

# Design of a Route Guidance System with Shortest Driving Time Based on Genetic Algorithm

UMIT ATILA<sup>1</sup>, ISMAIL RAKIP KARAS<sup>2</sup>, CEVDET GOLOGLU<sup>3</sup>, BEYZA YAMAN<sup>2</sup>,  
ILHAMI MUHARREM ORAK<sup>2</sup>

<sup>1</sup> Directorate of Computer Center  
Gazi University,  
06500 Besevler, Ankara  
TURKEY

<sup>2</sup> Department of Computer Engineering  
Karabuk University  
78050 Karabuk  
TURKEY

<sup>3</sup> Department of Mechanical Engineering  
Karabuk University  
78050 Karabuk  
TURKEY

umitatila@gmail.com, {ismail.karas, cgologlu, beyzayaman, imorak}@karabuk.edu.tr  
<http://muh.karabuk.edu.tr>

*Abstract:* - Nowadays, with the advancement of the technology on mobile devices, route guidance systems that assist drivers on the traffic have become widespread in daily life. For an accurate routing, a route guidance system should consider the effectual factors of traffic flow such as density and allowable velocity limits of the roads. With the increase of effectual factors and amount of nodes in road network, the computational cost increases. It is not proper to find exact optimal solution in real time for the road networks with excessive number of nodes using some well known deterministic methods such as Dijkstra's algorithm on navigation systems using mobile devices with limited processing speed and memory capacity.

This paper proposes a route guidance system and a Genetic Algorithm (GA) approach applied on this routing system to find the shortest driving time. Excluding classical methods, a gene search method of chromosomes named "first-matched-genes" on crossover operation had been introduced. The efficiency of the genetic algorithm was tested by applying on the networks with different sizes and a mobile application on the traffic network of Ankara was presented.

*Key-Words:* - Genetic algorithm; navigation; route guidance; shortest path; shortest driving time; optimization

## 1 Introduction

The logic behind navigation systems provides the users with the shortest path between *departure* and destination points. The downside of the current navigation systems is ignoring important decision variables including the traffic density and allowable velocity limits of the roads.

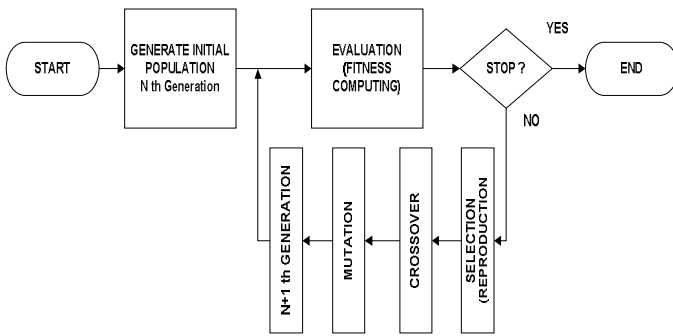
Shortest path problem can be defined as finding the shortest path between two vertices of a directed graph where each arc has been weighted. The shortest path is considered as one of the most fundamental network optimization problems. This problem comes up in practice and arises as a sub problem in many network optimization algorithms [1]. One of the most popular algorithm is the Dijkstra's algorithm conceived, which solves the shortest path problem in  $O(n^2)$  time on a

graph with  $n$  number of nodes and positive edge weights [2].

In last decade, there have been number of approaches used Genetic algorithm (GA) in the solution of the shortest path problems. GA is a heuristic technique developed by John Holland in 1975 based on genetic and natural selection principles [3]. Goldberg proved that GA is one of the powerful search methods in both theory and practice [4]. GA starts with generating an initial population by random selection of the individuals named chromosomes that each encodes the solution of the problem. Each chromosome that encodes a candidate solution of the problem is made with a combination of significant genes [5]. GA founded based on two fundamental evolutionary concepts:

- A Darwinian notion of fitness, which describes an individual's ability to survive
- Genetic operators, which determine the next generation's genetic makeup based upon the current generation [6].

The first step starts with obtaining the values that fitness function returns for each chromosome and selecting the best chromosomes of initial population, which will form the individuals of the next generation. The parents selected for regeneration are replaced by crossover operation and changed by mutation operation to produce child chromosomes. The chromosomes that are not passed through crossover or mutation and newly generated child chromosomes form a new population [5]. The generation of new *populations* repeats till defined number of times in advance or is being continued until not having better chromosomes (Fig.1).



**Fig.1** Steps of the Genetic Algorithm

Munemoto et al. implemented a GA, which is practically feasible on wired or wireless network environment [7]. Inagaki et al. proposed an algorithm with fixed length chromosomes [8].

Ahn and Ramakrishna reported that the algorithm proposed by Munemoto required a relatively large population for an optimal solution due to the constraints on the crossover mechanism. Furthermore, it was reported that this was not suitable for large networks or real-time communications [9]. On the other hand, they suggested that the algorithm proposed by Inagaki et al. required a large population to attain an optimal or high quality of solution due to its inconsistent crossover mechanism [8]. They proposed a GA for solving the shortest path problem that uses chromosomes with variable lengths. They showed that as the number of the nodes becomes more than 20, the computing time by adopting the GA was less than that by adopting the Dijkstra algorithm.

Hasan et al. produced a different solution for the shortest path problem using GA [10]. They employed a chromosome-coding scheme using node indices and

distance weights. The complete chromosome of a candidate was divided into node fields, which were equal to the number of nodes in the network. They proposed different crossover and mutation methods, which are appropriate their chromosome encoding. They tested their proposed algorithm on networks with 10, 20, 50 and 100 routers and results demonstrated consistent and speedy convergence for the tested scenarios.

This study presents a route guidance system and a GA approach applied on this routing system to find the shortest driving time. Proposed guidance system provides driving advice for the drivers considering not only the distances but also traffic density and the allowable velocity limits of the roads. Thus, it computes the shortest driving time instead of the shortest path.

The rest of the paper is organized in four sections: describing the shortest driving time problem and the genetic algorithm proposed to solve this problem; the basic design of the proposed route guidance system; experimental results of the genetic algorithm obtained from the networks with different sizes; and the general conclusions of the study.

## 2 Proposed Genetic Algorithm for Route Guidance System

### 2.1 Shortest Driving Time

It is possible to describe underlying topology of a network with directed graph  $G=(N,A)$  where  $N$  is the set of the nodes, of cardinality  $n$ , and  $A$  is the set of the arcs, of cardinality  $m$ . There is a cost  $T_{ij}$  for each  $(i,j) \in A$ .

These costs are defined in a cost matrix  $C=[T_{ij}]$ . Source and destination nodes are respectively shown as  $B$  and  $V$ . The connection information of the nodes with each other is described in an adjacency matrix  $I_{ij}$  shown below:

$$I_{ij} = \begin{cases} 1, & \text{if the link from node } i \text{ to node } j \text{ exist in adjacency list} \\ 0, & \text{otherwise.} \end{cases}$$

In shortest driving time problem, the cost is  $T_{ij}$  which defines the driving time from node  $i$  to node  $j$ . Using the above definitions, the shortest driving time problem can be formulated as a combinatorial optimization problem minimizing the objective function below as follows [9]:

$$\text{minimize} \quad \sum_{i=B}^V \sum_{\substack{i=B \\ i \neq V}}^V T_{ij} I_{ij}$$

(1)

subject to

$$\sum_{\substack{i=B \\ i \neq V}}^V I_{ij} - \sum_{\substack{i=B \\ i \neq V}}^V I_{ji} \begin{cases} 1, & \text{if } i=B \\ -1, & \text{if } i=V \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

and

$$\sum_{\substack{i=B \\ i \neq V}}^V I_{ij} \begin{cases} =0, & \text{if } i=V \\ \leq 1, & \text{if } i \neq V \end{cases} \quad (3)$$

In shortest driving time problem, cost  $T_{ij}$  is calculated as follows:

$d_{ij}$  = distance from node  $i$  to node  $j$

$v_{ij}$  = allowable velocity limit from node  $i$  to node  $j$

$\gamma_{ij}$  = traffic density from node  $i$  to node  $j$

On calculation of the driving time from node  $i$  to node  $j$ , allowable velocity limits and traffic densities from node  $i$  to node  $j$  are considered. In this case, driving time can be formulated as showing below:

$$T_{ij} = \frac{d_{ij}}{v_{ij} \times (1 - \gamma_{ij})} \quad (4)$$

### 2.2 Genetic Encoding

In the chromosome structure of proposed GA, node numbers of the route from source to destination are stored as positive integer numbers. Each locus of the chromosome represents an order of a node in a routing path. The chromosome length is static. The total number of nodes  $N$  is the length of each chromosome in the network. The node numbers that represent the routing path from source ( $B$ ) to destination ( $V$ ) are encoded in the chromosome. If the node number of the solution is smaller than the total node number  $N$ , unused genes of the chromosome is assigned by zero value. The chromosome encoding of the proposed GA is shown in Fig.2.

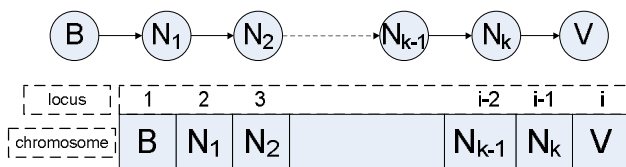


Fig.2 Chromosome encoding of a routing path

### 2.3 Creating Initial Population

To produce the initial population DFS (Depth First Search), algorithm is reorganized to produce random paths from source to destination.

### 2.4 Fitness Function

The fitness function is the object to be optimized. The fitness function must accurately measure the quality of the chromosome in the population and must have computational efficiency; therefore, the fitness function has a critical importance. The cost of the fitness function described in [9] rearranged according to the formula given in Section 2.1 to compute the shortest driving time as follows:

$f_i$  : fitness function of the  $i$  th chromosome

$g_i(j)$  :  $j$  th gene of the  $i$  th chromosome

$l$  : length of the chromosome

$D$  : distance between two nodes

$V$  : allowable velocity limit between two nodes

$Y$  : traffic density between two nodes

$$f(i) = \frac{1}{\sum_{j=1}^{u-1} \frac{D(g_{i(j)}, g_{i(j+1)})}{V(g_{i(j)}, g_{i(j+1)}) \times (1 - Y(g_{i(j)}, g_{i(j+1)}))}} \quad (5)$$

### 2.5 Selection (Reproduction) of a New Generation

The selection (reproduction) operator is intended to improve the average quality of the population by giving the high-quality chromosomes a better chance to be copied into the next generation. In this study, roulette wheel selection method, which is a proportionate selection method that picks out chromosomes based on their fitness values relative to the fitness of the other chromosomes in the population, was performed.

### 2.6 Crossover

The crossover operator generates new individuals called offspring, by recombining the genetic material of two individuals, deemed the parents. Individuals with higher fitness scores are selected with greater probability to be parents and “pass on” their genes to the next generation. In this study a new gene search method named "first-matched gene" is presented in which the first genes matched on two chromosomes as crossover points were selected. Difference of the crossover phase used in this study from classical crossover is that crossover points do not have to be in the same locus of chromosomes. In the end of the crossover operation, two child chromosomes is obtained. If these child chromosomes are infeasible, a repair function for

dealing with the infeasible chromosomes is performed. Steps of the crossover phase are shown in Fig.3.

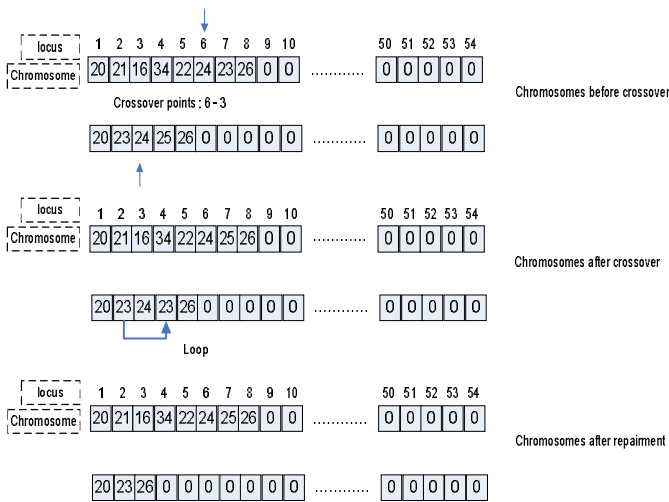


Fig.3 Crossover phase

**2.7 Mutation**

Mutation operation maintains the genetic diversity of the population and changes the genes of the selected chromosomes and keeps away from local optima. Through mutation, GAs can search previously unexplored sections of the solution space. Mutations consequently assure that the entire search space is connected [11].

A random path is generated using reorganized DFS algorithm from the mutation node to the destination node which is also used for creating initial population. This random generated path is exchanged with the genes starting from mutation point. The mutated chromosome may be feasible. In this case, a repair function for dealing with these infeasible chromosomes is performed. Steps of the mutation phase are shown in Fig.4.

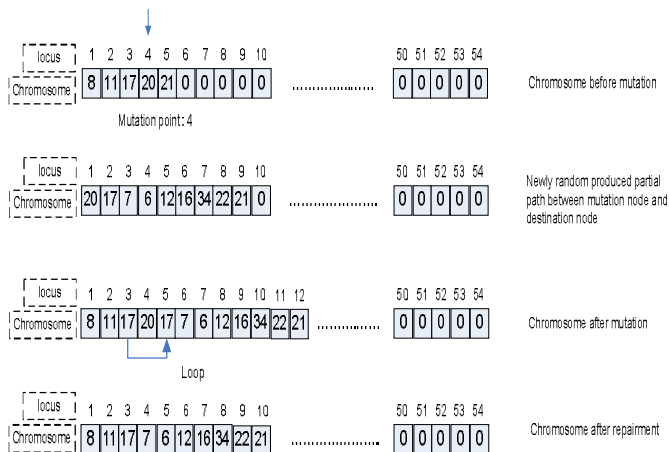


Fig.4 Mutation phase

**2.8 Termination of the Genetic Algorithm**

To terminate the genetic algorithm, the fitness value of the best chromosome on each generation is checked. If the fitness value of the best chromosome obtained does not change for 10 generations, the algorithm is stopped.

**3 Structure of the Designed Route Guidance System**

The system had been designed for navigation devices and pocket computers that use Microsoft Mobile Operating Systems. By using genetic algorithm, it is intended to avoid the loss on the performance of these devices with restricted memory and processor capacities even on large networks with thousands of nodes. Real-time traffic conditions are obtained from a XML service. This service produces XML based data about the traffic condition periodically. Client devices also take the data periodically. The maps that supposed to be used in the route guidance system should be loaded on the device in advance and therefore only the traffic density is obtained from the XML service over Internet connection in real-time. The route guidance system is shown in Figure 5.

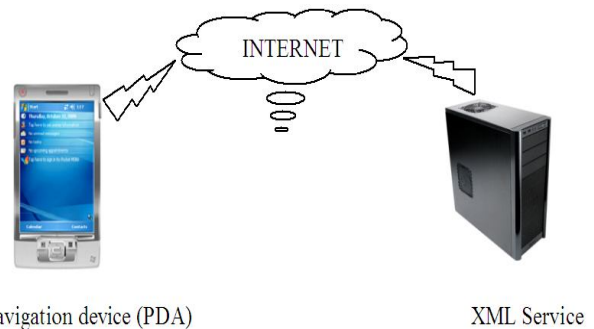


Fig.5 Route guidance system

The traffic density is simulated in the map loaded on the navigation device. Colors of the roads on the map are based on final velocity limits of the roads calculated by considering the traffic densities and allowable velocity limits of the roads. The roads are divided into four groups according to final velocity limits. The color codes of these groups are described in Table 1.

Table 1. Velocity ranges for coloring the map in route guidance system

Colors	Velocity Limit Ranges (km/h)
Red	0-30
Orange	30-50
Yellow	50-70
White	70-90

A sample map loaded on a pocket pc is shown in Fig.6. The obtained road after calculation is shown to users in blue color.



Fig.6 Map screen for the route guidance system

## 4 Results

The route guidance system had been developed using C# programming language. Windows Mobile 5.0 Pocket PC R2 emulator installed on Microsoft Visual Studio 2008 is used on experiments. Beside the traffic network of Ankara city with 54 nodes, random generated graphs with 10, 50, 250 and 1000 nodes used. Distances, velocity limits and updated traffic density obtained from XML service periodically generated randomly. Experiments done 100 times for each case are given on result tables.

Our first results are taken from the city map of Ankara with 54 common nodes. Performance measurement values like failure ratio and average generations of the results obtained by proposed genetic algorithm is listed in Table 2.

Table 2. Results obtained from 100 performs of proposed genetic algorithm on 10 different routes.

Source-Destination	Average Generation	Failure Ratio (%)
5-8	12	0
1-13	12	0
4-43	12.1	0
10-14	12.7	5
7-24	16.7	0
40-10	14	0
2-24	27.9	10
14-18	16	4
12-10	12	0
20-54	12	0

Test results performed on random generated graphs with 10, 50, 250 and 1000 nodes are presented below. Table 3 shows the average generations of the genetic algorithm to find the shortest driving time.

Table 3. Average generations to find the optimum path

		Number of Nodes			
		10	50	250	1000
Population Size	30	17.05	19.43	33.09	36.4
	50	14.16	19.41	30.57	33.2
	100	12.45	16.13	27.53	28.2
	200	12.01	13.22	25.24	27.6
	400	12	12.21	21.37	23.43
	800	12	12.01	18.14	20.12

As given in Table 3, even the number of nodes grew to 1000, the average generation to find the optimum path was not much than 36.4 in worst case and by increasing the population size it was seen that the algorithm found solutions in less generations. When the number of nodes grew by 10 times from 10 to 1000, the time needed for finding the solution grew only by nearly 2 times.

Table 4 shows the average difference of the exact routes and approximate routes found by the proposed genetic algorithm.

Table 4. Average difference of the exact routes and approximate routes in %

		Number of Nodes			
		10	50	250	1000
Population Size	30	0.6	20.7	140.1	159.3
	50	0	9.1	79.8	87.3
	100	0	3.6	57.1	71.2
	200	0	0.3	34.7	52.8
	400	0	0	20.2	33.1
	800	0	0	4	12.4

According to the results given in Table 4, approximate solutions found by the proposed genetic algorithm were very close to exact solutions with the node number 10 and 50. On a graph with 10 nodes, exact solutions obtained when the population size was 50 or greater. When the node number grew to 50, the exact solutions obtained with population starting from 400. By increasing the population size, it was seen that the algorithm got closer to exact solutions. When the node number was 1000, average difference of the exact and the approximate routes was cut from 159.3% to 12.4% with the increase of population size from 30 to 800.

## 5 Conclusion

When the amount of data being processed is too large on handheld devices, it is not proper to find exact solutions. The GA presented in this paper finds the acceptable approximate solutions effectively even on large networks while considering real-time information.

Results obtained from experiments shows that a route guidance system that computes the shortest driving time considering the real-time traffic information can be designed using GA for handheld devices produced with limited processor and memory capacities.

### References:

- [1] Xu MH, Liu YQ, Huang QL, Zhang YX, Luan GF, An improved Dijkstra's shortest path algorithm for sparse network, *Applied Mathematics and Computation*, 2007, Vol.185, No.1 pp.247-254.
- [2]Dijkstra EW, A Note on Two Problems in Connection with Graphs. *Numerische Mathematik*, 1959,Vol.1, No.1,pp.269-271.
- [3]Holland JH, *Adaptation in natural and artificial system*, The University of Michigan Press, 1975.
- [4]Goldberg DE, *Genetic Algorithms in Search Optimizations and Machine Learning*, Boston, Addison-Wesley Longman Publishing Co, 1989.
- [5]Whitley D, A genetic algorithm tutorial, *Statistics and Computing*, 1994, Vol.4, No.2,pp.65-85.
- [6]De Jong K, Learning with Genetic Algorithms: An Overview, *Machine Learning*, 1988, Vol.3, pp.121–138.
- [7]Munemoto M, Takai Y, Sato YA, A migration scheme for the genetic adaptive routing algorithm, *In Proceedings of the Systems, Man, and Cybernetics IEEE International Conference*, 1998, pp.2774–2779.
- [8]Inagaki J, Haseyama M, Kitajima H, A genetic algorithm for determining multiple routes and its applications, *In Circuits and Systems, Proceedings of the 1999 IEEE International Symposium*, 1999, pp. 137-140.
- [9]Ahn CW, Ramakrishna RS, A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations, *IEEE Transactions on Evolutionary Computation*, 2002, Vol.6, No.6,pp.566-579.
- [10]Hasan BS, Khamees MA, Mahmoud ASH, A Heuristic Genetic Algorithm for the Single Source Shortest Path Problem, *5th IEEE/ACS International Conference on Computer Systems and Applications*, 2007, pp.187-194.
- [11]Cedric D, Pawan D, Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks, *European Journal of Operational Research*, 2003, Vol.144,pp.27-38.