MATEMATIKA, MJIM, Volume 38, Number 1, 53–67

Analysis of Genetic Parameters and Operators in Solving Looping and No Looping Shortest Path

Muhammad Adam Abdullah and Md Yushalify Misro*

School of Mathematical Sciences, Universiti Sains Malaysia,
11800 Pulau Pinang, Malaysia

*Corresponding author: yushalify@usm.my

Article history
Received: 19 August 2021
Received in revised form: 26 April 2022
Accepted: 26 April 2022
Published on line: 30 April 2022

Abstract Studies on shortest paths are significantly impactful given its wide range of applications especially in transportation and route planning. This study provides an additional solution to various existing optimization methods by proposing a Genetic Algorithm (GA) approach incorporated with Haversine formula to find the solution to shortest path problems. Two cases are taken into account, namely Looping Shortest Path (LSP) and No Looping Shortest Path (NLSP). The algorithm is tested for a road map containing 20, 30, and 40 cities. The experiment is repeated several times to find the best combination of genetic parameters and operators for the problem under consideration.

Keywords Shortest Path; Genetic Algorithm; Haversine Formula.

Mathematics Subject Classification 90B06, 90B22

1 Introduction

The Travelling Salesman Problem (TSP) is a typical non-deterministic polynomial (NP) hard problem in designing the shortest path for travellers to visit each city without repetition. TSP considers a single vehicle that visits multiple places before returning to the original place. In contrast to Vehicle Routing Problem (VRP) which is focused on producing numerous routes to pass through all city nodes under vehicle capacity limitations, TSP is a single route node service combination problem with no vehicle capacity limitation.

Since March 2020, the world had been challenged with a global pandemic due to the emanation of Covid-19 virus, which had caused direct impacts on transportation [1]. In Malaysia, the government had implemented an effective restriction initiative known as Movement Control Order (MCO). Under these restrictions, citizens are required to stay at home, where outdoor physical activities are restricted. Therefore, more Malaysians had started to resort to e-commerce platforms in attempt to gain access to food, groceries, clothes, and other daily necessities. As a result, the demand for online shopping increased drastically, causing most e-commerce systems to crash, which then caused delay in deliveries. Courier services such as

J&T, DHL, and Grabfood are required to deliver their shipments to customers within the stipulated time. To achieve an optimal delivery process, the courier services must be able to make deliveries without repeating the paths.

Past studies attempted to solve TSP by developing several optimization methods in determining the shortest path. The Floyd-Warshall algorithm was applied to find the shortest path to a garage in [2]. This algorithm was used to calculate the shortest path for a traveller to find a garage from an unknown location. Another approach used Dijkstra and Floyd-Warshall's algorithm to determine the fastest travel time and the best route to the nearest hospital [3], where the algorithm was employed to discover the closest distance to the hospital. If there happened to be traffic accidents on the way to the hospital, the hospital's location will be redirected to another nearby hospitals.

Solving the TSP of multiple Unmanned Aerial Vehicles (UAVs) in forest fire fighting missions using Particle Swarm Optimization (PSO) was proposed by [4]. The algorithm in the research minimizes the distance travelled between each UAV's initial position and their assigned fire spots. In addition, [5] also used the PSO algorithm to control the fire rescue robots by finding their shortest path. The task was carried out by assigning each robot to each fire spot to help the agent choose their nearest path to minimize the travelling distance. Path planning for a mobile robot using PSO to find an optimal path in a known environment with static obstacles was proposed by [6]. This research aims to find feasible and optimal paths with respect to the distance covered and the robot's safety.

Next, [7] used Ant Colony Optimization (ACO) algorithm with the Internet of Vehicles for an intelligent traffic control system. Then, ACO was applied to a map in order to find the optimal routes to reach the destination. Recently, path planning in an Automated Guided Vehicle (AGV) based on intelligent parking system using improved ACO was proposed by [9]. In addition, [10] proposed a self-adaptive ACO with unique strategies to improve uncertain convergence time and random decisions. The main idea of this research is to select the first city and achieve the shortest path for TSP. Subsequently, [8] used Dijkstra's algorithm to find the interpolated shortest path of Bézier curves with control points to provide a smooth path planning curve.

A new path planning approach for emergency evacuation simulation using the Improved Artificial Bee Colony (IABC) algorithm and the Extended Social Force Model was proposed by [11]. The IABC algorithm improved the evacuation efficiency and supported building designs and evacuation management by employing grouping and exit selection strategies. Apart from that, [12] improved the IABC algorithm for mobile robot path planning to determine the optimum global path that satisfies the chosen criteria for shortest distance and collision-free with circular-shaped static obstacles in the robot environment.

Path planning for autonomous mobile robots using GA for finding a feasible path between two positions while avoiding obstacles in a static environment was solved by [13]. Apart from that, determining the distance between two locations on the world map based on longitude and latitude using the Haversine formula plays an important role, as mentioned in [14]. In another study, the Haversine formula was used for tracking schools by providing relevant information such as the school's location that is closest to the user [15]. However, there is insufficient research to determine the best combination of GA parameters for different numbers of chromosomes in discovering the optimal solution. Therefore, in this study, we would like to explore the relation between the number of chromosomes (in this case, the group of cities) and the parameters of

GA. It is an option for this study to use Google Maps in retrieving necessary data, such as road networks and coordinates of cities. However, there are road networks and paths in rural areas that are not covered in Google Maps. Hence, calculating the distance between two coordinates using Haversine formula is particularly more straightforward as opposed to Google Maps, which makes Haversine formula an applicable method for this study.

This research is organized in four sections. In Section 1, the introduction to TSP and proposed GA as our method to solve this problem are presented. We also provide various examples for solving the shortest path problem. The process of finding the possible shortest path using GA for Looping Shortest Path (LSP) and No Looping Shortest Path (NLSP) are presented in Section 2. Each case of LSP and NLSP is run several times with three different groups of cities. The total distance between each city is then computed using the Haversine formula. In Section 3, the results for each case of LSP and NLSP is discussed. Lastly, the conclusion and recommendation for future works are suggested in Section 4.

2 Methodology

Genetic Algorithm (GA) is a problem-solving method based on natural selection and genetics. In this research, GA is used to find an optimal solution for the shortest path problem. The problem is divided into two cases: Looping Shortest Path (LSP) and No Looping Shortest Path (NLSP).

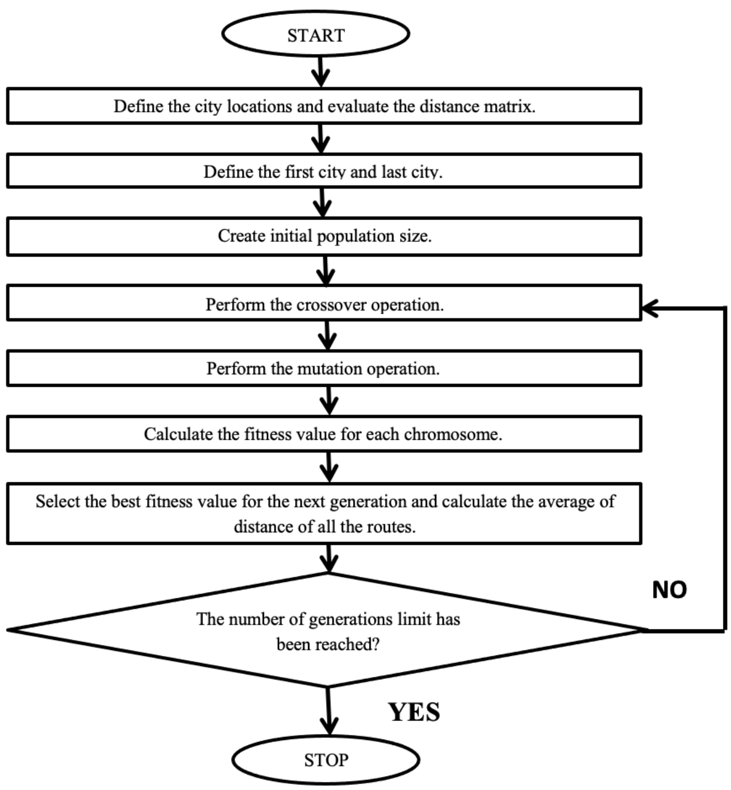

Figure 1: General flowchart of LSP and NLSP.

The general flowchart of LSP and NLSP is shown in Figure 1. Based on the general flowchart, the step to define the first and last city for LSP is skipped, while for NLSP, the entire step is followed. The general pseudocode for LSP and NLSP are shown in Figure 2. The same procedure is applied to all three different groups of cities. Each process further explicated in the next section.

```
Input = p : Population Size
        c : Number of pairs of chromosome required for crossover
        m : Number of chromosome required for mutation
        tg : number of generation
        Fc : Choose the first city
        Lc : Choose the last city.
Output = Min_Total_Distance; Optimal_Solution.

1.  Evaluate the distance matrix for each city.
2.  Choose the first city and last city.
3.  Create initial population size with first city and last city that has been chosen.
4.  While (generation limit NOT reached) do
5.          Perform crossover operation depending on number of pair of chromosome.
6.          Perform mutation operation depending on number of chromosome.
7.          Calculate the fitness value for each chromosome.
8.          Select the best fitness value for the next generation and calculate the average of the
            distance of all the routes in the corresponding generation.
9.  end.
10. Draw the Min_Total_Distance, average of all distance for each generation and the path of
    Optimal_Solution.
11. Write out the Min_Total_Distance and Optimal_Solution.
```

Figure 2: General pseudocode of LSP and NLSP.

2.1 Genetic Representation of LSP and NLSP

Each gene of all chromosomes represents a city. For LSP, a chromosome forms a path that begins and ends at the same city, passing through all cities. On the other hand, for NLSP, a chromosome forms a path with the chosen first city and last city.

For example, ten cities located in Seberang Perai, Penang, are chosen as depicted in Figure 3. First, possible chromosome representation for LSP with city number 7 as the first and the last city is shown in Figure 4. Next, possible chromosome representation for NLSP with cities number 9 and 6 as the first and last cities is shown in Figure 5.

Figure 3: An example of the location of 10 cities.

7	1	9	10	5	4	6	3	8	2	7

Figure 4: A chromosome representation of the LSP.

9	1	2	8	7	10	5	4	3	6

Figure 5: A chromosome representation of the NLSP.

2.2 Operation of Genetic Algorithm

The population size is an important parameter that directly influences the ability to search for an optimum solution in the search space. The population size is required to be increased for each iteration when the chromosome's length is larger (number of cities) to generate a good solution. A random value of the initial population size will be put in by the users, which will continuously be set up for each generation until it reaches the final generation for both cases of LSP and NLSP.

Chromosome 1	1	2	3	4	5	6	7	8	9	10
Chromosome 2	2	3	1	5	4	10	7	6	9	8
Child 1	1	2	3	4	5	10	7	6	9	8
Child 2	2	3	1	5	4	6	7	8	9	10

Figure 6: A pair of chromosome performing crossover.

A genetic operator called crossover combines the genetic information of two chromosomes to generate new offspring. The type of crossover used in this study is a one-point crossover.

First, the user will choose the number of chromosome pairs required for the crossover from the initial population size. Next, the crossover operation is performed on a pair of chromosomes to create new offsprings, as shown in Figure 6. In this study, if the crossover operation produced offsprings that generate infeasible path, the offsprings will be omitted.

Next, a genetic operator called mutation is used to keep genetic diversity in chromosome populations from one generation to the next. The process of mutation depends on the number of chromosomes required for mutation as specified by the user. This operator randomly selects a gene (city) position and randomly replaces it with another gene in the chromosome, as shown in Figure 7.

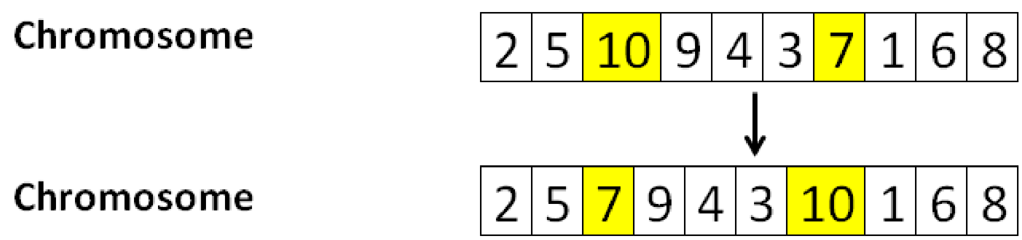

Figure 7: A chromosome performing mutation.

Lastly, a genetic operator called selection is used to choose the chromosome with the highest fitness value to survive for the next generation. After performing the crossover and mutation, the fitness value for each chromosome is calculated. From the calculation, a chromosome with the highest fitness value will be selected since it has the shortest total distance. This is made in accordance to the formula stated by (1), where the total distance of a chromosome is inversely proportional to the fitness value. Therefore, the selection of chromosomes with the highest fitness value will end when the generation limit is reached. The fitness value is defined as:

$$\text{fitness value} = \frac{1}{\text{the total distance of a chromosome}}. \tag{1}$$

2.3 Haversine formula

The distance of each city is calculated using the Haversine formula [14], defined by:

$$\mathrm{Dist}(i,j) = 2R\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\mathrm{lat}\,j - \mathrm{lat}\,i}{2}\right) + \cos(\mathrm{lat}\,j)\cos(\mathrm{lat}\,i)\sin^2\left(\frac{\mathrm{long}\,j - \mathrm{long}\,i}{2}\right)}\right) \times \frac{\pi}{180}, \tag{2}$$

where $Dist(i,j)$ = distance between city i and city j, R = the radius of Earth taken as 6371 km, $\mathrm{lat}\,i$ = latitude of city i, $\mathrm{lat}\,j$ = latitude of city j, $\mathrm{long}\,i$ = longitude of city i, $\mathrm{long}\,j$ = longitude of city j, and $\pi = 3.1416$.

The Haversine formula determines the great-circle distance between two points on a sphere, given their longitudes and latitudes. The calculated result of the total distance of a chromosome by the Haversine formula is approximately close to the real distance provided by Google Maps [14]. Therefore, the selection process finds and stores the highest fitness value (the route with the shortest distance) until the generation limit is reached.

2.4 Brute-Force Method (BFM)

BFM, or exhaustive search, is a general solving technique that systematically uses possible solutions to satisfy the problem. In this study, BFM is used to find the best combination for the parameters of GA by considering various combinations of the parameters.

3 Result and Discussion

This study is conducted on three groups of 20, 30, and 40 cities and the areas of Seberang Perai in Penang and Kulim in Kedah are selected for the experiment. These three different groups of cities are shown in Figure 8. The parameters of GA such as population size (**p**), number of chromosome pairs required for crossover (**c**), number of chromosomes required for mutation (**m**), and the total number of generation (**tg**) will affect the GA performance. Therefore, Brute-Force Method (BFM) is employed to identify the best parameter set for GA.

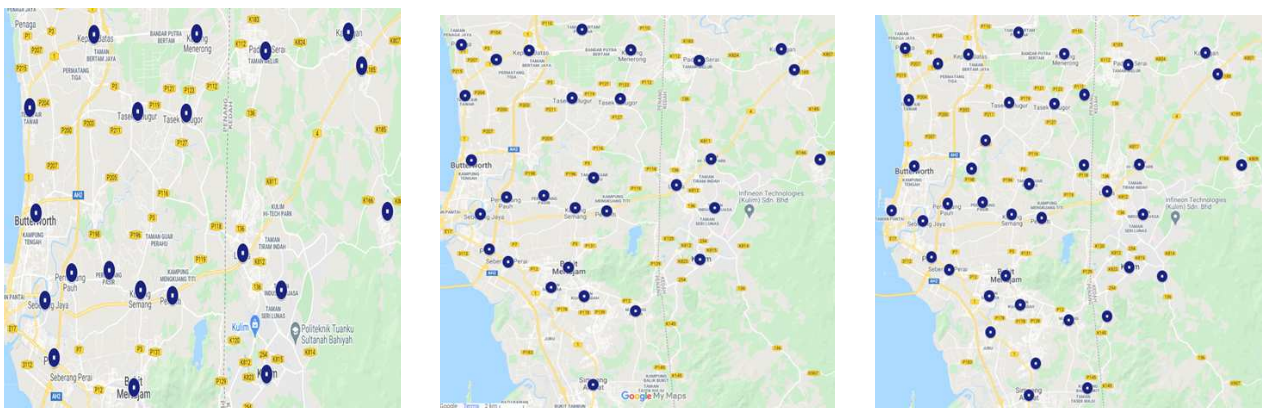

Figure 8: Location for 20 cities (left), 30 cities (middle), and 40 cities (right).

15 combinations of parameter values are considered in this study where only one parameter is changed in each combination except for combination 15. For example, from Table 1, in combinations 1 to 4, parameter **p** is changed. Then, in combinations 5 to 7, parameter **c** is changed. Next, in combinations 8 to 10 and combinations 11 to 13, parameter **m** and **tg** are changed, respectively. In combination 14, all parameters with the highest value were combined. Finally, in combination 15, we assigned an even higher value (in comparison to combination 14) for each parameter.

3.1 Case LSP

The best total distance travelled obtained is 98.6670 for combination 14 (by combining all parameters with the highest value). When the parameter values are increased (**p**=500, **c**=250, **m**=250 & **tg**=2000) in combination 15, as shown in Table 1, it yields the same result as for combination 14 (98.6670). The results from combinations 14 and 15 produced different paths with the same cost. From Figure 9 (bottom), notice that around 220th-230th generation, the blue curve plotted the minimum value until it reached the generation limit, implying that the algorithm has found the optimal solution. The best path for 20 cities is presented in Figure 9 (top).

Table 1: Result of LSP for 20 cities.

	p	c	m	tg	OPTIMAL PATH	TOTAL DISTANCE (KM)
1	**100**	20	10	100	13-8-7-3-11-4-12-9-1-2-10-5-15-16-6-14-18-17-19-20-13	158.9578
2	**200**	20	10	100	10-13-2-4-5-1-9-7-8-20-18-17-19-16-15-14-6-3-11-12-10	152.2176
3	**300**	20	10	100	1-5-10-9-12-2-7-16-20-19-18-17-15-8-6-14-4-11-3-13-1	149.6141
4	**400**	20	10	100	16-20-11-5-10-12-9-1-4-6-3-15-8-13-2-7-14-17-18-19-16	140.4103
5	100	**40**	10	100	10-4-3-6-14-17-19-18-15-11-1-9-12-7-13-20-16-8-2-5-10	147.1903
6	100	**60**	10	100	6-14-15-1-10-9-12-7-13-8-2-16-20-19-18-17-4-5-11-3-6	137.1318
7	100	**80**	10	100	3-7-13-12-9-10-8-2-1-5-4-6-11-14-17-18-19-20-16-15-3	132.6823
8	100	20	**20**	100	9-12-11-20-19-18-17-7-13-2-8-15-16-14-6-3-4-5-10-1-9	148.5620
9	100	20	**40**	100	4-5-10-9-12-15-19-17-18-14-2-13-1-7-8-16-20-3-6-11-4	139.9120
10	100	20	**60**	100	10-5-15-20-16-13-2-3-11-4-6-14-17-18-19-8-7-1-9-12-10	133.1592
11	100	20	10	**200**	9-1-6-14-3-11-2-7-4-5-10-8-15-19-17-18-20-16-13-12-9	145.7543
12	100	20	10	**300**	12-13-18-17-14-6-4-5-10-1-11-3-15-19-20-16-8-2-7-9-12	136.8678
13	100	20	10	**400**	4-6-19-18-17-14-3-11-1-9-12-13-2-8-16-20-15-7-10-5-4	126.0151
14	**400**	**80**	**60**	**400**	15-8-2-7-13-12-9-1-10-5-4-11-3-6-14-17-18-19-20-16-15	98.6670
15	**500**	**250**	**200**	**2000**	**14-6-3-11-4-5-10-1-9-12-13-7-2-8-15-16-20-19-18-17-14**	**98.6670**

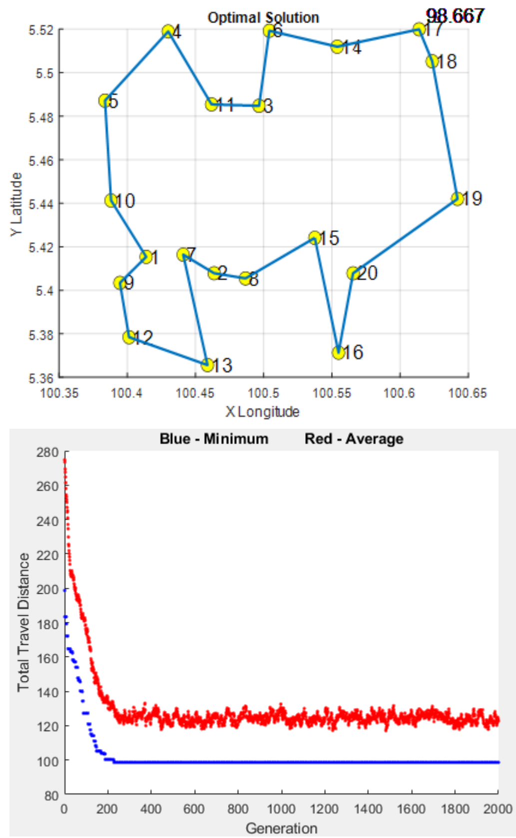

Figure 9: Best path of LSP (top) and the total travel distance of each generation of LSP (bottom) for 20 cities in combination 15.

Next, as presented in Table 2, when the parameter value is increased (**p**=500, **c**=250, **m**=250 & **tg**=2000) in combination 15, the results (total distance) showed a distinctive result compared to combination 14. There is a chance for the algorithm to search for other possible shortest paths when the parameter value increases. The best path for 30 cities, with the first and last cities being 27, is shown in Table 2 and Figure 10.

Table 2: Result of LSP for 30 cities.

	p	c	m	tg	OPTIMAL PATH	TOTAL DISTANCE (KM)
1	**100**	20	10	100	9-10-4-22-7-30-24-11-16-29-12-1-2-20-14-18-25-17-3-26-28-13-5-21-23-6-19-15-27-8-9	285.8088
2	**200**	20	10	100	24-23-6-3-20-16-26-9-11-25-14-15-7-5-22-4-21-12-1-10-27-29-28-17-19-18-8-13-30-2-24	265.6174
3	**300**	20	10	100	14-6-23-15-7-12-2-27-16-29-26-13-30-5-21-11-4-24-22-1-9-10-28-8-3-18-19-20-25-17-14	256.8302
4	**400**	20	10	100	15-20-3-6-11-24-2-8-28-13-27-30-29-14-18-19-25-16-12-26-9-7-22-1-5-4-10-21-23-17-15	247.3418
5	100	**40**	10	100	19-20-17-18-14-25-6-26-9-12-15-24-7-10-13-30-8-16-27-29-28-1-21-22-5-11-2-4-23-3-19	245.8217
6	100	**60**	10	100	21-4-2-1-10-9-26-8-25-20-16-30-13-28-12-24-15-29-27-6-23-18-19-17-14-3-22-5-11-7-21	239.0489
7	100	**80**	10	100	26-12-17-18-14-19-20-15-16-24-13-30-29-28-1-7-8-27-2-25-6-11-21-4-22-3-23-5-10-9-26	228.8303
8	100	20	**20**	100	24-8-15-4-12-26-28-7-10-21-5-2-13-1-22-23-6-3-11-16-25-17-14-19-18-20-30-27-29-9-24	254.2426
9	100	20	**40**	100	13-16-18-3-23-6-17-14-10-7-24-8-19-25-20-15-2-26-28-29-27-30-1-9-22-21-4-11-5-12-13	241.1701
10	100	20	**60**	100	12-7-3-6-14-17-18-8-16-15-20-27-29-26-9-11-23-2-13-28-30-10-4-21-22-5-24-25-19-1-12	239.0143
11	100	20	10	**200**	18-17-23-4-22-11-3-15-20-7-1-10-21-5-9-28-29-16-25-19-8-24-2-6-14-27-13-30-12-26-18	251.4039
12	100	20	10	**300**	4-21-5-22-23-14-25-19-28-13-26-7-16-20-15-10-1-3-17-18-6-9-12-30-29-27-8-2-24-11-4	243.2428
13	100	20	10	**400**	17-18-4-22-21-11-27-29-28-26-16-15-25-14-6-23-3-2-30-12-9-10-5-1-7-13-8-24-20-19-17	215.7660
14	**400**	**80**	**60**	**400**	3-15-20-25-19-17-18-14-6-1-9-12-26-13-30-28-7-2-24-8-16-27-29-10-5-22-21-4-23-11-3	167.8223
15	**500**	**250**	**200**	**2000**	**27-29-28-26-12-10-5-22-21-4-23-6-11-1-9-7-2-24-3-14-17-18-19-25-20-16-15-8-13-30-27**	**151.3608**

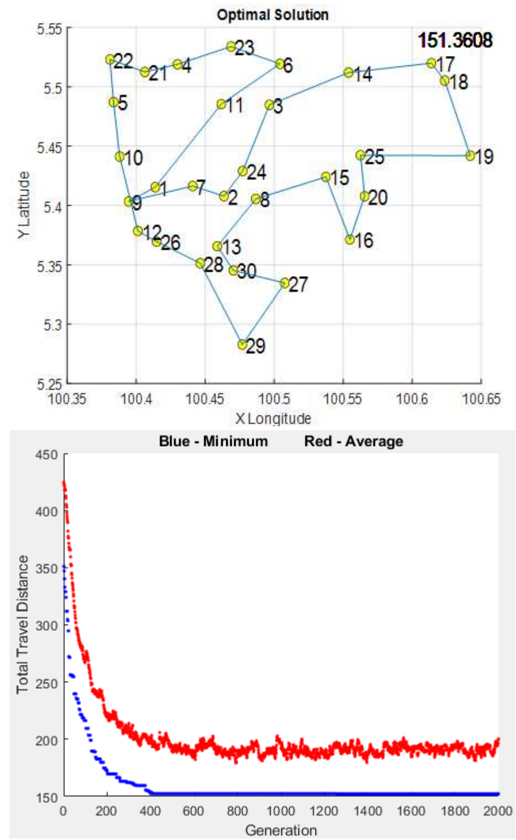

Figure 10: Best path of LSP (top) and the total travel distance of each generation of LSP (bottom) for 30 cities in combination 15.

In Table 3, when the parameter values is increased (**p**=500, **c**=250, **m**=250 & **tg**=2000) in combination 15, the results obtained were 190.0594. Similar to Table 2, even though we computed a higher value for each parameter of GA in combination 14, the algorithm still has a chance to search for alternative possible shortest paths using other different combinations of GA parameters. The best path with the first and last city is 10 for 40 cities is shown in Figure 11 (top). The blue curve is shown to have remain constant until the generation limits are achieved.

Table 3: Result of LSP for 40 cities.

	p	c	m	tg	OPTIMAL PATH	TOTAL DISTANCE (KM)
1	**100**	20	10	100	5-21-38-15-33-24-7-37-16-39-35-30-36-27-10-22-6-19-18-23-17-14-20-34-28-2-11-4-32-3-13-12-26-29-8-1-31-25-40-9-5	369.5219
2	**200**	20	10	100	38-7-33-17-14-18-19-26-31-11-1-24-28-5-21-10-8-6-3-15-37-13-30-39-16-29-27-36-34-35-2-20-25-22-23-4-9-12-40-32-38	361.5687
3	**300**	20	10	100	22-9-3-14-19-39-36-12-5-21-4-38-10-40-24-8-33-2-32-23-6-11-27-16-20-7-31-15-25-37-29-35-13-30-1-26-28-34-17-18-22	352.8923
4	**400**	20	10	100	24-29-36-10-22-32-21-4-3-31-6-8-1-23-5-11-38-40-9-2-12-26-27-30-35-7-28-33-14-17-18-39-34-13-15-19-37-16-25-20-24	347.9193
5	100	**40**	10	100	32-3-23-21-4-2-31-19-18-14-24-27-37-36-20-15-7-26-30-8-39-35-29-16-17-25-33-34-28-9-12-13-5-1-40-10-22-38-11-6-32	340.0800
6	100	**60**	10	100	6-31-21-11-4-10-16-36-28-1-26-12-13-2-34-35-27-15-14-17-18-19-39-25-37-29-9-38-32-5-22-23-24-8-7-30-40-33-20-3-6	326.4160
7	100	**80**	10	100	3-11-10-40-5-31-25-14-15-16-2-32-23-13-24-33-17-18-19-20-8-39-37-30-12-38-9-1-27-36-29-35-34-26-28-7-21-4-22-6-3	304.4456
8	100	20	**20**	100	32-21-22-11-18-17-25-15-34-13-5-12-1-7-4-23-16-20-8-28-27-35-29-3-31-6-33-9-40-37-39-19-14-30-36-26-2-24-38-10-32	354.1758
9	100	20	**40**	100	29-30-27-2-32-7-8-38-10-4-3-6-22-21-33-20-39-16-28-12-13-15-17-25-18-5-40-9-1-23-11-24-31-14-19-37-36-35-26-34-29	327.2042
10	100	20	**60**	100	14-18-17-5-32-31-3-23-4-21-22-37-39-19-35-34-12-40-9-7-38-6-20-25-36-29-27-33-24-15-16-30-28-13-8-2-26-1-10-11-14	314.2312
11	100	20	10	**200**	5-21-38-15-33-24-7-37-16-39-35-30-36-27-10-22-6-19-18-23-17-14-20-34-28-2-11-4-32-3-13-12-26-29-8-1-31-25-40-9-5	332.8981
12	100	20	10	**300**	36-28-30-34-27-39-37-13-2-24-25-20-15-33-8-11-4-10-26-29-35-7-9-1-19-18-17-14-31-3-5-22-21-38-12-40-32-6-23-16-36	291.5491
13	100	20	10	**400**	31-14-37-36-15-25-33-3-6-23-10-28-34-27-8-24-32-38-22-21-4-11-2-13-35-29-18-17-19-20-39-16-30-7-1-26-12-9-40-5-31	280.7454
14	**400**	**80**	**60**	**400**	38-33-31-23-6-14-18-17-19-20-15-25-16-13-28-27-35-29-36-37-39-8-32-5-22-21-4-11-3-24-2-7-1-9-30-34-26-12-40-10-38	216.7149
15	**500**	**250**	**200**	**2000**	**10-40-9-12-26-34-30-13-28-1-38-7-2-8-24-31-14-17-18-19-39-16-36-29-35-27-37-32-11-3-33-15-20-25-6-23-4-21-22-5-10**	**190.0594**

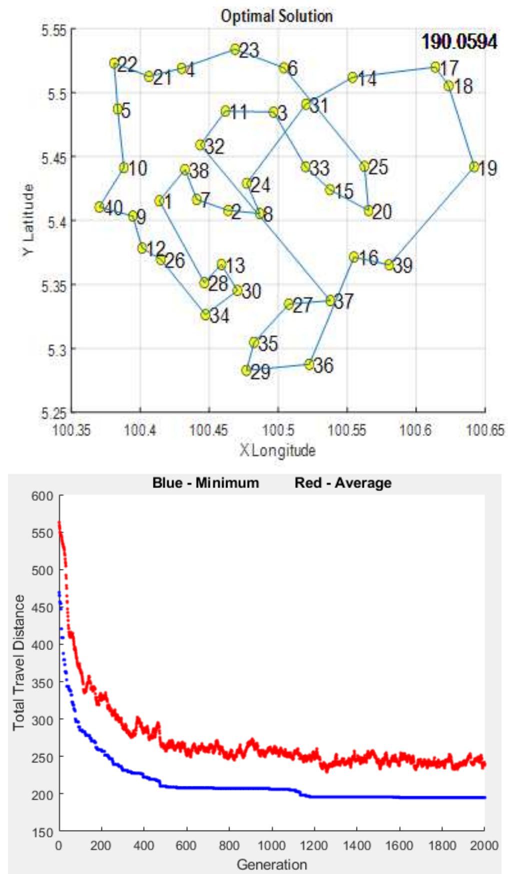

Figure 11: Best path of LSP (top) and the total travel distance of each generation of LSP (bottom) for 40 cities in combination 15.

3.2 Case NLSP

As previously mentioned, for NLSP, the user chooses the first and the last city. When the parameter values were increased in combination 15, the best total distance travelled obtained was 97.2089, as shown in Table 4. The optimal path for 20 cities of NLSP is shown in Figure 12 (top), with 7 and 18 as their first and last city, respectively.

The NLSP for 30 cities yields a total distance travelled of 143.3506 when the parameter value are increased (**p**=500, **c**=180, **m**=160, & **tg**=2000) in combination 15, as shown in Table 5. Figure 13 (bottom) shows that it requires 1640th-1650th generation for the algorithm to reach the optimal solution. Lastly, for 40 cities, when the parameter values were increased (**p**=500, **c**=180, **m**=160, & **tg**=2000) in combination 15, the best total distance travelled obtained is 182.4416, as demonstrated in Table 6. Here, Figure 14 (top) depicts the best path, with the first and last cities being 21 and 36, respectively. Meanwhile, Figure 14 (bottom) shows the blue curve plotted until the generation limit is reached at around the 1940th-1950th generation.

Table 4: Result of NLSP for 20 cities.

	p	c	m	tg	OPTIMAL PATH	TOTAL DISTANCE (KM)
1	**100**	20	10	100	7-1-9-2-10-4-5-13-12-15-20-16-8-11-3-17-6-14-19-18	157.1663
2	**200**	20	10	100	7-12-4-3-11-2-13-8-5-10-1-9-15-20-16-19-14-6-17-18	147.5789
3	**300**	20	10	100	7-12-2-5-10-9-1-13-8-15-6-4-11-3-14-20-16-19-17-18	131.8495
4	**400**	20	10	100	7-2-13-12-9-11-19-15-20-16-8-1-10-5-4-6-3-14-17-18	128.4789
5	100	**40**	10	100	7-15-13-12-9-10-1-2-3-6-5-4-11-8-19-16-20-14-17-18	147.2062
6	100	**60**	10	100	7-11-12-9-2-8-13-1-10-5-4-3-6-14-15-16-19-20-17-18	134.1472
7	100	**80**	10	100	7-9-1-12-13-16-19-20-15-8-2-10-4-5-11-6-3-14-17-18	122.6802
8	100	20	**20**	100	7-3-14-8-15-13-12-9-10-5-4-2-1-11-6-20-16-19-17-18	148.5477
9	100	20	**30**	100	7-5-4-10-9-1-12-13-20-15-16-19-17-14-6-2-8-11-3-18	143.3152
10	100	20	**40**	100	7-2-4-5-1-9-12-10-11-3-8-15-20-16-13-6-14-17-19-18	136.3097
11	100	20	10	**200**	7-4-11-6-3-19-20-16-15-8-2-13-12-9-1-10-5-14-17-18	125.0665
12	100	20	10	**300**	7-9-12-13-2-8-16-20-15-3-1-10-5-4-11-6-14-17-19-18	113.7228
13	100	20	10	**400**	7-1-9-12-13-16-20-14-6-3-11-4-5-10-2-8-15-19-17-18	111.8790
14	**400**	**80**	**40**	**400**	7-2-8-13-12-9-1-10-5-4-11-6-3-15-16-20-19-14-17-18	101.9473
15	**500**	**180**	**160**	**2000**	**7-2-8-13-12-9-1-10-5-4-11-3-6-14-15-16-20-19-17-18**	**97.2089**

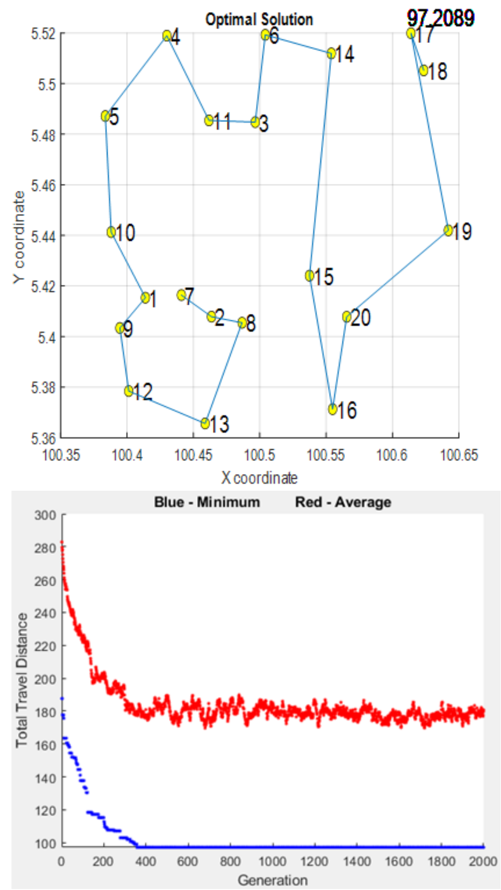

Figure 12: Best path of NLSP (top) and the total travel distance of each generation of NLSP (bottom) for 20 cities in combination 15.

Table 5: Result of NLSP for 30 cities.

	p	c	m	tg	OPTIMAL PATH	TOTAL DISTANCE (KM)
1	**100**	20	10	100	17-14-24-15-25-11-26-3-4-22-5-10-12-9-19 -20-16-27-28-7-1-2-8-13-30-29-18-6-23-21	246.6510
2	**200**	20	10	100	17-14-23-6-24-8-25-18-19-16-20-13-30-28 -27-15-2-12-26-1-7-29-10-9-11-5-3-22-4-21	232.8752
3	**300**	20	10	100	17-18-23-3-14-6-15-2-8-11-7-28-29-30-5-4 -24-12-27-16-20-19-25-13-26-9-1-10-22-21	230.3376
4	**400**	20	10	100	17-14-19-18-6-3-11-1-7-13-4-22-5-10-9-12 -20-16-30-28-29-8-24-25-15-27-26-2-23-21	227.7663
5	100	**40**	10	100	17-25-3-23-6-11-9-2-8-15-16-27-28-5-24- 20-19-18-14-22-4-29-30-13-26-10-7-12-1-21	248.6743
6	100	**60**	10	100	17-6-23-14-18-25-12-5-4-3-11-8-2-28-9-1- 24-15-20-30-27-19-16-29-13-26-7-10-22-21	240.3747
7	100	**80**	10	100	17-16-15-8-24-20-19-18-3-14-6-23-9-2-7-1 -25-30-13-28-27-29-12-26-5-11-4-10-22-21	235.1114
8	100	20	**20**	100	17-19-8-23-18-14-25-20-15-24-3-6-22-5-4- 11-7-26-13-2-1-30-16-29-28-9-27-12-10-21	249.0890
9	100	20	**30**	100	17-18-19-20-14-25-3-7-10-9-30-27-13-1-22- 5-6-8-2-4-23-16-15-29-28-12-26-24-11-21	241.6949
10	100	20	**40**	100	17-18-16-30-20-11-2-8-1-25-19-6-3-4-10-9- 28-13-27-29-26-12-7-24-15-14-23-5-22-21	235.5949
11	100	20	10	**200**	17-18-19-25-1-28-29-27-30-2-8-16-24-9-12- 26-13-7-5-6-14-3-23-24-15-10-22-11-4-21	223.4463
12	100	20	10	**300**	17-11-6-23-4-7-29-27-13-8-24-15-16-20-19- 18-14-25-12-9-3-2-26-30-28-1-10-5-22-21	215.9092
13	100	20	10	**400**	17-18-19-16-27-29-28-9-10-12-1-13-30-26- 20-2-8-7-24-15-25-14-6-20-3-22-5-11-4-21	192.4920
14	**400**	**80**	**40**	**400**	17-18-19-20-25-15-8-7-1-10-9-12-26-13-28 -30-2-3-6-11-4-23-14-16-29-27-24-5-22-21	174.1450
15	**500**	**180**	**160**	**2000**	**17-18-19-20-16-15-25-14-11- 7-1-10-9-12-26-28-13-30-29 -27-8-2-24-3-6-23-4-22-5-21**	**143.3506**

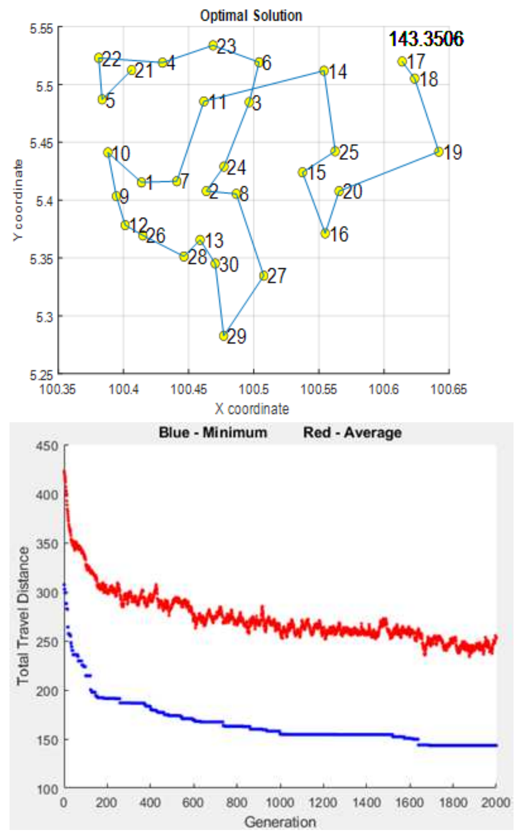

Figure 13: Best path of NLSP (top) and the total travel distance of each generation of NLSP (bottom) for 30 cities in combination 15.

As a heuristic search method, GA might not find the best solution after one run (result at combination 15 from Table 1-6). Therefore, GA needs to be run several times, where the best result achieved from all the runs should be selected as the best solution. If the number of runs is large enough, the best solution obtained could be considered optimal [16]. In this study, the GA using BFM's parameter set (combination 15 from Table 1-6) was run ten times for each case of LSP and NLSP. As a result, the best solution is achieved, where the details are shown in Tables 7 and 8. Hence, the best solution of LSP for 20, 30 and 40 cities are 98.6670, 151.3608 and 190.0594, respectively. Meanwhile, 97.2089, 143.3506 and 182.4416 are the best solutions for NLSP for 20, 30, and 40 cities, respectively.

Table 6: Result of NLSP for 40 cities.

	p	c	m	tg	OPTIMAL PATH	TOTAL DISTANCE (KM)
1	**100**	20	10	100	21-9-20-19-15-16-37-27-17-18-31-26-12 -29-39-3-4-32-33-23-6-11-2-24-25-10 -5-22-28-34-13-1-40-30-7-8-14-38-35-36	379.8084
2	**200**	20	10	100	21-22-29-34-35-37-15-11-6-31-3-16-19 -2-28-30-13-18-14-17-23-10-5-1-8-9-32 -4-7-40-24-33-39-25-20-38-26-12-27-36	359.0075
3	**300**	20	10	100	21-22-4-5-32-1-26-13-34-35-39-19-38- 15-3-11-16-33-9-40-23-6-25-14-31-8-7- 10-24-27-29-28-12-20-37-18-17-2-30-36	354.6371
4	**400**	20	10	100	21-3-25-16-39-26-5-1-30-10-9-13-40-17- 18-14-31-6-15-33-19-20-37-29-7-11-4- 22-8-24-12-34-28-32-38-23-2-27-35-36	352.8958
5	100	**40**	10	100	21-4-24-34-16-8-18-17-11-3-31-23-6-32- 30-29-27-28-13-25-39-35-20-2-9-7-22-5 -38-26-12-14-19-33-1-10-40-15-37-36	361.1300
6	100	**60**	10	100	21-23-11-3-12-2-10-9-38-13-34-28-29-35 -33-6-5-1-30-27-7-32-14-15-8-39-16-37- 24-17-18-25-20-19-31-4-22-40-26-36	331.9170
7	100	**80**	10	100	21-6-23-3-38-11-17-31-15-19-18-32-39- 16-20-37-33-14-25-7-26-27-28-8-2-1-12 -40-10-4-22-5-9-29-34-30-24-13-35-36	323.0823
8	100	20	**20**	100	21-38-4-33-31-25-15-26-40-22-10-12-1- 7-9-34-18-6-11-32-19-17-20-28-30-27- 39-8-24-14-16-37-2-5-23-3-13-29-35-36	363.9887
9	100	20	**30**	100	21-32-22-1-9-26-34-5-40-13-29-12-2-25- 17-19-16-37-3-4-23-6-31-14-10-38-15-8 -33-18-39-27-7-11-24-20-30-35-28-36	359.7035
10	100	20	**40**	100	21-10-9-40-14-17-19-31-16-18-25-15-13 -28-26-1-3-6-37-27-30-7-2-20-39-33-24- 8-38-22-23-4-5-12-32-11-29-34-35-36	327.3437
11	100	20	10	**200**	21-22-24-2-5-1-13-34-33-31-32-38-29-37 -8-7-10-40-4-23-11-16-20-3-14-17-18-25 -6-39-19-15-12-9-26-28-30-35-27-36	316.5966
12	100	20	10	**300**	21-23-31-6-3-33-11-4-30-35-29-39-16-15- 19-14-8-2-26-28-12-40-5-22-20-25-17-18 -24-10-38-32-9-1-7-27-34-13-37-36	294.4366
13	100	20	10	**400**	21-4-6-15-20-16-19-18-17-14-33-24-9-35 -34-32-3-31-23-11-25-8-39-37-13-2-7-1- 40-5-38-22-10-12-26-28-27-30-29-36	273.1390
14	**400**	**80**	**40**	**400**	21-22-4-23-6-3-1-2-8-33-24-5-26-12-9-7- 28-30-13-34-27-37-16-31-14-11-32-40-10 -38-15-20-17-18-19-25-39-35-29-36	246.1489
15	**500**	**180**	**160**	**2000**	**21-4-22-5-9-12-26-13-28-34-30-27-37-16- 39-20-25-15-33-24-2-7-1-40-10-32-38 -11-23-6-14-17-18-19-31-3-8-35-29-36**	**182.4416**

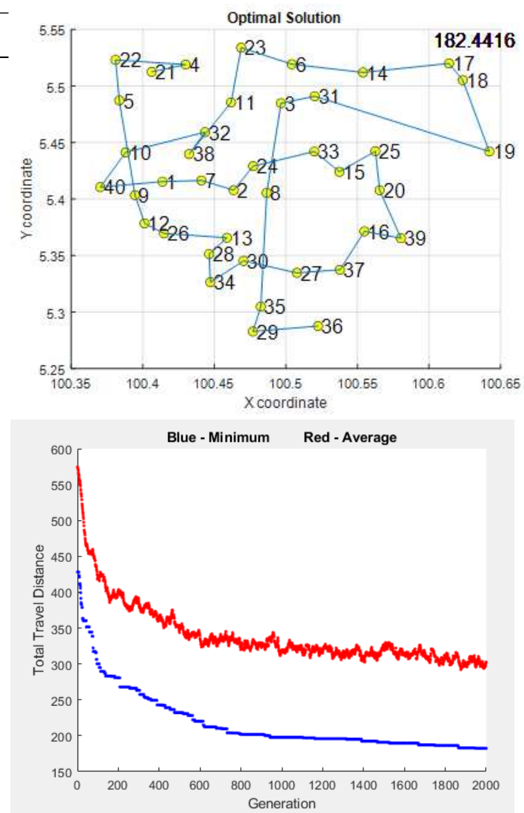

Figure 14: Best path of NLSP (top) and the total travel distance of each generation of NLSP (bottom) for 40 cities in combination 15.

Table 7: The best solution of LSP

NO. OF CITY	SOLUTION ACHIEVED FOR EACH RUN										BEST OPTIMAL SOLUTION (KM)
	1	2	3	4	5	6	7	8	9	10	
10	101.7359	98.6670	101.7359	98.6670	101.7359	98.6670	98.6670	101.7359	98.6670	98.6670	**98.6670**
20	151.3608	156.0507	151.6785	153.5134	153.5134	151.3608	156.3367	151.6785	151.6785	153.5134	**151.3608**
30	195.6831	190.0594	197.1556	200.2895	190.0594	191.3743	200.2895	190.0594	197.1556	195.6831	**190.0594**

Table 8: The best solution of NLSP

NO. OF CITY	SOLUTION ACHIEVED FOR EACH RUN										BEST OPTIMAL SOLUTION (KM)
	1	2	3	4	5	6	7	8	9	10	
10	98.7264	100.6948	100.6948	97.2089	98.7264	97.2089	100.6948	98.7264	97.2089	97.2089	**97.2089**
20	148.8811	147.0611	150.1859	143.3506	144.3346	147.0611	148.8811	150.1859	148.8811	143.3506	**143.3506**
30	200.8849	203.0738	182.4416	197.1363	203.0738	182.4416	197.1363	200.8849	197.1363	196.8559	**182.4416**

4 Conclusion and Future Work

In this paper, GA is applied to two different cases, which are LSP and NLSP. The main advantages of GA are their flexibility and robustness as a global search method. To find the best solution to this problem, a GA with chromosome representations and genetic operations is developed. BFM is used to obtain the best combinations of parameter sets by running it several times. In this research, the distance is computed using the Haversine formula. BFM successfully provides a satisfying result, given the random increment of parameter values. For future works, other relevant optimization algorithms such as PSO or ACO could be employed and compared with the results obtained in this paper, especially in terms of time computation and error. Additionally, the development of systematic strategies that could assist in finding optimal parameters is deemed a worthy research direction.

Acknowledgments

This research was supported by Universiti Sains Malaysia through the RUTeam Grant Scheme (1001/PMATHS/8580071) and the School of Mathematical Sciences, Universiti Sains Malaysia. In addition, the authors are very grateful to the anonymous referees for their valuable suggestions.

References

[1] Mogaji, E. Impact of COVID-19 on transportation in Lagos, Nigeria. *Transportation research interdisciplinary perspectives*. 2020. 3: 100154.

[2] Triana, Y. S. and Syahputri, I. Implementation Floyd-Warshall Algorithm for the Shortest Path of Garage. *International Journal of Innovative Science and Research Technology*. 2018. 3(2): 871–878.

[3] Mirino, A. E. Best routes selection using Dijkstra and Floyd-Warshall algorithm. *2017 11th International Conference on Information & Communication Technology and System (ICTS). IEEE* 2017. 155–158.

[4] Ghamry, K. A., Kamel, M. A. and Zhang, Y. Multiple UAVs in forest fire fighting mission using particle swarm optimization. *2017 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE*. 2017. 1404–1409.

[5] Ranaweera, D. M., Hemapala, K. U., Buddhika, A. G. and Jayasekara, P. A shortest path planning algorithm for PSO base firefighting robots. *2018 Fourth International Conference*

on Advances in Electrical, Electronics, Information, Communication and Bio-informatics (AEEICB). IEEE. 2018. 1–5.

[6] Adamu, P. I., Jegede, J. T., Okagbue, H. I. and Oguntunde, P. E. Shortest path planning algorithm–a Particle Swarm Optimization (PSO) approach. *2018 Proceedings of the World Congress on Engineering*. 2018. 1.

[7] Kumar, P. M., Manogaran, G., Sundarasekar, R., Chilamkurti, N. and Varatharajan, R. Ant colony optimization algorithm with internet of vehicles for intelligent traffic control system. *Computer Networks*. 2018. 144: 154–162.

[8] Safaruddin, M. S., and Misro, M. Y Multi-Objectives Path Planning using Bzier Curve. *Malaysian Journal of Mathematical Science*. 2021. 15(1): 45–59.

[9] Wang, X., Shi, H. and Zhang, C. Path planning for intelligent parking system based on improved ant colony optimization. *IEEE Access*. 2020. 8: 65267–65273.

[10] Ebadinezhad, S. DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem. *Engineering Applications of Artificial Intelligence*. 2020. 92: 103649.

[11] Liu, H., Xu, B., Lu, D. and Zhang, G. A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm. *Applied Soft Computing*. 2018. 68: 360–376.

[12] Kamil, R. T., Mohamed, M. J. and Oleiwi, B. K. Path Planning of Mobile Robot Using Improved Artificial Bee Colony Algorithm. *Engineering and Technology Journal*. 2020. 38(9A): 1384–1395.

[13] Lamini, C., Benhlima, S. and Elbekri, A. Genetic algorithm based approach for autonomous mobile robot path planning. *Procedia Computer Science*. 2018. 127: 180–189.

[14] Prasetya, D. A., Nguyen, P. T., Faizullin, R., Iswanto, I. and Armay, E. F. Resolving the shortest path problem using the haversine algorithm. *J. Crit. Rev*. 2020. 7(1): 62–64.

[15] Dauni, P., Firdaus, M. D., Asfariani, R., Saputra, M. I. N., Hidayat, A. A. and Zulfikar, W. B. Implementation of Haversine formula for school location tracking. *IOP Publishing*. 2019. 1402(7): 077028.

[16] Dao, S. D., Abhary, K., and Marian, R. Optimisation of partner selection and collaborative transportation scheduling in Virtual Enterprises using GA. *Procedia Computer Science. Elsevier*. 2014. 41(15): 6701–6717.