In The Name Of God

University of Tehran

Faculty of Engineering

Department of Surveying and Geomatics Engineering

Geospatial Information Systems (GIS) Division

Shortest path with genetic algorithms

Instructor : Dr.Samadzadegan

Teacher assistant : MC.s Hasani

Last Name : Ghandehary

First Name : Mehran

Student Number : 810389010

Email : m.ghandhary@yahoo.com

Cell : 09132668898

http: www.mohamadhamzeh.blogfa.com

Date Delivered : 30.10.2010

Abstract

This assignment aims to develop a genetic algorithm to solve a network routing problem. The algorithm has to find the shortest path between the source and destination nodes.
Dijkstra's algorithm is one of popular techniques to solve this problem. The developed genetic algorithm is compared with Dijkstra's algorithm to solve routing problem. Simulation results are carried out for both algorithms using MATLAB.

INTRODUCTION

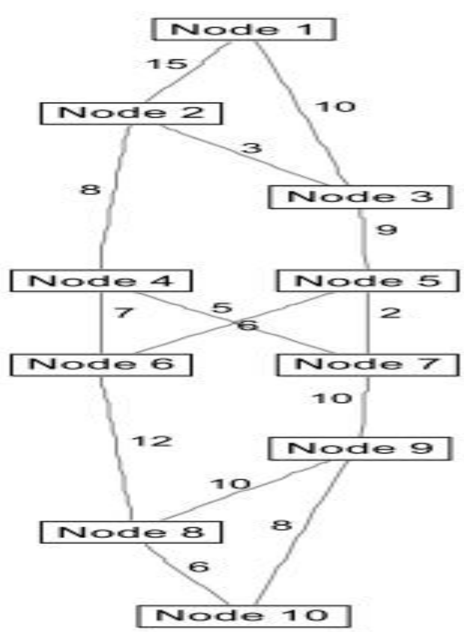
Genetic algorithms (GAs) are global search and optimization techniques modeled from natural selection, genetic and evolution. The GA simulates this process through coding and special operators. A genetic algorithm maintains a population of candidate solutions, where each candidate solution is usually coded as binary string called a chromosome. The best choice of coding has been shown to be a binary coding . A set of chromosomes forms a population, which is evaluated and ranked by fitness evaluation function. The fitness evaluation function play a critical role in GAs because it provides information how good each candidate. The initial population is usually generated at random. The evolution from one generation to the next one involves mainly three steps: fitness evaluation, selection and reproduction .
First, the current population is evaluated using the fitness evolution function and then ranked based on their fitness. A new generation is created with the goal of improving the fitness. Simple GA uses three operators with probabilistic rules: reproduction, crossover and mutation. First selective reproduction is applied to the current population so that the string makes a number of copies proportional to their own fitness. This results in an intermediate population.

Second, GA select "parents" from the current population with a bias that better chromosome are likely to be selected.
This is accomplished by the fitness value or ranking of a chromosome.
Third, GA reproduces "children" (new strings) from selected parents using crossover and/or mutation operators.

Crossover is basically consists in a random exchange of bits between two strings of the intermediate population. Finally, the mutation operator alters randomly some bits of the new strings. This algorithm terminates when an acceptable solution is found, when convergence criteria are met or when a predetermined limit number of iteration is reached.

PROPOSED ALGORITHM

The network under consideration is represented as a connected graph with N nodes. The metric of optimization is the cost of path between the nodes. The total cost is the sum of cost of individual hops. The goal is to find the path with minimum total cost between source node and destination node. Network topology for simulation are like below diagram.

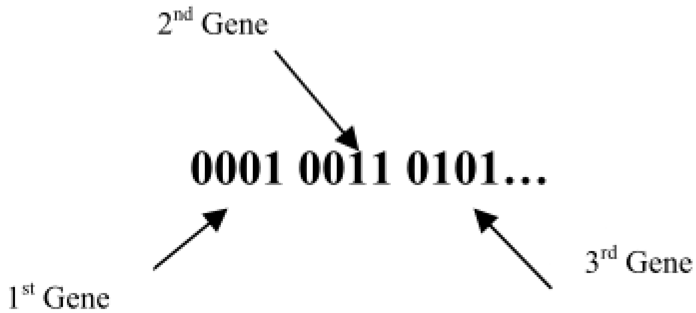


The path is encoded using binary numbers where each gene (node) in a chromosome is encoded by n bits binary. The number of bits (n) has to be sufficient to encode the network nodes.
For example for a network with 10 node we have :

Node	Binary code	Linked nodes
1	0001	2,3
2	0010	3,4
3	0011	5
4	0100	6,7
5	0101	6,7
6	0110	8
7	0111	9
8	1000	10
9	1001	8,10
10	1010	destination

A. Chromosomes and Initialization

A chromosome corresponds to possible solution of the optimization problem. Thus each chromosome represents a path which consists of a set of nodes to complete the feasible solution, as the sequence of nodes with the source node followed by intermediate nodes (via nodes), and the last node indicating the destination, which is the goal. The default maximum chromosome length is equal to the number of nodes times the gene length (4-bit binary code/gene). The network has 10 points where each point is coded in 4 binary bits. That means; the chromosome length is equal to 10X4=40 bits. The chromosome structure is given in Figure



B. Evaluation

The choice of a fitness function is usually very specific to the problem under condition. The evaluation function of a chromosome measures the objective cost function. The cost of a path indicated by the chromosome is used to calculate its fitness. Since the fitness should increase as the cost decreases. Thus, the fitness function (F) of a path is evaluated as defined in equation

$$F = \begin{cases} \dfrac{1}{\sum_{i=1}^{N-1} C_i\,(g_i, g_{i+1})} & ; \quad \text{Feasible path} \\ 0 & ; \quad \text{Infeasible path} \end{cases}$$

Where Ci(gi, gi+1) is the cost between gene gi and adjacent gene gi+1 in the chromosome of N genes (Nodes). The cost between linked nodes is given in Figure 2. If the path is not feasible, its fitness is equal to zero.

C. Operators

The algorithm uses the common two genetic operators: *crossover* and *mutation*. Crossover recombines two 'parent' generation. Two points crossover is used. Both parent paths are divided randomly into three parts respectively and recombined. The middle part of the first path between crossover bit positions and the middle part of the second path are exchanged to produce the new children. The crossover bit positions are selected randomly along the chromosome length between bit positions 5 and 36. These limits are chosen in order to keep the start and destination nodes without change during the crossover process. The mutation process is also applied to flip randomly a bit position in the chromosome (between bit position 5 and 36).

The pseudo-code of the proposed algorithm is given by:

BEGIN

Initialize the start and destination points
Generate randomly the initial population using via nodes in each chromosome
While NOT (convergence condition) DO
Evaluate the fitness for each chromosome in current population using equation (1)
Rank the population using the fitness values
Eliminate the lowest fitness chromosome
Duplicate the highest fitness chromosome
Apply randomly crossover process between current parents using the given probability, while keeping the start and end nodes without change in the population
Apply the mutation process with the given probability
Generate the new population
END
Output the best individual found
END