KINETIK, Vol. 3, No. 4, November 2018, Pp. 311-318
ISSN : 2503-2259
E-ISSN : 2503-2267

Genetic Algorithm Application for Navigation System on Virtual Shop

Andi Nagoro Syahputra*[1], Ali Sofyan Kholimi[2], Lailatul Husniah[3]
[1,2,3]Universitas Muhammadiyah Malang
andisuhat@webmail.umm.ac.id[*1], kholimi@umm.ac.id[2], husniah@umm.ac.id[3]

Abstract

Online shop becomes one of the updated common types of shop in the era of rapidly grown technology development, yet this system provides its drawbacks. One of the problems which commonly found is a great deal of scamming practices for profit-oriented aims. Virtual shop is defined as a virtual reality application allowing its user to experience shopping in virtual world. A navigation system is required to enable its users in finding out the expected product. Genetic algorithm is one of many search algorithms based on natural selection and natural genetics. On this algorithm, data and string are executed through process of selection to generate problem solving solution. Using this algorithm, the navigation system is proved to be successful on generating path leading to expected product in the virtual shop.

Keywords: *Navigation System, Virtual Shop, Genetic Algorithm*

1. Introduction

In order to live, human is undeniably required to fulfill various needs. Human, as social being, needs other people to fulfill daily need. Each human is impossible to fulfill his need by himself. In an effort to fulfill such need, human requires a place which provides easy access to human needs. According to KBBI (Indonesian Language Dictionary), store is defined as a tavern in the form of permanent building where people sell goods (food, snack and so on). As the time goes on, store is progressively developed into variety of types, one of which is online store or popularly recognized as online shop.

Online shop system is where buyer can shop easily at cyberspace providing convenience for its user to search the desired products. However, such system is unfortunately coupled with drawbacks, as there are two notable problems as encountered. The first problem is the sellers who commit scam practice for gaining profit. The second problem is the buyers who are left unsatisfied by the purchased products due to their misexpectations. Creating virtual stores based on virtual reality (VR) to minimize the previously given two online shop main problems; therefore, this practice is encouraged by the researchers.

Virtual Reality is a technology allowing its users to experience an artificial reality, by engaging the users to interact with some objects or room as virtually displayed [1]. Virtual Shop is part of a VR application where the users can shop as in the real world. Most virtual reality applications nowadays are focused on simulation and game genre. Meanwhile, VR application for online store remains uncovered.

In the previous study conducted by Karas et.al [2], the application of genetic algorithm was applied to find out the shortest driving time on mobile devices. The result presents that genetic algorithm can be applied to route guidance system on mobile device, which has limited processing and memory capacity. Abeysundara [3] studied the use of genetic algorithm to find out the shortest route for roadmap, resulting in a method to find and to produce more than one shortest paths to reach desired destinations. Made et.al [4] also conducted a study on the application of genetic algorithm to find out the shortest distance for goods distribution on Java island. Genetic algorithm is proven to be effective in the case of measuring the shortest path. Thus, researchers in this study proposed genetic algorithm to find out the shortest path for navigation system in virtual shop. It is expected that this navigation system will accommodate the users in easily searching desired products in virtual shops.

Syahputra, A., Kholimi, A., & Husniah, L. (2018). Genetic Algorithm Application for Navigation System on Virtual Shop.Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 3(4). doi:http://dx.doi.org/10.22219/kinetik.v3i4.668
Receive July 11, 2018; Revise July 20, 2018; Accepted September 26, 2018

2. Research Method

Genetic Algorithm is defined as A* search algorithm based on the mechanics of natural selection and natural genetics [5]. In this algorithm, a string or data is selected and processed to generate a problem solving conclusion [6]. The algorithm is initiated with initializing some population of randomly generated chromosome. Furthermore, algorithm calculates each chromosome fitness value to check whether algorithm termination meets the criteria. If the algorithm could not find the targeted solution, it will reiterate through selection, crossover, and mutation process. Figure 1 depicts the genetic algorithm flowchart.

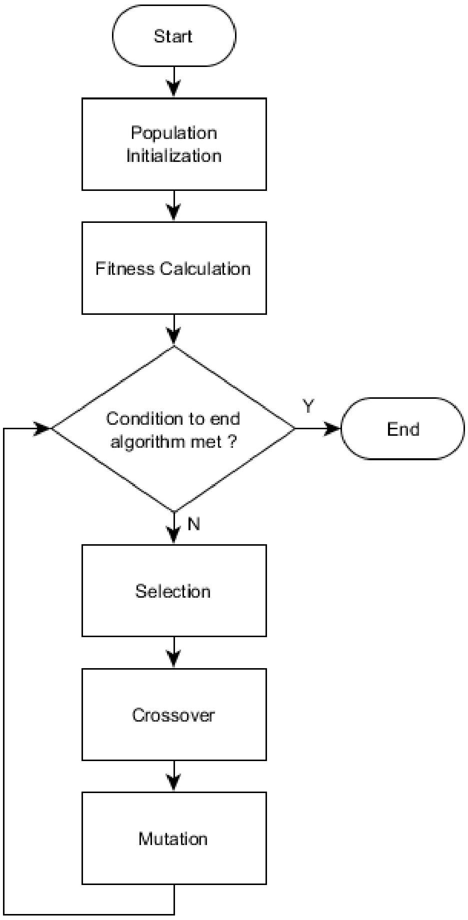

Figure 1. Genetic Algorithm Flowchart

Unlike classical search and optimization method, GA is initiated with multiple population sets of randomly generated solution. Once the population is generated, each solution is evaluated and selected to produce the most optimal solution for problem solving [7]. Large population value usually results in better solutions, although it requires better computing process [8]. Some parameters of GA in the algorithm (such as: population value, crossover probability, and mutation probability) must be precisely determined to achieve maximum results [9].

Chromosomes are identified as member of the population in the genetic algorithm. Each chromosome contains a solution for the problem. Each iteration of GA process is called generation. GA is usually iterated by 50 to 500 generations. Collection of all generations are called run [10]. Fitness function is a function to assess the expectedly good quality of a chromosome on solving the predetermined problem [11]. This function is unlimited to numerical problems, as non-numerical problems can also be solved. Fitness function is also identified as objective function and is denoted by *f(x)*. Selection operator is an operator used for generating a new generation selected from previous generations, consisting of both parent and derived chromosomes. Crossover operator is an operator that randomly selects one part of a chromosome and exchanges such part with another part of chromosome to form an offspring chromosome. The likelihood of crossover depends on p_c (crossover probability). Mutation operator is an operator used to modify one random value on a chromosome for creating diversity of solutions [12]. This operator is designed in such a way for restoring good genes on a chromosome likely disappearing during the selection and crossover process [13]. The likelihood of mutation depends on p_m (mutation probability).

There are several stages of research method conducted in this research, sorted by chronological order including: determining algorithm environment, determining design and implementation of algorithm, and determining system testing.

The environment in which the algorithm is applied is represented by virtual shop on the navigation system. In this system, the user is required to have brand and type of shoe they look for. The type of shoe will be utilized as a reference to determine the path from the closest node (from the user) to the destination node. Figure 2 displays the navigation screen on the virtual shop.

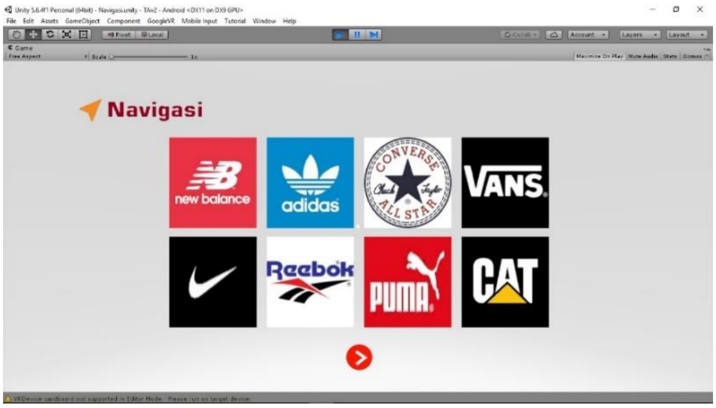

Figure 2. Navigation Screen on Virtual Shop

User is thus required to select one of the brands on the navigation screen. Afterwards, user needs to select one of the shoe types on the screen. Afterwards, the system will generate the path from the user position to the user selection on shoe based on genetic algorithm. Figure 3 demonstrates node mockup on a virtual shop.

There are 8 shelves in the virtual shop which contain 8 different shoe brands. The 8 brands of these shoes are Adidas, Caterpillar, Converse, New Balance, Nike, Puma, Reebok, and Vans. Each brand has 4 different types of shoes. As depicted in figure 3, there are horizontal and vertical nodes. These nodes are designed to speed up shoe search time. The details of genetic algorithm design are listed in the following points:

1. Each chromosome is designed to show the movement of the user on the virtual shop nodes. The movement of the user is limited to 4 directions: north, south, west, and east. These 4 directions are represented by 2 binary numbers as listed in the following table.

Table 1. Chromosome Design

Binary	Decimal	Directions
00	0	North
01	1	South
10	2	East
11	3	West

At the time when several binary numbers are initialized, it can be decoded into a series of directions to be followed by the user. The length of chromosome is 42. This value represents 21 steps in the node. The length of chromosome is the longest number of nodes, by which the user may travel to reach the destination node.

2. The fitness function to be applied is as seen Equaion 1.

$$F(x) = \frac{1}{|x_1 - x_0| + |y_1 - y_0| + 1} \quad (1)$$

In which: *F(x)* is fitness function, x_1 is x axis value of chromosome end position, x_0 is x value of chromosome initial position, y_1 is y axis value of chromosome end position, and y_0 is y axis value of chromosome initial position. The chromosome with a fitness value close to 1 is a chromosome whose position is closer to destination node. The chromosome with a fitness value equal to 1 is a chromosome which has reached destination.

3. The number of population in one generation is 84. This value is selected as referred to a study conducted by Matt Buckland, in which the number of population should be doubly initialized by the amount from the value of chromosome length [14].
4. The selection operator to be used is roulette-wheel selection. In this selection method, the chances of each chromosome to be selected are proportional to its fitness value. Roulette wheel selection is selected as referred to a study conducted by Anies Hannawati [15], stating that this selection method is the best selection method.
5. The crossover operator to be used is single point crossover with probability of 70%. This value is chosen by referring to a study conducted by Matt Buckland, in which a 70% is a good value for binary chromosomes [14]. A number between 0 and 1 is randomly initialized, in which if the number is less than crossover probability (70%), then random position along the length of chromosome is chosen in each bit, where the point is later swapped.
6. The probability of mutation is 0.001 (0.1%). This value is chosen by referring to a study conducted by Matt Buckland, pointing out that this value is considered as a good value for binary chromosomes [14]. A number between 0 and 1 is randomly initialized, in which if the number is less than mutation probability, then a bit within a chromosome is randomly selected and is flipped (from 0 to 1 or vice versa).
7. The algorithm will be stopped if any chromosome with a fitness value equal to 1 is found.

System testing is conducted to find out the fastest time on path-finding navigation system. There are 4 tests which will be conducted, which are: modifying population value, modifying crossover probability, modifying mutation probability, and comparing fitness value every ten seconds. There are 3 shoes which will be tested: type 4 Caterpillar shoe (4th shelves), type 4 Converse shoe (3rd shelves), and type 2 New Balance shoe (4th shelves).

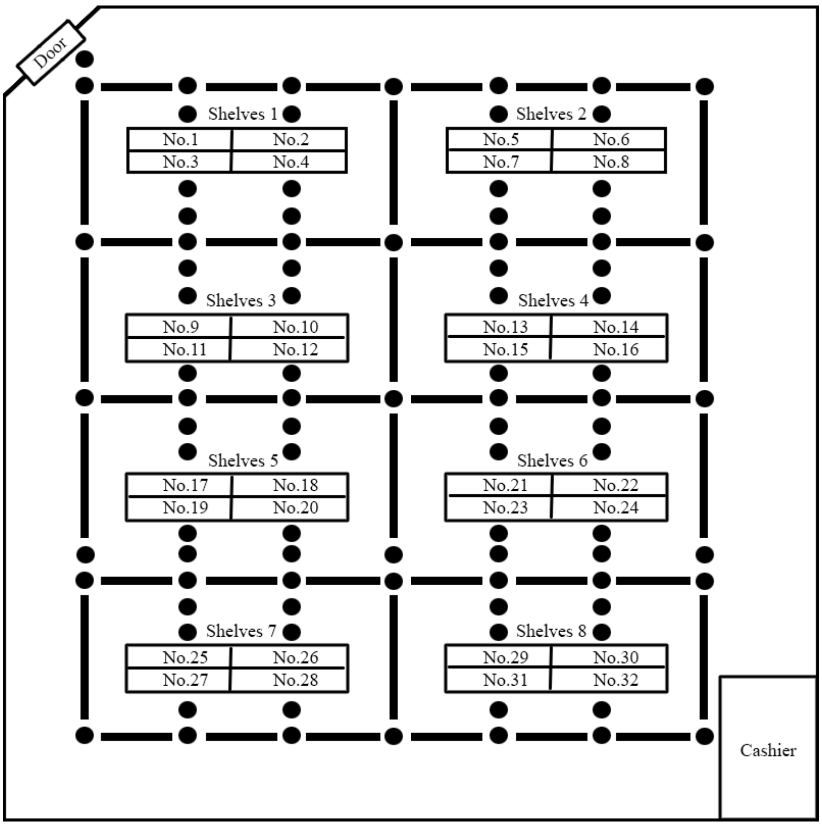

Figure 3. Node mock-up on Virtual Shop

3. Results and Discussion

3.1 Population Value Modify Test

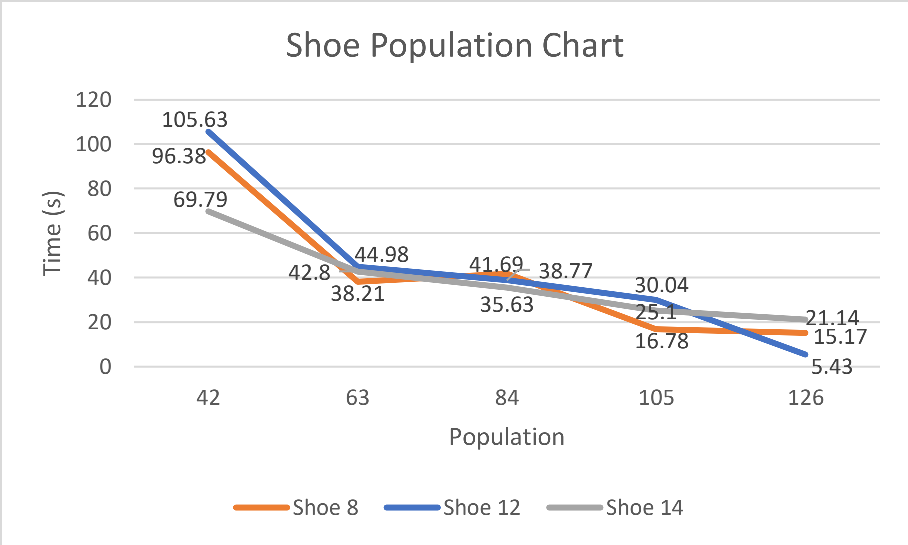

Figure 4. Shoe Population Comparison Chart

By this test, the population value is modified and determined as follows: 42, 63, 84, 105, and 126. The crossover probability used in the test is 0.7. The mutation probability used in the test is 0.1%. Figure 4 presents the result of population value modify test, in which the population 42 requires the longest time on path-finding (more than one minute). While on the population 126, the time required to find a path is less than 22 seconds.

3.2 Crossover Probability Modify Test

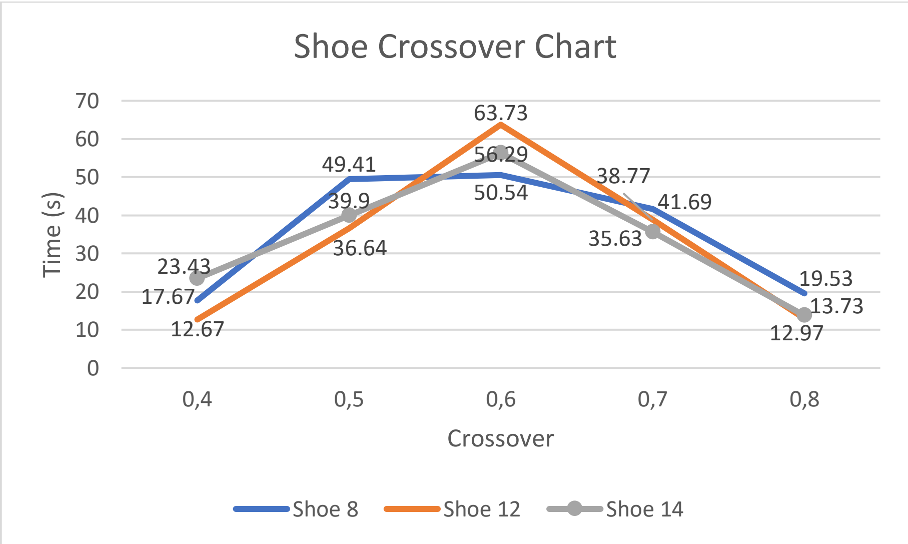

Figure 5. Crossover Probability Comparison Chart

By this test, the crossover probability is modified and determined as follows: 0.4, 0.5, 0.6, 0.7, and 0.8. The population value used in the test is 84. The mutation probability used in the test is 0.1%. Figure 5 illustrates the result of crossover probability modify test. Crossover probability whose values are 0.4 and 0.8 generate the shortest time compared to those in 0.5, 0.6, and 0.7 (crossover value of 0.6 generates the longest time).

3.3 Mutation Probability Modify Test

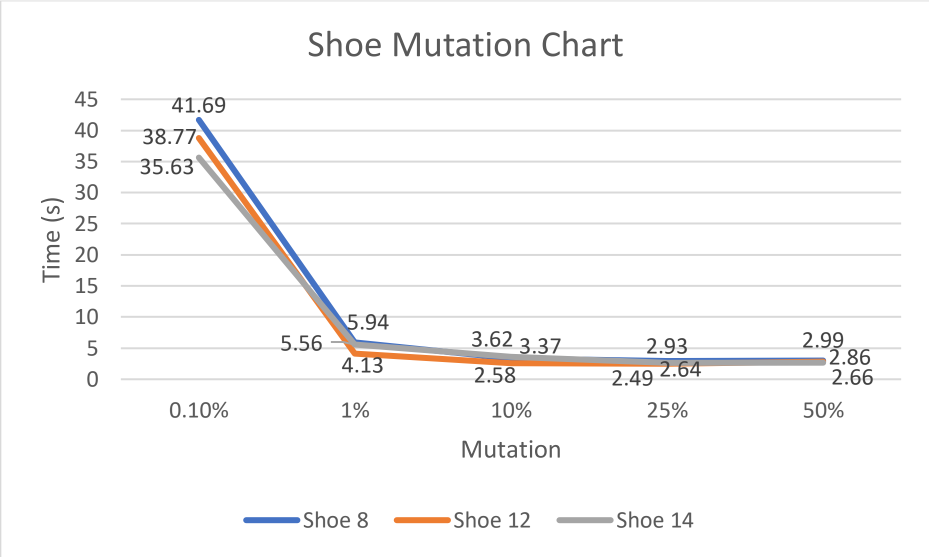

Figure 6. Mutation Probability Comparison Chart

By this test, the mutation probability is modified and determined as follows: 0.1%, 1%, 10%, 25%, and 50%. The population value used in the test is 84. The crossover probability used in the test is 0.7. Figure 6 demonstrates the result of mutation probability modify test, marking that the mutation values of 1%, 10%, 25%, and 50% generate better time on path-finding than that in the mutation value of 0.1% (mutation value of 25% generates the fastest time).

3.4 Fitness Value Comparison Test

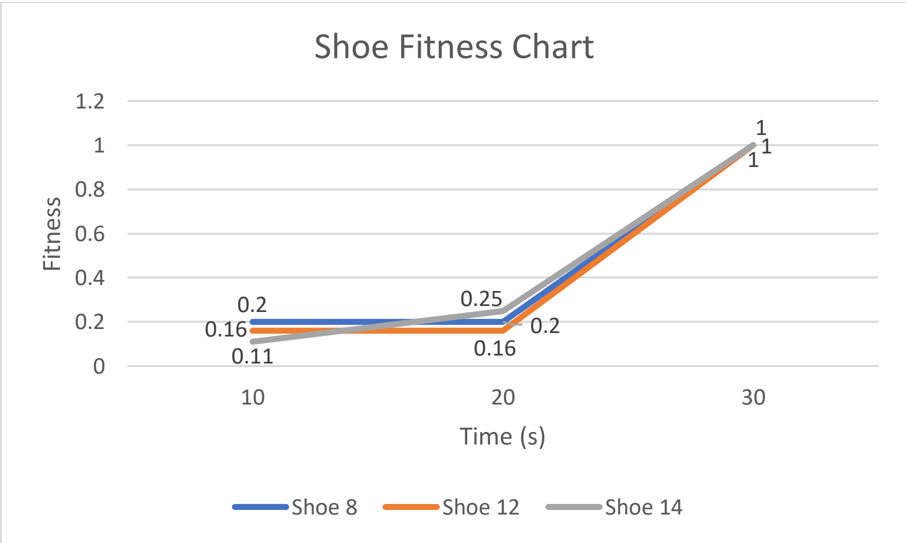

Figure 7. Fitness Value Comparison Chart

By this test, the fitness value for path-finding in 3 shoes every 10 seconds is compared. The population value used in the test is 84. The crossover probability used in the test is 0.7. The mutation probability used in the test is 0.1%. Figure 7 presents the result of fitness value comparison test noting that on the 10th and 20th second, the fitness value does not much change. However, at the 30th second the fitness value of 3 shoes show the value of 1 or in other words, the path-finding is accomplished.

4. Conclusion

Upon accomplishing the test, several conclusions are withdrawn. From the first test (population value modify test), it is concluded that greater population value used in the algorithm will lead to faster path-finding time. From the second test (crossover probability modify test), it is concluded that crossover value of 0.4 and 0.8 are the most optimal crossover values for a population of 84 chromosomes with a mutation value of 0.1%. From the third test (mutation probability modify test), it can be concluded that mutation value of 25% is the most optimal value for a population of 84 chromosomes with a crossover value of 0.7. From the fourth test (fitness value comparison test), it can be concluded that fitness value tends to increase as the path-finding time progresses. Research result as obtained from the experiments depicts that Genetic Algorithm can provide fairly fast path-finding time with certain parameters on virtual shop navigation system.

References

[1] J. Isdale, *"Introduction to VR Technology,"* IEEE Virtual Reality, 2003. Proceedings., Pp. 302, 2003.

[2] I. R. Karas and U. Atila, *"A Genetic Algorithm Approach for Finding the Shortest Driving Time on Mobile Devices,"* Scientific Research and Essays, Vol. 6, No. 2, Pp. 394–405, 2011.

[3] S. Abeysundara, B. Giritharan, and S. Kodithuwakku, *"A Genetic Algorithm Approach to Solve the Shortest Path Problem for Road Maps,"* Proc. Int. Conf. Inf. Autom., Pp. 272–275, 2005.

[4] I. D. Made, A. Baskara, and V. Nurcahyawati, *"ISSN 2089-8673 National Journal of Informatics Engineering (JANAPATI) JAVA ISLAND USING GENETIC ALGORITHM I,"* Introduction ISSN 2089-8673 National Journal of Informatics Engineering Education (JANAPATI) II. Fundamental theory," Vol. 1, Pp. 244–258, 2012.

[5] D. E. Goldberg and others, *"Genetic Algorithms in Search, Optimization, and Machine Learning,"* Addison Wesley Publisching, Vol. 412. 1989.

[6] D. E. Goldberg, *"David E. Goldberg-Genetic Algorithms in Search, Optimization, and Machine Learning-Addison-Wesley Professional (1989).pdf."* Pp. 432, 1989.

[7] K. Deb, *"An introduction to genetic algorithms,"* Sadhana, Vol. 24, no. 4–5, Pp. 293–315, 1999.

[8] C. W. Ahn and R. S. Ramakrishna, *"A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations,"* IEEE Transactions on Evolutionary Computation, Vol. 6, No. 6, Pp. 566–579, 2002.

[9] H. Fiori de Castro and K. Lucchesi Cavalca, *"Availability Optimization with Genetic Algorithm,"* International Journal of Quality & Reliability Management, Vol. 20, No. 7, Pp. 847–863, 2003.

[10] M. Melanie, *"An Introduction to Genetic Algorithms,"* Cambridge, Massachusetts London, England, Pp. 162, 1996.

[11] T. Lu and J. Zhu, *"A Genetic Algorithm for Finding a Path Subject to Two Constraints,"* Applied Soft Computing Journal, Vol. 13, No. 2, Pp. 891–898, 2013.

[12] Z. H. Ahmed, *"Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator,"* International Journal of Biometrics & Bioinformatics (IJBB), Vol. 3, No. 6, Pp. 96–105, 2010.

[13] T. Chen and C. Chen, *"Improvements of Simple Genetic Algorithm in Structural Design,"* International Journal for Numerical Methods in Engineering, Vol. 40, No. 7, Pp. 1323–1334, 1997.

[14] M. Buckland, *"AI Techniques for Game Programming,"* Vol. 1, No. 3. 2002.

[15] Anies Hannawati, Thiang, and Eleazar, "Optimum Route Search By Using Genetic Algorithms," Electrical Engineering Journal, Vol. 2, No. 2, Pp. 78–83, 2002.