

Developing a Genetic Algorithm for Solving Shortest Path Problem

SAEED BEHZADI, and ALI A. ALESHEIKH

Department of GIS, Faculty of Geodesy and Geomatics Engineering,

K.N. Toosi University of Technology

Valiasr Street, Mirdamad Cross, Tehran, P.C. 19967-15433

IRAN

Abstract: Routing is gaining prime importance in our increasingly mobile and highly Information Technology enabled world. The shortest path problem is widely applied in transportation, communication and computer networks. It addresses the challenges of determining a path with minimum distance, time or cost from a source to the destination. This paper introduces a novel Genetic Algorithm (GA) approach to solve the shortest path problem. A connected graph is assumed in which all the weights of the paths are positive. New mutations as well as crossover parameters are defined. The use of mutation parameters depends on the extent of the area under consideration, and the distance between the start and end points. In order to evaluate the proposed algorithm, an urban road map is selected. In all case studies, the algorithm was successful in determining the shortest path. The details of the experimental results are discussed and presented in the paper.

Key-Words: Genetic Algorithm, Shortest path problem, Mutation, Crossover, Graph theory.

1 Introduction

The problem of estimating a shortest path between two nodes is a well-known problem in network analysis. Shortest path algorithms are subject of extensive research, resulting in a number of approaches for various conditions and constraints [2, 4, 5]. The shortest path problem; finding the path with minimum distance, time or cost from a source to a destination; is one of the most fundamental problem in transportation networks. It arises in a wide variety of scientific and engineering problem settings, both as stand-alone models and as subproblems in more complex problem settings [1]. This paper presents a new Genetic Algorithm to solve the shortest path problem. Practical examples can be found in finding the shortest path (i.e. the shortest possible distance) between two nodes in a graph or two cities in a map with potential connections (assuming that the path distances are always positive). Typically, a transportation network is represented by a graph with each node representing a city and each edge being a path between two cities. 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.

Genetic algorithms are 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 [13]. 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 [7, 11]. 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 a 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 the genetic algorithm into a few small parts [10]. The algorithm stops when predefined conditions (for example the number of populations or improvement of the best solution) are met.

2 Genetic Algorithm

3 Implementation

Each node in the map is given a unique integer value index from 1 to i , where, i is the number of nodes in the map. Each individual is designed to represent a solution for the problem and it does not have repeated node indices. The length of the individual is selected to be equal to the number of nodes in the total map, because there may be cases such that the shortest path may contain the total number. For a map with i number of nodes the gene length is equal to i , where T_j is the j th node in the map, namely;

$$\text{Route_1: } [T_1, T_2 \dots T_j, \dots T_{(i-1)}, T_i]$$

Route_1 is a path that contains all nodes in the graph.

3.1 Coding

Permutation encoding is used in this problem. In permutation encoding, every chromosome is a string of numbers, which represents the number of nodes in a sequence. The number of nodes in each chromosome is not equal.

Integer vector $P = (v_1, v_2, \dots, v_k)$ is a chromosome to represent a path from nodes 1 and n . Because different paths include different nodes and arcs, the dimension of chromosome is not fixed. If (v_1, v_2, \dots, v_k) represents a path from nodes 1 to n , then the path includes $(1, v_1) \in A, (v_2, v_3) \in A, \dots, (v_k, n) \in A$ [12] then $X_{i,j}$ is defined as:

$$X_{i,j} = \begin{cases} 1, & \text{if } i = 1, j = v_1, \\ 1, & \text{if there exists } l \text{ such that } i = v_l, j = v_{l+1}, \\ 1, & \text{if } i = v_k, j = n, \\ 0, & \text{otherwise,} \end{cases}$$

For all $(i,j) \in A$. It is also easy to verify that $\{x_{ij} \mid (i,j) \in A\}$ obtained by this way is a path from nodes 1 to n . Conversely, let $\{x_{ij} \mid (i,j) \in A\}$ be a path from nodes 1 to n . A chromosome is obtained by the following procedure.

3.2 Initial population

The size of the population depends on the number of nodes in the graph and the length of each chromosome in the population. It is defined as:

$$\text{Chromosome_Length} = \frac{N_P \times D_{(\text{Start}, \text{End})}}{\sqrt{\text{Area}}} \times n \quad (1)$$

Where:

Chromosome_Length: The length of each Chromosome in the population

N_P: The number of points in the graph

$D_{(\text{Start}, \text{End})}$: The distance between Start and End points

Area: the area of the graph that is approximately computed by using minimum and maximum of x, y in the area. This component is defined as (2):

$$\text{Area} = (\text{Max_X} - \text{Min_X}) \times (\text{Max_Y} - \text{Min_Y}) \quad (2)$$

n : The flexibility of the length of the chromosome. It is approximately defined as an integer number between 1 and 4 that depends on the density of the points between start and end points. It is 4 when the density of the points between start and end points is more than other parts.

The first and last genes of each individual are consecutively the start and end points. Other genes in each chromosome are chosen randomly from the other points in the graph as they are not equal.

3.3 Fitness Function

The fitness function in this research is defined as [1]:

$$F(x) = \frac{1}{\text{Actual_length_of_the_path}} - \text{The_number_disconnected_path} \quad (3)$$

Considering the first part of the function, Actual_length_of_the_path is calculated by summing the route distances of the path. Therefore, the value for Actual_length_of_the_path is between 0 and 1. The_number_disconnected_paths counts the number of unreal path in the total path. Hence, the value for The_number_disconnected_paths is between 0 and i . It means that the function varies from $-i$ to $+1$.

If there is not a path between node t_1 and t_2 then the SLD (Straight Line Distance) between t_1 and t_2 is considered as the actual distance.

3.4 Fitness Scaling

Fitness scaling converts the raw fitness scores that are returned by the fitness function to values in a range that is suitable for the selection function. The selection function uses the scaled fitness values to select the parents of the next generation. The selection function assigns a higher probability of selection to individuals with higher scaled values [10].

After creating the initial population, fitness values for each individual are calculated. In this research,

proportional scaling is used to make the scaled value of an individual proportional to its raw fitness score.

3.5 Genetic Operator

Genetic operators mimic the process of heredity of genes to generate new offspring at each generation and play a very significant role in genetic algorithm [8]. In our algorithm, the crossover operator, mutation operator and selection are defined as:

3.5.1 Crossover operator

Crossover examines the current solutions to find better ones [6, 9]. Physically, crossover in the routing problems plays the role of exchanging each partial route of the two chosen chromosomes in such a manner that the offspring produced by the crossover will only be one route. This enforces the selection of one-point crossover as a good candidate

scheme for the proposed GA. One partial route connects the source node to an intermediate node and the other partial route joins the intermediate node to the destination node [3].

Let $P_1 = (v_1, v_2, \dots, v_k)$ and $P_2 = (v'_1, v'_2, \dots, v'_{k'})$; be two chromosomes. The crossover operation is: if there are common nodes between them, then randomly choose one, say $v_i = v'_{i'}$. The following two chromosomes are then produced:

$$(v_1, v_2, \dots, v_i, v'_{i'+1}, \dots, v'_{k'}), (v'_1, v'_2, \dots, v'_{i'}, v_{i+1}, \dots, v_k)$$

Which are also feasible chromosomes representing paths from nodes 1 to n. If there is no common node, then nothing is done [12]. This process is shown in Fig. 1.

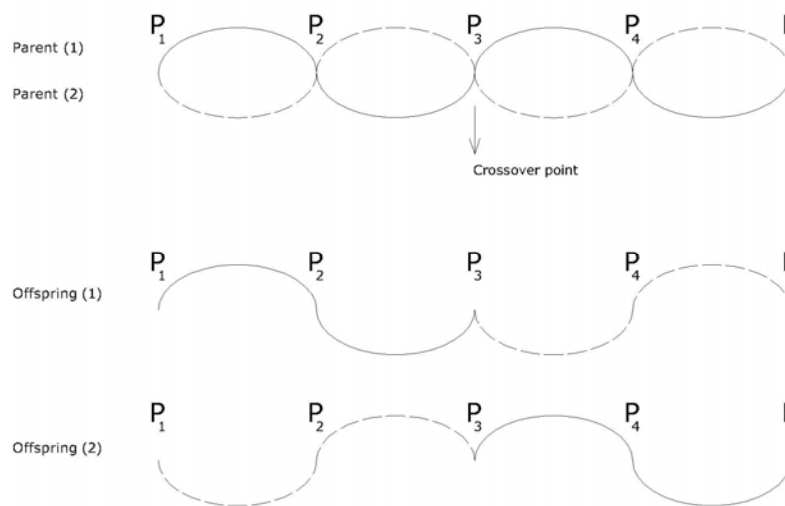


Fig. 1. Crossover operator

3.5.2 Mutation operator

In this research, three kinds of mutation are considered. The logic of using each kind of these mutation operators depends on the extension of the area and the distance between start and destination points. These kinds of mutation are defined as:

3.5.2.1 Internal mutation

In this case, mutation operator is performed by randomly choosing two chromosomes in the new generation to mutate. Then, the two selected genes will be exchanged. For example $(1\ 2\ 3\ 4\ 5\ 6\ 8\ 9\ 7)$ is a path that contains 1, 2, ..., 8, 9, 7 nodes. By using this operator two nodes are chosen (2 and 8) then they are changed.

$$(1\ 2\ 3\ 4\ 5\ 6\ 8\ 9\ 7) \Rightarrow (1\ 8\ 3\ 4\ 5\ 6\ 2\ 9\ 7)$$

3.5.2.2 External mutation

In this case, mutation operator is accomplished by randomly choosing one chromosome in the new generation to mutate. The selected gene will then be exchanged with another gene that was not in the path. For example $(1\ 2\ 3\ 4\ 5\ 6)$ is a path that contains 1, 2, ..., 6 nodes. By using this operator one node is chosen (2) then it is changed with an external point (9) that was not in the path.

$$(1\ 2\ 3\ 4\ 5\ 6) \Rightarrow (1\ 9\ 3\ 4\ 5\ 6)$$

3.5.2.3 Omitting mutation

In this case, mutation operator is done by randomly choosing one chromosome in the new generation to mutate. Then, the selected gene is omitted. For

example (1 2 3 4 5 6 8 9 7) is a path that contains 1, 2,..., 8, 9, 7 nodes. By using this operator one node is chosen (2) then it is omitted.

(1 2 3 4 5 6 8 9 7) \Rightarrow (1 2 3 4 5 6 8 9 7) \Rightarrow (1 2 3 4 6 8 9 7)

The use of each kind of the mutation operators depends on the problem at hand. For example if the

distance between start and end point is short, i.e. be more specific omitted mutation is preferred and if the distance between start and end point is long provide a number, the ration of using internal and external mutation is more than omitting ones. These kinds of mutations are shown in Fig. 2.

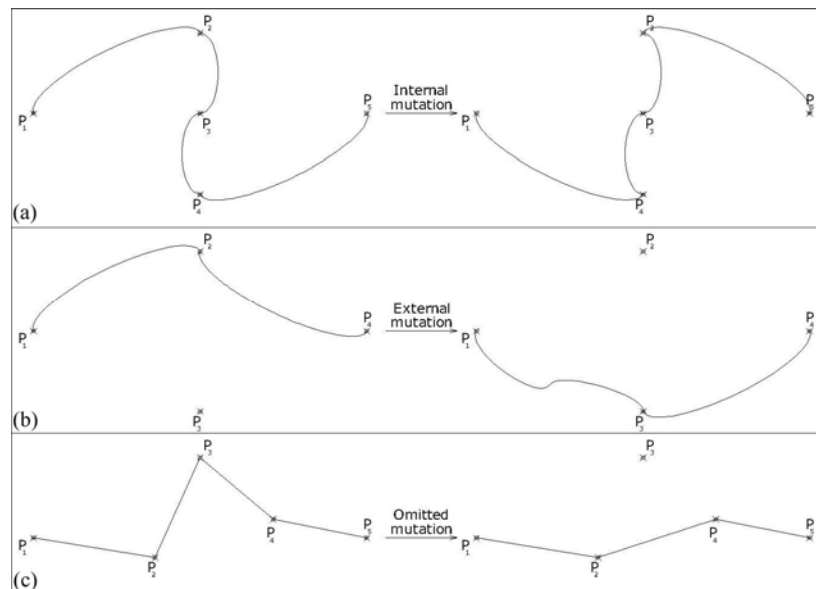


Fig. 2. a. Internal operation b. External operation c. Omitted operation

4 Experiments

To evaluate the performance of the outlined method, we performed experiments using actual road maps of parts of Tehran, the capital city of Iran (Figure.3). In this problem, the lengths of the arcs are

considered as weights of the arcs. The objective was to minimize the total weight of path. The result of this experiment is shown in Fig. 3.



(a)



(b)

Figure3. a. A part of the tested map b. Start and End point and the shortest path

4 Conclusions and recommendations

The main advantage of genetic algorithms is their flexibility. In most GAs, it is common that only one

crossover and mutation operator are defined for the problem but in this paper one crossover and three kinds of mutation operator were defined to solve the

shortest path problem. These operators are executed on a network. In each problem the crossover operator is always used but the ration of using each kind of these mutation operators depends on the extension of the area and the distance between start and destination points. By using this method the solution efficiency becomes higher.

Other crossover operations can be used in addition to the ones introduced in here. Such crossover operations and the use of each kind of the operators can be defined base on the specific problem like what is done for mutation operation. For increasing the efficiency of this algorithm, the first population can be defined base on the direction between start and end point. In this case, not only are the nodes of each gene chosen randomly but also they are restricted between start and end points.

References:

- [1] S. Abeyesundara, B. Giritharan, S. Kodithuwakku, A Genetic Algorithm Approach to Solve the Shortest Path Problem for Road Maps, *Proceedings of the International Conference on Information and Automation*, 2005, pp. 272-275.
- [2] C. Davies, P. Lingras, Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks, *European Journal of Operational Research* Vol. 144, 2003, pp. 27–38
- [3] M. R. Delavar, F. Samadzadegan, P. Pahlavani, A GIS-Assisted Optimal Urban Route Finding Approach Based On Genetic Algorithms, *Proceedings of the ISPRS*, 2004, Istanbul, Turkey.
- [4] E.W. Dijkstra, A note on two papers in connection with graphs, *Numeriske Mathematics* Vol. 1, 1959, pp. 269–271.
- [5] R.W. Floyd R.W., Algorithm 97: Shortest paths, *Communications of the ACM*, Vol. 5, 1962, 345.
- [6] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Welsey Publishing Company , 1989.
- [7] J. Holland, *Adaptatin in Natural and Artificial System*, University of Michigan Press, Ann Arbor, MI, 1975.
- [8] C. Hsinghua, G. Premkumar, C. Chu, Genetic algorithms for communications network design—an empirical study of the factors that influence performance, *IEEE Trans. Evol.Comput.* Vol. , No. 3, 2001, pp. 236–249.
- [9] X. Hue,. Genetic Algorithms for Optimization: Bachground and Applications. *Edinburgh Parallel Computing Centre*, Univ. Edinburgh, Scotland, Ver 1.0. , 1997
- [10] <http://www.mathworks.com/access/helpdesk/help/toolbox/gads/>
- [11] D. Whitley, A Genetic Algorithm Tutorial, *Computer Science Department, Colorado State University Fort Collins*, CO 80523
- [12] Ji. Xiaoyu , Models and algorithm for stochastic shortest path problem, *Applied Mathematics and Computation* Vol. 170, 2005, pp. 503–514
- [13] F. Hosseinali, and Ali A. Alesheikh, . Weighting Spatial Information in GIS for Copper Mining Exploration, *American Journal of Applied Sciences*. Vol. 5, No. 9, 2008, pp. 1187-1198.