An Adaptive Approach to Controlling Parameters and Population Size of Evolutionary Algorithm

Syafiul Muzid
Department of Information System, Faculty of Engineering, Universitas Muria Kudus, Indonesia
Email: syafiul.muzid@umk.ac.id

Abstract. Evolutionary algorithm (EA) is a part of evolutionary computing inspired by the theory of biological evolution. Evolutionary algorithms have three parameters that must be defined, namely population size, probability of crossing over, and probability of mutation. The absence of standard rules in setting the value of these parameters becomes a difficulty in utilizing evolutionary algorithms to solve optimization problems so that it can cause early convergence in obtaining local optimum values. This research was conducted to find a way to adjust the value of these parameters by using fuzzy logic to determine the probability of crossing and mutation probability and the method of determining population size based on the best fitness value. This method is called Population Resizing on Fitness Improvement Fuzzy Evolutionary Algorithm (PRoFIFEA). The testing of this method uses the problem of Traveling Salesman Problem (TSP) compared to the standard evolutionary algorithm method which states that the PRoFIFEA method is able to find a more optimal solution.

1. Introduction
Evolutionary algorithms is a soft computing techniques that are often used in solving optimization problems. Evolutionary algorithm is one part of evolutionary computing that is use to solve the problem of genetic population-based optimization inspired by the theory of biological evolution. Evolutionary algorithms have three parameters that must be define, namely population size, probability of crossing over, and probability of mutation. These three parameters must be carefully defined so that there is no optimum early or local convergences which is where individuals in the population converge on a local optimum solution so that the optimum results cannot be found. The absence of standard rules in setting the value of these parameters becomes a difficulty in the use of evolutionary algorithms to solve problems [1].

Population size is an important parameter in genetic algorithm algorithms [2][3]. The population size of classical genetic algorithms is determined constantly. Its size remains constant throughout the process [4]. If the size of the population is too small it will allow early convergence, and if it is too large it will result in the time needed for the genetic algorithm to produce the best solution. In determining the size of the population, appropriate intuition is needed and depends on the level of difficulty and complexity of the problem. The more complex and difficult the greater the population size [5].

The difficulties is to setting these parameter values is the utilization of the Population Resizing on Fitness Improvement Fuzzy Evolutionary Algorithm (PRoFIFEA) method, which uses the fuzzy logic method to determine the probability of crossing and mutation probability and the method of determining population size based on the best fitness value using the Population method Resizing on Fitness Improvement Genetic Algorithm (PRoFIGA) in the next generation.
In this paper, we will discuss the development of PRoFIFEA models that can adaptively adjust the three parameters, namely population size, probability of crossing and probability of mutation. To test the model developed it will be tested on optimization problems in the search for the shortest route or the so-called Travelling Salesman Problem (TSP).

2. Adaptive of Evolutionary Algorithm

2.1. Evolutionary algorithm

Soft Computing is an approach model for computing by imitating human reason and having the ability to reason and learn in an environment full of uncertainty [6]. Soft Computing is used to solve problems by using approaches in reasoning. The approach process can be carried out functionally or through random search.

One of the soft computing components are fuzzy systems, evolutionary computing or evolutionary algorithms, and reasoning with probability. The use of a soft computing component in problem solving can produce an acceptable result, but not necessarily a maximum [7].

The evolutionary algorithm is a soft computing model introduced by John Holland of the University of Michigan in 1975, where genetic algorithms are heuristic search techniques based on biological evolutionary mechanisms that Darwinian theories and genetic operations on chromosomes and are often used to solve optimization problems. The evolutionary algorithm consists of six main stages, namely: (1) Representation of chromosomes, (2) Population initialization, (3) Calculation of evaluation functions, (4) Selection process, (5) Genetic operators include crossover and mutation operators and (6) Determination of genetic algorithm control parameters, namely: population size, probability of crossing over, and probability of mutation.

2.2. Fuzzy evolutionary algorithm

Fuzzy evolutionary algorithm is a computational technique combined with genetic algorithms and fuzzy systems where the parameters used in evolutionary algorithms are generated from a fuzzy system. One of the fuzzy evolution algorithm models is the Xu model which uses input and output variables very clearly and easily understood [8].

Xu and Vukovich uses the fuzzy systems to determine parameters used by genetic algorithms. The developed model uses 2 fuzzy systems to determine the value of the probability of crossing over and the value of the probability of mutation [9]. Both fuzzy systems use population size and number of generations as input. Fuzzy rules used in determining the output value based on the conditions of the input in the fuzzy system in the Xu model show in Table 1 and Table 2.

| Table 1. Rules for probability of crossing over (pc) | Population Size |
|---------------------------------------------------|-----------------|
| Pc                                                | Small | Medium | Large |
| Generation                                        |       |        |       |
| Short                                             | Medium| Small  | Small |
| Medium                                            | Large | Large  | Medium|
| Long                                              | Very  | Very large | Large |

| Table 2. Rules for probability of mutation (pm)  | Population Size |
|-------------------------------------------------|-----------------|
| Pm                                               | Small | Medium | Large |
| Generation                                       |       |        |       |
| Short                                            | Large | Medium | Small |
| Medium                                           | Medium| Small  | Very  |
| Long                                             | Small | Very   | Very  |

The use of the fuzzy evolution algorithm of the Xu model has a weakness in determining the input value constraints especially in population size [10]. If the population size used is fixed then the value of the
probability of crossing over and the probability of mutation will not change so that it will allow for early convergence.

2.3. Population sizing theory
Population Resizing on Fitness Improvement Genetic Algorithm (PRoFIGA) used to control population size based on the best fitness value in each generation [2][3][11]. These changes mean that the size of the population can increase or decrease to a smaller size. The study explained that population size is an important parameter in genetic algorithms. If the size of the population is too small it will allow early to convergence, and if it is too large it will result in the time needed for the genetic algorithm to produce the best solution. The PRoFIGA model has the following 4 conditions:

a) Maximum fitness value increases
If the maximum fitness value in the current generation is better than the previous generation's maximum fitness value, the population size will increase by the following formula:

$$X = \text{increaseFactor} \times (\text{maxEvalNum} - \text{currEvalNum}) \times \frac{\text{maxFitness}_{\text{new}} - \text{maxFitness}_{\text{old}}}{\text{initMaxFitness}}$$

Where X is the percentage of growth for population size. increaseFactor is the value of the growth parameter that is in the interval (0.1), maxEvalNum is the maximum generation used in the running process, currEvalNum is the current generation, maxFitnessnew is the maximum generation fitness value of the current generation, maxFitnessold is the maximum fitness value of the previous generation and initMaxFitness is the maximum expected fitness value.

b) Maximum fitness value decreases
If the maximum fitness value of the current generation decreases the maximum fitness value of the previous generation, then the size of the population will shrink by the value of the depreciation percentage defined by the user. The percentage percentage of depreciation is a small percentage between 1 to 5 percent (1-5%) [2].

c) The maximum fitness value is the same
If the maximum fitness value of the previous generation and the current generation is of the same value, then the size of the population will shrink by the percentage value of the shrinkage defined by the user.

d) The maximum fitness value is the same within a certain time limit
Whereas if the maximum fitness value does not change or equal from the previous generation for several generations that have been determined then the size of the population will increase. Users can determine the percentage growth value equal to the condition if the maximum fitness value increases.

2.4. Adaptive of population size and parameters
PRoFIFE is a method used to overcome problems that arise when using genetic algorithms in solving problems. PRoFIFE is the development of Xu's model Fuzzy Evolutionary Algorithm it is combined with the Population Resizing model on Fitness Improvement Genetic Algorithm (PRoFIGA) [1]. The system architecture used in PROFIFEA can be see in Figure 1.

![Figure 1. PRoFIFE system architecture](image-url)
PRoFIFEA uses condition rules to determine the size of the new population using the condition rules used by the PRoFIGA concepts which consists of 4 (four) condition rules. In the use of evolutionary algorithms, the level of individual chromosome diversity in the population must be maintained properly, so that there is no individual homogeneity. Because this can lead to early convergence. The method or method used in PRoFIFEA to produce more optimum solution and prevent homogeneity of individuals or chromosomes in the population is as follows [1]:

a) If the size of the new population increases then new individuals will generate randomly according to the type of chromosome used by the number of population size increases.

b) If the size of the new population shrinks then the new population is formed by the following formula:

- 30% of the new population is taken from the old population that has the best fitness value.
- 30% of the other new population is taken from the old population that has the worst fitness value.
- The remaining of 40% is taken randomly from the old population.

2.5. Traveling Salesman Problem (TSP)

Traveling Salesman Problem (TSP) is one of the classic optimization problems that is difficult to solve conventionally. The completion of the TSP is to obtain the shortest path. TSP can be completed exactly but must calculate all possible routes, then choose one of the shortest routes [12]. If there are a number of n cities to visit, there are n city combination that will be compared to the distance of each city. Which will require a long computing time if the number of cities that must be visited is increasing.

3. Research Method

In this research, the software development method used is to use a prototype method and begin with a literature study. The stages of the research method are as follows:

- Literature study
  
  Literature study is include studying, exploring, and quoting theories or concepts from a number of literature, both books, journals that are relevant to the topic, focus or research variables related to genetic algorithms, fuzzy systems, and fuzzy evolution algorithms of Xu models, PRoFIGA models and PRoFIFEA.

- System Development
  
  This stage is the stage where a new system begins to develop which starts by analyzing existing theories, theories related to FEA and PRoFIFEA theories. System designs done to design the process and interface of the system to be developed. The program created by writing program code in the form of function modules and developing a graphic user interface (GUI) as well as integration of these function modules. The resulting program then tested with a TSP problem.

3.1. System Design

The PRoFIFEA flowchart in research has the combined flow stages of the Xu fuzzy evolution algorithm model with the PRoFIGA flowchart. The design is illustrated in the form of a flowchart as shown in Figure 2.
Case study of the problem to be solved is the problem of Traveling Salesman Problem (TSP) with a total of 20 cities with the provisions that several cities have only one destination route (one way). The fitness value used in this problem is finding the shortest distance as an example formula for calculating the distance between two cities in equation 4.

$$
||A - B|| = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}
$$

While the city points used in the TSP test are shown in table 3. While the list of one way routes can be seen in table 4 which means cities that are registered in one way routes can only go to predetermined cities.

| City  | Coordinate | City  | Coordinate |
|-------|------------|-------|------------|
| City-1 | 1 8        | City-11 | 17 16  |
| City-2 | 2 15       | City-12 | 17 28  |
4. Result and Discussion
In solving the TSP problem, it is necessary to set some parameters in the genetic algorithm as follows:
- The number of variables used is 1 piece, because it is a sequence of routes or city routes.
- The number of bits for each variable is 20 according to the number of cities.
- Initial population size: 100.
- Number of generations: 100 generations.
- The maximum expected fitness is 1 because the process carried out is minimization, the fitness value used is 1 / the distance.
- Minimum population: 10,
- Maximum population: 200,
- Growth factors: 0.1.
- Depreciation factor: 0.05.
- The generation limit does not improve the fitness value of 5 generations.
- The type of chromosome used is Permutation.
- The individual selection method is roulette wheel selection.
- The method of crossing over is Order.
- The mutation method is Insert.
- The best number of individuals maintained in the next generation is 2 generation.
- The running results graph used is the best individual chart.

To test whether the results found using the genetic algorithm PRoFIFEA model are better than the results of the standard genetic algorithm, then testing is done with a case study and setting the same parameters between the standard genetic algorithm and the PRoFIFEA genetic algorithm. After the running process, it can see the results found using the genetic algorithm PRoFIFEA model produces a better / shorter route than the results of the standard genetic algorithm. A comparison of the results of running of the two genetic algorithm models can see in Figure 3. The best fitness value of the standard genetic algorithm is show in a straight line in blue, while the best fitness value in the genetic algorithm in the PRoFIFEA model is show in the dots in red.

While Figure 4 shows the route path and route length resulting from the two genetic algorithm models in the Windows Command. Fitness value generated by the genetic algorithm PRoFIFEA model is 0.0069 that is more optimal than the fitness value generated by the standard genetic algorithm that is 0.0058. The length of the route produced by the standard genetic algorithm is 171.6663 while the length of the route produced by the genetic algorithm of the PRoFIFEA model is 143.4062 with the route: 16 17 11 15 19 18 14 9 10 7 6 5 1 4 2 3 8 12 13 20.
Based on the running process above, it can be conclude that the genetic algorithm PRoFIFEA model can produce a final solution that is more optimal than the final solution of the standard genetic algorithm.

5. Conclusions and Future Work

5.1. Conclusions

Based on the implementation and testing of this research, there can be several things of conclusions. The conclusions of this study are as follows:

- Hybrid technology between genetic algorithms and fuzzy logic and the PRoFIGA technique, or the so-called PRoFIFEA model, can improve the final solution or solution found.
- Parameter dynamization in PRoFIFEA makes it easy to solve optimization problems especially in the case of Traveling Salesman Problem (TSP) so that users have no difficulty in determining the value of the parameters used in the running process.

5.2. Future work

Research that has done still has many shortcomings that need to correct so that it requires suggestions for improvement. Some suggestions for further research are as follows:

- Further research is need to determine parameter boundary values used in the PRoFIFEA model such as minimum population size and maximum population size because the larger population size will cause longer running times.
- Hope that there will be other research for some other optimization problems aside from the Traveling Salesman Problem (TSP).

References

[1] S. Muzid, “Dinamisasi Parameter Algoritma Genetika Menggunakan Population Resizing On Fitness Improvement Fuzzy Evolutionary Algorithm,” Pros. SNATIF Ke-1, pp. 145–152, 2014.
[2] A. E. Eiben, E. Marchiori, and V. A. Valkó, “Evolutionary algorithms with on-the-fly population size adjustment,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3242, pp. 41–50, 2004.
[3] A. E. Eiben, Z. Michalewicz, M. Schoenauer, and J. E. Smith, “Parameter control in evolutionary algorithms,” Stud. Comput. Intell., vol. 54, pp. 19–46, 2007.
[4] E. B. Nababan, O. S. Sitompul, and Y. Cancer, “Genetic Algorithms Dynamic Population Size with Cloning in Solving Traveling Salesman Problem,” Data Sci. J. Comput. Appl. Informatics, vol. 2, no. 2, pp. 87–100, 2018.
[5] F. G. Lobo and C. F. Lima, “Adaptive population sizing schemes in genetic algorithms,” Studies in Computational Intelligence, vol. 54, Heidelberg: Springer-Verlag Berlin and Heidelberg GmbH & Co. K, pp. 185–204, 2007.
[6] J. S. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. London: Prentice Hall, 1997.

[7] S. Kusumadewi, *Analisis & Desain Sistem Fuzzy Menggunakan Toolbox Matlab*. Yogyakarta: Graha Ilmu, 2002.

[8] M. Tettamanzi, A., Tomassini, “Soft Computing,” *Technometrics*, vol. 44, no. 2, pp. 199–199, 2002.

[9] H. Y. Xu and G. Vukovich, “Fuzzy genetic algorithm with effective search and optimization,” in *Proceedings of the International Joint Conference on Neural Networks*, 1993, vol. 3, pp. 2967–2970.

[10] I. Muzid, S., Kusumadewi, S., dan Paputungan, “Matlab Toolbox for Fuzzy Evolutionary Algorithm.” Universiti Sains Malaysia, Kedah, 2009.

[11] T. Listyorini and S. Muzid, “Population resizing on fitness improvement genetic algorithm to optimize promotion visit route based on android and google maps API,” in *AIP Conference Proceedings*, 2017, vol. 1855.

[12] S. Puspitorini, “Penyelesaian Masalah Traveling Salesman Problem dengan Jaringan Saraf Self Organizing,” *Media Inform.*, vol. 6, no. 1, pp. 39–55, 2008.