

Fodder composition optimization using modified genetic algorithm

Vivi Nur Wijayaningrum, Wayan Firdaus Mahmudy

Faculty of Computer Science, Brawijaya University, Indonesia

Article Info

Article history:

Received Apr 21, 2018

Revised Sep 07, 2018

Accepted Feb 24, 2019

Keywords:

Fodder composition

Genetic algorithm

Livestock

Numerical methods

ABSTRACT

Determination of the fodder composition is a difficult process because it should simultaneously consider several constraints, such as minimizing the total cost of feed ingredients and maximizing the nutrient needs required by livestock. This study uses a modified genetic algorithm to solve the problem in order to obtain better results. The modification is done by applying numerical methods in generating an initial population of the genetic algorithm. Testing results show that the optimal parameters that can be used to produce the optimal solution are as follows: population size (popsize) is 300, generation number is 400, crossover rate (cr) value is 0.2, and mutation rate (mr) value is 0.6. The modified genetic algorithm provides an average fitness value of 0.142357, while the classical genetic algorithm provides an average fitness value of 0.094354. With additional computational time equal to 110 ms, the use of modified genetic algorithm offered has proven to provide a better result, with a higher fitness value compared with classical genetic algorithm.

*Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Wayan Firdaus Mahmudy,
Faculty of Computer Science,
Brawijaya University, Malang, Indonesia
Email: wayanfm@ub.ac.id

1. INTRODUCTION

The fodder is one factor that plays an important role in determining success in the livestock sector. It has a major contribution in the number of 50-70% of the total cost of livestock. The rise in prices of feed ingredients generally, especially soybeans, making the producers reconsider the fodder composition in order to increase the efficiency target of the fodder cost [1]. Therefore, additional efforts are needed to optimize the use of the feed in order to increase livestock products.

The problem of mixing fodder composition becomes difficult because it should simultaneously consider several constraints. On one hand, selecting the feed ingredients should consider the total cost of the feed ingredients to a minimum, but on the other hand, it should also maximize the nutrient needs required by livestock. The criteria for determining the composition of fodder formulation consist of the selection of feed ingredients and nutrient contents, so it is necessary to form a combination of feed ingredients to produce high nutrient content value [2].

Several techniques have been proposed to form the fodder composition formulations. Genetic algorithm is used to optimize the cost of feed ingredients in the mixing fodder composition by determining the proportion of feed ingredients randomly in mixing feed ingredients in the beginning of the calculation process to form an optimal feed ingredient solutions by considering constraints such as the availability of ingredients and overall total feed ingredients [3]. The mixing of fodder composition by using Goal Programming is done by forming a model of multi-criteria programming to determine the mix of feed ingredients to minimize costs, maximize nutrients and minimize the water content [4]. Mathematical Programming is used to solve the problems of poultry feed formulations in Nigeria. The mathematical equation is formed to minimize the total

cost of feed ingredients and consider constraints such as the number of fodder required, the minimum number of nutrients and energy that must be met, and the limited availability of feed ingredients [5].

The genetic algorithm can be used to solve problems in various fields of science and technology. It has several advantages, such as it can be used optimally in problems with continuous and discrete variables, it provides some optimal solutions, not just a single solution, and it works well on various types of optimization such as numerical data and analytical functions [6]. It also has the ability to generate good solutions for some complex problems [7]–[9]. However, initial values at the beginning of genetic algorithm process which is usually done at random must meet predetermined constraints because it can affect a given solution [10]. If the values in the initial population are not well chosen, then the right solution will be difficult to find [11], [12].

The genetic algorithm is selected to solve this problem because it is easy to use, simple, and has a wide search area [13]. In previous research, numerical methods such as Cramer's Rule, Gauss-Elimination, and Gauss-Jordan method were used to determine the values that will be used in the initialization process of the genetic algorithm. The calculation process was carried out by forming a linear equation system to obtain values that represented the number of feed ingredients to be used in the mixing of fodder composition [14]. In this research, the determination of the fodder composition solved using a modified genetic algorithm by applying a numerical method for the initialization of the initial chromosome. The proposed solution is to create a model of feed ingredients composition to meet the nutrient needs of livestock by minimizing the total cost of feed ingredients.

2. MODIFIED GENETIC ALGORITHM

The genetic algorithm (GA) is an optimization algorithm inspired by the process of natural selection. It is a general algorithm that can be applied to the large and complex problems [15] because it works based on population searching so that it can search globally and in all directions [16]. Like the other stochastic methods, there is no guarantee that the GA will find a global optimum solution for a problem. However, it can provide an acceptable solution well with a reasonable process [17].

The GA has a population consisting of several individuals or chromosomes which evolve with the selection rules which is determined to maximize the fitness value by minimizing the cost function [18]. The process starting from generating a random population consisting of several chromosomes, which each chromosome represents a problem solution. Then, the fitness value of each chromosome in the population is calculated. Afterwards, offspring are formed by performing crossover and mutation based on the crossover rate and mutation rate. Selection is performed to select a number of chromosomes to proceed to the next generation. The chromosomes are selected according to their fitness value. The population of the newly formed will replace the old population [15].

The proposed method is to modify the chromosome initialization process in the population. The modification is done by applying numerical methods in generating an initial population of the genetic algorithm. This mechanism is carried out by inserting the initial values obtain from the calculation using Cramer's Rule as a chromosome into the population. Then, the other chromosomes in the population will be generated randomly.

A chromosome from the calculation result using Cramer's Rule allows to produce a solution in an area close to the optimal solution in the search space so that it can provide a higher fitness value compared to other chromosomes. During the evolutionary process, the other chromosomes will be updated through the process of reproduction. Thus, the process of searching a solution will produce a fairly high fitness value at the beginning of time.

3. RESEARCH METHOD

The problem of fodder composition formulation is defined as the determination of the number of each feed ingredient to meet the required nutrient of livestock by considering the minimum total cost but still meet a minimum and a maximum number of feed ingredients required. Data of cattle used in this problem shown in Table 1. The nutrients consist of Dry Matter (DM), Digestible Crude Protein (DCP), Total Digestible Nutrients (TDN), Metabolizable Energy (ME), Calcium (Ca), and Phosphorus (P) shown in Table 2 [19].

3.1. Chromosome Representation

The GA has a population consisting of several possible solutions. Every possible solution is represented by a chromosome in the form of abstract representation. Encoding solutions into the chromosome is a process that is first performed in the GA. This process is not easy to do because of the determination of the proper representation of chromosomes will affect the overall process of the GA [20]. The effectiveness of the GA in the exploration of the search space will be strongly influenced by the representation of the chromosome used [21].

Table 1. Nutrient requirements

Weight (Kg)	Weight Gain (Kg)	DM (Kg)	DCP (Kg)	TDN (Kg)	ME (Mcal)	Ca (Kg)	P (Kg)
150	0	3.0	0.231	1.4	5.10	0.006	0.006
	0.25	3.8	0.400	1.8	6.56	0.012	0.009
	0.5	4.2	0.474	2.2	8.02	0.016	0.010
	0.75	4.4	0.589	2.6	9.55	0.021	0.013
	1	4.5	0.607	3.0	10.93	0.027	0.016
200	0	3.7	0.285	1.8	6.30	0.006	0.006
	0.25	4.5	0.470	2.2	8.10	0.011	0.009
	0.5	5.2	0.554	2.8	9.90	0.016	0.012
	0.75	5.4	0.622	3.2	11.7	0.021	0.015
	1	5.6	0.690	3.7	13.51	0.027	0.017
250	0	4.4	0.337	2.0	7.40	0.009	0.009
	0.25	5.3	0.534	2.6	9.52	0.012	0.010
	0.5	6.2	0.623	3.2	11.64	0.016	0.014
	0.75	6.4	0.693	3.8	13.78	0.021	0.017
	1	6.6	0.760	4.3	15.84	0.028	0.019
300	0	5.0	0.385	2.4	8.50	0.010	0.010
	0.25	6.0	0.588	3.0	10.90	0.015	0.011
	0.5	7.0	0.679	3.7	13.40	0.019	0.014
	0.75	7.4	0.753	4.3	14.80	0.023	0.018
	1	7.5	0.819	5.0	18.23	0.028	0.021

Table 2. Cost and nutrient content

Feed Ingredients	Price (N/Kg)	DM (%)	DCP (%)	TDN (%)	ME (Mcal/Kg)	Ca (%)	P (%)
Coconut meal	Rp 3500	86	21.6	66	2.48	0.08	0.67
Corn bran	Rp 4000	86	11.3	52	1.85	0.06	0.77
Rice straw	Rp 4000	40	4.3	40	1.35	0	0
Leucaena	Rp 3500	30	23.4	77	2.96	1.40	0.21
Thatch grass	Rp 3000	40	5.4	54	1.96	0.13	0.09
Elephant grass	Rp 3000	21	8.3	50	1.80	0.59	0.29

To formulate the fodder composition, the chromosome representation that is used is real code representation that describes the number of each feed ingredient used in the fodder mixing. The number of genes in a chromosome is the percentage of feed ingredients. Real-code representation is chosen because it has several advantages such as improving the efficiency of genetic algorithms because it does not need to convert the value of solution variables, less memory is needed, and there is freedom to use different genetic operators [22].

Suppose the feed composition will be used for cattle weighing 200 kg and the expected weight gain is 0.5 kg. The chromosome can be formed as shown in Table 3.

Table 3. Example of the chromosome representation

Coconut meal	Corn bran	Rice straw	Leucaena	Thatch grass	Elephant grass
0	12.27	0	13.22	48.19	0

In Table 3, the first gene with a value of 0 indicates the percentage of the first feed ingredient (coconut meal), the second gene with a value of 12.27 indicates the percentage of the second feed ingredient (corn bran), and so on up to the sixth gene with a value of 0 which indicates that the percentage of the eight-feed ingredient (elephant grass). The feed ingredients which have a value of 0 indicates that these feed ingredients are not used as fodder mixes.

These gene values have limits, they are 0 for the lower limit value and 50 for the upper limit value. If there is a gene which has a value less or more than the predetermined limit value, then the gene value will be set in accordance with the minimum or maximum limit value.

3.2. Fitness Function

The fitness value is used to measure the quality of the solutions produced by each individual based on the objective function [23]. In this research, the fitness function is calculated based on the total price of feed ingredients used in fodder compositions. In addition, the fitness function is also calculated based on the fulfillment of nutrient needs. If there are nutrients that have a number less than a predetermined number, then the number of nutrient deficiencies is used as a penalty value. This penalty technique is used to penalize the

infeasible solutions, which depends on the number of violation of the constraints that occur [24]. Fitness function can be calculated using (1) [8].

$$F = \frac{10000}{\sum_{i=1}^N \text{cost}_i + \sum_{j=1}^M \sum_{i=1}^N \text{penalty}_{ij}} \times 1000 \quad (1)$$

Cost is the total cost of feed ingredients used, penalty is the total value given for any deficiency number of nutrients, N is a number of feed ingredient, and M is a number of nutrients.

3.3. Crossover

Crossover is the process of combining the genetic information from two individuals [12]. It is used to exploit the current solution in order to get a better solution. This process begins with selecting two parents that will be crossed, then the genetic information from both parents are exchanged [25].

The crossover rate (cr) is used to determine how often the crossover process will be conducted. When the value of cr is 1, then all the offspring derived from the crossover. While the value of cr is 0 illustrates that the new population generated is exactly the same chromosomes copy of the previous population [11].

In this research, the crossover process is done by using extended intermediate crossover which produces two offspring of the combined value of the two parental chromosomes [26]. The offspring can be generated using (2) [27], with the value of α selected randomly in the interval [-0.25, 1.25].

$$\begin{aligned} \text{Offspring}_1 &= \text{Parent}_1 + \alpha (\text{Parent}_2 - \text{Parent}_1) \\ \text{Offspring}_2 &= \text{Parent}_2 + \alpha (\text{Parent}_1 - \text{Parent}_2) \end{aligned} \quad (2)$$

3.4. Mutation

Mutation is used to explore the whole search space. It is used to prevent the algorithm trapped in local optimum solution and maintain the diversity of individuals in the population. Low population diversity leads to premature convergence. This happens when all the individuals in the population reach suboptimal conditions so that the genetic algorithm cannot produce offspring that have better fitness values than their parents [11].

Mutation usually performed on selected individual to produce its new version by modifying the genetic material randomly [28]. A random mutation method used by increasing or decreasing the value of the selected genes with small random numbers [21], [29]. Suppose the range of values of variables x_i is $[\min_i, \max_i]$ and the resulting offspring is $C = [x'_1, \dots, x'_n]$, then the value of genes in offspring can be generated using (3), with a value of r selected randomly in the interval [-0.1, 0.1].

$$x'_i = x_i + r (\max_i - \min_i) \quad (3)$$

3.5. Selection

The selection process is done by comparing each chromosome in a population based on its fitness value. This process is used to select individuals used for the reproduction process in the next generation in order to generate new search areas [24].

In this research, the selection method used is a combination of random selection and elitism selection. The random selection is the simplest selection operator and has a low speed to produce a uniform population. In the random selection, each individual has an equal chance of $1/n$ where n is population size (popsize), so that the best and worst individuals in the population has an equal chance to move on to the next generation. While elitism selection has a high speed to produce a uniform population because this method ensures the best individual in the population currently can survive to move on to the next generation [27]. The combination of these two methods is used to prevent the dominance of the best individuals in the population which led to the acquisition of a local optimum solution.

4. RESULTS AND DISCUSSION

The parameter testing is conducted to determine the best parameter values used in the genetic algorithm in order to obtain the optimal solution. Setting the parameter values in the genetic algorithm is a determining factor in the balance of exploration and exploitation. The performance of genetic algorithm is not optimal if these parameters are not well defined, which results in premature convergence [30].

The parameter testing consists of population size testing, the generation number testing, the crossover rate (cr) value testing, and the mutation rate (mr) value testing. In this testing scenario, the data used are cattle with a weight of 200 kg and a target for weight gain of 0.5 kg.

The population size testing used to determine the proper chromosome number so it can produce the optimum solution for the completion of the optimization of fodder composition. The population size tested are 100, 200, 300, 400, 500, and 600. The number of generations used in this test is 500, with a combination of cr and mr are 0.3 and 0.4. The testing conducted 20 times for each population size testing scenario. Figure 1 shows the testing result of population size.

Figure 1 shows that the average fitness value will be increasing along with increasing the population size. This is because the genetic algorithm has a wider search area if the population size is large, so better average fitness value can be obtained. However, the fitness value does not increase significantly at a certain point. As shown in Figure 2, the average fitness value generated by population size 100 to 300 are increasing, but the fitness value tends to be stable when the population size more than 300, which means that 300 is the optimal population size.

The generation number testing used to determine the proper generation number so it can produce the optimum solution. The number of the generation tested are 100, 200, 300, 400, 500, and 600. The population size used is 300, which is the best population size based on test results as shown in Figure 1. While the combination of cr and mr used are 0.3 and 0.4. The testing conducted 20 times for each testing scenario of generation number. Figure 2 shows the testing result of generation number.

Figure 2 shows that the average fitness value will increase with the increasing of generation number. The average fitness value produced by the generation of 100 to 400 are increasing but tends to be stable when the generation number more than 400. This indicates that the optimal generation number is 400.

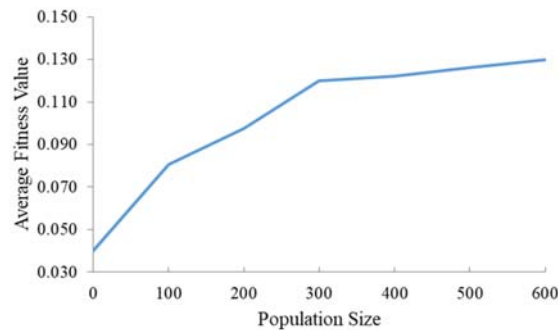


Figure 1. The Testing Result of Population Size

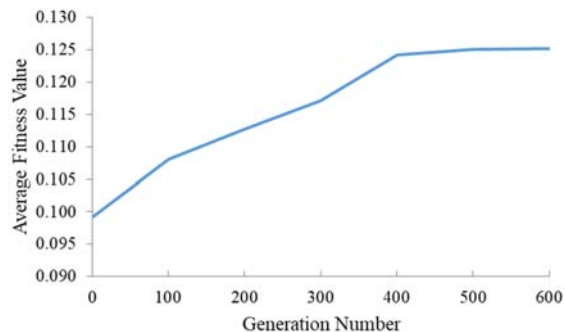


Figure 2. The Testing Result of Generation Number

The last testing is the testing of the value of cr and mr, which are used to determine the proper value of cr and mr so it can produce the optimum solution for the completion of this problem. The population size used is 300 and the number of generations used is 400 because these values are the best parameters based on

the testing results as shown in Figure 1 and Figure 2. In order to get a fair result, the combination of cr and mr used is $cr \times 2 + mr = 1$ since the crossover method used is extended intermediate, which produce two offspring in each process. Figure 3 shows the testing result of the value of cr and mr .

The testing results of cr and mr value in Figure 3 shows that the best parameters used to generate the optimal solution are the cr value is 0.2 and the value of mr is 0.6. The higher value of cr , the exploitation process will be faster and new solutions will be more quickly introduced to the population [31]. If the value of cr is too high, the genetic algorithm is not able to maintain the diversity of the population, so that it can produce a local optimum solution. Otherwise, if the value of mr is too high, the genetic algorithm will work like a random search method that causes lessening the effectiveness of the process of searching solutions [29].

Table 4 shows the comparison of the fitness value generated by the classical genetic algorithm and the modified genetic algorithm. The comparison of the fitness value indicates that resolving the problem of determining the fodder composition using the modified genetic algorithm gives better results with higher fitness value than the classical genetic algorithm, with little additional computational time.

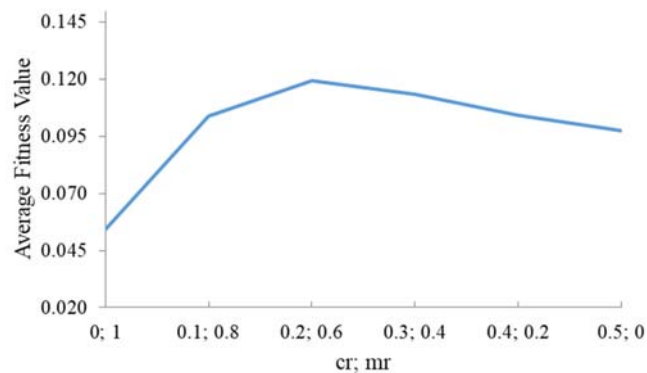


Figure 3. The Testing Result of the Value of cr and mr

Table 4. Comparison of the fitness value

Trial Number	Classical GA		Modified GA	
	Fitness	Time (ms)	Fitness	Time (ms)
1	0.095357	1775	0.149618	2586
2	0.087111	2013	0.139291	1863
3	0.099812	1782	0.124807	2052
4	0.083791	1766	0.128686	2178
5	0.096835	1831	0.133198	2052
6	0.099331	1864	0.154192	1764
7	0.088005	2102	0.143656	1899
8	0.103614	1755	0.167993	1941
9	0.098197	1842	0.122917	1923
10	0.092165	1774	0.120358	1837
11	0.080924	1942	0.151166	1948
12	0.095352	1882	0.126800	1888
13	0.095227	1902	0.142304	1953
14	0.106737	1816	0.182894	2082
15	0.102578	2030	0.125886	2104
16	0.085448	1853	0.163153	1856
17	0.104368	1966	0.183965	2024
18	0.094688	1964	0.123587	1904
19	0.085147	1881	0.127069	2040
20	0.092389	1887	0.135608	1932
Average	0.094354	1881.3	0.142357	1991.3

The average fitness value generated by the modified genetic algorithm is better than the classical genetic algorithm because the gene values used in the initialization process are the initial values close to the optimal solution. These initial values are the number of feed ingredients obtained from the calculation using numerical methods. Using these initial values, the genetic algorithm can improve its ability to find good areas in the search space quickly.

To find out the improvements made by the modified genetic algorithm, additional testing is performed to compare the results given by the classical genetic algorithm, the modified genetic algorithm, and Simulated Annealing. Comparison is done using the genetic algorithm optimal parameters as follows: population size is 300, the generation number is 400, the crossover rate value is 0.2, and the mutation rate value is 0.6. While the parameters used by Simulated Annealing are as follows: the iteration number is 100 and the temperature decrease rate is 0.75. Figure 4 shows the comparison result of the three algorithms.

Based on Figure 4, it can be seen that the use of the initial values obtained from the calculation using numerical methods during the initialization process gives an impact to the performance of the genetic algorithm. The increase in results given by classical genetic algorithms and Annealing Simulated at each iteration is not high enough, even Simulated Annealing tends to produce stable results, while the modified genetic algorithm has increased significantly over several times. Even in the early iterations, modified genetic algorithm has resulted in a high increase on the fitness value compared to the classical genetic algorithm. This makes the modified genetic algorithm able to achieve the optimal solution with a small number of iterations.

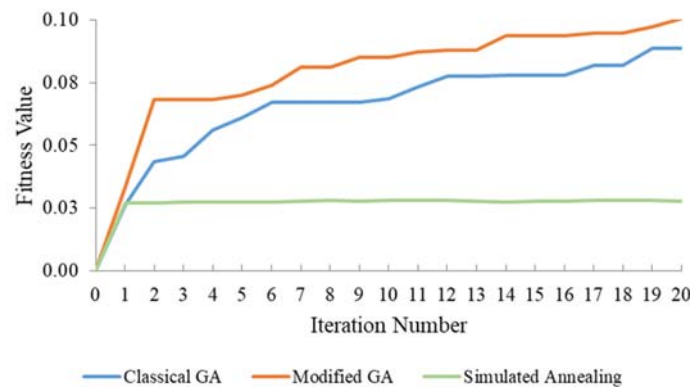


Figure 4. Comparison result of classical GA, modified GA, and simulated annealing

5. CONCLUSION

This research has found that the genetic algorithm is usable to solve the problem of determining the fodder composition. The modified genetic algorithm provides a better fitness value than the classical genetic algorithm. The parameters used to obtain the optimal solution are as follows: population size is 300, the generation number is 400, the crossover rate value is 0.2, and the mutation rate value is 0.6. The use of the initial values obtained from the calculation using numerical methods during the initialization process makes the genetic algorithm get higher fitness values at each iteration rapidly, so the genetic algorithm can achieve the optimal solution in few iterations.

In the next research, the problem of determining the fodder composition can be developed by considering the availability of feed ingredients in the warehouse so the number and type of feed ingredient to be used in the fodder mixing can be determined. With the increasing of the complexity of the problems to be solved, then the techniques required to get the solution will also be complex. Therefore, hybridization technique is necessary to do between the genetic algorithms with other algorithms to get a better solution. The process of finding solutions by using hybridization between the genetic algorithm with Tabu Search will be used in the future research [32]. To prevent premature convergence, the use of a fuzzy crossover operator can also be applied to the genetic algorithm [33].

REFERENCES

- [1] P. Spring, "The Challenge of Cost Effective Poultry and Animal Nutrition : Optimizing Existing and Applying Novel Concepts," *Lohmann Inf.*, vol. 48, no. 1, pp. 38–46, 2013.
- [2] R. A. Rahman, C. Ang, and R. Ramli, "Investigating Feed Mix Problem Approaches : An Overview and Potential Solution," *World Acad. Sci. Eng. Technol.*, vol. 46, no. 10, pp. 424–432, 2010.
- [3] M. A. Şahman, M. Çunkaş, Ş. Inal, F. Inal, B. Coşkun, and U. Taşkıran, "Cost Optimization of Feed Mixes by Genetic Algorithms," *Adv. Eng. Softw.*, vol. 40, no. 10, pp. 965–974, 2009.
- [4] Z. Babić and T. Perić, "Optimization of Livestock Feed Blend by Use of Goal Programming," *Int. J. Prod. Econ.*, vol. 130, no. 2, pp. 218–223, 2011.

- [5] V. Oladokun and A. Johnson, "Feed Formulation Problem in Nigerian Poultry Farms: A Mathematical Programming Approach," *Am. J. Sci. Ind. Res.*, vol. 3, no. 1, pp. 14–20, 2012.
- [6] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Second Edi. Luniver Press, 2010.
- [7] 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," in *IEEE International Conference on Computational Intelligence and Cybernetics*, 2013, pp. 126–130.
- [8] V. N. Wijayaningrum, W. F. Mahmudy, and M. H. Natsir, "Optimization of Poultry Feed Composition Using Hybrid Adaptive Genetic Algorithm and Simulated Annealing," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 2–8, pp. 183–187, 2017.
- [9] 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, pp. 180–186, 2016.
- [10] C. Faycal, M. E. Riffi, and B. Ahiod, "Hybrid Genetic Algorithm and Greedy Randomized Adaptive Search Procedure for Solving a Nurse Scheduling Problem," *J. Theor. Appl. Inf. Technol.*, vol. 73, no. 2, pp. 313–320, 2015.
- [11] D. Gupta and S. Ghafir, "An Overview of Methods Maintaining Diversity in Genetic Algorithms," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 5, pp. 56–60, 2012.
- [12] J. Magalhães-Mendes, "A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop Scheduling Problem," *WSEAS Trans. Comput.*, vol. 12, no. 4, pp. 164–173, 2013.
- [13] S. Malik and S. Wadhwa, "Preventing Premature Convergence in Genetic Algorithm Using DGCA and Elitist Technique," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 6, pp. 410–418, 2014.
- [14] V. N. Wijayaningrum and F. Utaminingrum, "Numerical Methods for Initialization in Fodder Composition Optimization," in *IEEE International Conference on Advanced Computer Science and Information Systems*, 2016, pp. 397–400.
- [15] P. Hoseini and M. G. Shayesteh, "Efficient Contrast Enhancement of Images using Hybrid Ant Colony Optimisation, Genetic Algorithm, and Simulated Annealing," *Digit. Signal Process. A Rev. J.*, vol. 23, no. 3, pp. 879–893, 2013.
- [16] S. Lianshuan and W. Huahui, "A Multi-Agent Self-Adaptive Multi-Objective Genetic Algorithm," *Int. J. Intell. Eng. Syst.*, vol. 8, no. 2, pp. 7–13, 2015.
- [17] S. Dündar and I. Şahin, "Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways," *Transp. Res. Part C Emerg. Technol.*, vol. 27, pp. 1–15, 2013.
- [18] M. Gendreau and J.-Y. Potvin, *Handbook of Metaheuristics*, Second Edi. Springer, 2010.
- [19] U. Umiyasih and Y. N. Anggraeny, "Technical Guidelines for Balanced Rations, Feed Strategies on Beef Cattle (In Bahasa)," *Pusat Penelitian dan Pengembangan Peternakan*, 2007.
- [20] V. Jorapur, V. S. Puranik, A. S. Deshpande, and M. R. Sharma, "Comparative Study of Different Representations in Genetic Algorithms for Job Shop Scheduling Problem," *J. Softw. Eng. Appl.*, vol. 7, no. 7, pp. 571–580, 2014.
- [21] 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," *Adv. Mater. Res.*, vol. 701, pp. 359–363, 2013.
- [22] A. Bedboudi, C. Bouras, and M. T. Kimour, "An Heterogeneous Population-Based Genetic Algorithm for Data Clustering," *Indones. J. Electr. Eng. Informatics*, vol. 5, no. 3, pp. 275–284, 2017.
- [23] Y. Xu, K. Li, J. Hu, and K. Li, "A Genetic Algorithm for Task Scheduling on Heterogeneous Computing Systems using Multiple Priority Queues," *Inf. Sci. (Nj.)*, vol. 270, pp. 255–287, 2014.
- [24] M. Mehra, M. L. Jayalal, A. J. Arul, S. Rajeswari, K. K. Kuriakose, and S. A. V. S. Murty, "Study on Different Crossover Mechanisms of Genetic Algorithm for Test Interval Optimization for Nuclear Power Plants," *Int. J. Intell. Syst. Appl.*, vol. 6, no. 1, pp. 20–28, 2014.
- [25] A. Otman and A. Jaafar, "A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem," *Int. J. Comput. Appl.*, vol. 31, no. 11, pp. 49–57, 2011.
- [26] A. J. Umbarkar and P. D. Sheth, "Crossover Operators in Genetic Algorithms: A Review," *ICTACT J. Soft Comput.*, vol. 6, no. 1, pp. 1083–1092, 2015.
- [27] A. P. Engelbrecht, *Computational Intelligence*. England: John Wiley & Sons. Inc., 2007.
- [28] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Second Edi. Springer, 2015.
- [29] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Real Coded Genetic Algorithms for Solving Flexible Job-Shop Scheduling Problem - Part II: Optimization," *Adv. Mater. Res.*, vol. 701, pp. 364–369, 2013.
- [30] F. Herrera and M. Lozano, "Fuzzy Adaptive Genetic Algorithms: Design, Taxonomy, and Future Directions," *Soft Comput.*, vol. 7, no. 8, pp. 545–562, 2003.
- [31] D. K. Dey, "Mathematical Study of Adaptive Genetic Algorithm (AGA) with Mutation and Crossover Probabilities," *Int. J. Adv. Comput. Technol.*, vol. 3, no. 5, pp. 765–768, 2014.
- [32] S. Khalid, "Optimized Aircraft Electric Control System Based on Adaptive Tabu Search Algorithm and Fuzzy Logic Control," *Indones. J. Electr. Eng. Informatics*, vol. 4, no. 3, pp. 149–164, 2016.
- [33] M. J. Varnamkhasti, L. S. Lee, M. R. A. Bakar, and W. J. Leong, "A Genetic Algorithm with Fuzzy Crossover Operator and Probability," *Adv. Oper. Res.*, vol. 2012, pp. 1–17, 2012.