GA-Based Multiple Route Selection for Car Navigation

Basabi Chakraborty

Faculty of Software and Information Science
Iwate Prefectural University
152-52 Aza Sugo, Takizawamura, Iwate 020-0193, Japan
basabi@soft.iwate-pu.ac.jp

Abstract. Search for optimal route from source to destination is a well known optimization problem and lot of good solutions like Dijkstra algorithm, Bellman-Ford algorithm etc. are available with practical applications. But simultaneous search for multiple semioptimal routes are difficult with the above mentioned solutions as they produce the best one at a time. Genetic Algorithm (GA) based solutions are currently available for simultaneous search of multiple routes. But the problem in finding multiple routes is that the selected routes resemble each other,i,e., partly overlap. In this paper a GA based algorithm with a novel fitness function has been proposed for simultaneous search of multiple routes for car navigation system avoiding overlapping. The simulation of the proposed algorithm and other currently available algorithms have been done by using a portion of real road map. The simulation results demonstrate the effectiveness of the proposed algorithm over other algorithms.

1 Introduction

Car navigation devices are widely used now as an information source for Intelligent Transportation Systems. One of the functionality of a car navigation system is route planning. Given a set of origin-destination pair, there could be many possible routes for a driver. A useful routing system for car navigation should have the capability to support the driver effectively in deciding on an optimum route to his preference. Search for optimal route from one point to another on a weighted graph is a well known problem and has several solutions. There are several search algorithms for the shortest path problem, breadth first search, Dijkstra algorithm, Bellman-Ford algorithm to name a few. Though these algorithms can produce stable solutions in polynomial time, they exhibit high computational complexity specially in changing real time environment. Moreover in case of navigation systems like flight route selection or car navigation, the shortest path may not be the best one from other considerations such as, traffic congestion, environmental problem or simply user's satisfaction and some of the parameters may vary with time. So for efficient car navigation in dynamic environment, we need to specify multiple and separate good choices with rank information. Simultaneous search for multiple short routes is difficult with the

S. Manandhar et al. (Eds.): AACC 2004, LNCS 3285, pp. 76–83, 2004.

above algorithms as they produce one solution (the shortest route) at a time and need to be rerun for alternate solutions which is computationally demanding and do not gurantee the successive shortest paths.

Genetic algorithms [1] are now widely used to solve search problems with applications in practical routing and optimization problems [2]. Some works [3–5] also have been reported for search of multiple routes for navigation systems using GA. But the problem in finding multiple semi optimal routes simultaneously is that the selected routes resemble each other i,e they partly overlap. Inagaki et.al [6] proposed an algorithm in which chromosomes are sequences of integers and each gene represents a node ID selected randomly from the set of nodes connected with the node corresponding to its locus number to minimize the effect of overlapping solutions. But the proposed algorithm requires a large solution space to attain high quality solution due to its inconsistent crossover mechanism. Inoue [7] proposed a method for finding out multiple different (non overlapping) short routes by dividing the road map in multiple areas and putting different weights in each of them so that the selected routes are through different areas of the map. But as their is no direct method for comparing the ovelapping of the selected paths this method is not guranteed to select minimally overlapped multiple shorter paths.

In this work a genetic algorithm has been developed for searching multiple non overlapping routes from starting point to destination point on a road map for use in car navigation system with the proposal of a new fitness function for ranking the probable solutions. The proposed algorithm has been evaluated against above mentioned algorithm using a real road map. In the next section a brief introduction to genetic algorithm and its use for multiple route selection is presented. The following section describes the proposed GA with new fitness function. Simulation experiments and results are presented in section 4. Section 5, the final section contains conclusion and discussion.

2 Genetic Algorithm

Genetic algorithms (GA) are adaptive and robust computational models inspired by genetics and evolution in biology. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to produce new solutions. GA are executed iteratively on a set of coded solutions called population initially randomly drawn from the set of possible solutions with three basic operators namely, selection, crossover and mutation in such a way that better solutions are evolved in each iteration. The goodness of a solution is measured by a problem dependent objective function called fitness function, the design of which is very critical for the success of a GA based algorithm in finding out optimal solution. Genetic algorithms have been applied for shortest path routing, multicast routing or dynamic routing, bandwidth allocation and in several practical optimization problems.

2.1 Multiple Route Selection by GA

GA can be used effectively for searching multiple routes from a real road map with a rank order i.e., shortest, second shortest, 3rd shortest and so on (k shortest path problem). The road map is first converted into a connected graph, considering each road crossing as a node in the graph and all such nodes are numbered. The roads in map are represented by links in the graph. The distance between any two crossings is considered as the weight of the link between the corresponding nodes. The starting point and the destination on the map are defined as the starting node and goal node on the graph. Any possible path from start node to goal node or destination node via other nodes is a possible solution and coded as a chromosome by using node numbers. However looping in the path is generally avoided.

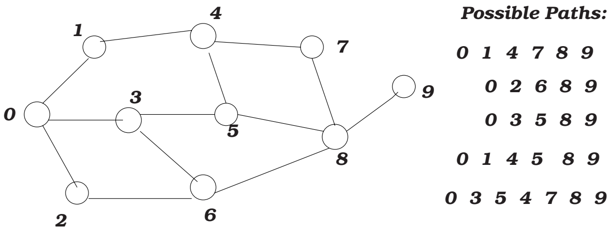

Fig. 1. Graphical Representation of road map and routes

Fig. 1 represents a simple graphical representation of a road map and the possible routes from source node 0 to the destination node 9. The general GA based algorithm for finding out m short routes simultaneously is described as follows [4, 7].

1. Coding the solution space: Population of chromosomes representing solution paths (routes) are generated by genetic coding. Chromosomes are equal length sequence of integers where each gene represents a node number. Genetic coding of the actual path through nodes has been done by changing the gene number in chromosome with the node number in the actual path sequence as shown in Fig. 2. Thus the integer 0 is changed by node number 3, 3 is changed by node number 5 and so on following the node sequence in the path 0 → 3 → 5 → 4 → 7 → 8 → 9. The circles in the coding of the path is to be replaced by randomly taking from the connected nodes to the node represented by the integer, i,e first circle is replaced by 0 or 4, the nodes connected to the node number 1.

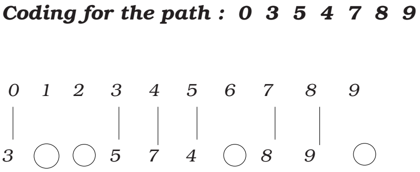

Fig. 2. Genetic Coding of the path

2. Setting fitness function: The fitness of any chromosome is measured by the following function:

$$fitness = \frac{1}{\sum_i^N rlength(i)} \tag{1}$$

   where $rlength(i)$ (the distance of the path between the crossings representing ith and $(i-1)$th nodes respectively) represents the weight of the link joining ith and $(i-1)$th nodes. N represents the total number of nodes in the path from start node to destination node.
3. Initialization of the population: A set of chromosomes are selected randomly from the pool of chromosomes for initial population. Genetic operations are carried out iteratively to generate better and better population until the termination condition is satisfied.
4. Genetic operation:
   (a) Selection: The fitness of individual chromosomes and the fitness of the population are evaluated respectively. Roulette wheel selection rule is used for selecting two parents.
   (b) Cross over and mutation with probability P_c and P_m are applied for generating new population from the selected parents.
   (c) Fitness evaluation :Fitness function is used to evaluate the new population.
   (d) The above steps are repeated until the preset number of iteration is achieved.
5. The individual chromosomes in the final population is evaluated and ranked. The best m chromosomes are taken as the best m solutions.

Now the selected routes by the above procedure may partly overlap. To avoid overlapping current method developed by others is to divide the road map into multiple areas and putting different weights in each of them so that the selected routes are through different areas of the map. To achieve this the fitness function has to be modified as follows.

$$fitness = \frac{1}{\sum_i^N rlength(i)\rho_i} \tag{2}$$

$$\rho_i = \begin{cases} \rho & \text{if } route(i) \in A \\ 1 & \text{otherwise} \end{cases}$$

where $0 < \rho < 1$ is the weight associated to the $route(i)$, the path from node i to node $(i-1)$ passing through the selected area of the map. But as their is no direct method for comparing the overlapping of the selected paths, this method is not guranteed to select minimally overlapped multiple shorter paths.

In the next section the proposed method for selecting non overlapping multiple short paths for car navigation has been presented.

3 Proposed GA-Based Multiple Route Search

In the proposed algorithm for selecting m routes simultaneously group of m routes are considered to be one set of solution and a new fitness function incorporating direct measures for non overlappingness of the routes belonging to a group has designed. The algorithm is as follows:

1. Initial population is formed by a number of groups of m chromosomes representing m solution paths (routes), randomly drawn from the whole solution space. One group of routes is considered one solution.
2. The objective function is designed to calculate fitness of a group as a whole. The fitness $F_r(L_r)$ of any solution path L_r(group member) is calculated as follows:

$$F_r(L_r) = \sum_{i}^{N_r} rlength(i) \tag{3}$$

where N_r is the number of nodes in the path L_r including the start node and goal node and $rlength(i)$ is the weight of the link connecting i th and $(i-1)$th node in the path.
The fitness of the group is defined from three factors as

$$F(G) = F_1(G) + F_2(G) + F_3(G) \tag{4}$$

$F_1(G)$ is designed for ensuring non overlapping of the individual solution paths and is defined as:

$$F_1(G) = GF(L) + X \tag{5}$$

where GF(L) represents the average of the fitness of the individual paths belonging to the group i,e

$$GF(L) = \frac{1}{m}\sum_{r=1}^{m} F_r(L_r) \tag{6}$$

and $X = \sum_r \sum_i x_{ri}$ where x_{ri} represents the number of nodes with in a radius R of the ith node in the rth path of a group that is taken by any of the r th path of the same group. R is problem dependent and should be chosen by trial and error. The term X is a penalty term to ensure seperation of the individual path s in a group.

$$F_2(G) = GF(L) \times \frac{2z}{p_1} \tag{7}$$

$$F_3(G) = \begin{cases} GF(L) \times \frac{3z}{p_2} & \text{if } p_2 > C \\ \frac{GF(L)}{2} & \text{if } p_2 \leq C \end{cases} \tag{8}$$

where z represents the average number of nodes of m paths i,e $z = \sum_r N_r$, p_1 and p_2 represents the average of total number of nodes in any group of $\frac{m}{2}$ paths and total number of nodes in all the paths of the group respectively. C is a constant depending on the total number nodes on the graph and should be chosen by trial and error.
Equation. 7 and 8 represent the penalty terms due to the number of nodes shared by the routes in a group. Here the smaller value of the objective function corresponds to better fitness.

3. For crossover operation, the parent groups are selected according to Roulette wheel selection procedure using group fitness. Then the actual parents are selected, one each from two parent groups by the same procedure using individual fitness values. Then crossover position is selected randomly and crossover is done with a predefined probability of crossover P_c.
4. Mutation is done randomly with a probability P_m at a random position to generate different solution path inside a group.
5. The new groups are formed by changing the group members of the selected parent groups with a probability of P_g. The groups are evaluated by group evaluation function and better groups (ranked via evaluation function) are selected for next generation of population.
6. The process is repeated a pre-defined number of times to get the best solution group.

Now due to randomness in the process, there is a small possibility of generating non-existent solution paths. Those solution are discarded eventually as they do not contribute to the fitness function.

4 Simulation and Results

Simulation experiments have been done using a portion of the real road map Fig. 3 by applying Dijkstra's method, proposed algorithm and Inagaki's algorithm. In all the cases 4 alternate paths are selected. The road map is first converted into a graph with 130 nodes. The number of candidate paths from start(X) to destination node(Y) are found to be 100 (no looping is considered). First Dijkustra's method has been used to find out shortest, 2nd shortest, 3rd shortest and 4th shortest routes. Secondly Inagaki's method has been used to devide the road map into 4 parts and using different values of ρ of Eq. 2 for selecting 4 non overlapping paths from the different a reas of the map. Finally proposed algorithm has been used to select the optimal group of 4 short paths with minimum overlapping. The selected genetic parameters are represented in Table 1. In both the methods the parameters P_c and P_m have been changed to several values and the optimum values are noted in the table.

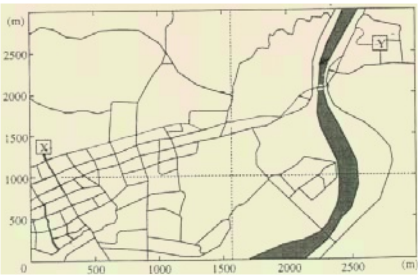

Fig. 3. A portion of road map used for simulation

Table 1. Setting of parameters of GA

Method	Population size	No. of iteration	P_c	P_m	P_g
Inagaki	100	40	0.9	0.5	
Proposed	150	40	0.8	0.6	0.2

Table 2. Comparative Performance of different Algorithms

Algorithm	Time taken	Number of overlapping nodes	Average weight of the path
Dijkstra	.014s	15	243
Inagaki	.182s	5	275
Proposed	.177s	2	257

4.1 Results and Discussion

Table 2. shows the comparative results of the different algorithms. The time taken is the average run time of the same computer used for simulation of the three algorithms. Average weight of the path in the 4th column is calculated from the weight of the links between nodes. The actual distance in the road map is converted to weight between nodes by taking 100m as 1 unit of weight. Lesser average weight represents shorter paths. Dijkstra algorithm takes much shorter time compared to other algorithms for finding out the shortest route and it also is able to find out better paths in terms of distance. But successive short routes are highly overlapped. Both Inagaki's method and the proposed algorithm take longer time than Dijkstra's algorithm but alternate routes can be found out simultaneously. The proposed GA is found to be better than Inagaki's method as only 2 nodes are shared by the individual paths in the alternate routes compared to 5 nodes shared in alternate routes selected by Inagaki's method. The average run time and the weight of the path in both the methods are nearly equal, proposed method being slightly better. The average weight of the selected path is also close to the average weight of the paths found out by Dijkustra algorithm. Fig. 4 and Fig. 5 represents the simulation results on the road map by Inagaki's method and the proposed method respectively.

5 Conclusion

In this work a Genetic Algorithm based solution technique for finding out m routes simultaneously has been proposed. Simultaneous multiple route selection is difficult by popular optimization technique like Dijkstra algorithm. Currently available GA based algorithm can produce multiple routes simultaneously but selected routes resemble each other. In this work a new GA based algorithm is developed for finding out multiple routes simultaneously with minimal overlapping by grouping m routes as one set of solution and designing fitness functions in such a way that it penalizes the function for overlapping. Simulation experiments on a piece of real road map demonstrates the efficiency of the algorithm over other algorithms in finding out nonoverlapping multiple routes. The usefulness of the proposed algorithm can be better understood in the problem of finding out multiple routes in dynamic environment. At present simulations are carried out for finding out multiple routes dynimically which I hope to report in near future.

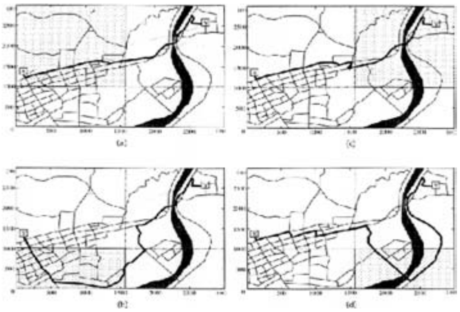

Fig. 4. Simulation by Inagaki's method

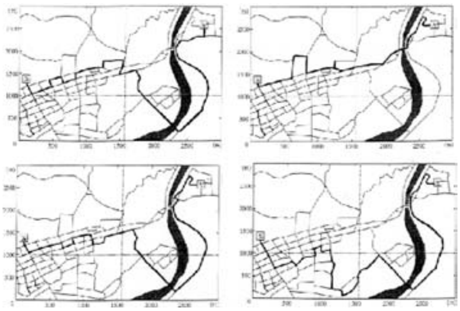

Fig. 5. Simulation by Proposed method

References

1. David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley, 1989.
2. Chang, W. A., "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations", IEEE Trans. on Evolutionary Computations, Vol. 6, no. 6,p p. 566-579, December 2002.
3. Kanoh, H. and Nakamura, T.,*Knowledge based Genetic Algorithm for Dynamic Route Selection*, in Proceedings of International Conference on Knowledge based Intelligent engineering Systems and Allied Technologies, pp. 616–619, 2000.
4. Inagaki, J. et, al., *A genetic algorithm for determining multiple routes and its applications* in Proc. IEEE Int. Symp. Circuits and Systems, pp. 137–14 0, 1999.
5. Tanaka, M. et,al., *A multi-Route Planning for Aircraft Using Genetic Algorithms* in Trans of IEICE, J85 D-I, No. 8, pp. 767–775, 2002. (in Japanese)
6. Inagaki, J. et, al.,*A method of Determining Various Solutions for Routing Application with a Genetic Algorithm*, in Trans of IEICE, J82-D-I, No.8. pp.11 02-1111, August 2002.(in Japanese)
7. Inoue, Y., *Exploration Method of Various Routes with Genetic Algorithm*, Master's Thesis, Information System Engineering, Kochi Institute of Technology, 2001.(in Japanese)