International Journal of Information System & Technology
Akreditasi No. 36/E/KPT/2019 | Vol. 3, No. 2, (2020), pp.302-308

# The Determination of Shortest Path Using Genetics Algorithm Assisted Matlab

*Hendra Cipta[1], Rina Widyasari[2]*
*[1,2]Mathematics Study Program, Universitas Islam Negeri Sumatera Utara, Medan*

*[1]hendracipta@uinsu.ac.id, [2]rina_widyasari@uinsu.ac.id*

***Abstract***

*The problem in taking the shortest path for a road driver is an interesting thing. This paper is an explanation of how to find the shortest path using a genetic algorithm in order to achieve the best solution. Then it shows how to implement the genetic algorithm using MATLAB program. As an example, it is given a connected directed graph containing 20 vertices and 41 arcs where these vertices are assumed to be cities..*

***Keywords:*** *Shortest Path, Genetic Algorithm, Directed Graph*

## 1. Introduction

The problem of the shortest path is a routing network where a road driver wants to determine the shortest path between two cities based on the available alternative paths, where the destination city is only one. This problem itself uses graph representation to model the represented problem so that it is easier to solve. The problem is how to visit the vertices in the graph from the initial vertices to the final vertices with the minimum weight, where in this case the weight used is the distance and the cities visited are assumed to be connected graphs between one city and another [8].

Several algorithm methods that have been developed to solve the shortest path problem include the Djikstra algorithm, the Floyd-Warshall algorithm, and the Bellman-Ford algorithm [1]. In the 70s, a new algorithm known as genetic algorithm (GA) appeared, which is one of the branches of Artificial Intelligence (AI). This genetic algorithm was introduced by John Holland from the University of Michigan which was later popularized by one of his students, David Goldberg, so that genetic algorithms began to be widely used in various fields, including to solve optimization problems [9].

In genetic algorithms, there is something called the reproduction process, the interbreeding process, the mutation process and the selection process. The reproduction process is a process to form new offspring by inheriting the same traits from the parent chromosome or in other words, a duplication process that does not eliminate the old chromosome characteristics [2]. There are several recommendations that can be used, to determine the parameter value [3], among others: For problems that have a large enough solution area, De Jong recommends the control parameter values: (popsize; pc; pm) = (50; 0.6 ; 0.001), if the average fitness of each generation is used as an indicator, then Grefenstette recommends: (pop size; pc; pm) = (30; 0.95; 0.01), if the fitness of the best individual is monitored in each generation, then the proposal is: (popsize; pc; pm) = (80; 0.45; 0.01).

In solving the shortest path problem that is designed using a genetic algorithm it is necessary to perform a needs analysis. In detail, the analysis of the shortest path genetic algorithm is described in section 2.After carrying out the analysis, the genetic algorithm is implemented to obtain the best solution which is presented.

## 2. Research Methodology

In running an application including the input process and the output process, the simulation system for determining the shortest path includes problem identification to find

ISSN : 2580-7250

out what is the problem and what must be solved to apply genetic algorithms in the shortest path problem.

The input of the application in determining the shortest path, in the form of parameters required in the genetic algorithm, namely: Data on the number of vertices that are input (N), where N is determined by the user and where each point is a connected and directed graph. connected graph), the parameters are the size of the population (Popsize) = 50, the chance of crossover ($p_c$) = 0.6, the chance of mutation ($p_m$) = 0.05, the maximum generation = 100.The process needs to be carried out to determine the shortest path problem include: the process of making vertices, the process of determining the city distance, the process of calculating the fitness function, selection, crossover, mutation, to determining the population results, finally, the process of selecting the shortest path. Meanwhile, the output data obtained from the application process in determining the shortest path with a genetic algorithm is the shortest path of the 20 vertices and 41 arcs that have been determined along with the distance between the cities and the minimum path length and the average fitness graph is also obtained.

John Holland in Simple Genetic Algorithm said that to implement genetic algorithms in finding the shortest path, there are 5 things that must be done, namely chromosome coding, population initialization, determining fitness values, selection process, cross-breeding process and mutation process. The genetic algorithm procedure for determining the shortest path is as follows [4], [7]:

Genetic algorithm procedure for determining the shortest path
(1) Set parameter number of vertices (N), max. Generation, popsize and $p_c$.
(2) Enter the distance
(3) Chromosam coding
(4) Evaluate the firness value
    If the optimization criteria have not been achieved, go to (5)
    Else go to (8)
(5) Make a selection, go to (6)
(6) Perform crossovers, go to (7)
(7) Perform mutation, return to (4)
(8) Sort the shortest path

## 3. Result and Discussion

After analyzing the needs of the genetic algorithm to determine the shortest path, the algorithm is implemented according to the procedure and the program used in the application is the MATLAB R2017a program.

### 3.1. Chromosome Coding

Chromosome coding is a difficult process in genetic algorithms. This is because the coding process for each problem is different because not all coding techniques are suitable for every problem. This coding process produces a sequence which is then called a chromosome. In this case, permutation encoding is used. At this stage, the vertices to be visited are given serial numbers, then formed into a chromosome containing genes that represent the serial number of all the cities as shown below.

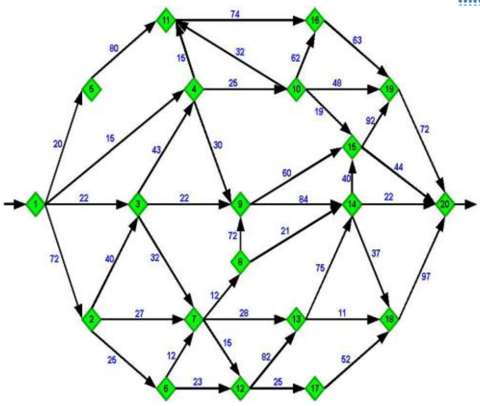

**Figure 1. Directed and weighted graph**

From Figure 1 above, the vertex distance can be represented in the form of a matrix, which is as follows:

$$A=\begin{bmatrix}
0&72&22&15&20&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&40&0&0&25&27&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&43&0&0&32&0&22&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&30&25&15&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&80&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&12&0&0&0&0&23&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&12&0&0&0&15&28&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&72&0&0&0&0&21&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&84&60&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&32&0&0&0&19&62&0&0&48&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&74&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&82&0&0&0&25&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&75&0&0&0&11&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&40&0&0&37&0&22&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&92&44\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&63&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&52&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&97\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&72\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

### 3.2. Initialization Population

In the individual initialization stage, a population containing a number of chromosomes will be generated. Each chromosome contains a number of genes. The input for this function is the population size (number of chromosomes in the population) and the number of genes in one chromosome. The population size depends on the problem to be solved and the type of genetic operator to be implemented. After the population size has been determined, then the chromosomes must be initialized in that population. Chromosome initialization is carried out randomly, however, we must pay attention to the solution domain and the existing problem constraints. In this population, there are members of the population called chromosomes which contain solution information of the many alternative solutions to existing problems. The individual evaluation stage aims to calculate the fitness value of an individual x. This function is very dependent on the problem to be solved. In this problem, what is discussed is determining the shortest path of 20 vertices and 41 arcs, therefore the fitness value that can be used is 1 divided by the total distance.

In this case, what is meant by total distance is the total distance between one city and another which is a solution. The variables used to find the fitness value are population, number of genes, and distance between cities on a chromosome. From the above problems, we do population initialization and individual evaluation as much as the population size is 50, the crossover probability and mutation chance are 0.6 and 0.05 respectively and the maximum generation is 100.

>> Popsize: 50

Pc: 0.600

Pm: 0.050
MaxGen: 100

### 3.3. Fitness Probability Value

The fitness probability is the calculation of each fitness value on each chromosome in a population against the total number of fitness values. The formula used is:

$$\frac{fitness\ value}{total\ fitness} \tag{1}$$

The cumulative value of the fitness probability obtained is 1, this happens because when viewed based on the definition of probability theory, the probability value ranges from the interval 0-1 where, this means that the resulting probability value cannot be more than 1.The maximum probability value generated must be of 1. For this case, the probability of fitness and cumulative value is obtained using a population size of 50, the crossover and mutation opportunities are 0.6 and 0.05, respectively, and the maximum generation is 100.

**Table 1. Fitness Probability**

| No | Pk | qk | No | Pk | qk |
|---|---|---|---|---|---|
| 1 | 0.01593 | 0.01593 | 26 | 0.01657 | 0.50801 |
| 2 | 0.03228 | 0.0482 | 27 | 0.02101 | 0.52902 |
| 3 | 0.01721 | 0.06541 | 28 | 0.01027 | 0.53929 |
| 4 | 0.02032 | 0.08573 | 29 | 0.01715 | 0.55644 |
| 5 | 0.01027 | 0.096 | 30 | 0.01709 | 0.57352 |
| 6 | 0.02481 | 0.12081 | 31 | 0.01598 | 0.5895 |
| 7 | 0.01721 | 0.13802 | 32 | 0.00957 | 0.59907 |
| 8 | 0.01492 | 0.15294 | 33 | 0.02041 | 0.61948 |
| 9 | 0.01403 | 0.16697 | 34 | 0.0168 | 0.63628 |
| 10 | 0.01685 | 0.18382 | 35 | 0.01368 | 0.64996 |
| 11 | 0.02599 | 0.20981 | 36 | 0.01044 | 0.6604 |
| 12 | 0.03292 | 0.24273 | 37 | 0.01614 | 0.67653 |
| 13 | 0.0147 | 0.25743 | 38 | 0.03011 | 0.70664 |
| 14 | 0.01387 | 0.2713 | 39 | 0.01843 | 0.72507 |
| 15 | 0.01364 | 0.28494 | 40 | 0.04794 | 0.77301 |
| 16 | 0.02329 | 0.30823 | 41 | 0.02759 | 0.8006 |
| 17 | 0.02214 | 0.33037 | 42 | 0.03207 | 0.83266 |
| 18 | 0.02286 | 0.35324 | 43 | 0.01809 | 0.85075 |
| 19 | 0.03086 | 0.3841 | 44 | 0.02204 | 0.8728 |
| 20 | 0.02318 | 0.40728 | 45 | 0.01836 | 0.89115 |
| 21 | 0.02024 | 0.42752 | 46 | 0.01727 | 0.90842 |
| 22 | 0.01048 | 0.43801 | 47 | 0.02351 | 0.93193 |
| 23 | 0.01709 | 0.45509 | 48 | 0.0327 | 0.96464 |
| 24 | 0.01757 | 0.47267 | 49 | 0.01885 | 0.98348 |
| 25 | 0.01878 | 0.49144 | 50 | 0.01652 | 1 |

### 3.4. Selection Process

The selection process is carried out by evaluating each chromosome based on its fitness value to get the best ranking. At this stage, random numbers will be generated, then chromosomes are selected according to their cumulative value. The first step is the cumulative value that has been obtained compared to the generated random number. A chromosome will be selected if the random number generated is in its cumulative interval.

### 3.5. Crossover

One of the most important components in a genetic algorithm is crossover. A chromosome leading to a good solution can be obtained by crossing two chromosomes. At this stage, select two parent chromosomes that will undergo random cross-breeding, then determine the intersection point. After that, exchange information from the two chromosomes based on the predetermined intersection points.

### 3.6. Mutation Process

In this mutation process, a randomly selected gene pair on a chromosome is exchanged. This pair exchange is carried out on two genes in a chromosome. If the random number [0,1] generated is less than the mutation probability, then the gene value will be exchanged with the value of another randomly selected gene. This is true for all genes in chromosomes. The value of the mutation probability is:

$$\frac{1}{number\ of\ genes} \qquad (2)$$

For the case above, the chance for mutation is $\frac{1}{20} = 0.5$. There are 3 variables used in this mutation process, namely chromosome, permutation and *n*.

### 3.7. Determination of the Best Path With Matlab

At this stage, the mutation probability is 0.05. The population size is 50. The number of genes is 20. The maximum generation is 100 and the probability of interbreeding (pc) is set to 0.6. The result obtained is the best path and the average fitness graph is also obtained as shown in Figure 2.

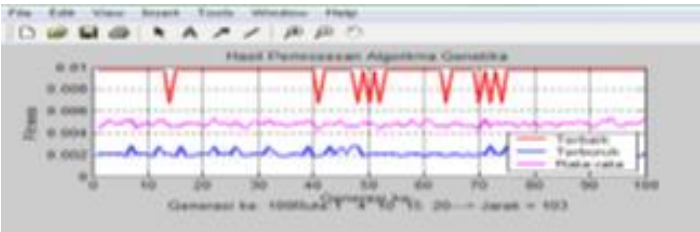

**Figure 2. Average fitness graph**

In the shortest path problem above, the reference consists of 20 vertices and uses the GUI on MATLAB R2017a. In this display, an example of the results of program execution is displayed with a population size of 50 and a maximum of 100 generations. The results obtained can be seen in Figure 3.

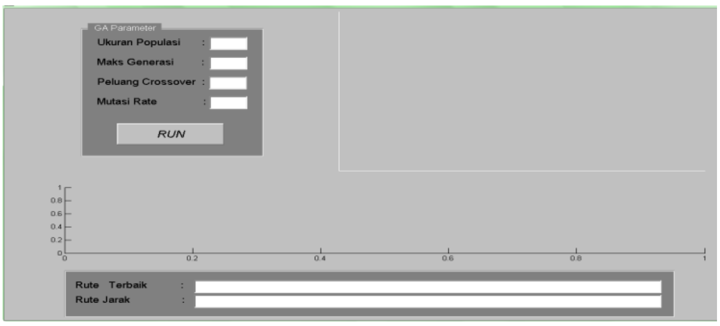


**Figure 3. Display before input of genetic algorithm parameters**

By inputting the population size is 50. The maximum generation is 100, the mutation and crossover probability are 0.05 and 0.6 respectively, then the results are obtained as shown in Figure 4.

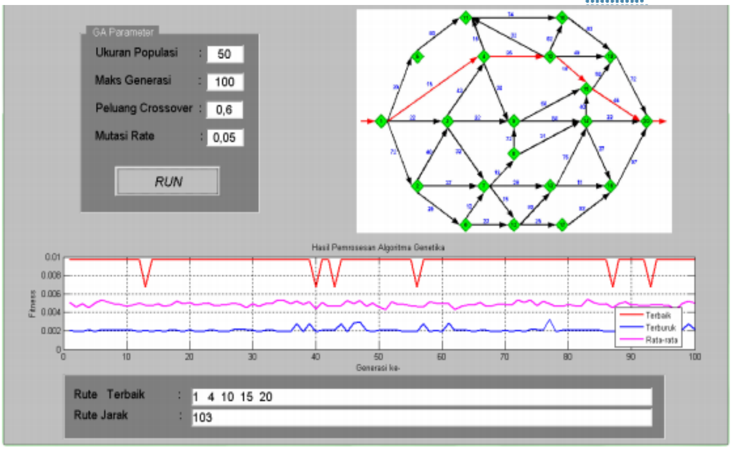


**Figure 4. Display after input of genetic algorithm parameters**

## 4. Conclusion

From the tests have been done, it can be concluded that by not requiring to go through all paths, the optimization results of the shortest path case can produce a higher level of optimization. For the case of shortest path problem, chromosomes are designed using the principle of randomization, where the calculation is carried out only for the paths that are traversed from the source city and destination city, with the remaining chromosomes (the untreated city) functioning as a complement so that the chromosomes are not cut off. The shortest path length resulted in the shortest path problem in Figure 1 is 103. With lines 1-4-10-15-20. The advantage of genetic algorithms compared to conventional search methods on the shortest path problem is that solutions can be obtained at whatever generation and whenever desired and genetic algorithms do not take a long time to process.

## Acknowledgments

We thank Mathmatics department in *Universitas Islam Negeri Sumatera Utara Medan* for facilitating the research

## References

[1] Mutakhiroh, I dan Wiryadinata, R,. “Pemanfaatan Metode Heuristik Dalam Pencarian Jalur Terpendek Dengan Algoritma Semut dan Algoritma Genetik. Seminar Nasional Aplikasi Teknologi Informasi”. Vol. 1. (2007). pp. 1907-5022.
[2] Hannawati, ,A. “Pencarian Rute Optimum Menggunakan Algoritma Genetika”. Jurnal Teknik Elektro Fakultas Teknologi Industri-Universitas Kristen Petra. Vol. 2 (2005). pp. 78 -83.
[3] Kusumadewi, S. “Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik” (2005).. Yogyakarta: Graha Ilmu.
[4] Goldberg, D. E. “*Genetic Algorithms in Search, Optimization & Machine Learning”*. (1989). New York: Addison-Wesley.
[5] Wilson, R. J., and Watkhins, J. J. “Graph An Introductionary Approach, A First Course in Discrete Mathematics”. (1990). John Willey and Sons, New York.
[6] Ahmed, Z.H. “Genetic Algorithm for Travelling Salesman Problem using Sequential Constructive Crossover Operator”. International Journal of Biometrics & Bioinformatics. Vol. 3. (2005). pp. 96-105.
[7] Chartrand, G., and Zhang, P. ” Introduction to Graph Theory” ( 2005). New York: McGraw-Hill.
[8] Conradie, W. and Goranko, V. “Logic and Discrete Mathematics”. (2015) .Published. John and Wiley.

[9] Tettamanzi, A. and Tomassini. "Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems". .(2015). Heidelberg: Springer.

## Authors

**1st Author**
***Hendra Cipta***
*Mathematics Study Program, Universitas Islam Negeri Sumatera Utara, Medan*

**2nd Author**
***Rina Widyasari***
*Mathematics Study Program, Universitas Islam Negeri Sumatera Utara, Medan*