	0.6244287846532304
3	0.8	0.2	0.6415168703054328
4	0.7	0.3	0.6672733216846897
5	0.6	0.4	0.6775241304953841
6	0.5	0.5	0.6938093972311976
7	0.4	0.6	0.6951620993246688
8	0.3	0.7	0.6950206375328978
9	0.2	0.8	0.6837869424975724
10	0.1	0.9	0.6747470681063829
11	0	1	0.6027972532168469

Table 4. Worker schedule for 30 days

Day	Worker		
	Morning	Afternoon	Night
1	[3, 9, 13]	[12, 5, 2]	[6, 8, 7]
2	[12, 10, 1]	[13, 4, 6]	[9, 5, 8]
3	[2, 3, 1]	[4, 6, 8]	[11, 5, 10]
4	[1, 9, 12]	[6, 7, 3]	[13, 10, 5]
5	[8, 2, 11]	[10, 12, 3]	[5, 1, 9]
6	[7, 11, 13]	[2, 6, 3]	[4, 8, 5]
7	[12, 1, 3]	[9, 4, 7]	[8, 10, 6]
8	[3, 5, 12]	[7, 9, 8]	[4, 2, 6]
9	[11, 13, 5]	[1, 4, 2]	[6, 10, 7]
10	[11, 1, 12]	[4, 2, 6]	[5, 8, 9]
11	[6, 10, 3]	[13, 8, 2]	[9, 7, 4]
12	[8, 12, 3]	[1, 4, 11]	[13, 10, 7]
13	[1, 12, 3]	[8, 9, 5]	[13, 4, 6]
14	[8, 11, 9]	[2, 4, 5]	[13, 7, 3]
15	[10, 6, 9]	[1, 12, 3]	[2, 13, 11]
16	[7, 9, 6]	[13, 5, 4]	[2, 8, 12]
17	[1, 10, 7]	[9, 11, 3]	[13, 5, 12]
18	[9, 1, 4]	[8, 12, 7]	[11, 3, 2]
19	[7, 4, 5]	[2, 1, 3]	[13, 10, 9]
20	[11, 7, 1]	[12, 2, 6]	[10, 9, 3]
21	[6, 12, 7]	[5, 2, 4]	[1, 13, 3]
22	[8, 12, 5]	[7, 3, 6]	[9, 11, 10]
23	[8, 12, 6]	[10, 4, 9]	[3, 7, 1]
24	[11, 6, 5]	[1, 8, 13]	[12, 4, 10]
25	[13, 11, 5]	[10, 6, 1]	[3, 2, 4]
26	[13, 1, 9]	[11, 2, 8]	[4, 12, 6]
27	[2, 13, 11]	[5, 8, 7]	[12, 4, 10]
28	[2, 13, 11]	[5, 8, 7]	[12, 4, 10]
29	[1, 9, 6]	[5, 11, 8]	[2, 3, 12]
30	[4, 11, 1]	[2, 8, 12]	[7, 10, 5]

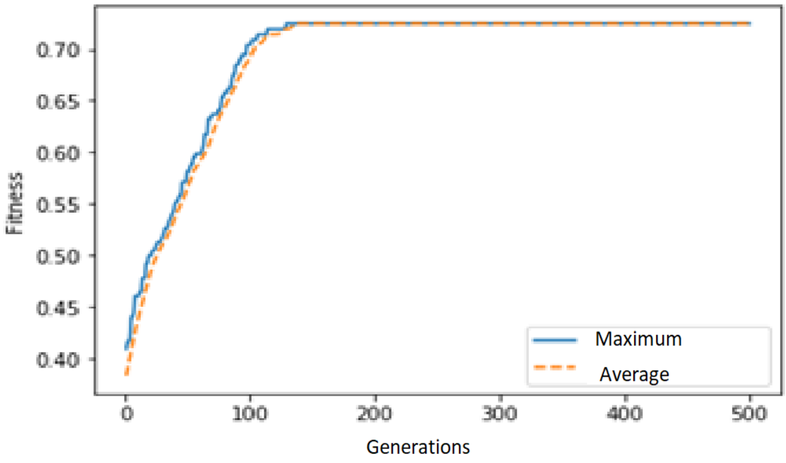

Fig. 3. Result of number of generations testing

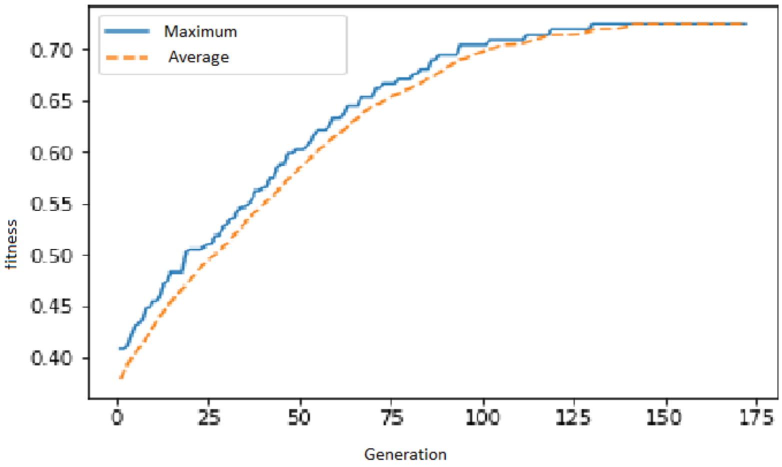

Fig. 4. Result of genetic algorithm running using the best parameter values

IV. Conclusion

Genetic algorithms for optimizing worker scheduling can be implemented with integer chromosome representation. The chromosome representation consists of the worker's ID, date of work, and daily shift. An initial stage is required to obtain the best parameter values for the genetic algorithm. The best parameter values are required to ensure that the genetic algorithm produces a good worker’s schedule in a reasonable amount of time. Using the best parameter values, the genetic algorithm obtained a solution with a smaller number of penalties in the generations of 172. The next study will consider adding local searches into the genetic algorithm’s cycle to produce a better solution.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, *Real coded genetic algorithms for Solving Flexible Job-Shop Scheduling Problem- Part I: Modelling*, vol. 701. 2013.

[2] H. Kikuchi *et al.*, "Association of overtime work hours with various stress responses in 59,021 Japanese workers: Retrospective cross-sectional study," *PLoS One*, vol. 15, no. 3, p. e0229506, Mar. 2020.

[3] K. Bhui, S. Dinos, M. Galant-Miecznikowska, B. de Jongh, and S. Stansfeld, "Perceptions of work stress causes and effective interventions in employees working in public, private and non-governmental organisations: a qualitative study," *BJPsych Bull.*, vol. 40, no. 6, pp. 318–325, Dec. 2016.

[4] Z. Jia, J. Yan, J. Y. T. Leung, K. Li, and H. Chen, "Ant colony optimization algorithm for scheduling jobs with fuzzy processing time on parallel batch machines with different capacities," *Appl. Soft Comput.*, vol. 75, pp. 548–561, 2019.

[5] L. Wang, J. Cai, M. Li, and Z. Liu, "Flexible Job Shop Scheduling Problem Using an Improved Ant Colony Optimization," *Sci. Program.*, vol. 2017, p. 9016303, 2017.

[6] C. Gallo and V. Capozzi, "A Simulated Annealing Algorithm for Scheduling Problems," *J. Appl. Math. Phys.*, vol. 7, Oct. 2019.

[7] F. Chahyadi, A. Azhari, and H. Kurniawan, "Hospital Nurse Scheduling Optimization Using Simulated Annealing and Probabilistic Cooling Scheme," *Indones. J. Comput. Cybern. Syst.*, vol. 12, no. 1, pp. 21–32, 2018.

[8] A. Dabah, A. Bendjoudi, and A. AitZai, "An efficient Tabu Search neighborhood based on reconstruction strategy to solve the blocking job shop scheduling problem," *J. Ind. Manag. Optim.*, vol. 13, no. 4, pp. 2015–2031, 2017.

[9] A. I. Awad, N. A. El-Hefnawy, and H. M. Abdel_kader, "Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments," *Procedia Comput. Sci.*, vol. 65, pp. 920–929, 2015.

[10] H. Jiang, J. Liu, H.-W. Cheng, and Y. Zhang, "Particle swarm optimization based space debris surveillance network scheduling," *Res. Astron. Astrophys.*, vol. 17, no. 3, p. 30, 2017.

[11] S. Thevenin and N. Zufferey, "Learning Variable Neighborhood Search for a scheduling problem with time windows and rejections," *Discret. Appl. Math.*, vol. 261, pp. 344–353, 2019.

[12] W. Jomaa, M. Eddaly, and B. Jarboui, "Variable neighborhood search algorithms for the permutation flowshop scheduling problem with the preventive maintenance," *Oper. Res.*, 2019.

[13] M. Samà, A. D′Ariano, F. Corman, and D. Pacciarelli, "A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations," *Comput. Oper. Res.*, vol. 78, pp. 480–499, 2017.

[14] R. Rody, W. F. Mahmudy, and I. P. Tama, "Using Guided Initial Chromosome of Genetic Algorithm for Scheduling Production-Distribution System," *J. Inf. Technol. Comput. Sci.*, vol. 4, no. 1, pp. 26–32, 2019.

[15] M. L. Seisarrina, I. Cholissodin, and H. Nurwarsito, "Invigilator Examination Scheduling using Partial Random Injection and Adaptive Time Variant Genetic Algorithm," *J. Inf. Technol. Comput. Sci.*, vol. 3, no. 2, pp. 113–119, 2018.

[16] H. Algethami, R. L. Pinheiro, and D. Landa-Silva, "A genetic algorithm for a workforce scheduling and routing problem," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 927–934.

[17] V. Meilia, B. D. Setiawan, and N. Santoso, "Extreme Learning Machine Weights Optimization Using Genetic Algorithm In Electrical Load Forecasting," *J. Inf. Technol. Comput. Sci.*, vol. 3, no. 1, pp. 77–87, 2018.

[18] A. Rahmi, W. F. Mahmudy, and M. Z. Sarwani, "Genetic Algorithms for Optimization of Multi-Level Product Distribution," *Int. J. Artif. Intell.*, vol. 18, no. 1, pp. 135–147, 2020.

[19] V. N. Wijayaningrum and W. F. Mahmudy, "Optimization of ship's route scheduling using genetic algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 2, no. 1, 2016.

[20] L. R. Abreu, J. O. Cunha, B. A. Prata, and J. M. Framinan, "A genetic algorithm for scheduling open shops with sequence-dependent setup times," *Comput. Oper. Res.*, vol. 113, p. 104793, 2020.

[21] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Hybrid genetic algorithms for multi-period part type selection and machine loading problems in flexible manufacturing system," *2013 IEEE Int. Conf. Comp. Intl. Cyb. (CYBERNETICSCOM)*, Dec. 2013.

[22] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*. New York: John Wiley & Sons, Inc., 2000.

[23] B. F. Rosa, M. J. F. Souza, S. R. de Souza, M. F. de França Filho, Z. Ales, and P. Y. P. Michelon, "Algorithms for job scheduling problems with distinct time windows and general earliness/tardiness penalties," *Comput. Oper. Res.*, vol. 81, pp. 203–215, 2017.