Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-30 (2018) 620–625

GA Based Traveling Salesman Problem Solution and its Application to Transport Routes Optimization

U. Hacizade, I. Kaya

Department of Computer Engineering, Halic University, Sütlüce District., Imrahor Str., No: 82, Beyoğlu-Istanbul, TURKEY (e-mail: ulviyehacizade@halic.edu.tr; kayaismail@hotmail.com)

Abstract: In this study, the well-known traveling salesman problem (TSP) with relational optimization problem is defined as a method of solution used in genetic algorithms (GA). TSP is a problem about finding the shortest route starting from one city and turning back to the same city while considering only one pass through each city (points, nodes or components) where the distances between each city are known. In this study, the GA based method is developed for solving the travelling salesman problem, and applied to the problem of finding the optimal route for Istanbul Electricity Tram and Tunnel Operations (IETT) audit team which provides public transport services in Istanbul

Keywords: Optimization, Genetic Algorithms, Traveling Salesman Problem, Optimal Route Determination, Route Planning, Public Transport Services.

1. INTRODUCTION

Recently the determination of optimal route is developing and gains importance while solving practical problems. Determination of an appropriate route, reaching the destination in time and energy saving have significance on real world applications.

Route determination is being studied for years by scientists and engineers. There have been a lot of research about this subject and it still keeps its popularity. The research done so far is mostly about Travelling Salesman Problem (TSP). The TSP is arguably the most prominent problem in combinatorial optimization, operations, research and theoretical computer science which receives much attention because of its applications in industrial nowadays, transport and service problems. In this problem, a salesman starts to move from an arbitrary place called depot and returns there after visiting *n* nodes. The objective is to minimize the total distance travelled by the salesman.

TSP is a problem about finding the shortest route starting from one city and turning back to the same city while considering only one pass through each city (points, nodes or components) where the distances between each city are known.

TSP and its derivate problems which can be modeled in similar manner are attracting the attention of numerous researchers who are focused on mathematics, artificial intelligence and physics. Probably travelling salesman problem is the most well-known optimization problem. The models which are used for these types of problems are known to be very hard or almost impossible to solve by hand next to the fact that their computer based solutions requires long time

In Literature, there are optimization approaches like local search algorithms, genetic algorithms (Goldberg,1989), tabu search (Fiechter, 1994), ant colony optimization (Angus and Hendtlass, 2005) and artificial neural networks for solving Travelling Salesman Problem (Johnson and McGeoch, 1997).

In order to solve the traveling salesman problem through GAs, Takahashi (2005) proposed a method of changing crossover operators (CXO), which can flexibly substitute the current crossover operator for another suitable crossover operator at any time. It is shown that changing crossover operators at arbitrary time according to city data structure is available to improve the performance of GAs.

Raichaudhuri and Jain (2010) address the problem of selecting route to a given destination on an actual map under a static environment by using a GA. It is shown that, the best route selection problem in network analysis can be solved with GA through efficient encoding, selection of fitness function and various genetic operations. Crossover is identified as the most significant operation to the final solution.

Helshani (2015) used a GA to determine the optimum route on Google map and solve the Travelling Salesman problem.

Lin et al. (2016) present an improved hybrid genetic algorithm (HGA) to solve the two-dimensional Euclidean traveling salesman problem, in which the crossover operator is enhanced with a local search. The proposed HGA provides much better reasonable structure design to compensate for the balance between global and local search contributes to efficiently improve the evolutionary process in terms of the convergence rate and the solution quality.

Hussain et al. (2017) propose a new crossover operator for traveling salesman problem to minimize the total distance.

10.1016/j.ifacol.2018.11.224

This approach was linked with path representation, which is the most natural way to represent a legal tour.

The hybrid algorithm for solving the travelling salesman problem is proposed in (Eswarawaka et al., 2015). This algorithm describes the combination of GA and the local search technique. To improve the performance of finding an optimal solution from huge search space, the authors incorporated the use of tournament and rank as selection operators, and inver-over operator mechanism for crossover and mutation.

In study (Bortas et al., 2018) the factors that affect the distribution of the goods flows are analysed and for the selection of the optimal transport route an optimization model is built by using the fuzzy logic, according to the criteria of minimizing the costs and negative impact on the environment.

Analysing above mentioned works, it can be said that the complexity of the exact algorithm for solving the Travelling Salesman Problem increases with the number of places to visit. For this reason GA is offered as the optimal solution of travelling salesman problem. In this study, the GA based method is developed for solving Travelling Salesman Problem, and applied to the problem of finding the optimal route for Istanbul Electricity Tram and Tunnel Operations (IETT) audit team which provides public transport services in Istanbul.

2. THE OPTIMAL ROUTE FINDING FOR IETT

The GA based method which is developed for solving Travelling Salesman Problem is applied to the problem of finding the optimal route for IETT audit team. Considering the lack of some data related with the problem some assumptions made to obtain these data. Specifically the distance between each bus stop does not exist, however the coordinates (longitude, latitude) are present. With using these coordinates, the distance between bus stops are calculated. Google Maps API is used for this calculation.

In this study, the optimization steps of the GA are applied on the Travelling Salesman Problem. The application is developed in Microsoft Visual Studio 2010, with using Asp.Net framework and C# programming language, JavaScript and Google Maps API.

All bus stops which belong to IETT are shown in the "Show All Stops" page on the map. Screenshots are given in Fig. 1. Application shows the bus stop's name when the stop is selected on the map. "Choose Stop From Map" screen shows the stops with their numbers next to their names. "Choose Stop From Map" screen is shown in Fig. 2 with all the stops selected.

Application consists of 4 pages. Firstly, the area has to be chosen in the application's "Choose Stop" page. Afterwards the district should be selected inside the area. The bus stops in the district are selected multiple. When the related values are submitted, the button "Calculate" should be clicked. The following page has "Stop" button which stops the iteration at that moment.

When the iteration steps chosen as 25113 from the choose stops page, for the 78 bus stops case, the optimal route and the total distance between the stops are shown in Figure 3.

As seen from Fig.3, the optimal route when 78 bus stops chosen, is comprised of the following sequence of stops: 6-57-16-37-73-70-35-56-55-38-19-32-75-33-50-60-61-18-5-8-62-9-63-40-14-67-49-48-11-23-66-24-65-43-15-10-21-34-47-69-68-59-58-31-72-36-30-71-53-77-52-22-12-44-13-26-45-46-27-74-3-51-7-20-76-54-78-17-42-2-39-25-64-41-4-29-28.

In addition, the optimal routes with 38541;51729 and 871972 iteration steps when 78 bus stop chosen are investigated in this study. The best route distance in different iteration steps are given in Table 1.

Table 1. Iteration steps vs best route distance

Iteration Steps	Best Route Distance (km)
25113	78.78
38541	70.48
51729	65.35
871972	50.79

As seen in Table 1, different iteration steps give different distances. Generally, when iteration steps are increasing, the distance is decreasing. It is observed that the distance is decreasing when steps increase which means better route found.

3. SAMPLE TRAVELLING SALESMAN PROBLEM

To investigate the application of genetic algorithms to the Travelling Salesman Problem, at first only 4 stops selected. The initial goal is to find the shortest route while visiting each stop only once. It is assumed that the inspector has no obligation to return where he/she started. The stops are connected to each other and the distance between them are given. With these terms, the minimum total distance will be calculated. The stops and the distance between these stops are given in Table 1. Distances between the stops are given in Table 2 as an n x n matrix (n is the stop number)

Table 2. Distances between the stops

Stops	0	1	2	3
0	0	3	4	7
1	3	0	5	4
2	4	5	0	6
3	7	4	6	0

1st Step

At t=0 in G(t) generation, a primitive population with N chromosomes are generated randomly. Each chromosome is a possible solution of the problem. Lets take N=6 for this example. Population which contains 6 chromosomes is generated randomly. The randomly generated 6 chromosomes are shown below:

v_1 =[1 2 3 0]

v_2 =[1 3 0 2]

v_3 =[2 1 3 0]

v_4 =[3 2 0 1]

v_5 =[0 1 3 2]

v_6 =[2 0 1 3] .

For all v_i, i ∈ [1,6], chromosomes the conformity value f(v_i) is calculated with the formula below:

$$f(v_i) = \frac{1}{g(v_i)} . \quad (1)$$

Here $g(v_i)$ is the total distance traveled by the inspector. It is very clear that the smaller value $g(v_i)$ gets the chromsome's conformity value higher proportionally.

for v_1 = [1 2 3 0]

g(v_1) =5 km +6 km +7 km = 18 km;

for v_2 = [1 3 0 2]

g(v_2) =4 km +7 km +4 km = 15 km;

for v_3 = [2 1 3 0]

g(v_3) =5 km +4 km +7 km = 16 km;

for v_4 = [3 2 0 1]

g(v_4) = 6 km +4 km +3 km = 13 km;

for v_5 = [0 1 3 2]

g(v_5) = 3 km +4 km +6 km = 13 km;

for v_6 = [2 0 1 3]

g(v_6) = 4 km +3 km +4 km = 14 km;

$$f(v_i) = \frac{1}{g(v_i)} \rightarrow f(v_1) = \frac{1}{18} = 0.06$$

$$f(v_2) = \frac{1}{15} = 0.07$$

$$f(v_3) = \frac{1}{16} = 0.06$$

$$f(v_4) = \frac{1}{13} = 0.08$$

$$f(v_5) = \frac{1}{13} = 0.08$$

$$f(v_6) = \frac{1}{14} = 0.07$$

Since $f_{\max} = \frac{1}{13}$ the first genereation's best route chromosomes are v_4 and v_5 .

2nd Step

The sum of the conformity values is calculated with the formula below:

$$F = \sum_{i=1}^{6} f(v_i) = 0.06+0.07+0.06+0.08+0.08+0.07 = 0.42$$

Afterwards the probability of a chromosome to be selected to the G(t) generation (t=t+1) calculated and roulette wheel designed.

$$p_1 = \frac{f_1}{F} = \frac{0.06}{0.42} = 0.14 \qquad q_1 = 0.14$$

$$p_2 = \frac{f_2}{F} = \frac{0.07}{0.42} = 0.17 \qquad q_2 = 0.31$$

$$p_3 = \frac{f_3}{F} = \frac{0.06}{0.42} = 0.14 \qquad q_3 = 0.45$$

$$p_4 = \frac{f_4}{F} = \frac{0.08}{0.42} = 0.19 \qquad q_4 = 0.64$$

$$p_5 = \frac{f_5}{F} = \frac{0.08}{0.42} = 0.19 \qquad q_5 = 0.83$$

$$p_6 = \frac{f_6}{F} = \frac{0.07}{0.42} = 0.17 \qquad q_6 = 1.00.$$

As we mention before, in natural selection only the appropriate species can survive, reproduce and transfer its genes to the next generation. Genetic Algorithms uses an analogous approach. Unlike the nature, the size of the population remains the same in the next generation. For instance while v_1 and v_2 have low selection probability, v_4 and v_5 have enough chance to be selected.

In the example, a population with 6 chromosomes initially generated. In this way, to keep next generation's population the same with the previous one 6 numbers will be generated (this is similar to have 6 different spins in a roulette wheel). In [0-1] interval 6 numbers are generated randomly.

r_1 = 0.12 r_2 = 0.30 r_3 = 0.82 r_4 = 0.60 r_5 = 0.03 r_6 = 0.09.

Roullete wheel spins 6 times. Every i step the comparison below executed. If $r_i < q_1$ the i th chromosome replaces with the first chromosome from the previous generation. Otherwise, i th chromosome replaces with the j th chromosome which justifies the $q_{j-1} < r_i \le q_j$ condition. The first population of the second generation which created at the result of the selection process is given below:

$v_1 = v_{1eski}$ = [1 2 3 0]

$v_2 = v_{2eski}$ = [1 3 0 2]

$v_3 = v_{5eski} = [\ 0\ 1\ 3\ 2\ ]$

$v_4 = v_{4eski} = [\ 3\ 2\ 0\ 1\ ]$

$v_5 = v_{1eski} = [\ 1\ 3\ 0\ 2\ ]$

$v_6 = v_{1eski} = [\ 1\ 3\ 0\ 2\ ]$.

The chromosomes grew out of the selection be subjected to crossover with a specific probability. New chromosomes grow out of the crossover swaps with the old chromosomes.

3rd Step

Crossover aim is to obtain chromosomes with the better conformity values. Crossover operators in GA implement a mechanism that mixes the genetic material of the parents. There is no obligation to apply crossover to all members of the population. To pass some of the members of the population to the next population without crossover, the crossover ratio is designated. For each individual in the population a random real number between 0 and 1 is generated. If the generated number of the individual is less than the crossover value then crossover is applied to this particular individual. If the number is greater than the crossover ratio no crossover will be applied to the individual.
Crossover will be applied to all chromosome pairs selected as parent chromosomes. After a randomly chosen crossover point the parts of chromosomes replace with each other. As a result we obtain two offsprings. For example, when the crossover possibility p_c=0.5, the expected amount of the chromosomes which are going to be crossed over will be 2 which is calculated as p_c x pop_size = 0.5 x 6 = 3, 3 – 1 = 2. The selection of the chromosomes is done by using the steps below. Firstly 6 random numbers are generated: r_1 = 0.12 r_2 = 0.72 r_3 = 0.82 r_4 = 0.60 r_5 = 0.53 r_6 = 0.09. Afterwards all the numbers are compared with the p_c. If the random number is less than p_c, the chromosome which is represented with this random number is selected . To illustrate , v_1 and v_6 will be crossed over due to the fact that $r_1 < p_c$ ve $r_6 < p_c$. After randomly selected 2nd position gene exchange takes place between chromosomes.

$v_1 = [\ 1\ 2\ 3\ 0\ ] \rightarrow [\ 1\ 3\ 0\ 2\ ]$

$v_6 = [\ 1\ 3\ 0\ 2\ ] \rightarrow [\ 1\ 2\ 3\ 0\ ]$.

However v_1 and v_4 crossover results are given below:

$v_1 = [\ 1\ 2\ 3\ 0\ ] \rightarrow [\ 1\ 2\ 0\ 1\ ]$

$v_4 = [\ 3\ 2\ 0\ 1\ ] \rightarrow [\ 3\ 2\ 3\ 0\ ]$.

As experienced above, the new chromosomes are not valuable for the solution of Traveling Salesman Problem. Because the inspector should travel around all the stops and he/she should pass from each stop only once. However, observing the newly created chromosomes shows us the fact that the inspector could not pass from some of the stops or pass twice from the same stop. In v_1 we observe that the 1st stop visited twice and in v_4 it is clear that the 3rd stop never visited. In this negative type of case, a second operation is needed namely the inappropriate chromosomes that do not ensure the limitations should be standardized. For standardization a simple rule can be used: "Replace the first recurring stop with the lowest numbered unvisited stop in a chromosome". This standardization method is the first thing that comes to mind. Another rule could be used for standardization. In Genetic Algorithms almost everything is unplanned and unscheduled like the nature. In the example the problematic chromosomes are standardized as below:

$v_1 = [\ 1\ 2\ 3\ 0\ ] \rightarrow [\ 1\ 2\ 0\ 1\ ] \rightarrow [\ 1\ 2\ 0\ 3\ ]$

$v_4 = [\ 3\ 2\ 0\ 1\ ] \rightarrow [\ 3\ 2\ 3\ 0\ ] \rightarrow [\ 3\ 2\ 1\ 0\ ]$.

The new population formed as a result of crossover is given below:

$v_1 = [\ 1\ 3\ 0\ 2\ ]$

$v_2 = [\ 1\ 3\ 0\ 2\ ]$

$v_3 = [\ 0\ 1\ 3\ 2\ ]$

$v_4 = [\ 3\ 2\ 0\ 1\ ]$

$v_5 = [\ 1\ 3\ 0\ 2\ ]$

$v_6 = [\ 1\ 2\ 3\ 0\ ]$.

To ensure the genetic variability within the population formed as a result of the crossover mutation is applied with a known probability value.

4th Step

Mutation operation takes place after crossover operation. Mutation operation is used to make random changes from 0 to 1 or from 1 to 0 in a chromosome's genes. Mutation ratio is used to determine the selection ratio of the genes in binary coded mutation. To determine which genes are going to be mutate random r numbers generated with the same amount of individual's genes. In the example, due to using permutation encoding, mutation is executed by swapping two genes location in a randomly selected chromosome. For instance, when mutation probability is p_m=0.3, p_m x pop_size = 0.3 x 6 = 1.8 ≈ 2 genes will be subjected to mutation. 6x4 = 24 random numbers generated. The gene which justifies $r_i < p_m$ condition and the one just after that are selected for the mutation. 24 random numbers are generated within the interval [0-1].

r_1 = 0.42 r_2 = 0.82 … r_{22} = 0.23, r_{23} = 0.32 r_{24} = 0.69.

Since $r_{22} < p_m$ 6th chromosome's 2nd and 3rd genes are swapped:

$v_6 = [\ 1\ \mathbf{2}\ \mathbf{3}\ 0\ ] \rightarrow [\ 1\ 3\ 2\ 0\ ]$.

The new chromosomes formed after mutation replaces with the old chromosomes. As a result of the mutation the new population is given below:

$$v_1 = [\ 1\ 3\ 0\ 2\ ]$$

$$v_2 = [\ 1\ 3\ 0\ 2\ ]$$

$$v_3 = [\ 0\ 1\ 3\ 2\ ]$$

$$v_4 = [\ 3\ 2\ 0\ 1\ ]$$

$$v_5 = [\ 1\ 3\ 0\ 2\ ]$$

$$v_6 = [\ 1\ 3\ 2\ 0\ ].$$

5th Step
All the conformity values are calculated for each chromosome in the population:

$$g(v_1) = 15 \text{ km}\, f(v_1) = \frac{1}{g(v_1)} = \frac{1}{15} = 0.06$$

$$g(v_2) = 15 \text{ km}\, f(v_2) = \frac{1}{g(v_2)} = \frac{1}{15} = 0.06$$

$$g(v_3) = 16 \text{ km}\, f(v_3) = \frac{1}{g(v_3)} = \frac{1}{16} = 0.06$$

$$g(v_4) = 13 \text{ km}\, f(v_4) = \frac{1}{g(v_4)} = \frac{1}{13} = 0.07$$

$$g(v_5) = 15 \text{ km}\, f(v_5) = \frac{1}{g(v_5)} = \frac{1}{15} = 0.06$$

$$g(v_6) = 14 \text{ km}\, f(v_6) = \frac{1}{g(v_6)} = \frac{1}{14} = 0.07.$$

Since $f_{2.\max} = \frac{1}{13}$, v_4 is the chromosome which contains the best routes in second generation.

6th Step
If the generation number which is defined at the beginning is not reached then you should return back to the second step.

7th Step
All $f_{t.\max}$ values obtained as a result of each t generation compared with each other. As a result of comparison, the chromosome which provides the highest conformity value points out the best solution (best route). In the example, 4th chromosome represents the best route: 3-2-0-1 at the end of the second generation.

4. CONCLUSION

The main aim of this study is to develop an effective method for the field audit team which inspects certain points (bus stops) to speed up the process and reduce the time needed. Furthermore, it is also aimed to develop effective software which is based on Genetic Algorithms approach. In this context, the most well-known traveling salesman problem with relational optimization problem is defined as a method of solution used in genetic algorithms. Traveling Salesman Problem is a problem about finding the shortest route starting from one city and turning back to the same city while considering only one pass through each city (points, nodes or components) where the distances between each city are known. There are many methods for solving the Travelling Salesman Problem. TSP algorithms to solve problems of medium size needed for very long periods are not very satisfactory. Instead, heuristic methods results in the shortest period of time are near optimum.

In this study, the Genetic Algorithm based method is developed for solving Travelling Salesman Problem applied to the problem of finding the optimal route for Istanbul Electricity Tram and Tunnel Operations (IETT) audit team which provides public transport services in Istanbul.

REFERENCES

Angus, D. and Hendtlass, T. (2005). Dynamic ant colony optimisation. *Applied Intelligence* , 23(1), 33–38.

Bortas, I., Brnjac, N. and Dundović, Ć. (2018). Transport routes optımızatıon model through applıcatıon of fuzzy logic. *Promet-Traffic & Transportation,* 30(1), 121-129.

Eswarawaka, R., Mahammad, S.K.N., Reddy, B.E. (2015). Genetic annealing with efficient strategies to improve the performance for the np-hard and routing problems. *Journal of Experimental & Theoretical Artificial Intelligence*, 27(6), 779–788.

Fiechter, C.N. (1994). A parallel tabu search algorithm for large traveling salesman problems. *Discrete Applied Mathematics,* 51(3), 243-267.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addisson-Wesley Publishing Company Inc., Reading .

Johnson, D.S. and McGeoch, L.A. (1997).The Traveling Salesman Problem: A Case Study in Local Optimization. In E.H.L. Aarts and J.K. Lenstra, ed., *Local Search in Combinatorial Optimization,* 215-310. Wiley, New York.

Helshani, L. (2015). An android application for google map navigation system, solving the travelling salesman problem. Optimization throught genetic algorithm. Proc. of FIKUSZ '15 Symposium for Young Researchers.

Hussain, A., Muhammad, Y.S., Sajid, M.N., Hussain, I., Mohamd Shoukry, A.M., and Gani, S. (2017). Genetic algorithm for traveling salesman problem with modified cycle crossover operator. *Hindawi Computational Intelligence and Neuroscience,* Article ID 7430125.

Lin, B., Sun, X. and Salous, S. (2016). Solving travelling salesman problem with an improved hybrid genetic algorithm. *Journal of Computer and Communications,* 4, 98-106.

Raichaudhuri, A., Jain, A. (2010). Genetic algorithm based logistics route planning. *International Journal of Innovation, Management and Technology,* 1(2), 205-208.

Takahashi, R. (2005). Solving the traveling salesman problem through genetic algorithms with changing crossover operators. The Fourth International Conference on Machine Learning and Applications.

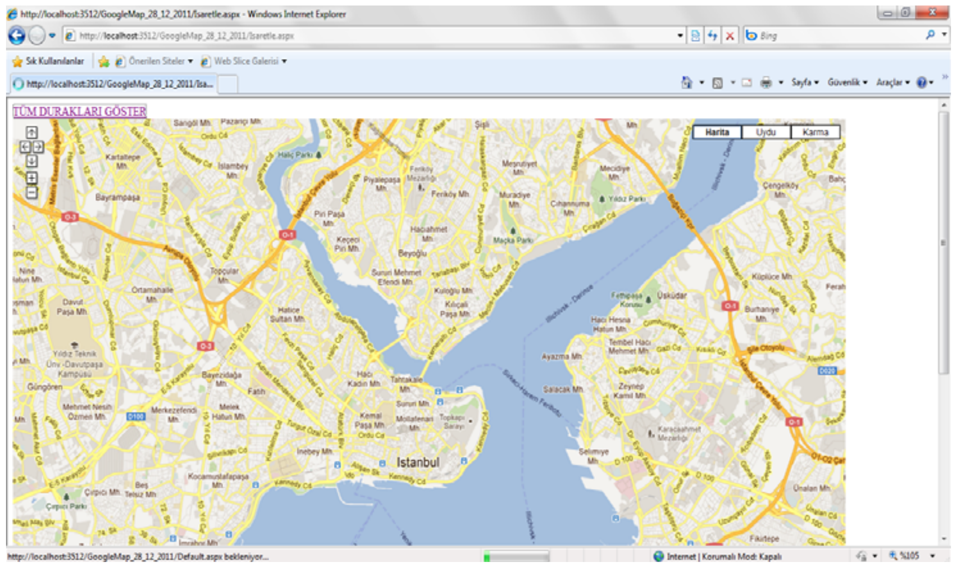

Fig. 1. All bus stops which belong to IETT on the map.

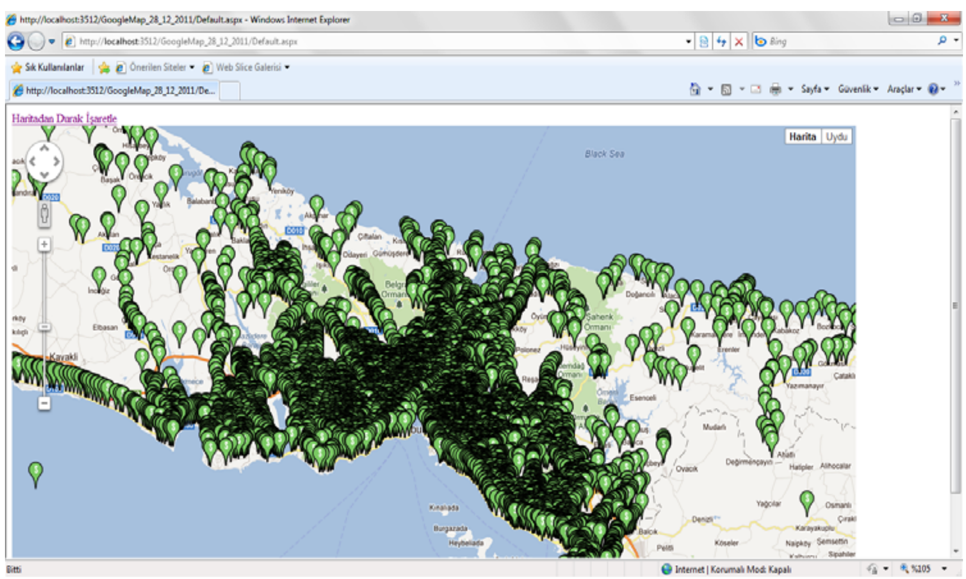

Fig. 2. All bus stops on the map.

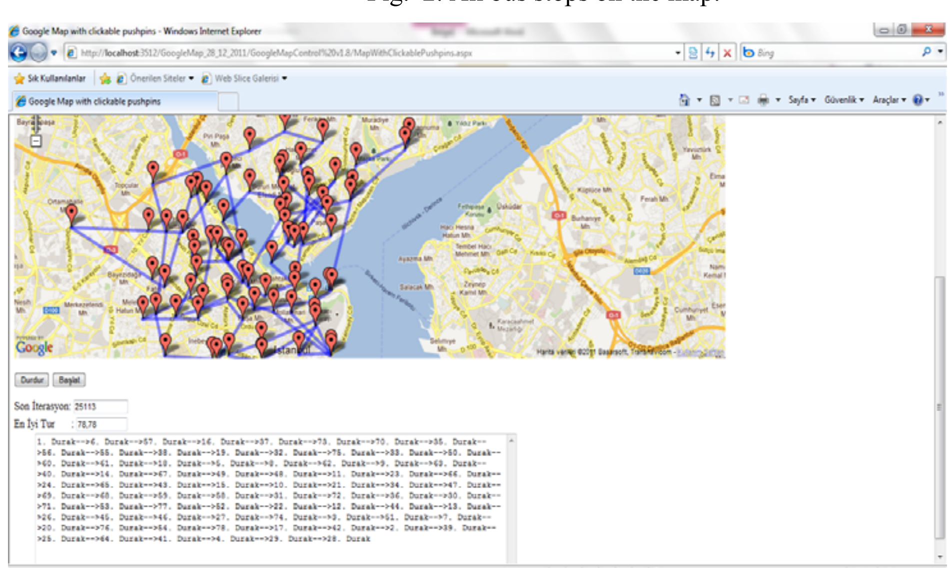

Fig. 3. The route with 25113 iteration steps when 78 bus stops chosen