



# The Optimization of Routes Using Evolutionary Algorithms in Public Transportation Systems

Salih Serkan Kaleli<sup>1\*</sup>, Mehmet Bayğan<sup>2</sup>, Abdullah Naralan<sup>3</sup>

<sup>1</sup> Ardahan University, Department of Office Management and Secretary, Ardahan, Turkey

<sup>2</sup> Ardahan University, Department of Computer Engineering, Ardahan, Turkey

<sup>3</sup> Recep Tayyip Erdoğan University, Department of Research Information System, Rize, Turkey

salihserkankaleli@ardahan.edu.tr, mehmetbaygin@ardahan.edu.tr, abduallah.naralan@erdogan.edu.tr

## Abstract

This study aims to examine, regulate, and update the land transportation of the Erzurum Metropolitan Municipality (EMM), Turkey using computerized calculation techniques. In line with these targets, some critical information has been obtained for study: the number of buses, the number of expeditions, the number of bus lines, and the number and maps of existing routes belonging to EMM. By using the information that has been obtained, this study aims at outlining specific outputs according to the input parameters, such as determining the optimal routes, the average travel, and the journey time. Once all of these situations were considered, various optimization algorithms were used to get the targeted outputs in response to the determined input parameters. In addition, the study found that the problem involved in modeling the land transport network of the EMM is in line with the so-called "traveling salesman problem," which is a scenario about optimization often discussed in the literature. This study tried to solve this problem by using the genetic algorithm, the clonal selection algorithm, and the DNA computing algorithm. The location data for each bus stops on the bus lines selected for the study were obtained from the EMM, and the distances between these coordinates were obtained by using Google Maps via a Google API. These distances were stored in a distance matrix file and used as input parameters in the application and then were put through optimization algorithms developed initially on the MATLAB platform. The study's results show that the algorithms developed for the proposed approaches work efficiently and that the distances for the selected bus lines can be shortened.

**Keywords:** Optimization, Evolutionary Algorithms, Traveling Salesman Problem, Genetic Algorithm, Management Information Systems

## Toplu Taşıma Sistemlerinin Evrimsel Algoritmalarla Optimizasyonu

### Özet

Bu çalışma, Erzurum Büyükşehir Belediyesi'nin (EBB) Türkiye kara ulaşımını bilgisayarlı hesaplama teknikleri kullanarak incelemeyi, düzenlemeyi ve güncellemeyi amaçlamaktadır. Bu hedefler doğrultusunda, çalışma için bazı önemli bilgiler: otobüs sayısı, sefer sayısı, otobüs hattı sayısı ve EBB'ye ait mevcut güzergâh sayısı ve haritaları elde edilmiştir. Bu çalışma, elde edilen bilgileri kullanarak, optimal rotaların belirlenmesi, ortalama seyahat ve yolculuk süresi gibi girdi parametrelerine göre belirli çıktıların ana hatlarını çizmeyi amaçlamaktadır. Tüm bu durumlar göz önüne alındığında, belirlenen girdi parametrelerine karşılık hedeflenen çıktılar elde etmek için çeşitli optimizasyon algoritmaları kullanılmıştır. Çalışma, EBB' nin ulaşım ağının modellenmesindeki problemin, literatürde sıklıkla tartışılan optimizasyonla ilgili bir senaryo olan "gezgin satıcı problemi" ile uyumlu olduğunu bulmuştur. Çalışmada genetik algoritma, klonal seçim algoritması ve DNA hesaplama algoritması kullanılarak bu problem çözülmeye çalışılmıştır. Çalışmada seçilen otobüs hatlarındaki her bir durak için konum bilgisi EBB'den alınmış ve bu koordinatlar arasındaki mesafeler bir Google API üzerinden Google Maps kullanılarak elde edilmiştir. Bu mesafeler bir mesafe matrisi dosyasında saklanmış ve uygulamada giriş parametreleri olarak kullanılmış daha sonra MATLAB platformunda geliştirilen optimizasyon algoritmalarına aktarılmıştır. Çalışmanın sonuçları, önerilen yaklaşımlar için geliştirilen algoritmaların verimli çalıştığını ve seçilen otobüs hatları için mesafelerin kısaltılabileceğini göstermektedir.

**Anahtar kelimeler:** Optimizasyon, Evrimsel Algoritmalar, Gezgin Satıcı Problemi, Genetik Algoritma, Yönetim Bilişim Sistemleri

\* Corresponding Author.

E-mail: salihserkankaleli@ardahan.edu.tr

Received : 11 June 2021

Revision : 29 Nov. 2021

Accepted : 17 Dec. 2021

## 1. Introduction

The rapid and recent developments in computer and internet technology have enabled many processes to be done fully automatically and quickly. In addition, in light of these developments, although the problems encountered in daily life can occur rapidly, they can also be resolved much more easily using this technology. In parallel with this situation, it was inevitable that optimization processes would be used for constantly encountered problems in daily life. The optimization concept refers to a process developed every day and aims to facilitate the processes involved in everyday life. These can be used by human beings to use time more efficiently and solve problems in their work. In addition, by developing this concept every day and moving it to a higher level, the number of problems can be solved can be increased (Deng et al. 2017). At this point, the developments in computer technology and the continuous increases made in terms of processor power have downsized a large amount of computing time, which is a frequently encountered problem when utilizing optimization processes. Using this increase in processor power, problems that require a vast solution space can be calculated in a much shorter time and in a much more accurate way.

Optimization processes are encountered various fields, such as engineering, design, financial planning, holiday planning, computer science, and industrial computing (Sundararaghavan et al. 2010). Human beings have always aimed to maximize or minimize a purpose. For example, the effort at minimizing current expenses while trying to maximize the profit of an enterprise is something wanted by many human beings. For this reason, humankind has made it a goal to choose the most suitable solution for almost every problem. The aim of achieving the best price-performance balance when purchasing goods and when running or performing more than one business in parallel with each other over a specific period forms the basis of having an optimization process.

There are various optimization methods studied in the literature. These methods are generally based on numerical calculations, and the problems with a vast and possible solution area can be solved in a concise amount of time. Thanks to the development of these methods, completely tailored solutions can be produced and specific targets for success can be achieved. Examples of the best known of these methods include the genetic algorithm, the ant colony algorithm, particle swarm optimization, the differential development algorithm, the clonal selection algorithm, and the DNA computing algorithm. Although there are various algorithms in the literature, there are some basic steps involved in optimization algorithms. A block diagram showing these steps is given in Figure 1.

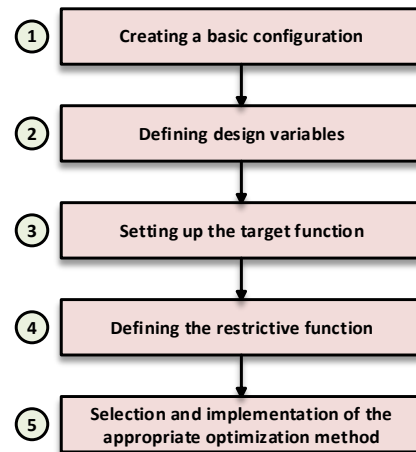
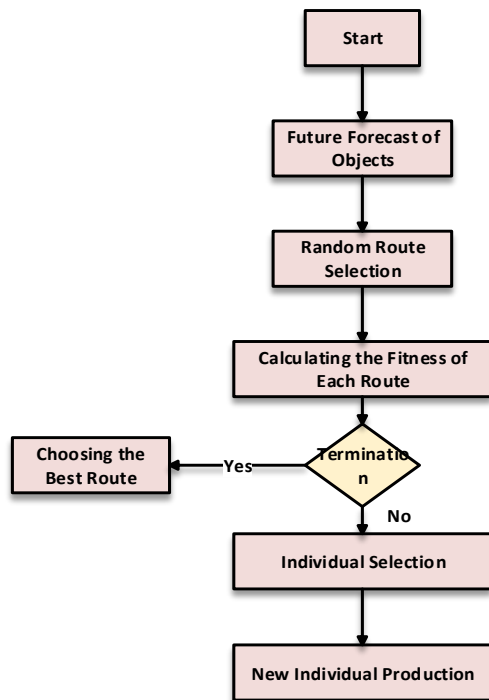


Figure 1. Steps to Follow for Optimization Problem

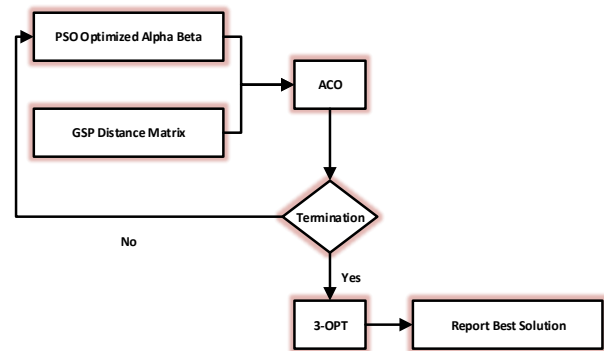
As shown Figure 1, the first step in an optimization problem is to create the basic configuration. After this step, it is necessary to determine the design variables and define them for the problem. In the third step toward the solution, the objective function should be determined and be suitable for the problem. After this process, the restrictive statements in the problem should be defined in the application, and the problem should be designed according to these constraints. In the last stage of the system, a suitable optimization method should be selected and then implemented. Optimization algorithms are used in many different fields (Guo et al. 2007). One of these studies sought to solve the traveling salesman problem by using the genetic algorithm. In the study, crossover and mutation steps were used and the mutation process was carried out in two stages in terms of scrolling and adding. After the crossover process was complete, the individuals were checked and an attempt was made to determine if any deterioration to their condition had occurred. The approach proposed within the scope of the study has been tested in two different cities, 40 and 100. It has been observed that the proposed method contributes significantly to solving the GSP (Chen 2013)-(Bolat and Cortés 2011) proposed an optimization approach that was based on the Genetic Algorithm and the Tabu Search algorithm for a group of elevator systems. In this study, the floor numbers and cabin numbers were grouped and tests were carried out in different combinations of these groups. In the proposed approach, it was observed that the average wait time and the average travel times of the passengers were lowered, and it was also found that the Genetic Algorithm yielded better results than the Tabu search algorithm. Another study on this subject, by (Groba et al. 2015), used the combination of a prediction technique and the genetic algorithm, which is an intuitive method, for the solution to the dynamic traveling salesman problem. In this study, route optimization was done for a scenario in which the targets were constantly moving. For this purpose, a genetic algorithm was created that feeds in Newton's motion equation. The study results proved that the trajectory approach using the prediction-

based genetic algorithm provides better and more effective results than the methods commonly used, as noted in the literature. A flow diagram summarizing the trajectory predictions based on the approach that used the genetic algorithm method developed within the scope of the study is shown in Figure 2.



**Figure 2.** Genetic Algorithm Application for Dynamic TSP

In a study conducted by (Baygin and Karakose 2013), the genetic algorithm, the clonal selection algorithm, and the DNA computing algorithm were used, and the timing of a group of elevator systems was checked. For this purpose, an elevator system with 20 floors and five cabins was simulated and examined using the three different optimization methods. The study aimed to decrease the average wait time of the passengers, decrease the average amount of their travel time, and decrease the amount of energy that was consumed by the cabins. In another study that took up the traveling salesman problem, a hybrid approach was proposed and was based on the particle flock algorithm, the ant colony algorithm, and the 3-Opt algorithm. In this study, the particle flock algorithm was used to optimize the parameters affecting the performance of the ant colony algorithm. In addition, the 3-Opt intuitive method, the other method used in the study, was used to develop localized solutions. The experimental studies conducted in this research showed that the proposed approach yielded better results than many others found in the literature in terms of the quality and accuracy of the solution (Mahi et al. 2015). A block diagram that summarizes this study is shown in Figure 3.



**Figure 3.** Ant Colony and 3-Opt Algorithm Based Optimization Process for TSP

This study, which was conducted in Turkey, aims at using optimization algorithms to optimize actively used bus lines in the city of Erzurum. In this context, certain lines have been optimized, and the purpose of the study was to find out how to provide a faster, better, and more effective service for bus passengers. First, six different line routes actively used by the municipality were selected and the GPS data for each of the stops on these lines were obtained. The GPS data were sent to Google Maps via a Java program using the Google API, and the distance matrix files were drawn from Google Maps. This distance was used to calculate the conformity function in the matrix file for the optimization algorithms. After these processes, the clonal selection algorithm, the genetic algorithms, and the DNA computing algorithms were developed from an artificial immune system in a MATLAB environment. These previously selected routes were subjected to optimization. After all of these processes, new routes were configured for each of the line routes and these were compared with the currently used routes. As a result of the simulations carried out, significant improvements were achieved on the six different routes. The distance for each of the routes currently in use has been significantly shortened. In addition, the results have been marked on Google Maps and the new routes are visualized.

The traveling salesman problem is examined in the second part of this study. In the third section, the algorithms used for the optimization process are provided. The results of the simulations carried out are provided in the fourth part of the study and these results are given comparatively. In the fifth and last part of the study, further results and suggestions are included.

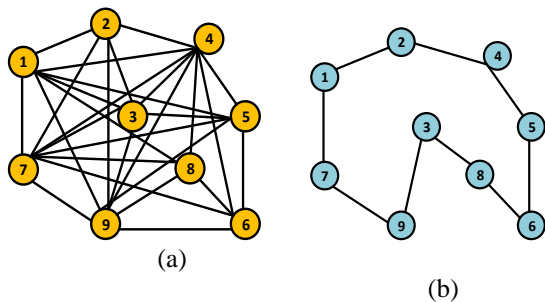
## 2. The Traveling Salesman Problem

The traveling salesman problem (TSP) is a scenario in which the purpose of a salesman is to sell his goods in an “n” different city. While performing this process, the traveller should visit the city only once and not visit the previously visited city. Determining the minimum length of the road route within the scope of these objectives and constraints is the primary purpose for solving the traveler salesman problem(Asadpour et al. 2010). A simple

version of the traveling salesman problem is summarized below (NARALAN et al. 2017);

- A traveler wants to sell his or her goods in “n” different cities. The salesperson wants to visit the cities as soon as possible and wants to visit each city alone. The problem aims to offer the seller the shortest route that has the least cost.
- In the first city, the seller has the right to choose between “n-1” different city roads.
- In the second city, the seller has the right to choose between “n-2” different city roads. In this way, the cities that the vendor has visited are subtracted from the total of the city “n,” and the process is continued again in the same way. When the problem is considered in general, the total “(n-1)!” are the many situations or possibilities.

Consequently, if the seller is assumed to be going to these cities in the opposite direction, “(n-1)! / 2” has to be chosen from the different routes taken at each step. In Figure 4, an exemplary version of the traveling salesman problem is given (NARALAN et al. 2017).

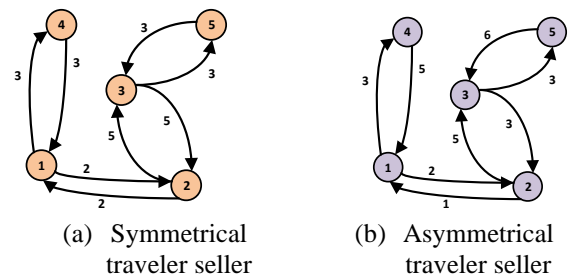


**Figure 4.** An Example Traveling Salesman Problem and Solution

As shown in Figure 4, the numbers represent the cities and the lines represent the routes between the cities. For example, a seller has to choose between  $29! / 2$  different tours for a tour of the 30 cities. From this point of view, the travelling salesman problem is the type of problem that belongs in the NP-Tam class (Kovács et al. 2018). The possibilities of the routes to be followed by the salesman, who wants to go from the number 1 city to the number 9 city as soon as possible, is shown in Figure 4. If the total number of cities to be visited by the salesman is shown with “n,” and the round-trip distances for these “n” cities that he visited may be the same or different (Wang et al. 2016). The main reasons for the different round-trip distances between the cities can be listed, including situations such as having to take a compulsory direction in traffic, the existence of traffic jams, and the compatibility of the round-trip lanes to be used. In other words, the cost calculated to go from “A” to “B” and the calculated cost to reach “A” from “B” may not always be the same. The literature defines this situation as the symmetrical / asymmetrical traveling salesman problem (Hasan Söyler 2007).

The symmetrical traveling salesman problem involves having a distance between the two cities equal to the

round-trip length. Depending on the situation, the costs to be spent for the distances during the journey are also equal. The opposite of the symmetric traveling salesman problem is the asymmetrical traveling salesman problem (Nguyen et al. 2002). There are some cases in which symmetrical problems are not always valid. Especially in big cities, traffic jams, one-way roads, bad weather, bad road conditions, etc. For reasons such as these, the definition of the problem will need to be more elaborate (Saji and Riffi 2016). Considering all of these situations, the problem of asymmetrical traveling salesman involves a scenario in which the round-trip distance between two cities or points and the cost to be covered at these distances are not equal (Choong et al. 2019). A block diagram illustrating the symmetrical and asymmetrical traveling salesman problem is presented in Figure 5.



**Figure 5.** Symmetric / Asymmetric Traveler Salesman Problem Status

Figures 5 (a) and (b) present the symmetrical and asymmetrical traveling salesman problem according to their distances. Figure 5- (a) shows that the round-trip distances between the same two-knot pairs are the same. On the other hand, in the symmetrical travelling salesman problem, the distances between the nodes remain the same for the round-trip and return. The opposite of this situation is given in Figure 5-(b). In this example, the round-trip distances between any two pairs of nodes are different from each other. When the data obtained within the scope of this study are examined, it can be observed that the optimization problem of the EMM public transportation network coincides with the asymmetric traveling salesman problem. For this reason, the study mainly aims to solve the problem of the asymmetric traveling salesman.

### 3. Proposed Method

Within the scope of this study, the optimization methods that can ascertain the optimal road routes for bus lines that are frequently encountered and used in daily life were researched and applied. For this purpose, data from the EMM Public Transportation Branch Directorate were provided, and the data were made ready for use in practice. After these steps, the distance between the coordinates of these stops was obtained using an application prepared using the Google API, and a distance matrix table was created. Three different



optimization methods were researched and applied in detail. In this context, the genetic algorithm, the DNA computing algorithm, and the clonal selection algorithm from an artificial immune system, which is one of the methods frequently used, as noted in the literature, were utilized. The approach proposed in this study was tested on six different routes and it aimed to shorten the routes that the buses actively follow. A block diagram that summarizes the system's flow in line with all of these objectives is shown in Figure 6.

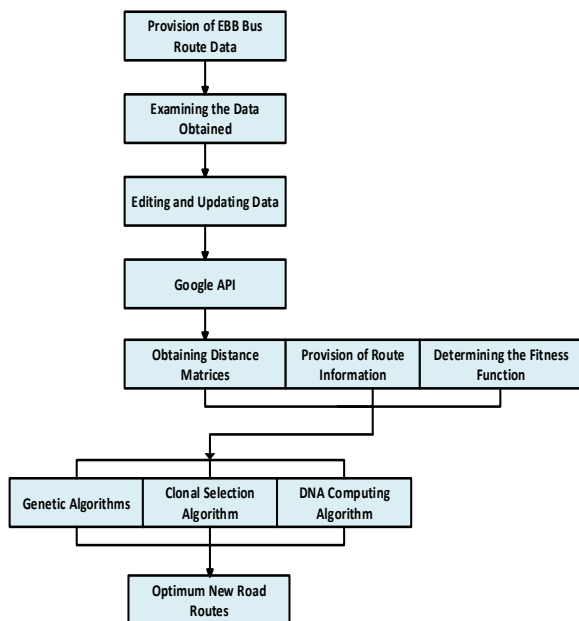


Figure 6. Steps of the Proposed Approach

As can be seen in Figure 6, three different optimization algorithms were used in the proposed method. These methods are the genetic algorithm, the clonal selection algorithm, and the DNA computing algorithm. The details for each of these optimization methods are presented below in the subsections.

### 3.1 The Genetic Algorithm

The genetic algorithm method consists of seven steps (Malhotra et al. 2011). A flow chart that summarizes the steps of the genetic algorithm is given in Figure 7, and the details for each of these steps are given below.

- **Random Start Population Creation:** The first step in using the genetic algorithm is to generate a random population based on the problem. Although there are various methods noted in the literature, e.g., binary coding and permutation coding, permutation coding was preferred for this study. Therefore, within the scope of this study, a 100-element population was created.
- **Calculate Fitness:** The next stage taken while using the genetic algorithm is to determine the purpose function and subject the individuals that were randomly created in the previous step to this function. The purpose function used in the application is presented in Equation 1.

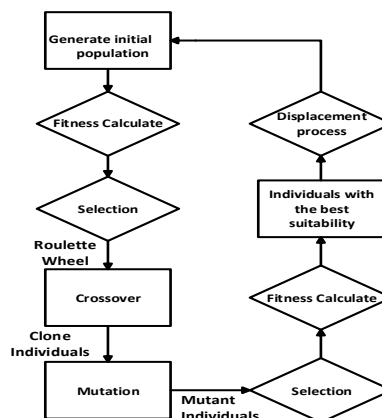


Figure 7. Genetic Algorithm Steps (Suman 2015)

$$\sum_{i=1}^{n-1} MT(P_i, P_{i+1}) \quad (1)$$

$n$  = Total number of stops  
 $P$  = Individual in the population  
 $MT$  = Distance table

- **Selection:** At this stage in using the genetic algorithm, the individuals are ranked from the best to the worst, depending on their fitness values. After this ranking process, a selection process is carried out. There are various selection approaches taken up in the literature. The best known of these selection procedures are called elitism, roulette wheel, and tournament selection (Mohammed et al. 2017). In this study, the roulette wheel method was preferred.

- **Crossover Process:** In this step in using the genetic algorithm, the individuals selected in the previous step are subjected to the crossover process. The primary purpose here is to cross different individuals with the best suitability and to obtain child individuals with good suitability, but in a different structure (Hiassat et al. 2017). In this study, a single-point crossover was applied.

- **Mutation Process:** In this part of the genetic algorithm, a mutation process is applied to the child individuals obtained at the crossover stage. The mutation process is carried out in reverse proportion to the suitability of the individuals.

- **Calculate Fitness and Change:** The next step in using the genetic algorithm after the mutation phase is the calculation and change process. The re-eligibility of the new child individuals obtained is calculated and subjected to displacement with the individuals in the randomly generated population at the beginning (Priyo Anggodo et al. 2016).

- **Termination:** In this section, which constitutes the last stage in using the genetic algorithm, how long the algorithm will work is specified and, at the end of this period, the aim is to terminate the algorithm. In studies carried out for this purpose, the termination criteria of the genetic algorithm were determined in terms of 1000 iterations.

### 3.2 The Clonal Selection Algorithm

A flow diagram showing the steps involved in the clonal selection algorithm, taken from the artificial immune system, is provided in Figure 8. In addition, the steps for this method are as shown below (Muthreja and Kaur 2018).

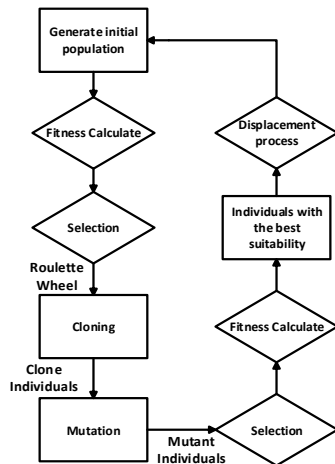


Figure 8. Clonal Selection Algorithm (Guney et al. 2007)

- **Initially Population Create:** The first step involved in the clonal selection algorithm is to generate random populations, just as in the genetic algorithm. In this step, in which permutation coding is preferred, again as in the genetic algorithm, random routes are created in terms of the population size specified by the user(Xu et al. 2016).
- **Calculate Fitness:** This stage involved in the clonal selection algorithm is the same process as in the genetic algorithm, and the calculation function given in equation 1 is used.
- **Selection:** At this stage in the clonal selection algorithm, the individuals with the best suitability are selected and these parent individuals are transferred to the next step, the crossover step. Again, in this algorithm, the roulette wheel method is preferred.
- **Cloning:** After the selection process is performed based on the suitability value, the next step is cloning, that is, the copying process of the selected “n” individuals(Shrikrishna et al. 2018). The primary purpose here is to copy the more well-suited individuals and fewer of the well-suited individuals. In this way, a new and temporary population can be obtained.
- **Mutation:** This is the process of replacing individuals in the temporary population by replicating the mutation process if certain conditions occur(Muthreja and Kaur 2018).
- **Calculate Fitness and Change:** The re-eligibility value of the temporary population that has been subjected to mutation is calculated and subjected to change.
- **Termination:** The last stage involved in the clonal selection algorithm is termination. This stage is the section that determines how long the clonal selection algorithm will operate.

### 3.3 The DNA Computing Algorithm

A flow diagram that summarizes the processes for the DNA computing algorithm is given in Figure 9.

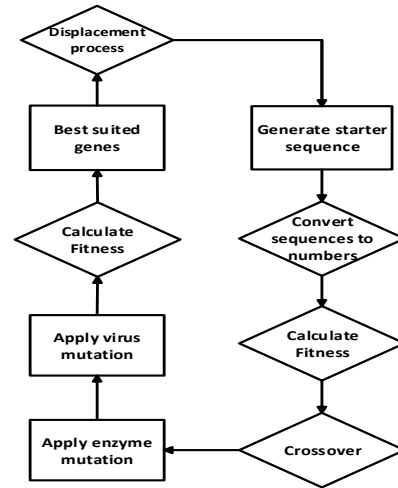


Figure 9. DNA Computing Algorithm Flow Diagram

- **Generating Random DNA Sequences:** The first step involved in the DNA computing algorithm is to generate random DNA sequences. DNA molecules are used in the creation of DNA sequences. These molecules are adenine, thymine, guanine, and cytosine. The numerical equivalents of these molecules are given below.

$$\begin{aligned}
 A &= 0, G = 1, C = 2, T = 3 \\
 AAA &= 0x4^0 + 0x4^1 + 0x4^2 = 0 \\
 TTT &= 3x4^0 + 3x4^1 + 3x4^2 = 63
 \end{aligned}$$

As can be seen from this example, by bringing the three molecules side by side, DNA sequences of up to 63 stops can be obtained. In this study, the maximum number of stops in the selected lines is 60, and all of these stops can be expressed in terms of the three-string DNA sequences. In this context, when creating a random population, the DNA sequences of three are combined side by side and a DNA helix given in the total equation 2 is obtained. A sample DNA sequence randomly generated for an 18-stop route is given in Figure 10.

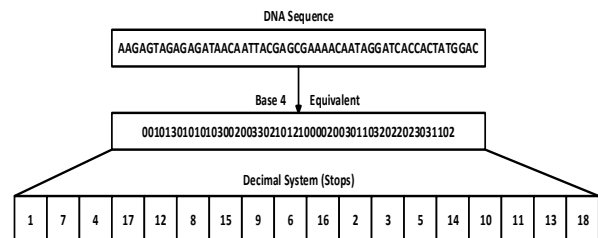


Figure 10: DNA Computing Algorithm Flow Diagram

- **Numerical Value Cycle:** The DNA helix obtained in the first stage of the application is converted into real numerical values in the second stage of the application(Zhang 2018).
- **Calculate Fitness:** A conformity calculation is carried out as in the other two optimization methods, and

Equation 1 is used for this purpose.

- **Crossover Process:** In the fourth stage used in the DNA computing algorithm, the crossover process is applied to the selected DNA sequences. The crossover is done on DNA strands, not on the numerical values.
- **Enzyme and Virus Mutation:** In the DNA computing algorithm, a two-stage mutation process is applied to the DNA sequences. The first of these processes is the enzyme mutation, which is the process of deleting any DNA sequences randomly selected from the DNA sequence. The second is a virus mutation, which involves replacing a randomly deleted DNA sequence with a newly produced DNA sequence (Dodge et al. 2020).
- **Calculate Fitness and Change:** This process is performed in the DNA computing algorithm and is applied just as in the other two methods. At this stage, The mutated DNA sequences are converted back into numerical data and their suitability is calculated (Ibrahim et al. 2018).
- **Termination:** A termination criterion is also used for the DNA computing algorithm. With this termination process, the DNA computing algorithm is stopped at a certain point and the result obtained is recorded as the solution. As with the other methods, 1000 iterations were preferred in using the DNA computing algorithm.

In this study, three different optimization methods were used for solving the traveling salesman problem. These methods were tested for the actively used routes, and the results obtained have been compared. The comparative steps for each of the algorithms used in the application are presented in Table 1.

**Table 1.** Comparison of the steps of the algorithms used

Step	GA	CSA	DNA
<b>Start</b>	- Determining the type of coding - Determination of fitness functions, population size and termination criteria		
<b>1</b>	Creating random populations	Creating random populations	Creating random populations
<b>2</b>	Calculate Fitness	Calculate Fitness	Calculate Fitness
<b>3</b>	Selection process	Selection process	Selection process
<b>4</b>	Crossover	Cloning	Crossover
<b>5</b>	Mutation	Mutation	-Enzyme mutation -Virus mutation
<b>6</b>	Calculate Fitness	Calculate Fitness	Calculate Fitness
<b>7</b>	Displacement	Displacement	Displacement
<b>End</b>	Repeat process from step 3 according to termination criteria		

#### 4. Simulation Results

Six line routes, which the EMM public transportation administration actively uses, have been optimized using

the steps outlined in this study. In the optimization processes, three different methods were used: clonal selection, genetics, and the DNA computing algorithm. In this study, the distance matrices are used in the calculation process, the results are then obtained, and then a comparison of the three optimization methods is made. The optimization methods take the distance matrix files as the input parameters. These distance matrix files were obtained through the Google API using GPS coordinates, and sample GPS data for one of the lines used is presented in Table 2.

**Table 2.** Comparison of the steps of the algorithms used

Stop	Coordinate	Stop	Coordinate
<b>1</b>	39.902638 41.274524	<b>10</b>	39.911979 41.266006
<b>2</b>	39.89879 41.270129	<b>11</b>	39.908586 41.265283
<b>3</b>	39.90003 41.267221	<b>12</b>	39.901406 41.266290
<b>4</b>	39.90942 41.265402	<b>13</b>	39.892800 41.248494
<b>5</b>	39.91174 41.265663	<b>14</b>	39.898357 41.262598
<b>6</b>	39.907539 41.278788	<b>15</b>	39.899936 41.266736
<b>7</b>	39.906147 41.286567	<b>16</b>	39.8981894 1.270431
<b>8</b>	39.905089 41.290898	<b>17</b>	39.897225 41.272914
<b>9</b>	39.911879 41.272602	<b>18</b>	39.898594 41.275351

As can be seen from Table 2, the D1 route consists of 18 stops in total. The GPS coordinates of these stops are given in Table 2. Within the scope of the application, a total of six different lines are used, and the properties for each of these lines are given in Table 3.

**Table 3.** The route information used within the scope of the application

Route No	D1	G1	G10	G3	G5	G7
<b>Stop Number</b>	18	42	60	45	57	28

The distances between the stops are different from each other. This is due to the problem of being asymmetrical. In other words, in this problem, the transportation distance from point "A" to point "B" and the distance from point "B" to point "A" may not be the same. For this reason, the distance matrix files previously obtained for each route are used for calculating the compliance function. As a result of touring the line routes made within the scope of this study, all of the bus stop coordinates for the six lines selected were obtained in a way that is closest to reality by considering the possible error margins (one to three meters) that may occur in the GPS fixing device. The real data that was obtained has been tested with the optimization methods and new routes that have shorter lengths than the existing routes have been identified. The conformity value changes obtained from the algorithms

for the selected routes are presented in Figure 11.

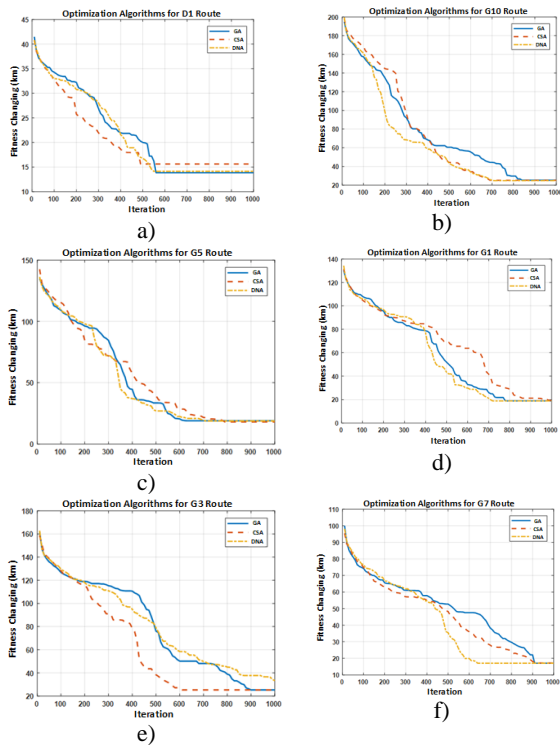


Figure 11. The changes of the suitability of algorithms for routes

In Figure 11, the compatibility changes of the optimization algorithms for each line are presented. As can be seen from the figure, each algorithm behaves differently for each of the lines. The final results obtained from these algorithms are presented in detail in Table 4.

As shown in Table 4, there has been an improvement on almost all of the lines. The genetic algorithm provides the optimum improvement for three lines, the clonal selection algorithm does so for two lines, and the DNA Computing Algorithm does so for three lines. The new optimized routes obtained from the algorithms, the daily voyages carried out on these lines, and the daily gains based on these voyages are given in detail in Table 5.

In addition, the new optimized routes that were obtained from the algorithms are presented in Table 6.

As can be seen in Table 6, changes have been experienced in all of the routes. In the final stage of the application, the new routes obtained, along with the existing routes, are drawn on Google Maps, and the results for the application is visualized. An example image for route D1 is given in Figure 12.

Table 4. The change of distance based on the algorithms (km)

	Lines					
	D1	G7	G1	G3	G5	G10
Current Distance (Km)	16.2710	17.1850	18.8170	25.8130	20.1150	26.7470
Genetic Algorithm	<b>13.8350</b>	<b>17.0340</b>	18.8170	<b>25.3020</b>	18.7720	25.3110
Clonal Selection Algorithm	15.6110	17.1850	18.8170	<b>25.3020</b>	<b>18.0490</b>	25.2350
DNA Computing Algorithm	14.1270	<b>17.0340</b>	18.8170	<b>25.3020</b>	18.5810	24.8480

Table 5. The difference states varying by distance (km)

Routes	Previous Distance (Km)	Next Distance (Km)	Percent (%)	Number of Voyages (Daily)	Daily Acquisition (Km)
D1	16,2710	13,8350	14,9	16	38,976
G1	18,8170	18,8170	0,0	16	0
G10	26,7470	24,8480	7,09	14	21,168
G3	25,8130	25,3020	1,97	15	7,665
G5	20,1150	18,0490	10,27	15	30,990
G7	17,1850	17,0340	0,87	18	2,718
				<b>Total</b>	<b>101,517</b>
				<b>Average</b>	<b>16,920</b>

Table 6: The exit routes obtained as a result of the application

D1	1	7	8	6	9	10	11	4	5	2	3	12	13	14	15
	16	17	18												
G1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	31	32	33	34	35	36	37	38	39	40	41	42			
G10	1	2	3	5	56	6	7	8	55	9	54	10	13	14	49
	16	47	18	19	20	21	22	23	24	25	26	27	28	29	30
	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
	17	46	48	15	50	51	52	12	53	11	4	57	58	59	60



G3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	16	17	35	19	20	21	22	23	24	25	26	27	28	29	30
G5	31	32	33	34	18	36	37	38	39	40	41	42	43	44	45
	1	4	54	5	53	6	52	7	47	11	46	12	13	14	15
	16	17	18	19	20	21	22	35	23	25	26	27	28	29	30
G7	31	32	33	34	24	36	37	38	39	40	41	42	43	44	45
	10	48	9	49	8	50	51	55	3	2	56	57			
	1	2	3	4	5	6	7	8	9	10	11	12	13	15	16
	14	17	18	19	20	21	22	23	24	25	26	27	28		



Figure 12. An example image for route D1

## 5. Results and Discussion

Today, public transportation systems are actively used in many countries around the world. In particular, railway, seaway, and road transportation are the services preferred by people when using public transport systems. One of the main reasons why these services are especially preferred by people is their desire to complete their daily work faster, easier, and in a shorter amount of time. As can be expected, people need to arrange these routines according to specific days and times in order to, do their daily routines as quickly as possible. When considering all of these situations, the concept of optimization emerges.

The optimization process is basically the selecting the best or near best solution from a vast solution space. The optimization process, which is used in many different fields such as financial planning, computer science, and industrial computing, is encountered even if people regulate their daily work. In this study, an application that can determine the optimal route for municipal bus lines, which is a service offered by the municipalities and is frequently encountered in daily life, is introduced. For this purpose, a problem based on real data was solved by using data provided by the EMM Public Transportation Branch Directorate. The data provided was used in applications made during this study using Google API and Google Maps. These routes, subjected to testing, have been examined, optimized, and visualized on

Google Maps.

During this research, various restrictions were identified and some findings were identified. In this context, the research was carried out only for six of the line routes. Applying this method for all of lines the would provide significant results in terms of ascertaining the reliability of the applied methods. In addition, looking at the real-world experience of the test results obtained from the research on some pilot routes would confirm the applicability and performance of the results.

In addition, it was determined that some of the stations were located very close to each other and some of them were located very far from each other when the stops were physically visited in order to confirm the data provided by the Metropolitan Municipality. In addition, when considering the climatic structure of Erzurum, the distance between the stops is relatively far from each other, which significantly reduces the ease of access to the buses. For this reason, the stop locations need to be revised. During this research, it was observed that the majority of the passengers using the buses travel by foot. With the implementation of the optimization process on these lines, the return of the earnings obtained on km basis as an increase in the number of trips will allow the passengers to travel more comfortably and more efficiently.

When considering the energy consumption problems today, buses need to achieve the minimum amount of

fuel consumption. Depending on the situation, both the stops' locations and the buses' routes should be reviewed and updated periodically. In addition, the problem of global warming directly affects many living spaces. The biggest reason for this situation is the consumption of fossil fuels. At this point, minimizing the emissions of harmful carbon gases from these buses will provide many societal and environmental benefits.

## ACKNOWLEDGEMENT

In this study, we thank the Erzurum Metropolitan Municipality Public Transportation Administration for the data it provided.

## REFERENCES

- Asadpour, A., Goemans, M. X., Mađry, A., Gharan, S. O., and Saberi, A. 2010. "An  $O(\log n / \log \log n)$ -Approximation Algorithm for the Asymmetric Traveling Salesman Problem," *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* (June), pp. 379–389. (<https://doi.org/10.1137/1.9781611973075.32>).
- Baygin, M., and Karakose, M. 2013. "Immunity-Based Optimal Estimation Approach for a New Real Time Group Elevator Dynamic Control Application for Energy and Time Saving," *The Scientific World Journal* (2013). (<https://doi.org/10.1155/2013/805343>).
- Bolat, B., and Cortés, P. 2011. "Genetic and Tabu Search Approaches for Optimizing the Hall Call - Car Allocation Problem in Elevator Group Systems," *Applied Soft Computing Journal* (11:2), pp. 1792–1800. (<https://doi.org/10.1016/j.asoc.2010.05.023>).
- Chen, P. 2013. "An Improved Genetic Algorithm for Solving the Traveling Salesman Problem," *Proceedings - International Conference on Natural Computation* (6), pp. 397–401. (<https://doi.org/10.1109/ICNC.2013.6818008>).
- Choong, S. S., Wong, L. P., and Lim, C. P. 2019. "An Artificial Bee Colony Algorithm with a Modified Choice Function for the Traveling Salesman Problem," *Swarm and Evolutionary Computation* (44:December 2017), pp. 622–635. (<https://doi.org/10.1016/j.swevo.2018.08.004>).
- Deng, W., Zhao, H., Zou, L., Li, G., Yang, X., and Wu, D. 2017. "A Novel Collaborative Optimization Algorithm in Solving Complex Optimization Problems," *Soft Computing* (21:15), Springer Berlin Heidelberg, pp. 4387–4398. (<https://doi.org/10.1007/s00500-016-2071-8>).
- Dodge, M., MirHassani, S. A., and Hooshmand, F. 2020. "Solving Two-Dimensional Cutting Stock Problem via a DNA Computing Algorithm," *Natural Computing* (3), Springer Netherlands. (<https://doi.org/10.1007/s11047-020-09786-3>).
- Groba, C., Sartal, A., and Vázquez, X. H. 2015. "Solving the Dynamic Traveling Salesman Problem Using a Genetic Algorithm with Trajectory Prediction: An Application to Fish Aggregating Devices," *Computers and Operations Research* (56), Elsevier, pp. 22–32. (<https://doi.org/10.1016/j.cor.2014.10.012>).
- Guney, K., Babayigit, B., and Akdagli, A. 2007. "Position Only Pattern Nulling of Linear Antenna Array by Using a Clonal Selection Algorithm (CLONALG)," *Electrical Engineering* (90:2), pp. 147–153. (<https://doi.org/10.1007/s00202-006-0056-9>).
- Guo, Z., Koehler, G. J., and Whinston, A. B. 2007. "A Market-Based Optimization Algorithm for Distributed Systems," *Management Science* (53:8), pp. 1345–1358. (<https://doi.org/10.1287/mnsc.1060.0690>).
- Hasan Söyler, T. K. 2007. *Karınca Kolonisi Algoritması İle Gezen Satıcı Probleminin Çözümüğ*.
- Hiassat, A., Diabat, A., and Rahwan, I. 2017. "A Genetic Algorithm Approach for Location-Inventory-Routing Problem with Perishable Products," *Journal of Manufacturing Systems* (42), The Society of Manufacturing Engineers, pp. 93–103. (<https://doi.org/10.1016/j.jmsy.2016.10.004>).
- Ibrahim, G. J., Rashid, T. A., and Sadiq, A. T. 2018. "Evolutionary DNA Computing Algorithm for Job Scheduling Problem," *IETE Journal of Research* (64:4), pp. 514–527. (<https://doi.org/10.1080/03772063.2017.1362964>).
- Kovács, L., Iantovics, L. B., and Iakovidis, D. K. 2018. "IntraClusTSP-An Incremental Intra-Cluster Refinement Heuristic Algorithm for Symmetric Travelling Salesman Problem," *Symmetry* (10:12). (<https://doi.org/10.3390/sym10120663>).
- Mahi, M., Baykan, Ö. K., and Kodaz, H. 2015. "A New Hybrid Method Based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt Algorithms for Traveling Salesman Problem," *Applied Soft Computing Journal* (30), Elsevier B.V., pp. 484–490. (<https://doi.org/10.1016/j.asoc.2015.01.068>).
- Malhotra, R., Singh, N., and Singh, Y. 2011. "Genetic Algorithms: Concepts, Design for Optimization of Process Controllers," *Computer and Information Science* (4:2), pp. 39–54. (<https://doi.org/10.5539/cis.v4n2p39>).
- Mohammed, M. A., Abd Ghani, M. K., Hamed, R. I., Mostafa, S. A., Ahmad, M. S., and Ibrahim, D. A. 2017. "Solving Vehicle Routing Problem by

- Using Improved Genetic Algorithm for Optimal Solution,” *Journal of Computational Science* (21), Elsevier B.V., pp. 255–262. (<https://doi.org/10.1016/j.jocs.2017.04.003>).
- Muthreja, I., and Kaur, D. 2018. “A Comparative Analysis of Immune System Inspired Algorithms for Traveling Salesman Problem,” *2018 World Congress in Computer Science, Computer Engineering and Applied Computing, CSCE 2018 - Proceedings of the 2018 International Conference on Artificial Intelligence, ICAI 2018*, pp. 164–170.
- NARALAN, A., KALELİ, S. S., and BAYĞIN, M. 2017. “Shortest Path Detection Using Clonal Selection Algorithm for Erzurum Metropolitan Municipality,” *Mugla Journal of Science and Technology* (3:2), pp. 138–142. (<https://doi.org/10.22531/muglajsci.357621>).
- Nguyen, H. D., Yoshihara, I., Yamamori, K., and Yasunaga, M. 2002. “Greedy Genetic Algorithms for Symmetric and Asymmetric TSPs,” *IPSI Trans. Mathematical Modeling and Its Applications* (43:10), pp. 165–175.
- Priyo Anggodo, Y., Kartika Ariyani, A., Khaerul Ardi, M., and Firdaus Mahmudy, W. 2016. “Optimization of Multi-Trip Vehicle Routing Problem With Time Windows Using Genetic Algorithm,” *Journal of Environmental Engineering and Sustainable Technology* (3:2), pp. 92–97. (<https://doi.org/10.21776/ub.jeest.2017.003.02.4>).
- Saji, Y., and Riffi, M. E. 2016. “A Novel Discrete Bat Algorithm for Solving the Travelling Salesman Problem,” *Neural Computing and Applications* (27:7), Springer London, pp. 1853–1866. (<https://doi.org/10.1007/s00521-015-1978-9>).
- Shrikrishna, K., B, N. V. N. K., and Shyamasundar, R. K. 2018. *Security Analysis of EMV Protocol*, (2:December), pp. 69–85. (<https://doi.org/10.1007/978-3-319-72344-0>).
- Suman, S. K. 2015. “Genetic Algorithms: Basic Concepts and Real World Applications,” *International Journal of Electrical, Electronics and Computer Systems* (November). ([https://scholar.google.co.in/citations?view\\_op=view\\_citation&continue=/scholar%3Fhl%3Den%26as\\_sdt%3D0,5%26scilib%3D1&citilm=1&citation\\_for\\_view=dj9eYFMAAAAJ:UebtZRa9Y70C&hl=en&oi=p](https://scholar.google.co.in/citations?view_op=view_citation&continue=/scholar%3Fhl%3Den%26as_sdt%3D0,5%26scilib%3D1&citilm=1&citation_for_view=dj9eYFMAAAAJ:UebtZRa9Y70C&hl=en&oi=p)).
- Sundararaghavan, P. S., Kunnathur, A., and Fang, X. 2010. “Sequencing Questions to Ferret out Terrorists: Models and Heuristics,” *Omega* (38:1–2), Elsevier, pp. 12–19. (<https://doi.org/10.1016/j.omega.2009.01.002>).
- Wang, J., Ersoy, O. K., He, M., and Wang, F. 2016. “Multi-Offspring Genetic Algorithm and Its Application to the Traveling Salesman Problem,” *Applied Soft Computing Journal* (43), Elsevier B.V., pp. 415–423. (<https://doi.org/10.1016/j.asoc.2016.02.021>).
- Xu, Z., Wang, Y., Li, S., Liu, Y., Todo, Y., and Gao, S. 2016. “Immune Algorithm Combined with Estimation of Distribution for Traveling Salesman Problem,” *IEEJ Transactions on Electrical and Electronic Engineering* (11), pp. S142–S154. (<https://doi.org/10.1002/tee.22247>).
- Zhang, Y. 2018. “The Image Encryption Algorithm Based on Chaos and DNA Computing,” *Multimedia Tools and Applications* (77:16), Multimedia Tools and Applications, pp. 21589–21615. (<https://doi.org/10.1007/s11042-017-5585-x>).