■ 論 文 ■

Using Evolution Program to Develop Effective Search Method for Alternative Routes

진화 프로그램을 이용한 효율적인 대체경로 탐색방법 연구

CHOI, Gyoo Seok
(Professor, Dept. of Computer Science, Chungwoon University)

SEO, Ki Sung
(Professor, Dept. of Electronic Engineering , Seokyeong University)

PARK, Jong Jin
(Professor, Dept. of Internet & Computer, Chungwoon University)

목 차

Ⅰ. Introduction
Ⅱ. Route finding problem
  1. Road network
  2. Alternative paths
  3. Existing approaches
Ⅲ. Validity index for alternative paths
Ⅳ. EP based Route finding
  1. Evolution program
  2. Representation and generation process of a path individual
  3. Genetic operator
Ⅴ. Computational experiments
Ⅵ. Conclusions
References

Key Words : alternative path, validity index, evolution program, genetic operators, route finding

요 약

도시 내 교통혼잡이 증가됨에 따라 최단경로 탐색방법뿐만이 아니라 동일 목적지까지의 여러 가지 경로(준최단경로)를 제시해 줌으로써 교통량을 효과적으로 분산시킬 수 있는 대체경로 탐색기법에 대한 관심이 고조되고 있다. 본 논문에서는 대체 경로의 유효성을 평가하는 성능지표를 제안하고, 복수개의 우수해 탐색에 유리한 진화 프로그램에 기초한 효과적인 대체경로 탐색기법을 제시한다. 기존 방법(k-th 최단경로 방법)의 문제점이었던 대체 경로들간의 유사성이 제안된 방법에서는 해결된다. 가상 도로망을 통한 컴퓨터 시뮬레이션의 결과로서 제안된 방법이 기존 방법보다 교통량 분산(경로들간의 상이성)측면에서 훨씬 더 우수함을 확인하였다.

본 연구는 청운대학교 교내연구비 지원에 의해서 수행되었음.

Ⅰ. Introduction

As the traffic congestion in a metropolitan area such as Seoul has been serious due to an increasing number of vehicles beyond road capacity, various types of ITS project is being developed. The route guidance, an essential part of ATIS(Advanced Traveler Information Systems) and ATMS(Advanced Traffic Management Systems) in ITS, is regarded as an effective strategy for relieving road congestion and better utilizing existing road networks.[1] The shortest path problem has been researched over the last thirty years,[3)-5)] and is given a great deal of weight in the route guidance system. But in the aspect of traffic management, it would rather be desirable to present different routes than a single shortest route to drivers whose travel routes (Origin-Destination pair) are similar, so that it may elevate the transport efficiency of overall road network by decentralizing the entire traffic volume. Therefore, for efficient route guidance, several alternative paths should be provided in addition to the shortest one. It is often required to provide multiple paths to disperse the traffic volume and to satisfy users' preferences such as easiness of driving, beautiful surrounding view of a road, a toll-free road, etc. Algorithms for finding multiple paths from the given origin and destination —effective K-paths — have been studied, and recently applied in finding realistic alternative paths.[6)-8)]

However, the results may not be satisfactory. That is, most of paths adjacent to the shortest path are likely to be selected using the method since apart paths are more costly. To put it differently, there exists a network in which many links of the alternative paths are overlapped with those of the shortest path until a large k is reached. This is a fundamental problem of the k-th shortest path algorithm. This method fails to provide various alternatives, and thus can not satisfy the various users preferences and the spreading of traffic volume. For instance, users may sacrifice flow time to satisfy their own preferences such as easiness of driving or good scenery.

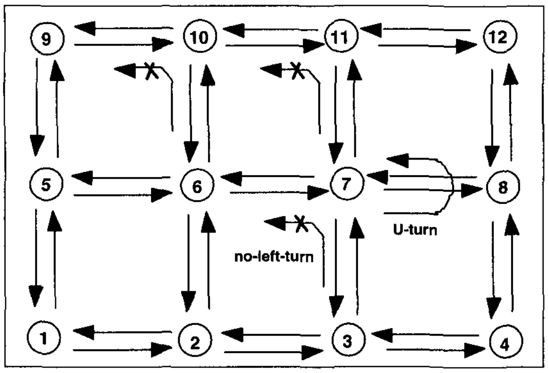
This paper presents a evolution program based search technique that satisfies the purposes mentioned above. We implemented efficient genetic operators — path crossover and path mutation for path processing. The suggested method can provide multiple alternatives which are nearly optimal. Similarities among the paths are reduced due to the evolution program's ability to search multiple solutions, being provided well devised fitness functions for the time costs and spread of paths. This paper also suggests effective criteria which can be used to evaluate the effectiveness of several route alternatives. In addition to the existing two criteria, the ratio of common nodes and the ratio of common links, a new criterion, being the area surrounded by the paths, is used. Finally, the performance of the suggested technique is evaluated with the virtual road network model by computer simulation.

Ⅱ. Route finding problem

1. Road network

A road network is represented by a graph G=(N, L), consisting of a set of nodes $N=\{n_1, n_2, \ldots, n_m\}$ and a set of links $L=\{l_1, l_2, \ldots, l_n\}$. A link lk in L represents an ordered pair of nodes (i,j). Let o and d be two given nodes of (N, L). The chain from o to d in (N, L) is a sequence of nodes and links, of the form $\{o=n_p, l_p, n_{p+1}, l_{p+1}, \ldots, l_{q-1}, n_q=d\}$, and is called a path. Moreover, when a path has the smallest link cost among all the paths, it is said to be the shortest path.

A typical city road network is grid shaped as shown in ⟨Figure 1⟩. There are some constraints in real road networks such as U-turns, P-turns, and no-left-turns. For example, paths including loops can also be selected as shortest paths. As shown in ⟨Figure 1⟩, with origin 3 and desti-



〈Figure 1〉 Representation of road network

nation 9, a path 3-7-8-7-6-5-9 including a U-turn can be selected as the shortest path.

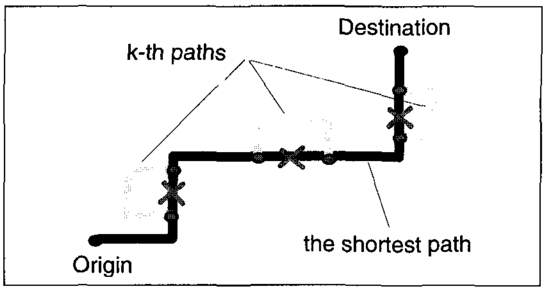
2. Alternative paths

Several paths may exist that have costs equal to or more than that of the shortest path. These are said to be alternative paths. There are some reasons that it is necessary to generate alternative paths in addition to the shortest one. First, from the viewpoint of traffic management center(controls traffic volume and provides traffic information for certain area), providing multiple alternatives for drivers with similar origins and destinations would disperse the traffic volume and thus increase the route efficiency.

Second, drivers may have various criteria(i.e. time, distance, congestion degree, toll-fees, difficulty, scenery) on account of specific objectives. They can choose personally optimal route among the multiple paths on their preferences. For example, a driver may be satisfied with a route which is not the shortest path but has a quiet and beautiful scenery. Therefore, it is required to provide multiple alternative paths satisfying the purposes at costs not significantly greater than the cost of the shortest path(nearly optimal path).

3. Existing approaches

In this paper, Shier's k-th shortest path algorithm[8] was chosen in order to compare it with EP's. The algorithm is as follows.



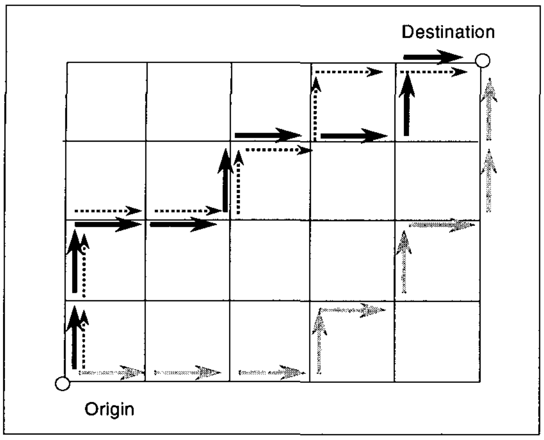
〈Figure 2〉 Example for similar k-th shortest paths with the shortest one

```
Procedure k-th shortest path
begin
   Initialize k-vector of path length
   Put all nodes on the list
   (all components are made temporary)
   and set i = 1
   while (this list is not empty) do
   begin
      Find the smallest temp component TMP
         for node i
      For each node j adjacent from i
         insert if possible the value TMP + c(i,j)
         into the k-vector for node j
      Make permanent the component TMP
         for node i
      Remove node i from the list
         if there are no more temp components
   end
end
```

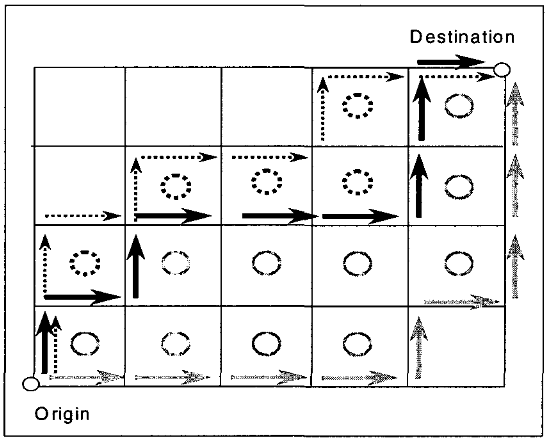
As shown in 〈Figure 2〉, it is highly probable that, except for some links, k paths similar to the shortest path would be selected in this method.

Ⅲ. Validity index for alternative paths

In 〈Figure 3〉, the dark arrows represent the shortest path and the dotted and grey arrows represent the alternative paths. As shown, alternative paths may or may not be overlapped with the shortest path. It means that diverse kinds of paths exist; fully different(represent by the grey



〈Figure 3〉 Several kinds of alternative paths



〈Figure 4〉 The area surrounded by paths

arrows), partially overlapped or mostly overlapped.

The more overlapped with the shortest path, the less meaningful the alternative paths become in satisfying users' preferences or in spreading the traffic volume.

The number of overlapped links or nodes has been used to evaluate the differences between the shortest path and the alternatives. However a new index is required to more effectively differentiate the validity of the alternatives.

If the purpose is to spread the traffic volume, crossed or neighboring alternatives may not be effective even though they are different from the shortest path according to the above criteria.

As shown in 〈Figure 4〉, the bigger the area surrounded by the shortest path and the alternatives, the more apart the paths and thus, the more dispersed the traffic.

Considering all these, we used the following three indices for validity.

CNR : common nodes ratio(number of common nodes /total number of nodes for the shortest path)

CLR : common links ratio(number of common links/ total number of links for the shortest path)

ASP : Area surrounded by paths

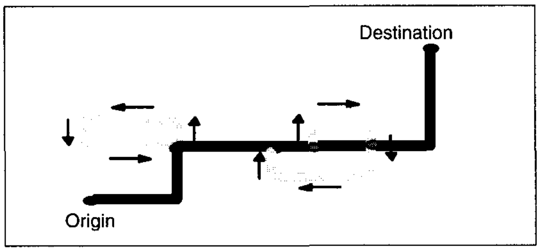
Ⅳ. EP based Route finding

1. Evolution program

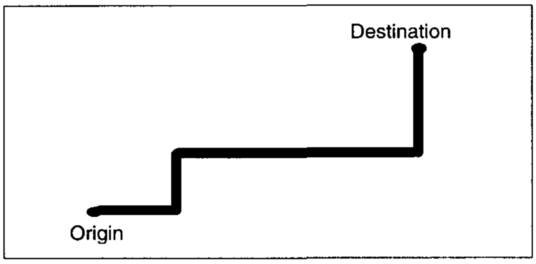
Nowadays, Evolution Program(EP), as a sort of Genetic Algorithm(GA), is effectively applied to the search of a global optimal solution in diverse complex areas which are multi-variable and non-linear, because it is difficult to find an optimal solution in problems with multi-variable.[9)-11)] EP finds a global optimal solution to the complex problem through evolution of individuals(potential solutions) during many generations by genetic method. EP approach can provide multiple alternatives at one trial, because it is based on the concept of evolution of path individuals and presents multiple superior ones in the final generation. Thus, we think, EP approach can be effectively applied in the problem which finds multiple superior alternative paths.

The alternative search technique using EP finds superior multiple paths in order of the time costs and spread of paths by suitable fitness function. That is, superior individuals for alternatives' validity are maintained as paths evolve so that selecting k path individuals in the final generation would provide effective alternatives. The evolution program used in searching for alternative paths consists of the following steps.

```
Procedure EP for Alternative paths
begin
    t = 0
    Initialize Path(t)
    Evaluate Path(t)
    while (not termination-condition) do
    begin
        t = t + 1
        Select Path(t) from Path(t-1)
        Path-crossover Path(t)
        Path_mutation Path(t)
        Remove unnecessary loop in Path(t)
        Evaluate Path(t)
    end
end
```



(a) Before removal



(b) After removal

〈Figure 5〉 Example for removal of unnecessary loop in path

If unnecessary loops are included after the genetic operation, they are removed as shown in 〈Figure 5〉.

2. Representation and generation process of a path individual

start node	.....	node i	......	terminal node	-1

〈Figure 6〉 The structure of a path individual

```
procedure PathMaking(origin :integer, destination: integer)
  const
     max_length = {appropriate  maximum length of
                        a path }
  /* for  excluding the generation of a  too long path */

  var
  /* pnode: previous node,  cnode: current node,
   nnode: next node  in the course of path expanding  */

 pnode, cnode, nnode: integer;
 ptr : integer;  /* path array pointer */
  path : array  [ 0.. MAX_LENGTH]  of  integer ;
  begin
    /* Initialize parameters */
     ptr = 0; cnode = origin;  pnode = -1;
     path[ptr] = origin;  ptr = ptr+1;
   while  cnode ! = destination  do
   begin
     if  cnode = origin   then
     /* get next_node of cnode among its
         neighbor  nodes freely */
      nnode = GetNextNode(cnode);  end  {if};

     else
         nnode = FindNextNode(cnode,pnode,destination) ;
end  {else} ;

     /* Input node data and update parameters  */
      path[ptr] = nnode;    ptr = ptr + 1 ;
     if (ptr = MAX_LENGTH)    then
      pnode = -1;  cnode = origin;  ptr = 1;   end {if}
     else  pnode = cnode;    cnode = nnode;  end {else} ;
  end {while};
   path[ptr] = -1;
   /* for representing the end of  path */

 /* check and delete  circular sub-path which   may
   contain in a given   path  */
  KillLoop(path);

 end  {procedure  PathMaking}
```

path individual(chromosome) in the proposed method is a node set. because road network in this paper is represented by a node-oriented network. The structure of a path individual is described as an array of node informations which consists of a path. The end of a path is represented by adding -1(flag) for easy data processing, because the length of each path between origin and destination can be different(See ⟨figure 6⟩).

How to generate initial individual(chromosome) is important in the genetic algorithm. The generation process of initial path individual is a method which randomly selects current node and neighbor node repeatedly, starting from a origin node until current node reaches to destination node. In this process, neighbor nodes violating the turn constrains(no left turn, etc.) is excepted from selection. Also a neighbor node is selected, considering of relative distance and direction to a destination node for raising efficiency. Circular sub-path(loop) which can be contained as a part of a path in the process is deleted by a special procedure(KillLoop). The initial population of path individuals(chromosomes) is made by repeating the generation process of a path individual. The entire procedure of generating a initial path represented by a pseudo code is as follows.

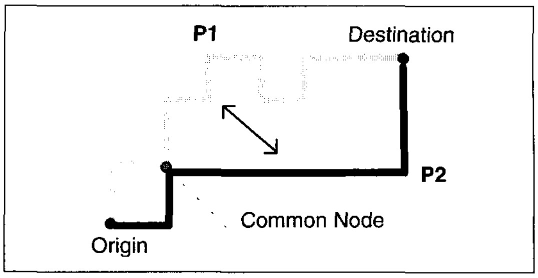
3. Genetic operator

EP method normally needs two genetic operator, crossover and mutation. We devised new genetic operators for producing diverse path individuals. *In the crossover operation, combining two individuals* must not break the route. Crossover points in individuals(P1 and P2) are determined at a common node.

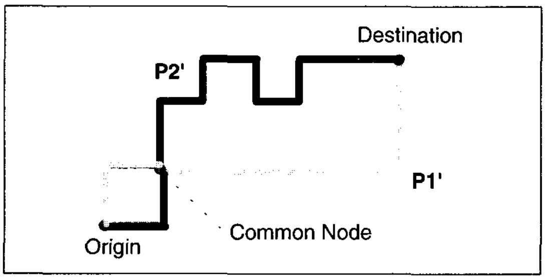
If two individuals have a common node, the latter segments are exchanged and two new children(P1′ and P2′) are created(⟨Figure 7⟩). EP method is effective when provability of crossover(Pc) is above 0.5 in many case. And so we used Pc=0.7 in the experiment for simulation.

If more than one node exist, segments between the common nodes are randomly selected(⟨Figure 8⟩).

The mutation operator randomly selects two points on a path. With these points as the origin and the destination, a new partial path is created and replaces the original segment.(⟨Figure 9⟩).

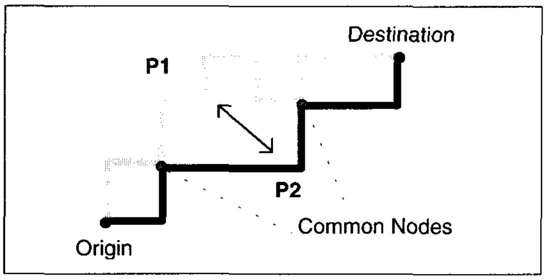


(a) Before crossover

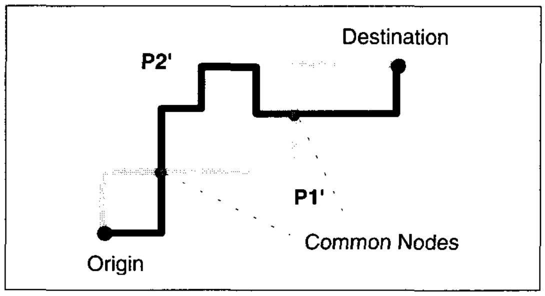


(b) After crossover

⟨Figure 7⟩ Path crossover operator

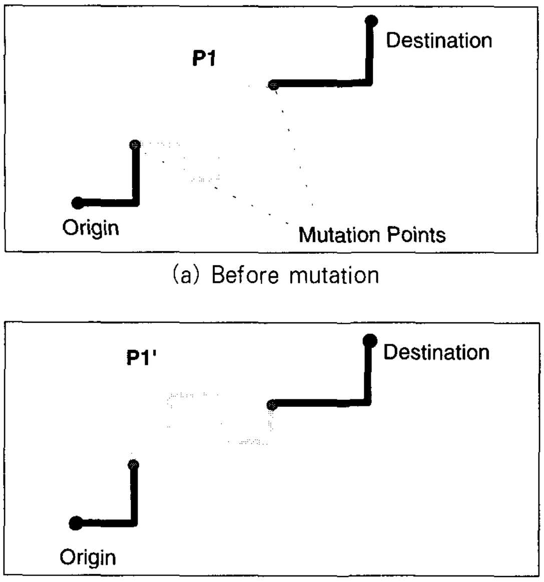


(a) Before crossover



(b) After crossover

⟨Figure 8⟩ Path crossover operator with two common nodes



(a) Before mutation

(b) After mutation

〈Figure 9〉 Path mutation operator

EP method is effective when provability of mutation (P_m) is below 0.5 in many case. And so we use P_m=0.3 in the experiment for simulation.

V. Computational experiments

In this section we describe some computational experiments comparing the proposed method with k-th shortest path algorithm approach. All algorithms were written in the C language(Visual C++) and run on a Pentium PC. In order to simulate the efficiency of the proposed algorithm, test rectangular grid networks were selected. Size of test network varied with 10×10, 20×20, and 30×30 nodes.

All the costs of routes used in this analysis are based on flow time for given origin and destination pairs in the test network. 〈Table 1〉 indicates the validity index for 5 shortest alternative paths of two algorithms.

From the visible standpoint, the appropriate number of alternative paths that should be provided or displayed to the user is 3 to 5. Due to the characteristics of the k-th shortest path algorithm, the k paths provided to the users(k=5) are quite similar to the shortest path.

As shown in the table, for all the three indices, the EP method provides more dispersed paths than the k-th shortest path algorithm. ASP is measured by the number of grids. 〈Tables 2〉 and 〈Table 3〉 shows similar results of the validity index for 10 and 20 shortest alternative paths.

〈Table 1〉 Evaluation index by EP and k-th shortest path algorithm(average of 5 shortest paths)

Network	Algorithm	index		
		CNR	CLR	ASP
10×10	k-th	0.77	0.66	2.6
	EP	0.49	0.40	6
20×20	k-th	0.94	0.90	1.8
	EP	0.80	0.74	8.8
30×30	k-th	0.94	0.92	2.4
	EP	0.82	0.78	14.8

〈Table 2〉 Evaluation index by EP and k-th shortest path algorithm(average of 10 shortest paths)

Network	Algorithm	index		
		CNR	CLR	ASP
10×10	k-th	0.79	0.67	2.3
	EP	0.51	0.42	6.6
20×20	k-th	0.93	0.89	2
	EP	0.80	0.74	8.4
30×30	k-th	0.94	0.91	2.6
	EP	0.62	0.58	35

〈Table 3〉 Evaluation index by EP and k-th shortest path algorithm(average of 20 shortest paths)

Network	Algorithm	index		
		CNR	CLR	ASP
10×10	k-th	0.72	0.57	3.4
	EP	0.45	0.36	8.1
20×20	k-th	0.91	0.85	2.8
	EP	0.76	0.69	9.5
30×30	k-th	0.93	0.89	3.0
	EP	0.54	0.49	41.4

〈Table 4〉 Average costs of k paths by EP and k-th shortest path algorithm

Network	Algorithm	k		
		5	10	20
10×10	k-th	655	659	666
	EP	703	705.5	710.5
20×20	k-th	1507	1508	1509
	EP	1554	1556	1559
30×30	k-th	2534	2535.5	2537.5
	EP	2648	2673.5	2679

The results for average costs of EP's and k-th's for various k alternative paths are summarized in 〈Table 4〉. It shows the average costs of the k paths with varying k values. Regarding average costs, the k-th algorithm shows a little less amount than the EP method. This result is attributed to the fact that path dispersion as well as the cost are considered in EP. Considering that the difference is small, it is highly probable that it would not be a main factor in choosing the paths.

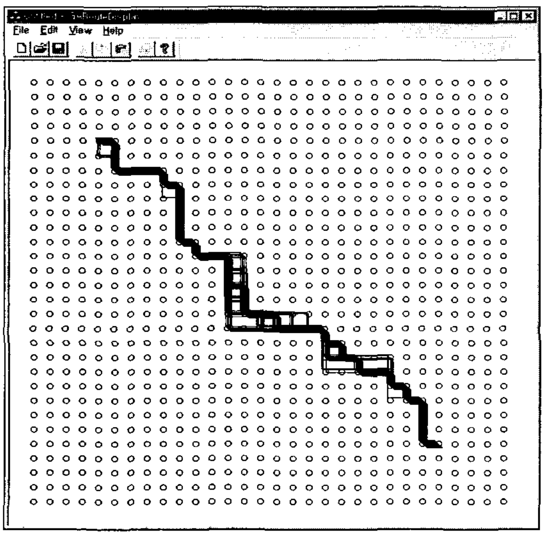
〈Figures 10〉 and 〈Figures 11〉 show the graphic displays of the 50 alternative paths for 30×30 network obtained by two algorithms. From the figures, dispersion of the alternative paths by the k-th algorithm and the EP method are visually identified.

Ⅵ. Conclusions

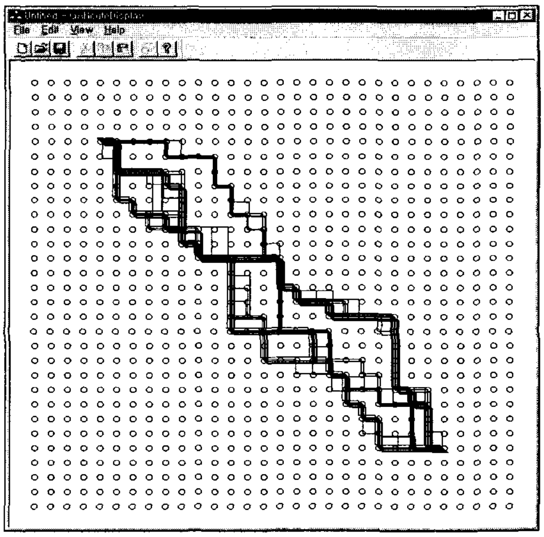
Most present route guidance systems are designed to provide the shortest route between O-D. But, for efficient traffic management and route guidance, it is necessary to provide several alternative paths in addition to the shortest. Also various attributes of a road network such as easiness of driving, surrounding scenery of a road, etc. should be considered. Thus, future systems may be developed to allow users for having guidance choices among some alternative routes. Therefore, we presented a new approach to using genetic algorithms in finding multiple alternative paths.

The suggested method satisfied the validity indices for the effectiveness of alternative paths more than the k-th shortest path algorithm. It also resolved disadvantages of the existing methods such as similarities among the paths. Therefore, the method can provide solutions to such traffic problems as road congestion, while satisfying diverse driver needs.

In order to enhance the validity of alternatives, the development of more systematic techniques for alternative paths including parallel EP, is subject of further research. Moreover, an analysis of



〈Figure 10〉 The graphic display for alternative paths (k=50) by k-th shortest path algorithm



〈Figure 11〉 The graphic display for alternative paths (k=50) by EP the computation time and memory space required is also needed.

References

1. T. Kitamura, M. Kobayashi, K. Takeuchi(1995), "The Dynamic Route Guidance Systems of UTMS", Proc. of The Second World Congress on Intelligent Transport Systems 95 YOKOHAMA.
2. B. Liu(1997), "Route Finding by Using Knowledge About the Road Network", IEEE Trans. on Systems, Man, and Cybernetics-Part A, Vol.27, No.4.
3. E. W. Dijkstra(1959), "A Note on Two Problems in Connection with Graphs", Numer. Mathemetics, Vol.1, pp.269~271.
4. B. V. Cherkassky, A. V. Goldberg, T. Radzik, "Shortes path algorithms: Theory and experimental evaluation", Mathematical Programming, Vol.73, pp.129~174.
5. James A. McHugh(1990), Algorithmic Graph Theory : Shortest Paths, Prentice-Hall Inc.
6. L. R. Rilett, Dongjoo Park(1996), "Identifying Unique Routes in Transportation Network", Proc. of The Third World Congress on Intelligent Transport Systems 96 Orlando.
7. J. Y. Yen(1971), "Finding the k Shortest Loopless Paths in a Network", Management Science, Vol.17.
8. R. D. Shier(1979), "On Algorithms for Finding the k Shortest Paths in a Network", Networks, Vol.9.
9. K. Uchimura, H. Sakaguchi(1995), "Vehicle Routing Problem in Physical Distribution using Genetic Search", Proc. of The Second World Congress on Intelligent Transport Systems 95 YOKOHAMA.
10. Z. Michalewicz(1992), Genetic Algorithms+Data Structures=Evolution Programs, Springer-Verlag.
11. David E. Goldberg(1989), Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Co.Inc.
12. G. S. Choi, J. J. Park, K. S. Seo(1998), "Driver-preference Optimal Route Search based on Genetic-Fuzzy Approaches", Proc. of 5th World Congress on ITS 98 SEOUL.

♧ 주 작 성 자 : 최규석
♧ 논문투고일 : 2001. 9. 27
논문심사일 : 2002. 2. 20 (1차)
2002. 3. 21 (2차)
2002. 4. 1 (3차)
심사판정일 : 2002. 4. 1
♧ 반론접수기간 : 2002. 8. 30