Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems

Chu-Hsing Lin [1], Jui-Ling Yu [2], Jung-Chun Liu [1], Wei-Shen Lai [3], Chia-Han Ho [1]
[1]*Department of Computer Science and Information Engineering,*
Tunghai University, Taiwan
[2]*Department of Applied Mathematics, Providence University, Taiwan*
[3]*Department of Information Management, Chienkuo Technology University, Taiwan*
[1]*{ jcliu, chlin, g95280077 }@thu.edu.tw*
[2]*jlyu@pu.edu.tw*
[3]*WeiShenLai@gmail.com*

Abstract

The route guidance system, which provides driving advice based on traffic information about an origin and a destination, has become very popular along with the advancement of handheld devices and the global position system. Since the accuracy and efficiency of route guidance depend on the accuracy of the traffic conditions, the route guidance system needs to include more variables in calculation, such as real time traffic flows and allowable vehicle speeds. As variables considered by the route guidance system increase, the cost to compute multiplies. Since handheld devices have limited resources, it is not feasible to use them to compute the exact optimal solutions in real time systems by some well-known algorithm, such as the Dijkstra's algorithm, which is usually used to find the shortest path with a map of reasonable numbers of vertices.

To solve this problem, we propose to use the genetic algorithm to alleviate the rising computational cost. We use the genetic algorithm to find the shortest time in driving with diverse scenarios of real traffic conditions and varying vehicle speeds. The effectiveness of the genetic algorithm is clearly demonstrated when applied on a real map of modern city with very large vertex numbers.

1. Introduction

It has become a common practice to use PDAs as route guidance [1-2]. Usually, the shortest path is provided by route guidance systems to advise traffic users how to reach the destination from the origin. Depending on the real time situations on the suggested route, including traffics congestions, road conditions, speed limits, and behaviors of the drivers, the traveling time might be saved or not.

The shortest path route provided by route guidance systems is not necessarily the optimal path since it is computed mainly based on the shortest distance, but other variables, for example, traffic congestions, driving speed limits might have significant effects and need to be included in the computation. The incurred computational cost by taking many traffic variables into consideration might consume too much time and resources of handheld devices. In general, the computing power and memory of handheld devices are limited. One method to solve this dilemma is to do all the computations in a host server, but once the communication between handheld devices and the host server is disrupted, so is the route guidance application. The alternative method is to use some algorithm that can provide approximate answers with relatively lower computational cost. In this paper, we propose to use genetic algorithm to find the optimal path. Because we take the allowable vehicle speeds into consideration, the shortest time instead of the shortest path is used for route guidance.

The rest of this paper is organized as follows. In Section 2, the background information is given. In Section 3, the shortest driving time problem is defined. Experimental settings and simulation results will be shown in Section 4. Section 5 concludes this paper.

2. Background

2.1. Shortest Path Problem

In graph theory, the shortest path problem can be generalized as the single-source shortest path problem, in which the shortest path from a source vertex and all other vertices in the graph is found. The well-known algorithms for this include the Dijkstra's algorithm and Bellman-Ford algorithm.

Dijkstra's algorithm solves the shortest paths problem when the edge weights are nonnegative, which are applicable to the conditions of route guidance systems. We will use the Dijkstra's algorithm to find the shortest driving time when the vertex number is not too large. The running time of implementing the Dijkstra's algorithm by storing vertices in an ordinary linked list is of $O(|V|^2+|E|)$, where |V| is the number of vertices and |E| is the number of edges. The running time might be unacceptable when the vertex numbers become too large. In this situation, other alternative methods can be used to find approximate solutions. The genetic algorithm is one of them.

2.2. Genetic Algorithm

The genetic algorithm is a search technique to find exact or approximate solutions [3-10]. It has its origin from the theory of evolution in nature. In 1975, John H. Holland worked out the genetic algorithm in a book called "Adaptation in Natural and Artificial Systems". The genetic algorithm has on a wide scope of applications, including genome biology, economics, game theory, pattern recognition, neural networks, fuzzy theory, etc.

Figure 1 show Steps of the genetic algorithm, which are described as follows:

- Initialization of population
- Choice of fitness function and evaluation of fitness of each individual in the population
- Selection of best ranked part to reproduce
- Breeding new generation by crossover or mutation and giving birth to offspring
- Replacement of worst ranked part of population with the offspring
- Repeating this generational process until a termination condition has been reached.

Chang et al. proposed a genetic algorithm to solve for the shortest path problem in 2002[6]. The problem is described as follows:

The multi-hop network can be defined as a directed graph **G** = (**N**, **A**), where **N** denotes the set of n nodes (vertices) and **A** denotes the set of edges. The cost matrix is denoted as **C** = [$\mathbf{C}_{ij}$], where $\mathbf{C}_{ij}$ associates the cost from node i to node j. The origin is **S** and the destination is **D**. The link indicator, I_{ij}, indicates whether a route exists between node i and node j. If there is a route, then $I_{ij} = 1$, otherwise, $I_{ij} = 0$.

They show that by using the genetic algorithm, after nine generational processes, the optimal solution is found for a network with 20 nodes. The genetic algorithm converges fast and it needs just a few generational processes to find the optimal solution. As the number of nodes increases more than 20, the computing time by adopting the genetic algorithm is less than that by adopting the Dijkstra's Algorithm. So, it is more feasible to use the genetic algorithm to find the optimal path in a complicated real life map with thousands of nodes on handheld devices with constrained resource.

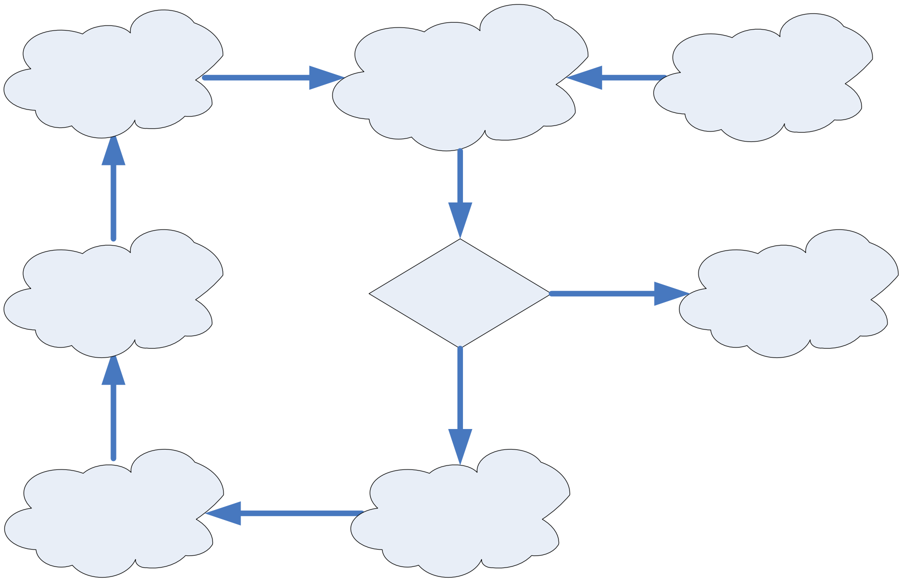

Figure 1. Schematic genetic algorithm

2.3. Intelligent Transportation System (ITS)

The Intelligent Transportation System (ITS) provides drivers with route guidance and map information, and it can be used to support the intelligent driving system by sensing technologies or by the driving information feedback from the users [11][12][15]. As shown in Figure 2, the intelligent driving system guides the driver to the destination by updating the optimal traveling route according to real time traffic information.

Replacement

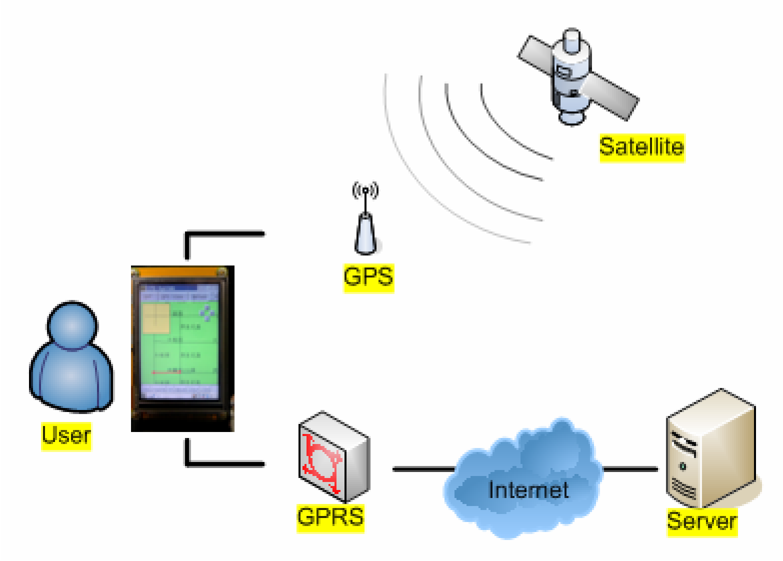

Figure 2. The intelligent driving system

Mutation

3. Shortest Driving Time

The Shortest Driving Time (SDT) problem can be formulated into an integer programming model. The notation list of its mathematical model is given as follows:

Problem parameters:

N	set of all nodes
A	set of all links
S	source node, $\in N$
D	destination node, $\in N$
i,j	index of node, i,j , $\in N$
$<i,j>$	node i to node j, directional
E_{ij}	link node i to node j
d_{ij}	distance of node i to node j
v_{ij}	velocity of node i to node j

Problem decision variables:

T_{ij}	cost time of node i to node j, $\in R^+$
U_{ij}	binary, 1 if the link from node i to node j exists in the routing path, 0 otherwise
t	total drive time, $\in R^+$

3.1. Shortest Driving Time

In the traditional shortest path problem, the cost between node i and node j is the distance d_{ij}. By including the speed limits, v_{ij}, which is the highest driving speed from node i to node j, the cost of time can be denoted as $T_{ij} = d_{ij} / v_{ij}$. By replacing the cost of distance with the cost of time, we can use Dijkstra's algorithm to find the shortest driving time as follows:

$$\text{Minimize } t = \sum_{i=S}^{D} \sum_{j=S}^{D} T_{ij} U_{ij} \quad \text{subject to}$$

$$\sum_{\substack{j=S \\ j \neq i}}^{D} U_{ij} - \sum_{\substack{j=S \\ j \neq i}}^{D} U_{ji} = \begin{cases} 1, & \text{if } i = S \\ -1, & \text{if } i = D \\ 0, & \text{otherwise} \end{cases}$$

$$\sum_{\substack{j=S \\ j \neq i}}^{D} U_{ij} \begin{cases} \leq 1, & \text{if } i \neq D \\ = 0, & \text{if } i = D \end{cases}$$

$$U_{ij} \in \{0,1\}, \text{ for all } i.$$

$$T_{ij} = d_{ij} / v_{ij}$$

To find the shortest driving time, we also use the following steps based on the genetic algorithm proposed by Chang et al.: [6][13][14]

Step 1. Genetic Representation:

Chromosomes with various lengths are used. The maximum length is N, which can be set first. Chromosomes start at S and end at D. For example, we can represent a route as chromosome ($S \rightarrow N_1 \rightarrow N_2 \rightarrow \ldots \rightarrow N_{k-1} \rightarrow N_k \rightarrow D$).

Step 2. Population Initialization:

Chromosomes of the first generation are produced by considering the population size and the procedure to initialize the population. The heuristic initialization or random initialization is adopted.

Step 3. Fitness Function:

The fitness function is defined as

$$f = \frac{1}{\sum d_{ij} / v_{ij}}$$

Where f represents the fitness value of the chromosome, d_{ij}, the distance of node i to node j, and v_{ij}, the allowable velocity on the edge of node i to node j.

Step 4. Selection:

Pair-wise tournament selection without replacement is used.

Step 5. Crossover:

The crossover is done by randomly finding the crossover point in two chromosomes.

Step 6. Mutation:

The probabilities of mutation are set at a range of 5% to 10%. It is used to produce new route.

Step 7. Termination:

Repeating the generational processes until a termination condition is achieved.

4. Experimental Settings and Results

We used ARM 9 S3C2410 embedded system as the portable device, and a desktop PC as the ITS server, which provide allowable driving speeds according to traffic conditions. We performed two sets of experiments, one on a virtual square matrix map and one on a real city map.

We adopted the Dijkstra’s algorithm and the genetic algorithm to compute the shortest driving time. But as the number of nodes increased, the memory space required by the Dijkstra’s algorithm went beyond the limited memory on the embedded system. So, we only list the experimental results by adopting the genetic algorithm.

4.1. Matrix Map Case

The virtual maps of square matrix had sizes of 4 x 4, 8 x 8, 16 x 16, and 32 x 32.

As shown in Figure 3, the origin was at the upper left corner and the destination was at the lower right corner. The distances between nodes were fixed at 20. The speed limits were varied from 2 to 10. The probability of mutation was set to be 8 %, and the limit of generational processes was set at 30. All experiments are done 1,000 times.

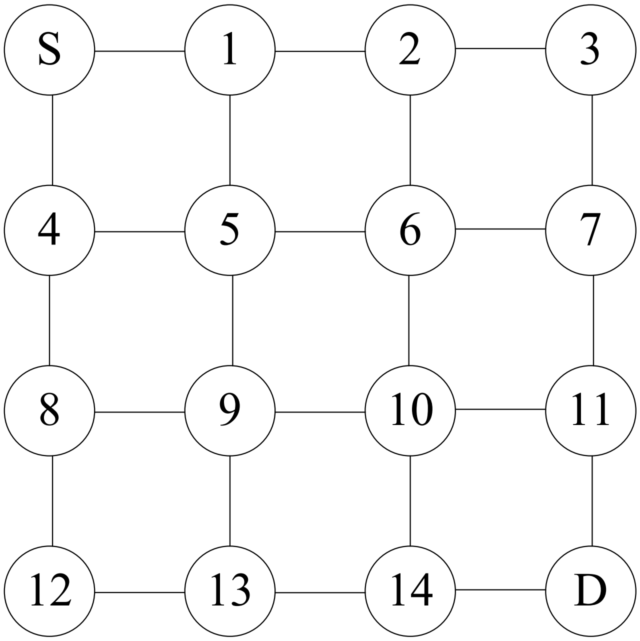

Figure 3. Virtual map of Square matrix.

Table 1 gives the average number of generational processes to find the shortest driving time. It shows that the genetic algorithm converges very fast, less than 20 generations even when the number of nodes growing to 1024. Also for a complex map with thousands of nodes, the average number of generational processes for the genetic algorithm beginning to converge is still small enough for real time applications.

Table 1. Average generations to find optimum path

		Number of nodes			
		16	64	256	1024
Number of chromosome	20	3.31	5.19	9.38	13.97
	40	5.06	7.29	9.95	17.33
	60	6.33	8.18	10.59	18.21
	80	6.58	9.22	11.16	18.30
	100	6.60	10.27	12.04	18.38
	120	6.64	10.46	12.44	18.48
	140	6.71	11.64	13.27	18.77
	160	6.89	11.92	14.08	19.05

Approximately optimal routes are found when the genetic algorithm begins to converge. Table 2 gives the average of difference of the approximate route and the exact route. It shows that when the number of nodes is small, 16 or 64 nodes in this example, the approximate solutions are very close to the exact solution, which is less than 5%. Also it helps by increasing the number of chromosomes. But, when the number of nodes is large, 1024 nodes in this example, the difference of the approximate and exact solutions grows to about 50%.

Table 2. Average difference of approximate and exact routes (in %)

		Number of nodes			
		16	64	256	1024
Number of chromosome	40	12%	34%	47%	56%
	80	3.6%	23%	44%	52%
	160	1.2%	18%	39%	51%
	320	<1%	13%	36%	51%
	640	<1%	7%	31%	49%
	1280	<1%	5%	26%	48%

4.2. Real Map Case

We used the city map of Taichung, Taiwan in real map experiments. It has 8039 nodes and five levels of road according to speed limits. The highways are classified as LV5 and the slowest speed limit roads are classified as LV1. Information of the speed limits are provided from the ITS server in real time.

The experimental results are listed in Table 3. It shows that the genetic algorithm converges relatively fast even with large amount of data, so it is feasible to use genetic algorithm on handheld devices to find approximately optimal paths. It also shows that the number of chromosomes is critical when the number of nodes is very large. By increasing the number of chromosomes from 40 to 1024, the difference between approximate and exact solutions is cut from 223% to 48%.

Table 3. Results on real map by genetic algorithm

		Avg. number of generation	Avg. difference
Number of chromosome	40	22	223%
	80	26	180%
	160	28	103%
	320	29	86%
	640	31	68%
	1280	33	48%

5. Conclusions

The route guidance system has very wide applications. An ideal route guidance system not only provides shortest path information based on the static information of the distance of the origin and the destination, but also offers dynamic solutions based on the ITS information collected from the sensors and the users. The shortest driving time approach, which can be computed by genetic algorithm on the handheld device, is feasible to be used in the route guidance system.

The computation power and memory space on the handheld devices are limited. It is not practical to search for the exact optimal solution on them with massive amounts of traffic data. The genetic algorithm that can be used to search approximate solutions is able to effectively find the approximately optimal path in complicated situations when many variables affecting the traffic are included in calculation.

6. Acknowledgement

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 95-2218-E-001-001, NSC95-2218-E-011-015, NSC 97-2115-M-126-003, and NSC 97-2221-E-029 -023.

7. References

[1] S. Feng and C. L. Law, "Assisted GPS and its impact on navigation in intelligent transportation systems", in Proc. Intelligent Transportation Systems, 2002, pp. 926-931

[2] V. Chang, "Web Service Testing and Usability for Mobile Learning", in Networking, International Conference on Systems and International Conference on Mobile Communications and learning Technologies, 2006, pp. 221-227

[3] S. Xu and J. C. Bean, "A Genetic Algorithm for Scheduling Parallel Non-identical Bath Processing Machines", in Computational Intelligence in Scheduling, 2007, pp. 143-150.

[4] M. Munemoto, Y. Takai, and Y. Sato, "A migration scheme for the genetic adaptive routing algorithm", in Proc. IEEE Int. Symp. Circuits and Systems, 1999, pp. 137-140.

[5] J. Inagaki, M. Haseyama, and H. Kitajima, "A genetic algorithm for determining multiple routes and its applications", in Proc. IEEE International Symposium on Circuits and Systems, 1999, vol. 6, pp. 137-140

[6] C. W. Ahn and R.S. Ramakrishnal, "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations", in Evolutionary Computation, IEEE Transactions, 2002, pp. 566-579.

[7] M. Gen and L. Lin, "A New Approach for Shortest Path Routing Problem by Random Key-based GA", in Proc. Genetic and evolutionary computation, July 2006, pp. 1411-1412

[8] Z.J. Li, J. J. Garcia-Luna-Aceves, "Quality of service in wireline networks: A distributed approach for multi-constrained path selection and routing optimization", 2006

[9] Q.F. Zhang and Y.-W. Leung, "An orthogonal genetic algorithm for multimedia multicast routing", in Evolutionary Computation, IEEE Transactions, 1999, pp. 53-62

[10] Y. Chengjun and J. Vuanwei, "Solving the Problem of the Link Optimizing and Delay-constrained Multicast Routing Based on GA", in Control Conference Chinese, 2006, pp. 1783-1786.

[11] H. Housroum, T. Hsu, R. Dupas and G. Goncalves, "A hybrid GA approach for solving the Dynamic Vehicle Routing Problem with Time Windows", in Information and Communication Technologies, April 2006, pp. 787-792

[12] H. Kanoh and N. Nakamura, "Route Guidance with Unspecified Staging Posts Using Genetic Algorithm for Car Navigation Systems", in Proc. Intelligent Transportation Systems, 2000, pp. 119-124

[13] Y.-L. Xu, M.-H. Lim and M.-J. Er, "Investigation on Genetic Representations for Vehicle Routing Problem", in Systems, Man and Cybernetics, IEEE International Conference, 2005, pp. 3083-3088

[14] C. Mattiussi and D. Floreano, "Analog Genetic Encoding for the Evolution of Circuits and Networks", in Evolutionary Computation, IEEE Transactions, 2007, pp. 596-607

[15] Y.-U. Chung and D.-H. Cho, "Enhanced soft-handoff scheme for real-time streaming services in intelligent transportation systems based on CDMA", in Intelligent Transportation Systems, IEEE Transactions, 2006, pp. 147-155

Authors

Chu-Hsing Lin received both of his B.S. and M.S. degrees in applied mathematics from National Tsing Hua University and National Chung Hsing University, respectively. In 1991, he received his Ph.D. degree in computer sciences from National Tsing Hua University, Taiwan. Since then he has been a faculty of the Department of Computer Science and Information Engineering, Tunghai University. Dr. Lin is currently a professor and the chair of the CSIE department of Tunghai University. From 1995 to 1999, he has ever been the Director of the Computer Center of Tunghai. He has also been one of the Board Directors of the Chinese Information Security Association (CCISA) from 2001 till now. Dr. Lin has published over 50 papers in academic journals and international conferences. He has received over twenty project grants from government departments and private companies in recent years. In 2006, he was awarded the Outstanding Instructor Award of Master & Ph.D. Thesis by the IICM (Institute of Information & Computing Machinery). He was the winner of the 1991 Acer Long-Term Award for Ph.D. Dissertation. His current research interests include multimedia information security, wireless ad hoc networks, embedded systems applications.

Jui-Ling Yu received her Ph.D. degree in Department of Mathematics from Michigan State University, USA, in 2005. She currently is an assistant professor in the Department of Applied Mathematics at the Providence University, Taichung, Taiwan, since 2005. Her current research interests include numerical methods, industrial mathematics, and mathematical biology.

Jung-Chun Liu received his B.S. degree in electrical engineering from National Taiwan University in 1990. He received M.S. and Ph.D. degrees from the Electrical and Computer Science Engineering Department at University of Texas at Austin, in 1996 and 2004, respectively. He is an assistant professor in the Computer Science Department at the Tunghai University, Taiwan. His research interests include digital signal processing, VLSI design, RF and microwave engineering, watermarking, embedded systems, and computer networks.

Wei-Shen Lai received his B.S. and M.S. degrees in computer science and information engineering from Feng Chia University and National Chiao Tung University, respectively. In 2002, he received his Ph.D. degree in computer science and information engineering from National Chiao Tung University, Taiwan. In 2004, he has been a faculty of the Department of Information Management, Chienkuo Technology University. His current research interests include network security and cryptography.

Chia-Ren Li received both of his B.S. and M.S. degrees in computer science and information engineering from Tunghai University in 2005 and 2008, respectively. Under the instruction of his adviser Professor Chu-Hsing Lin, he has published two international conference papers. The topics of his research interests include security of firewall, mobile networks, routing protocols, embedded system, and routing guidance. His current research focus on the analysis and improvement of the routing algorithm for embedded systems applications.