Solving Shortest Path Problems Using Genetic Algorithms

Shatha Abdullah Rasheed

University of Baghdad/ College of Arts

Abstract

In this paper we considered solving shortest path problem; this problem is widely applied in transportation and communication networks. The objective of this problem is to determining a path with minimum distance, time or cost from source to the destination. However, practical transportation networks will become much more complicated and needed to solve efficiently. Roadway and telephone systems are the examples of them. In this work we use genetic algorithms to solve the shortest path problem. The proposed algorithms were tested on random generated shortest path problems. The experimental results are very encouraging and show that genetic algorithms a good approach for such kinds of difficult to solve problems.

Key words: genetic algorithm, shortest path problem

1. Introduction

One of the most common problems encountered in analysis of networks is shortest path problem: finding a path between two designated nodes having minimum total length or cost. It is a fundamental problem that appears in many applications involving transportation, routing and communications. Several good methods have been proposed to solve the problem to optimality.

This study provides an interesting alternative: using genetic algorithms to find out a shortest path. The purpose is not, of course, to compare the genetic algorithms with conventional algorithms, because genetic algorithms will be unable to compete. Shortest path problem forms an important base for other relevant problems, such as multiple objective shortest path problems, nonlinear objective shortest path problems, multiple fuzzy objective shortest path problems, network flow problems, and shortest path based network analysis and optimization problems. These problems are difficult to solve in general and there is no exact optimal solution method known yet. A common feature of these problems is that they may be solved, either exactly or approximately, by identifying shortest path. Therefore, these studies will provide a base for constructing efficient solution procedures for shortest path-based network optimization problems. The most thorny problem when applying genetic algorithms to this problem is how to encode a path in a graph into a chromosome [1].

This paper presents a new genetic algorithm to solve the shortest path problem. Genetic algorithm is appealing as a solution, since it deviates from traditional algorithms that try to compare every possibility to find the best solution that might be a time consuming algorithm for a graph containing a large number of nodes and edges [2].

2. The Shortest Path Problem

The shortest path problem is defined as that of finding a minimum-length (cost) path between a given pair of nodes. Shortest path problem is a classical research topic. It was proposed by Dijkstra in 1959 and has been widely researched. The Dijkstra algorithm is considered as the most efficient method. It is based on the Bellman optimization theory. But when the network is very big, then it becomes inefficient since a lot of computations need to be repeated. Also it can not be implemented in the permitted time [3,4].

3. Genetic Algorithm

There are many heuristic algorithms used to solve complex optimal problems. Genetic algorithms (GA) are one of the most powerful and successful methods in stochastic search and optimization techniques based on the principles of the evolution theory [5].

The genetic algorithm search approach is now the underlying solution methodology of many optimization systems and has fundamentally changed the way many organizations operate [6].

Genetic algorithm is inspired by Darwin’s theory about evolution. The genetic algorithm is an optimization solution that is based on natural selection. The genetic algorithm repeatedly changes a population of individual solutions. At each step, the genetic algorithm chooses individuals randomly from the current population to be parents and uses them to reproduce the children for the next generation. Over successive generations, the population "evolves" to an optimal solution.

The algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a wish, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness- the more suitable they are the more chances they have to reproduce. Basically, several random sets of parameters are considered for an algorithm, and fitness value (optimization value) is calculated for each. Based on the fitness values, the best sets are mixed (Selection, Crossover and mutation are combined) together and new sets are again applied to the algorithm until an optimal parameter(s) is obtained. This effect is usually obtained by breaking genetic algorithm into a few small parts. The algorithm stops when predefined conditions (for example the number of populations or improvement of the best solution) are met[2,7].

4. The Proposed Work

If we want to use genetic algorithm to solve the practical problems, we need solve how to coding and create the initial population, how to evaluate the fitness value, how to evaluate the fitness value, how to select, crossover and mutation. Here is the basic idea for the shortest path problem using genetic algorithm.

a- Problem definition

In the graph G(V,E), any two nodes (Vi,Vj) correspond to a weight value d(Vi,Vj). If d(Vi,Vj) is not exist, then d(Vi,Vj)=1000. A road from node S to T is an (S,T) path, the length of a path is the sum of the weight of all the edges in one path.

The metric of optimization is distance of path between the nodes. The total distance is the sum of the weight of all the edges in one path. The goal is to find the path with minimum total distance between source node(S) and destination node (T).

b- Chromosome encoding and initial population generation

In the proposed genetic algorithm, the chromosome is represented as a string of numbers, which represents the nodes' numbers in a sequence, the number of nodes in each chromosome is not equal.

In the chromosome structure of the proposed genetic algorithm, 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. Because of different paths include different nodes and arcs, the length of chromosome not fixed. The node numbers that represent the routing path from source (S) to destination (T) are encoded in the chromosome.

The first step is to initialize the first population (generation) that you will use as a base. This algorithm uses random initialization to create the first generation. Because purely random generation is not feasible for shortest path problem the algorithm attempts to be as random as possible. The start reach gene by adding the source node. Then they randomly choose a node that has an edge from the source. Then they repeat the process from that node and so on, making sure not to add a node twice. If they get to a point where they can not add a node without a repeat, they backtrack until there is a new node they can add that is not one they have tried previously. They continue this until the destination is found.

c- Fitness Function

The fitness function is the object to be optimized. The fitness function must have accurately measured the quality of the chromosome in the population and must have computational efficiency; therefore the fitness function has a critical importance.

The evaluation function takes a path in the population. It gets the distance between each node pair in the path, by calling a function to read from the distance array. Add them together and returns the sum as the cost of the path.

In this work, we need to create the cost matrix *A*. cost matrix reflects the distance between each node. a_{ij} is the distance $d(Vi,Vj)$ between point *Vi* to point *Vj*. Because the traffic network is abstracted to an undirected graph, so the cost matrix is a symmetric matrix. The value 1000 represents that there is no direct link between these nodes, for example, a graph as shown in figure 1:-

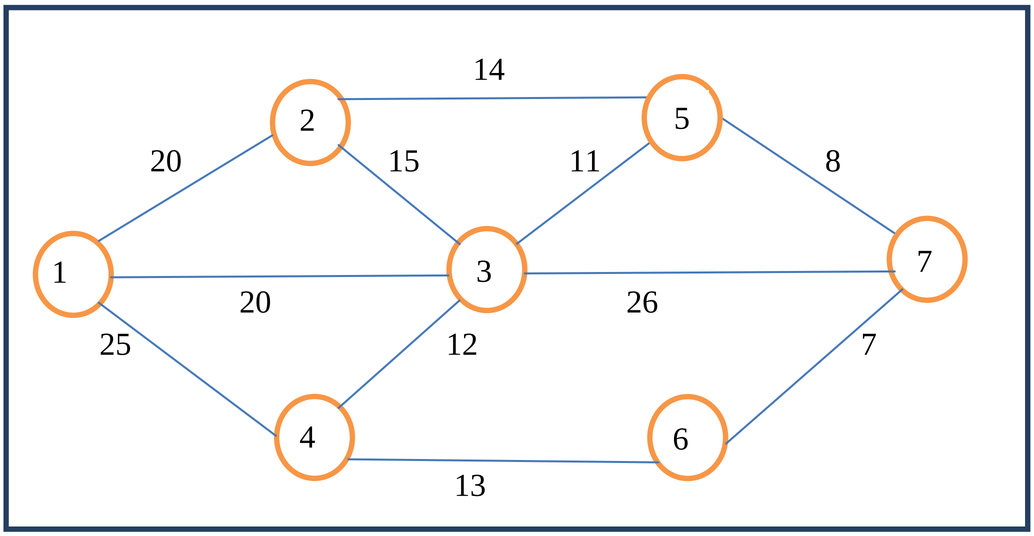

Figure (1): Sample Network Topology

The cost matrix of the figure 1 is:-

$$\begin{bmatrix} 1000 & 20 & 20 & 25 & 1000 & 1000 & 1000 \end{bmatrix}$$

$$A = \begin{matrix} 20 & 1000 & 15 & 1000 & 14 & 1000 & 1000 \\ 20 & 15 & 1000 & 12 & 11 & 1000 & 26 \\ 25 & 1000 & 12 & 1000 & 1000 & 13 & 1000 \\ 1000 & 14 & 11 & 1000 & 1000 & 1000 & 8 \\ 1000 & 1000 & 1000 & 13 & 1000 & 1000 & 7 \\ 1000 & 1000 & 26 & 1000 & 8 & 7 & 1000 \end{matrix}$$

After obtaining the cost matrix of a graph, the fitness function can be defined as following:-

$$f(x) = \sum_{n=1}^{N-1} A(x(n), x(n+1))$$

x: a chromosome

N: the length of chromosome

d- Selection

It is a feature of Genetic algorithm for selecting parents for next generation. Current work is based on roulette wheel selection. In roulette wheel selection, the individual is selected based on the relative fitness with its competitors. This is similar to dividing the wheel into a number of slices. Fittest chromosomes get larger slice [4]. For a given population X, if the fitness of each chromosome is *fi(x)*, the number of chromosome in X is M, so the probability of each chromosome is selected is as following:

$$P_i = f_i(x) / \sum_{K=1}^{M} f_k(x)$$

After that, a random number between 0 and 1 is generated, if the generated random number is equal or less than the P_i, the chromosome (i) is passed on to the new generation directly. The process is continued as population size increases.

e- Crossover

Crossover operation is applied to obtain better chromosomes. The crossover operator combines sub parts of two parent chromosomes and produces offspring that contains some part of both the parent genetic material [4].

With some probability, the program mates the two individuals. The crossover function takes two parents to mate. It looks for the common points in the parents. The common nodes are where these two paths intersect. Among the common points, the program selects one of them randomly. It makes the crossover from that point. The crossover operation is illustrated below.

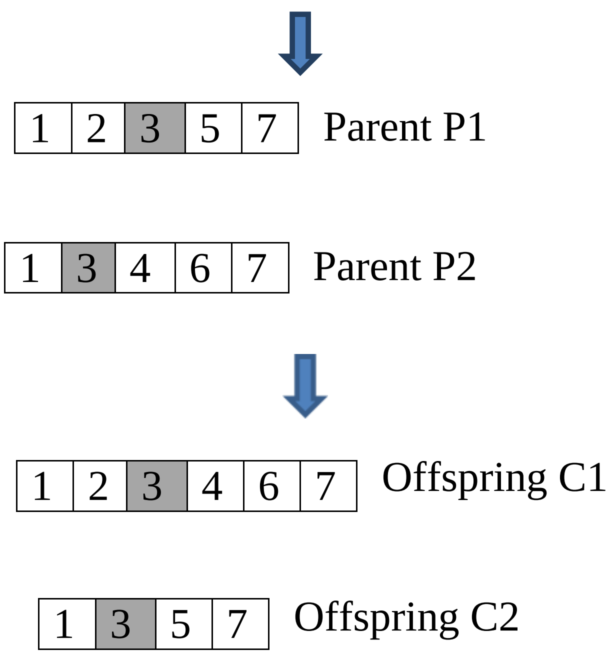

Figure (2): Crossover Operator

f- Mutation

Mutation operation maintains the genetic diversity of the population and changes the genes of the selected chromosomes, thereby keeping them away from local optima [7].

The mutation operation is needed with certain probability Pm. The operation is carried out with random selection in the gene.

Mutation operation can be bit flipping, interchanging, inversion, insertion, reciprocal exchange or others. This paper uses insertion method. In case of insertion a node is inserted at random position in the string. This is because a node along the optimal path may be eliminated through crossover. Using insertion, it can be brought back. Once mutation is completed, the offspring generated by mutation have to be validated with the same process used in crossover.

The mutation operation is illustrated below:-

Parent	1	2	3	7	
Offspring	1	2	3	5	7

Figure (3): Mutation Operator

G- Termination Criteria

To terminate the genetic algorithm, we use two termination criteria:-

1-The fitness value of the best chromosome on each generation is checked, if the fitness value of the best chromosome obtained does not change of ten generations, the algorithm is stopped.

2-The number of generations is equal to the maximum generation number that we have proposed in this work, and we must check, up to what number of generation there is improvement in chromosome fitness.

H-Steps of the proposed work

The pseudo code of the proposed work is as follows:-

1. Initialize crossover rate, mutation rate, population size, maximum_generation
2. Read the number of nodes and cost matrix
3. Generate the initial population
4. I=0, count=0.
5. Calculate the fitness values of the chromosomes
6. find the minimum_fitness in generation(I)

   // first see who survives to mate
7. J=0
8. Roulette wheel selection
9. Crossover on selected chromosomes.
10. Mutation on selected chromosomes.
11. Calculate the fitness values of the new chromosomes
12. J=J+1
13. If the number of chromosome (J) in generation (I)< population size go to step 8
14. I=I+1
15. find the minimum_fitness in generation(I)
16. if minimum_fitness in generation(I)= minimum_fitness in generation(I-1) then count:=count+1
17. If I<maximum_generation or count<10 go to step 7 else stop

5- The Results and Conclusions

In most situations the initial population may contain routs that are impossible to traverse in practical situations. The selection criteria for the new generations are totally based on the fitness value given to each individual. But the changes in probabilities of recombination operators may produce surprising results. So it is convenient to keep the operators' values in a certain acceptable range.

The current work is based on network consisting 7 nodes. We have taken population size 15 in first generation. By selecting chromosomes based on roulette wheel selection and application of genetic algorithm operators generations are performed.

After the path to all nodes from source node 1is computed, the set of paths to a specific node will be displayed. Let the destination node 7. the following table is the set of paths from node 1 to 7. the optimal path returned is 1-2-5-7 with fitness value is 42.

Route	Fitness value
1-4-6-7	45
1-3-7	46
1-2-5-7	42
1-2-3-7	61
1-2-3-5-7	54
1-2-3-4-6-7	67
1-4-3-7	63
1-3-4-6-7	52
1-4-3-5-7	56
1-2-5-3-7	71
1-2-5-3-4-6-7	77

Table (1): Routes from source node (1) to destination node(7)

The main advantage of genetic algorithms is their flexibility and robustness as a global search method. GA explore solution space in multiple directions at once . Its well suited for solving problems were the solution space is huge and time taken to search exhaustively is very high. GA has ability to solve problems with no previous knowledge. The performance of GA is based on efficient representation, evaluation of fitness function, population size, crossover rate, mutation probability and the selection method. Current work can be improved by using some intelligent approach for initial population and using better crossover, mutation probabilities.

6-Refernces

[1] Mitsuo Gen, Run wei Cheng and Dingwei wang, Genetic Algorithms for solving shortest path problems, IEEE, pp. 401-406, 1997.

[2] Saeed Behzadi and Ali A. Alesheikh, Developing a Genetic Algoritm for Solving Shortest Path Problem, WSEAS International conference on URBAN planning and Transportation (UPT'07),Heraklion, Crete Island, Greece, July 22-24, 2008.

[3] Bilal Gonen, Genetic Algorithm Finding the Shortest Path in Networks, University of Nevada, Reno.

[4] Rakesh Kumar and Mahesh Kumar, Exploring Genetic Algorithm for Shortest Path Optimization in Data Networks, Global Journal of Computer Science and Technology, Vol. 10 Issue 11 (Ver. 1.0), pp. 8-12, October ,2010.

[5] Yinzhen Li, Ruichun He and Yaohuang Guo, Faster Genetic Algorithm for Network paths,ORSC& APORC, pp. 380-389, 2006.

[6] Basela S. Hasan , Mohammad A. Khamees and Ashraf S. Hasan Mahmood, A Heuristic Genetic Algorithm for the Single Source Shorteset Path Problem, IEEE, pp. 187-194, 2007.

[7] Ismail Rakip Karas and Umit Atila, A Genetic Algorithm Approach for Finding the Shortest Driving Time on Mobile Devices, Scientific Research and Essays Vol. 6(2), pp. 394-405, 18 January , 2011.

حل مشكلة اقصر مسار باستخدام الخوارزميات الجينية

شذى عبد الله رشيد

جامعة بغداد/كلية الآداب

الخلاصة

درسنا في بحثنا هذا حل مشكلة اقصر مسار، وهذه المشكلة تستخدم بشكل واسع في النقل وشبكات الاتصالات ، ان هدف هذه المشكلة هو تحديد المسار بأقصر مسافة، بأقل وقت او كلفة من المصدر الى الهدف ، مهما كانت شبكات النقل اكثر تعقيدا" فأنها بحاجة الى الحل بكفاءة. ان طرق النقل وأنظمة الهاتف هي أمثلة عليها. وفي عملنا هذا استخدمنا الخوارزمية الجينية لحل مشكلة اقصر مسار . والخوارزمية المقترحة تم تطبيقها على مسارات تم توليدها بشكل عشوائي وكانت النتائج مشجعة وبينت لنا ان الخوارزمية الجينية هي طريقة جيدة لحل المشاكل الصعبة.