Optimisation of Rice Fertiliser Composition using Genetic Algorithms

Retno Dewi Anissa [a, 1, *], Wayan Firdaus Mahmudy [a, 2], Agus Wahyu Widodo [a, 3]

[a] Faculty of Computer Science, Universitas Brawijaya
Jl. Veteran, Malang, 65145, Indonesia
*[1] retnodewianissa@yahoo.com *; [2] wayanfm@ub.ac.id; [3] a_wahyu_w@ub.ac.id*
* corresponding author

ARTICLE INFO

ABSTRACT

There are so many problems with food scarcity. One of them is not too good rice quality. So, an enhancement in rice production through an optimal fertiliser composition. Genetic algorithm is used to optimise the composition for a more affordable price. The process of genetic algorithm is done by using a representation of a real code chromosome. The reproduction process using a one-cut point crossover and random mutation, while for the selection using binary tournament selection process for each chromosome. The test results showed the optimum results are obtained on the size of the population of 10, the crossover rate of 0.9 and the mutation rate of 0.1. The amount of generation is 10 with the best fitness value is generated is equal to 1,603.

I. Introduction

Fertiliser is very important for rice growth and evolution. It has one or more elements, used as a replacement of element in plants. Fertilisation can be interpreted as a soil nutrient enhancer. Fertiliser is divided into several types based on where they are come from [1]; such as inorganic, organic and biological fertiliser. There are two groups of inorganic fertilisers based on the content of nutrients: single and compound fertiliser [2][3][4]. Single fertiliser only had one content. There are three kinds of single fertiliser which had the main nutrients such as nitrogen (N), phosphorus (P), potassium (K) and magnesium (Mg). Some examples of single fertilisers are nitrogen, phosphorus, potassium, and magnesium fertiliser. On the other hands, compound fertiliser is a mixture of two or more nutrients. Some examples of mixture is blended again into NP and NPK.

In rice planting, the right composition of fertilizer is essentially needed. If the composition given is not correct, the growth and quality of rice produced is inadequate or the costs incurred become more wasteful and even unaffordable [5][6]. Farmers need an appropriate combination in order to obtain good results and minimum expenses of using fertiliser. The combination should consist of s nitrogen (n), phosphorus (p) and potassium (k). This composition is one of the solutions used in composing the optimal fertiliser for rice plants. Often, farmers do trial and error or follow others experience to get the right composition of fertiliser. The problem will arise if the raw material for fertilizer changes, then the farmer must change the composition that he knew beforehand. They must try harder to get the right composition of fertiliser. Many farmers fail in this case and lose a lot of costs. Accordingly, it is necessary to have the right composition of fertilisers in order to obtain high yields and affordable budgets.

Plants fertiliser combinations issue has been done by several methods such as goal programming [2], and meta-heuristic algorithms [7]. This study uses a genetic algorithm to optimise the fertiliser composition since it has been known as a powerful optimization method [8][9][10][11]. The method is used to produce optimal results with the reproduction processes such as determining population size, crossover rate, chromosome, reproductive and fitness value. The results of the genetic algorithm process expected to help determine the optimal combination of fertiliser.

II. Materials and Methods

Genetic algorithms or evolutionary algorithms is an optimizing technique that mimics biological evolution. There are several individuals in the population. From several generations, the individual has a role as a parent, which acts as a principal reproduction which can produce offspring. The individual has evolved and the new individual can have a great chance to survive the natural selection and survive. Better offspring tend to be generated from the individual who has a good offspring or sometimes is not better, but more likely from generation to generation will form a better population [7]. Genetic algorithm stages include initialization, reproduction, evaluation and selection [12][13] [14], is shown in Figure 1.

In the genetic cycle will be discussed how the process of solving the problem in optimizing the composition of rice fertiliser. There are several steps to resolve the problem using Genetic Algorithms. These stages can be seen in Figure 2. The genetic algorithm process consists of the main processes which include initialization, reproduction, evaluation, and selection.

Initialization begins with creating certain chromosomes randomly that can represent the solution of problems that need to be solved. The structure of the chromosome is illustrated in Table 1. It is also used as a generator of new solution set by randomly composed of a number of chromosome strings
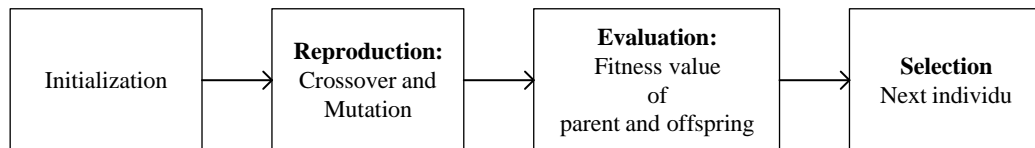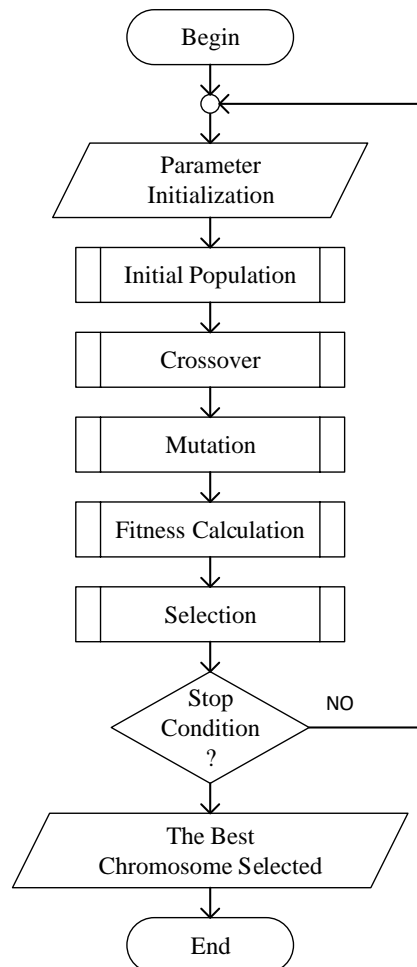

Fig. 1. Cycle diagram of genetic algorithm

Fig. 2. Flowchart of genetic algorithm

Table 1. Implementation of chromosome representation

Individual	Chromosome									Total Price (Rp)	Fitness
	1	2	3	4	5	6	7	8	9		
P1	1	5	8	6	2	4	7	9	3	623,800	1.603

and use it as a population. In this case, it needs a population size (popSize). The value contained in popSize can be used to determine the number of individuals/chromosomes which can accommodate within a population. The number of chromosome representations is determined by the choice of fertilizer available.

There are nine chromosomes formed, five initial chromosomes are used to calculate the amount of fertilizer, content, total price needed and fitness value. Reproduction is used to form or produce offspring from individuals in the population. There are two kinds of genetic operators here: crossover and mutation. This study uses one cut point crossover. Crossover levels must be determined so that the value used to express the ratio of offspring can be produced in the process crossover to produce *cr(crossover rate) x popSize* of offsprings. Illustration of the crossover process can be seen in Table 2 and the steps can be seen in Figure 3.

Reciprocal exchange mutation is used in this study. The mutation rate (*mr*) must be determined in advance to get a comparison between the amount of offspring that can be generated, and the parent involved in the process. The number of offspring that produced can be calculated by *mr* x *popSize*. Illustration of the mutation process can be seen in Table 3 and the steps can be seen in Figure 4. Evaluation is used for calculating the fitness of each chromosome, the greater fitness results of the chromosome means it produces a better result as a potential solution. Within a chromosome always

Table 2. Ilustration of crossover

Individual	Chromosome								
	1	2	3	4	5	6	7	8	9
P1	**1**	**5**	**8**	6	2	4	7	9	7
P3	**7**	**6**	**3**	**2**	**4**	5	1	8	9
C1	1	5	8	7	6	3	2	4	9
Random parents: 1.5									
Cutpoint : 3									
P2	**1**	**2**	**9**	4	5	6	7	3	8
P5	2	**5**	**8**	**4**	9	**3**	1	6	**7**
C2	1	2	9	5	8	4	3	6	7
Random parents: 2.3									
Cutpoint : 3									

Table 3. Ilustration of mutation

Individual	Chromosome								
	1	2	3	4	5	6	7	8	9
P1	**1**	5	8	6	2	**4**	7	9	3
C3	4	5	8	6	2	1	7	9	3
Random parents: 1									
Random point : 1 and 6									
P4	7	**4**	9	5	6	**2**	1	3	8
C4	7	2	9	5	6	4	1	3	8
Random parents: 4									
Random point : 2 and 6									
P5	2	5	8	4	**9**	3	1	6	7
C5	2	5	8	4	1	3	9	6	7
Random parents: 2									
Random point : 3 and 6									

```
                        ┌─────────┐
                        │  Begin  │
                        └─────────┘
                             ↓
                   ╱─────────────────╲
                  ╱  For i=0,          ╲
                  ╲  i < length of     ╱
                   ╲  crossover       ╱
                     └──────────────┘
                             ↓
                  ┌────────────────────┐
                  │ Random parent as   │
                  │ parent of crossover│
                  └────────────────────┘
                             ↓
                   ┌──────────────────┐
                   │  Cuttpoint = 3   │
                   │  Initialize array│
                   └──────────────────┘
                             ↓
                    ╱────────────────╲
                   ╱   For m=0,        ╲
                   ╲   m < cuttpoint   ╱
                     └───────────────┘
                             ↓
                    ┌─────────────┐
                    │ Add parent 1│
                    └─────────────┘
                             ↓
                        ( m++ )
                             ↓
                   ╱──────────────────╲
                  ╱  n=0, n < length of ╲
                  ╲  chromosome         ╱
                    └──────────────────┘
                             ↓
                    ┌─────────────┐
                    │ Add parent 2│
                    └─────────────┘
                             ↓
                        ( n++ )
                             ↓
                  ┌───────────────────┐
                  │ Calculate fitness │
                  └───────────────────┘
                             ↓
                        ( i++ )
                             ↓
                        ┌───────┐
                        │  End  │
                        └───────┘
```
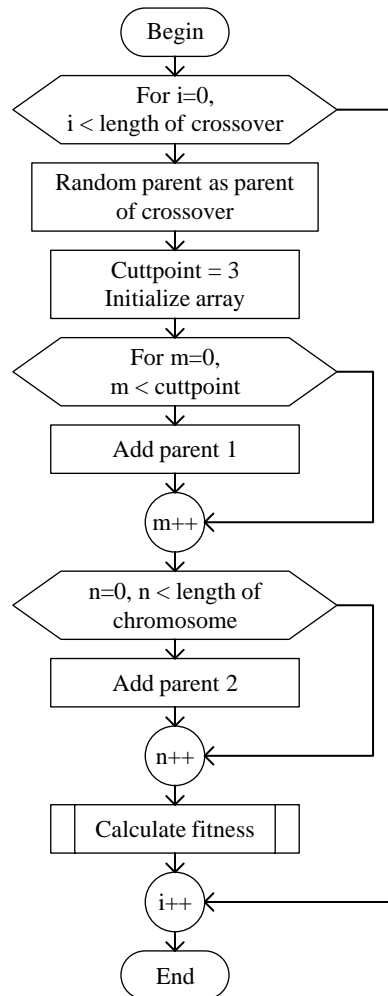
Fig. 3. Flowchart of crossover process

contains the value of fitness and some other property of individuals who have a gene called formers such as name, age, address, and another period of property. Fitness values can be expressed in (1).

$$\text{Fitness} = 1000 / ((\text{Total price}/1000) + (1000 * \text{penalty})) \tag{1}$$

Fertilizer composition always has ideal value requirements for the elements it contains (such as the need for Nitrogen (N) content, Phospor (P) content, and Potassium (K) content. Every violation of these ideal values is stated as a punishment written in (2).

$$\text{Penalty} = \text{need of N} + \text{need of P} + \text{need of K} \tag{2}$$

Selection is used to select the individual that can be sustained for the next generation coming from the set of the population and offspring. Probability function used to select an individual who needs to be maintained. Individuals with higher fitness values have greater chances of being elected to represent the next generation. In this study, the selection used is binary tournament selection. And flowchart of selection process is represented in Figure 5.

III. Results and Discussions

Implementation of this study using the programming language C# desktop-based. The interface made as easy as possible so that users only enter some parameter values to get the results of the optimization. There are four interfaces that can be accessed by the user, data of fertiliser, fertiliser recommendation reference, input and calculation results of genetic algorithms, and genetic algorithms computation.
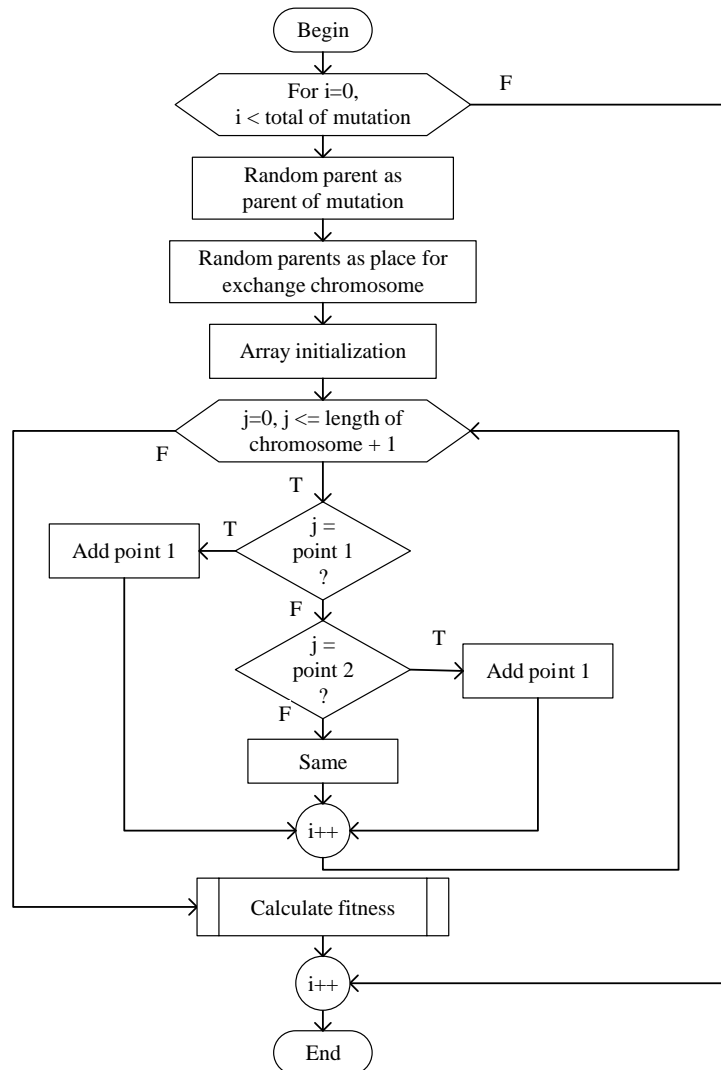
Fig. 4. Mutation flowchart

User inputs the parameter values in the main interface on the menu input and calculation results of genetic algorithms and genetic algorithms. Input interface and display the results of the genetic algorithm in Figure 6. Afterwards, users can see the process of genetic algorithm calculation that occurs from the input. Interface genetic algorithm calculation displayed in Figure 7.

The population size test is conducted to determine the size of the population which can produce the best fitness as well as the effect of population size on the value of fitness. For testing the generation of a population was 10 and multiplied by 10, from 10 to 100 population. While the value of the crossover rate and mutation rate used by $cr = 0.4$ and $mr = 0.6$. Each experiment will be performed 10 times and average fitness will be calculated

Figure 8 shows that the fitness value increases in population size 10 to 20 and produces the same fitness value on the size of 20 to 100 after 10 attempts. In the experiment, population size produces an average of the best fitness with the best fitness value of 1.603. The higher fitness produced better results for the calculation of genetic algorithms. It can be concluded that the value of fitness influenced by the size of the population, the greater the value of the size of the population, no increases fitness values occurred and may affect the computation time that occurs in the software. These experiments generate the optimal solution in the population from 20 to 100 with an average fitness of 1.603.

Generation testing are conducted to determine the size of the generation which can produce the best fitness as well as the influence of the generation number on the value of fitness. For testing, the population amount was 10 and multiplied by 10 to 100 generations. While the value of the crossover rate and mutation rate used by $cr = 0.4$ and $mr = 0.6$; each experiment will be performed 10 times and average fitness will be calculated.
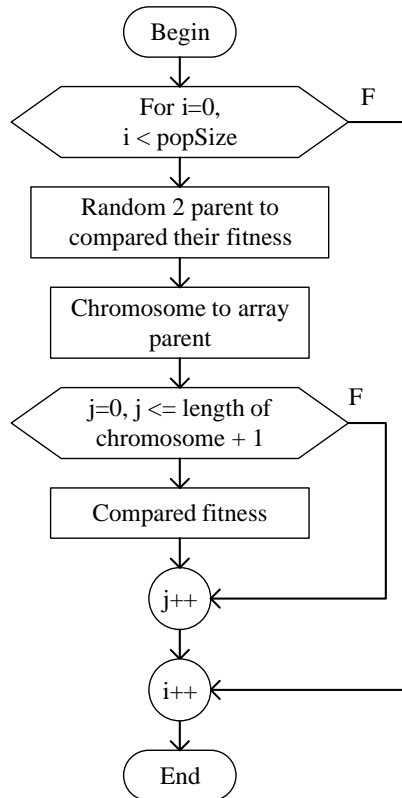
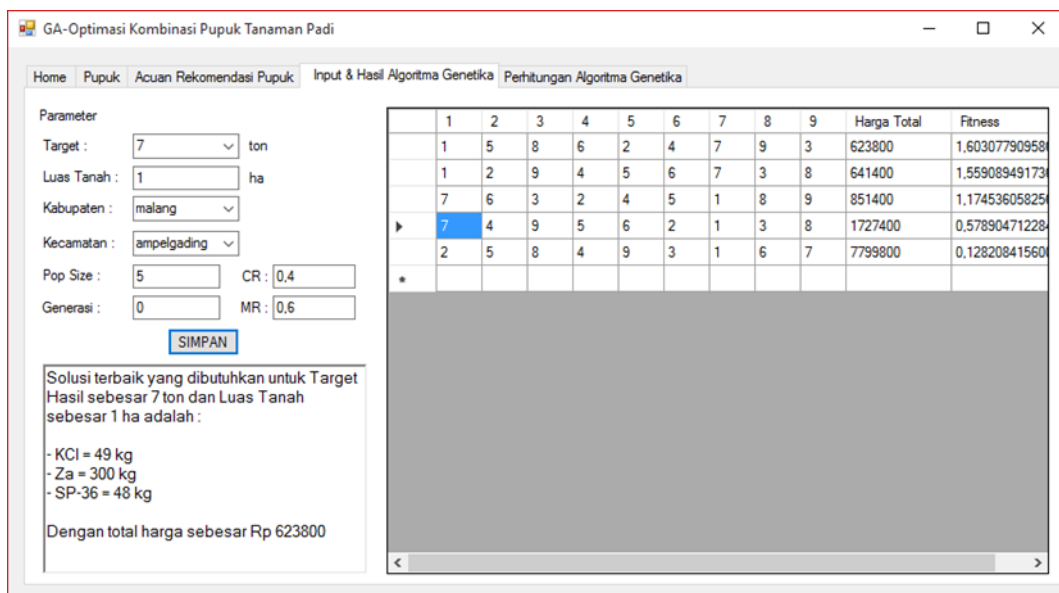Fig. 5. Flowchart of selection process

Fig. 6. Graphical user interface of genetic algorithm input

On the Figure 9 show that the fitness value increased in the size of the generation of 10 to 30 and 70 to 100, while generation reached the lowest fitness value on the size of the generation of 10. The trial generation size produces an average generation of fitness of 1.603 and fitness 1.603 as the best value. The higher the fitness value produced, the better the results of the calculation of genetic algorithms. It can be concluded that the value of fitness is influenced by the size of the generation, the greater the value of the size of the generation of better fitness values occurred. These experiments generated the optimal solution in the population 30 to 50 and 70 to 100 with an average of 1.603.
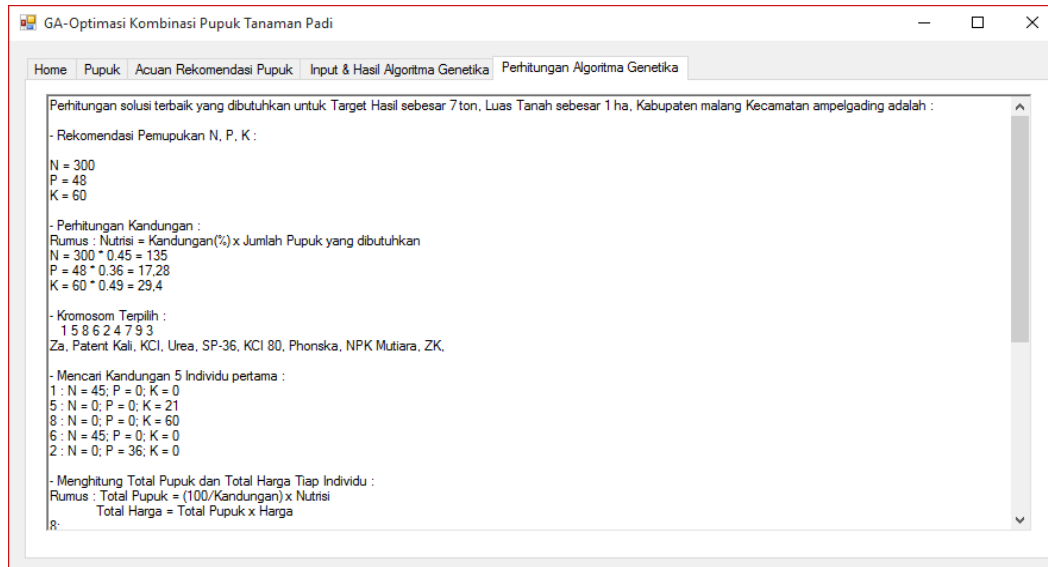
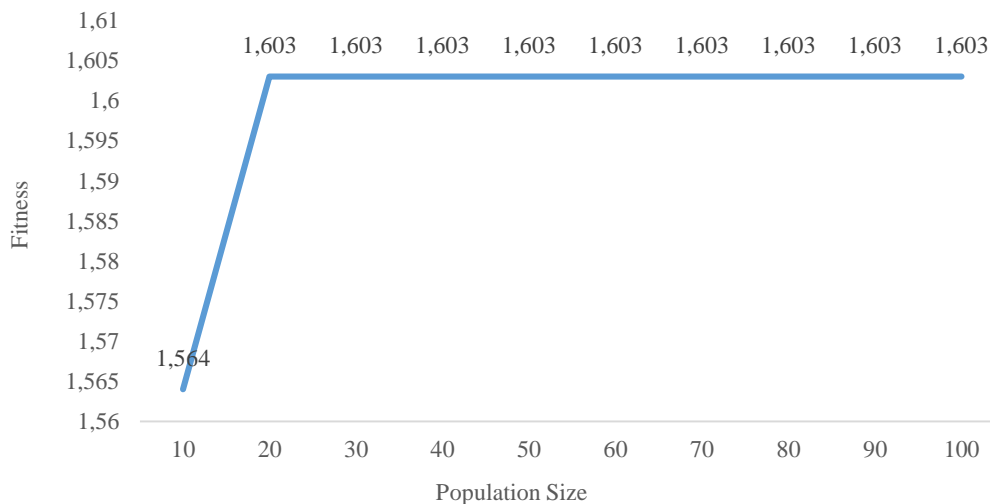Fig. 7. Graphical user interface of genetic algorithm calculation

Fig. 8. Graph of population size testing results

Cross over and mutation rate is conducted to determine the size of the crossover rate and mutation rate which can produce the best fitness and influence the amount of crossover rate and mutation rate of the value of fitness; for the testing population amount used was 10 and the generation was 10. While the value of the crossover rate and mutation rate used added by 0.1 from 0 to 1. Each experiment will be performed 10 times and calculated the average fitness.

On Figure 10, it is shown that the combination of crossover rate and mutation rate affect the fitness. The fitness value of the highest increases in the size of the crossover rate and mutation rate of 0.1 and 0.9. While generating the lowest fitness value on the size crossover rate and mutation rate of 0.9 and 0.1. Trial size crossover rate and mutation rate produce an average fitness of 1.603 and fitness of 1.603 as the best value. The higher the fitness value produced the better the results of the calculation of genetic algorithms. It can be concluded that the value of fitness was influenced by the size of the crossover rate and mutation rate. These experiments generate the optimal solution in the population 30 to 50 and 70 to 100 with an average of 1,603.

Best parameters s test has 3 different scenarios using 3 different cases. Genetic algorithm parameter values used are the best parameters for the results of population trials, generation, crossover rate and mutation rate. The first thing is to test the best parameters. The population size used was 20 and the generation size was 30. While the crossover rate was 0.1 and the mutation rate was 0.9. Each experiment will be carried out 3 times.
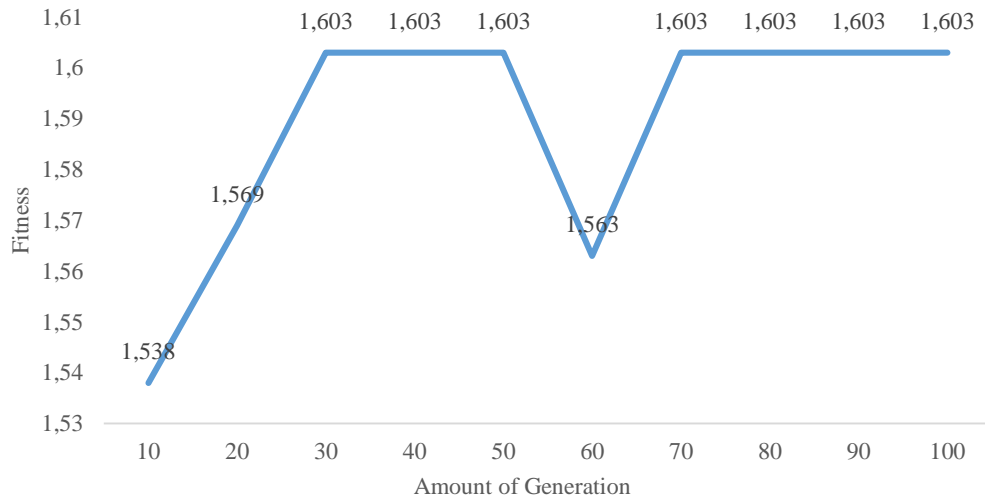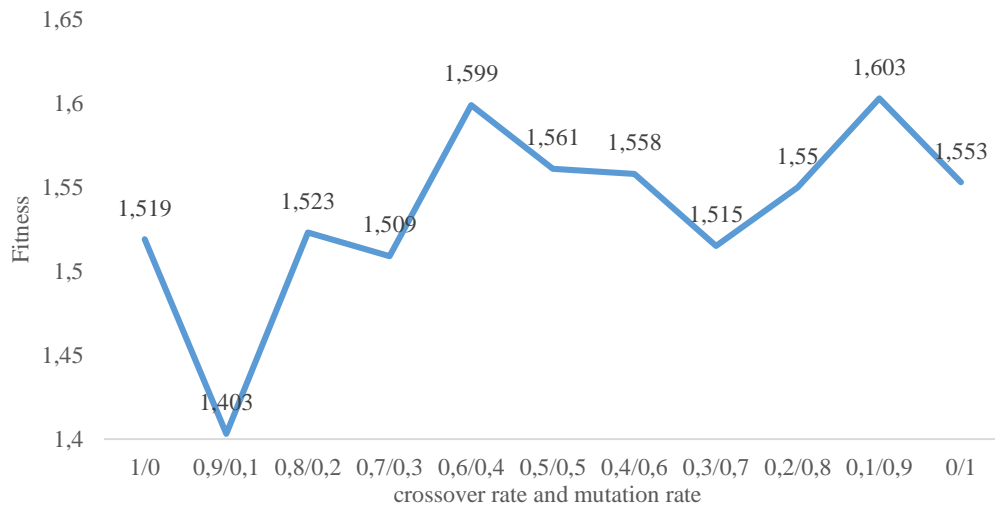
Fig. 9. Generation testing results

Fig. 10. Graph of crossover rate and mutation rate testing results

- Trial case 1. Target of 7 tons, Land Area of 1 ha, Malang Regency and Ampel Gading District. The best solution needed is: KCI = 49 kg, Za = 300 kg, SP-36 = 48 kg, Total Price IDR 623,800.00. The best chromosomes obtained in case 1 trial are shown in Table 4.
- Trial case 2. Target is 8 tons, Land Area is 1 ha, Malang Regency and Ampel Gading District. The best solution needed is: KCI = 58 kg, Za = 300 kg, SP-36 = 54 kg, Total Price IDR 655,600.00. The best chromosomes obtained in case 2 trial are shown in Table
- Trial case 3Target is 9 tons, Land Area is 1 ha, Malang Regency and Ampel Gading District. The best solution needed is: KCI = 66 kg, Za = 300 kg, SP-36 = 60 kg, Total Price IDR 685,200.00. The best chromosomes obtained in case 2 trial are shown in Table

In case 1, for Target 7 tons, Land 1 ha, in Malang and District Ampel Gading can be seen that the best fitness shown is 1.603. The total of the best price is Rp 623,800. In case 2 for Target 8 tons, Land 1 ha, in Malang and District Ampel Gading can be seen that the best fitness shown is 1.525. The total of the best price is Rp 655,600. While the test solution in case 3 for Target 9 tons, Land 1 ha, in Malang, and District Ampel Gading can be seen that the best fitness shown is 1.459. The total of the best price is Rp 685,200. Measurement of the best solutions on the issue composition of rice plant fertilizer is to look at the highest fitness value. The highest fitness is obtained from the smallest penalty value with minimum price. From the testing that has been done, a genetic algorithm can be used to solve optimisation problems fertiliser composition with a good rice crop.

Table 4. Chromosomes obtained from case 1 trials

Individual	Chromosome									Total Price (Rp)	Fitness
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		
P1	2	4	8	1	9	3	6	7	5	623,800	1.603

Table 5. Chromosomes obtained from case 2 trials

Individual	Chromosome									Total Price (Rp)	Fitness
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		
P1	2	8	4	1	3	9	7	5	6	655,600	1.525

Table 6. Chromosomes obtained from case 3 trials

Individual	Chromosome									Total Price (Rp)	Fitness
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		
P1	1	2	5	8	4	6	3	7	9	685,200	1.459

IV. Conclusion

From the experiments conducted in this study, it can be concluded that a genetic algorithm can be used to compose the rice plant fertiliser. The settlement of this problem in the genetic algorithm is to use the representation of a real code chromosome, reproduction method used are one cut point crossover and random mutation, and binary tournament for the selection process. Moreover, the quality of the best solution in this study was measured by looking at the highest fitness value. The highest fitness value having the smallest penalty value and the minimum price. On testing can be seen that the best solutions for the fitness value of the result is 1,603 and the price of Rp 623,300.-. Furthermore, tests have been conducted to examine the influence of genetic parameters on the fitness value are test population, test of generation amount and test combinations of crossover rate (cr) and the mutation rate (mr) and using the best parameters at the trial population produces optimal population values on population size 20 and produce the best fitness of 1,603. In the test generation produce optimal value generation to generation size 30 and produce the best fitness of 1,603. While the trials combine crossover rate (*cr*) and the mutation rate (*mr*) is best found in combination *mr cr* 0.1 and 0.9 with the fitness value 1,603. The conclusion is the higher value of the population will produce better fitness value, the higher the value generation does not make much impact to produce the best fitness value and the higher value of the crossover rate (*cr*) and the mutation rate (*mr*) affect the value fitness but it is not necessarily can yield optimal results. The time needed to process the algorithm effect on the higher value it will be a long process of computation.

Additional information

No additional information is available for this paper.

References

[1]   P. Lingga and M. Marsono, *Petunjuk Penggunaan Pupuk*. Jakarta: Penebar Swadaya, 2007.

[2]   N. Hassan, K. Hassan, S. Yatim, and S. Yusof, "Optimizing fertilizer compounds and minimizing the cost of cucumber production using the goal programming approach," *Am. J. Sustain. Agric.*, vol. 7, no. 2, pp. 45–49, 2013.

[3]   N. C. Lu, Y. H. Cheng, Y. T. Wang, and J. Cheng, "Dynamic propagation problems on mode III asymmetrical interface crack," *Harbin Gongye Daxue Xuebao/Journal Harbin Inst. Technol.*, vol. 39, no. 11, pp. 1710–1714, 2007.

[4]   P. Du Jardin, "Plant biostimulants: definition, concept, main categories and regulation," *Sci. Hortic. (Amsterdam).*, vol. 196, pp. 3–14, 2015.

[5]   R. Budiono, P. G. Adinurani, and P. Soni, "Effect of new NPK fertilizer on lowland rice (Oryza sativa L.) growth," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 293, no. 1, 2019.

[6]   V. B. Bado, K. Djaman, and V. C. Mel, "Developing fertilizer recommendations for rice in Sub-Saharan Africa, achievements and opportunities," *Paddy Water Environ.*, vol. 16, no. 3, pp. 571–586, 2018.

[7]   N. Sivakumar, T. Amudha, and N. Thilagavathi, "Development of a Novel Bio Inspired Framework for Fertilizer Optimization," in *2019 Amity International Conference on Artificial Intelligence, AICAI 2019*, 2019, pp. 175–181.

[8]   A. N. Fauziyah and W. F. Mahmudy, "Hybrid Genetic Algorithm for Optimization of Food Composition on Hypertensive Patient," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, pp. 4673–4683, 2018.

[9]   N. Metawa, M. K. Hassan, and M. Elhoseny, "Genetic algorithm based model for optimizing bank lending decisions," *Expert Syst. Appl.*, vol. 80, pp. 75–82, Sep. 2017.

[10]   A. Hiassat, A. Diabat, and I. Rahwan, "A genetic algorithm approach for location-inventory-routing problem with perishable products," *J. Manuf. Syst.*, vol. 42, pp. 93–103, Jan. 2017.

[11]   C. Bharathi, D. Rekha, and V. Vijayakumar, "Genetic algorithm based demand side management for smart grid," *Wirel. Pers. Commun.*, vol. 93, no. 2, pp. 481–502, Mar. 2017.

[12]   A. Sharma, R. Preet, P. Singh, and P. Lehana, "Evaluation of the accuracy of genetic algorithms in orientation estimation of objects in industrial environment," *Int. J. Sci. Tech. Adv.*, vol. 1, no. 4, pp. 7–14, 2015.

[13]   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, p. 113, 2018.

[14]   G. Prasad, D. Singh, A. Mishra, and V. H. Shah, "Genetic algorithm performance assessment by varying population size and mutation rate in case of string reconstruction," *J. Basic Appl. Eng. Res.*, vol. 4, no. 2, pp. 157–161, 2017.