

지능형 주행 안내 시스템을 위한 유전 알고리즘에 근거한 최적 경로 탐색 알고리즘

An Optimal and Genetic Route Search Algorithm for Intelligent Route Guidance System

최규석, 우광방

(Gyoo-Seok Choi and Kwang-Bang Woo)

Abstract : In this thesis, based on Genetic Algorithm, a new route search algorithm is presented to search an optimal route between the origin and the destination in intelligent route guidance systems in order to minimize the route traveling time. The proposed algorithm is effectively employed to complex road networks which have diverse turn constrains, time-delay constraints due to cross signals, and stochastic traffic volume. The algorithm is also shown to significantly promote search efficiency by changing the population size of path individuals that exist in each generation through the concept of age and lifetime to each path individual. A virtual road-traffic network with various turn constraints and traffic volume is simulated, where the suggested algorithm promptly produces not only an optimal route to minimize the route cost but also the estimated travel time for any pair of the origin and the destination, while effectively avoiding turn constraints and traffic jam.

Keywords : genetic algorithm, optimal route search algorithm, turn constraints, route guidance system system

I. 서론

도시내 교통혼잡 증가에 따라 기존 도로의 운영효율을 극대화하기 위해 선진 각국에서는 최근의 전자,정보,통신 등 첨단기술을 이용하여 ITS(Intelligent Transportation System)라고 하는 지능화된 첨단 교통시스템의 개발에 박차를 가하고 있다[3][4]. 지능형 주행안내시스템은 ITS의 일환으로 개발되는 시스템으로서 거리상의 최단 경로를 안내하는 단순 정적 경로안내 차원을 넘어서서 통신 시스템과 연계하여 동적으로 변하는 교통 상황 및 도로가 갖는 각종 제약 사항을 반영하여 운전자에게 목적지까지 시간상의 실질적 최적 주행로를 안내하는 것을 목표로 하는 첨단 운전자 지원시스템을 말한다[1][5] (그림 1). 이에 따라, 본 논문에서는 동적 교통상황 및 각종 제약(좌회전 금지 등) 사항이 존재하는 복잡한 도로 교통망에 대해서 주행시간 측면에서 출발점에서 목적지까지의 실질적인 최적 주행로 및 그에 따른 추정 주행시간을 효과적으로 도출해내는 알고리즘을 개발하고자 한다.

최단 경로탐색(Shortest Path Search)문제는 그래프 이론(Graph Theory) 및 인공지능의 한 분야로서 많이 연구되어 왔으며, 지금까지 개발된 최단경로 탐색 알고리즘으로는 Dijkstra 알고리즘, Floyd 알고리즘, A* Algorithm 등이 있다[7]. Dijkstra 알고리즘은 트리 탐색과 같은 방법으로 그래프를 확장하는 탐색 절차를 사용하여 시작점 x로 부터 종점 y에 이를 때까지 보다 먼 정점들을 계속적으로 탐색해나감에, 종점이 시작점에서 멀어지는 경우 탐색 영역이 커져서 계산 부담이 매우 증가하는 문제점이 있다. Floyd 알고리즘은 그래프내의 모든 두 정점간의 최단 거리를 도출해 내는 방법으로서 임의의 두 정점간의 경로를 도출하는데 모든 두 정점간의 최단거리를 구해야 하므로 효율적이지 못한 문제점이 있다. A* 알고리즘은 Dijkstra 알고리즘의 발전된 변형으로서 탐색 과정중 현재 노드에서 도착지까지의 잔여 경로비용의 최소 추정치(lower bound estimation)

을 이용하여 탐색공간을 줄이는 방법이나 거리 비용보다는 통행시간 비용이 중요한 도로 교통망에서는 남은 경로에 대한 추정이 거의 불가능하므로 동적 교통량을 갖는 도로망의 최적경로 탐색에는 적합하지 않다. 이러한 기존 최단 경로 탐색 알고리즘들은 네트워크의 연결 구조가 고정된 회전 제약이 없는 선형 그래프 형태의 단순 네트워크에서의 최단거리 탐색에 유용하며 또한, 순환서브경로(circular sub-path)는 경로 비용을 증가시키므로 원칙적으로 배제한다. 그러나, 현실적인 도로 교통망은 회전 제약에 의해 진입로에 따라 이웃 노드로 연결될 수도 있고, 연결이 안되기도 하는 연결 구조가 가변적인 복잡한 네트워크이며, 회전 제약 및 교통량 변수에 의해 최적 경로는 U턴과 같은 순환서브경로를 필연적으로 포함하게 되는 경우가 종종 발생된다. 또한, 교차로 상에서 신호등에 의한 통행지연이 전체 주행시간에 많은 영향을 미치는데 기존 알고리즘에서는 이것을 전혀 반영하지 않고 링크상의 비용만을 고려하여 최단 경로를 산출하므로 실제적인 최적 경로의 도출에는 어려움이 있다.

본 논문에서 제안한 알고리즘은 유전 알고리즘[6][8]에 근거하여 신호 교차로에 의한 통행지연, 회전 제약사항, 교통량에 의한 링크 통행시간 정보 등의 다양한 도로교통 상황을 효과적으로 반영하여 주행시간 관점에서 목적지까지의 실질적인 최적 경로를 산출해내는 것을 목표로 한다. 제안 알고리즘은 유전 알고리즘에 근거하므로 탐색과정의 병렬처리가 가능하며, 따라서 트랜스퓨터와 같은 멀티 프로세서 시스템을 이용하면 대폭적인 성능향상이 가능한 우수한 장점을 갖는다. 본 논문의 구성은 다음과 같다. 2장에서는 알고리즘 전개를 위한 도로망 표현 및 데이터구조에 대해서 설명하고 3장에서는 제안하는 최적경로 탐색 알고리즘에 대해서 기술한다. 4장에서는 제안된 알고리즘을 가상 도시가 도로망에 적용한 결과를 분석하고 마지막으로 5장에서 결론을 맺는다.

II. 도로망 표현 및 데이터 구조

일반적으로 도로망은 교차로를 나타내는 노드성분과 교차로간 도로구간을 나타내는 링크성분을 갖는 격자형태의 무

게있는 방향 그래프(weighted digraph)로서 표현될 수 있다. 도로망의 교차로 표현은 1개 교차로를 단일 노드로 처리하는 방안이 있을 수 있고 1개 교차로를 4개의 노드로 분리, 표현하여 회전의 실제 상황을 모델링 작업에 반영할 수도 있다. 본 논문에서는 경로 계산 및 데이터 처리의 부담이 적은 단일 노드에 의한 교차로 표현 방식을 이용하여 도로망을 표현하였고, 회전 상황 처리를 위해 회전 정보를 부가하였다. 제안된 최적경로 탐색 알고리즘의 컴퓨터 처리를 위한 노드 중심의 도로망 데이터 구조를 그림 2와 같이 설계하였다.

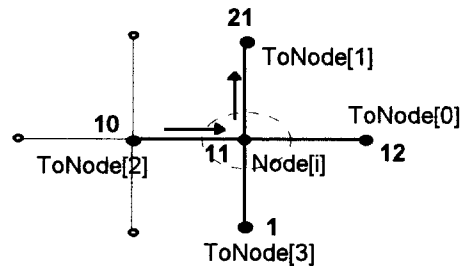


그림 3. 두 노드간의 접속각도의 표현예.
Fig. 3. Representation example of connection angle between two nodes.

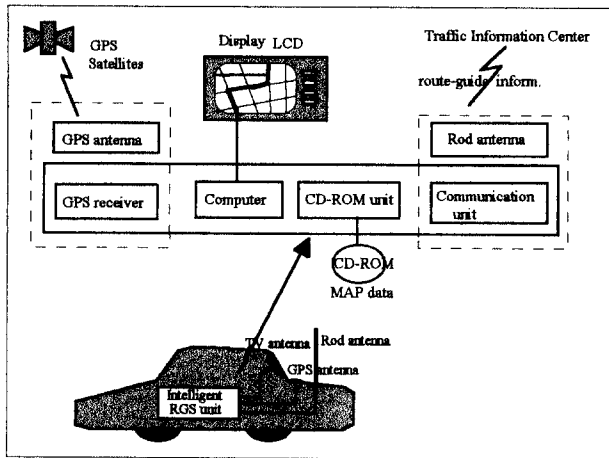


그림 1. 지능형 주행안내시스템 구성도.
Fig. 1. Configuration of Intelligent route guidance system.

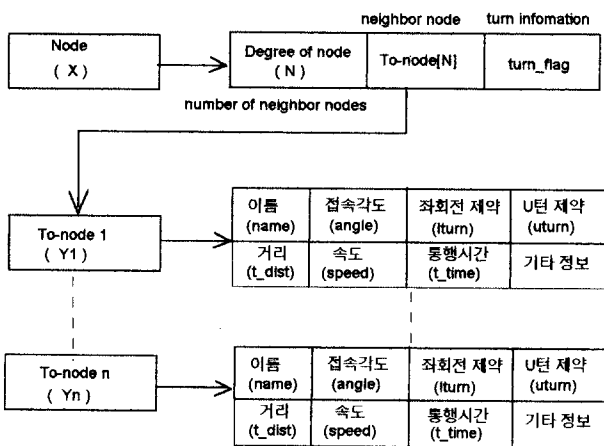


그림 2. 도로망 정보의 데이터 구조.
Fig. 2. The data structure of road network topology.

III. 제안하는 최적경로 탐색 알고리즘

3.1 회전제약 정보 및 경로 개체의 표현

본 논문에서 전개하는 알고리즘에서는 두 노드간의 접속각도(FromNode[i].ToNode[j].angle)를 이용해 주행경로내에서의 회전 상황을 파악한다. 접속각도는 현재 노드와 그 이웃 노드가 링크에 의해 접속되는 각도로서 그림 3에 나타난 바와 같이 반시계 방향으로 각도가 증가되는 방식으로 정의하였으며, 이에 따라 회전 상황은 두 노드간의 접속각도 차에 의하여 효과적으로 파악된다. 접속각도 차이가 0도이면 직진, 90도이면 좌회전, 180도이면 U턴, 270도(또는 -90도)이면 우회전 상황임을 알 수 있다. 이때, 적절한 마진(MARGIN) 값을 설정하여 접속각도 차이가 정확히 일치하지 않는 경우(예: 80도)를 대비한다.

노드 관점에서 이웃 노드에 대한 회전 제약정보는 그림 4에 나타난 바와 같이 turnflag를 이용해 크게 9가지 경우로 구분하고, 각각의 경우에 대해서 해당 데이터 필드(lturn, uturn) 값을 적절히 설정하므로써 회전제약 정보가 효과적으로 표현 및 처리되도록 하였다. 탐색결과로 얻은 경로 개체는 경로를 구성하는 노드집합에 의해 표현되며, 경로 배열에 저장되어 처리된다. 이때, 경로개체의 길이가 각각 다르므로 데이터 처리가 용이하도록 경로의 끝을 나타내는 플래그로서 -1을 중점 노드 다음에 삽입한다.

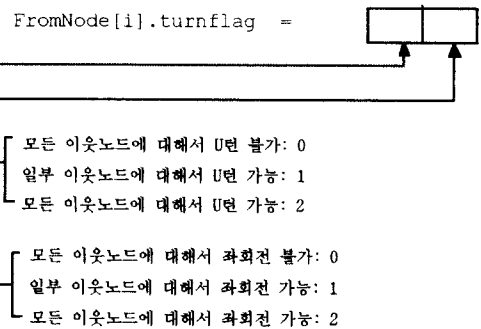


그림 4. 회전제약 정보의 표현방법.
Fig. 4. Representation method of turn constraints information at a node.

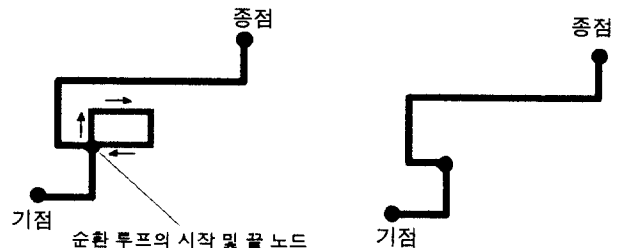


그림 5. 순환서브경로 제거 프로시저 실행예.
Fig. 5. Execution example of circular sub-path deletion procedure.

3.2 경로개체의 생성방법

제안된 기중점(origin-destination)간 경로 생성방법은 기점 노드에서 부터 시작하여 종점 노드에 이를 때까지 현재 노드와 그 이웃 노드를 확률적 랜덤선택에 의해 노드를 계속적으로 반복해서 획득해 나가는 방법이다. 이 과정에서 설정된 회전 제약(U턴 금지 및 좌회전 금지)에 따라 이웃노드중 일부를 선택 대상에서 제외시키며 탐색효율을 높이기 위해서 종점 노드와의 상대적 거리를 고려하여 다음 노드를 생성한다. 여기서, 생성된 경로개체는 필연적으로 여러 가지 형태의 순환서브경로가 기중점간 경로의 일부로서 포함될 가능성이 높으므로, 탐색 효율을 증가시키기 위해서 특정 프로시저(KillLoop)을 이용해 이를 조사하여 제거하도

록 하였다(그림 5). 이때, U턴이 가능하도록 3개 노드 이상으로 구성된 순환서브경로만을 제거하도록 하였다.

회전 제약을 고려한 구체적인 경로 개체의 생성 절차는 다음과 같다.

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

/*get next_node of cnode considering to turn
constraints at the node and the relative
distance to destination */

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 }

3.3 경로재생 연산자의 설계

본 논문에서는 유전 알고리즘에 기반하여 기종점간의 하나의 가능 경로를 경로 개체(염색체)로 보고 유전자 교환을 통한 생물체에서의 진화과정과 유사하게 경로 염색체간의 노드 정보 상호교환 및 서브 경로의 변이를 통해 경로의 진화를 꾀하는 방식으로 경로탐색의 효율을 증진시키도록 하였다(그림 6).

경로 교배 연산자는 교배된 서브 경로가 원래의 경로에 연결될 수 있어야 하므로 두 경로의 공유 노드를 중심으로 형성된 구간에서 임의의 서브 경로를 선택하여 그 경로간의 노드 정보를 상호교환함으로써 새로운 두 경로 개체를 만들

어 낸다. 경로 돌연변이 연산자는 하나의 경로간에 존재하는 임의의 두 점을 선택하여 이 두 점을 시작점과 끝점으로 하는 새로운 서브 경로를 진출한 경로 생성 절차에 의거, 랜덤하게 생성하여 이를 원래의 서브 경로와 대치시키는 방법으로 하나의 새로운 경로 개체를 생성시킨다.

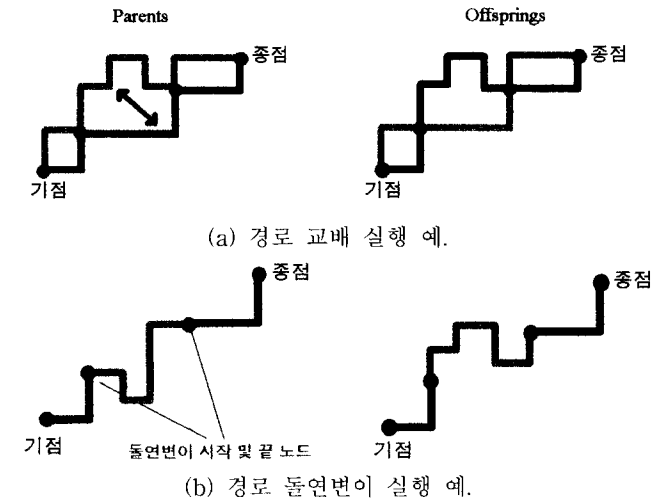


그림 6. 경로재생 연산자의 실행 예.

Fig. 6. Execution example of path reproduction operator.

3.4 교차로 통과비용 및 경로비용 산출

실질적인 최적 경로를 도출하기 위해서 각 교차로에서의 신호 현시 (display) 및 신호시간 스케줄에 근거하여 신호등에 의한 통행지체를 추정하여 이를 최적 경로 탐색에 효과적으로 반영되도록 하였다. 일반적으로 각 교차로에서 최단 지체 시간은 0 이고 최장 지체시간(Max_CrossDelayTime)은 (신호주기 시간 - 회전 허용시간) 이다. 따라서, 신호 스케줄에 의한 교차로 통과비용은 (1)에 나타난 바와 같이 두 값의 산술 평균값에 순수 회전비용을 더한 값으로 추정하였다. 이때, 교차로에서의 회전제약의 유무, 교차로 종류(삼거리, 사거리 등), 접근 방향 및 회전 방향(직진, 좌회전 등)에 따라 최장 지체시간이 달라지므로 설정된 도로망 정보 및 경로 구성 노드간 상호관계 추적을 통해 이를 효과적으로 파악하여 통과비용 산정에 반영되도록 하였다. 또한, 혼잡 정도가 심화된 경우 신호등에 의한 통가지체는 더욱 커지므로 해당 링크의 속도정보를 참조하여 이를 이용해 보정계수 ((2))를 산출하여 교차로 통과비용 산정에 반영((3))되도록 하였다.

$$\text{cross_cost} = (\text{Max_CrossDelayTime}) / 2 + \text{TurnCost} \quad (1)$$

where, U-TurnCost > LeftTurnCost > Straight>

$$\text{RightTurnCost}$$

if RightTurn, Max_CrossDelayTime = 0

if link_speed < THRESHOLD_SPEED

$$\text{correct_factor} = 2.0 - \text{link_speed} / \text{THRESHOLD_SPEED} \quad (2)$$

otherwise, correct_factor = 1.0

$$\text{Cross_cost} = \text{cross_cost} * \text{correct_factor} \quad (3)$$

기종점간 경로비용은 경로 배열내의 모든 노드에 대해서 노드간 링크비용(통행시간) 및 각 노드(신호 교차로)에서의 통과비용을 계산한 총합으로 하였다. 또한, 연산자에 의한 경로 재생과정에서 제약을 어기는 불합리한 경로가 발생될 수 있으므로 이 과정에서 불합리한 경로에 대해서는 별점

(PENALTY)이 경로 비용에 추가 되도록 한다.

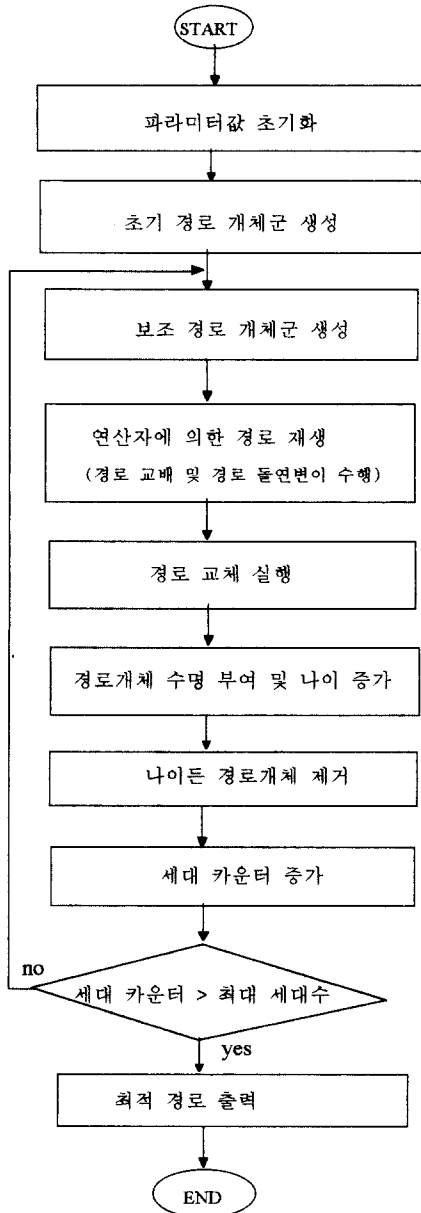


그림 7. 제안 알고리즘에 의한 최적경로탐색 수행절차.
Fig. 7. Execution procedure of optimum route search by the proposed algorithm.

3.5 전체 알고리즘의 수행절차

본 논문에서 제안하는 알고리즘은 유전 알고리즘에 기반하여 경로 개체집단의 크기를 매 세대마다 변화시킴으로써 조기 수렴으로 인한 탐색능력 저하 가능성을 배제하고 효율적 탐색이 수행되도록 하였다. 전체적인 최적경로 탐색절차는 그림 7에 나타난 바와 같다. 먼저 경로 교배 및 돌연변이 확률, 초기 개체군 및 최대 개체군 크기 등 각종 파라미터들의 값을 설정하고, 초기화 시킨다. 다음으로 초기 경로 개체군(population)을 생성하고, 보조 경로 개체군을 생성한다. 이어서, 설정된 교배 확률 및 돌연변이 확률을 적용, 경로 교배 및 경로 돌연변이를 수행한다. 위의 각 과정에서 각 경로를 평가하여 현재의 최소값을 갱신하는 경로는 최소비용 경로배열에 그 정보를 보유하며, 현재의 최적 경로를 포함한 일정 갯수의 우수한 경로를 특정 장소(ELITE_POOL)에 저장한다. 그림 7(e)의 과정에서는 탐색 효율을 극대화하기 위해 현재 존재하는 경로 개체군내의 모든 경로 염색체를 평가하여 설정된 수만큼의 열등한 경로 개체를 특

정 장소에 이미 보유하고 있는 우수한 경로로 대체시킨다. 그 다음 과정에서는 위의 과정 중에서 새로이 생성된 신생 경로만을 대상으로 경로 비용을 평가하여 그 값에 반비례하여 경로 개체의 수명을 부여함으로써 우수한 개체는 오래 살아 남도록 하여 탐색능력 향상을 도모한다. 또한, 각 경로 개체의 나이를 조사하여 그것의 수명보다 더 나이가 많은 경로 개체는 개체 집단에 삭제하여 경로 개체 집단의 크기를 조절한다. 이어서 다음 세대로의 이행을 위해 세대 카운터를 하나 증가시킨다. 만일 설정된 최대 세대수 값과 비교하여 이보다 작으면 계속적인 탐색을 위해 그림 7(c)로 분기하여 반복 루프를 수행한다. 최대 세대수 보다 클 때에는 최종 과정으로서 특정 배열에 보유하고 있는 최소비용 경로를 최적 경로로서 출력하고 탐색작업을 종료한다.

IV. 시뮬레이션 및 결과 분석

본 장에서는 제안된 최적경로탐색 알고리즘을 단순 네트워크 및 가상 도시교통망에 적용한 시뮬레이션 결과를 분석한다. 시뮬레이션은 C언어로 구현한 시뮬레이터(그림 8)를 이용하여 PC(Pentium 90)상에서 수행하였다.

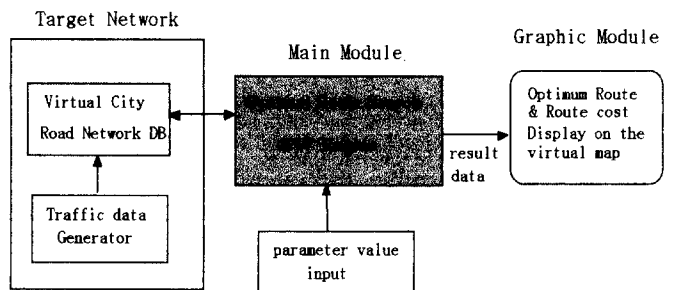
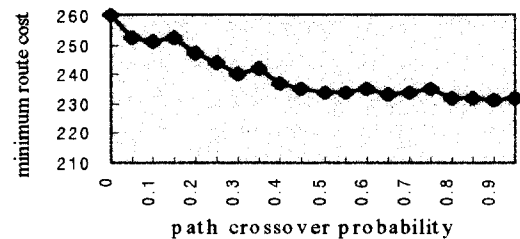
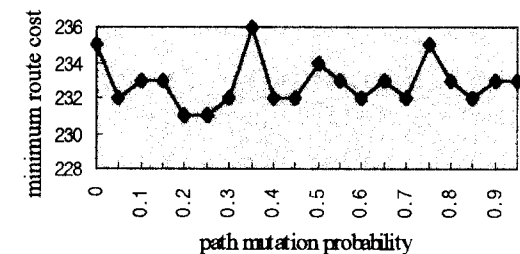


그림 8. 최적경로 탐색 시뮬레이터의 구조.
Fig. 8. Structure of optimal route search simulator.



(a) 경로교배 확률 변화에 따른 성능 변화.



(b) 경로돌연변이 확률 변화에 따른 성능 변화.

그림 9. 경로재생 연산자 확률값 변화에 따른 탐색 성능변화.

Fig. 9. Search performance change by the probability change of path reproduction operator.

단순 네트워크는 회전 제약이 없는 100개의 노드와 360개의 링크(양방향 포함)를 포함하는 정방향 격자형태의 방향 그래프(digraph)로 구성하였고, 그 링크비용은 링크를 이루는 기준 노드와 그 이웃 노드의 인덱스 정보를 이용하여 단순

하게 부여하였다. 단순 네트워크에 제시된 알고리즘을 적용하여 시뮬레이션을 수행하였을 때, 모든 기종점에서 항상 최소 비용을 보장하는 기존 알고리즘(Floyd, Dijkstra)에서 도출된 값과 동일한 값을 나타내었다. 부가해서 경로재생 연산자의 적용 확률을 변화시키면서 도출된 최소 비용을 비교한 결과 그림 9에 나타난 바와 같이 교배확률 0.9, 돌연변이확률 0.2에서 최소값을 보였다. 이 결과에서 경로 교배 연산자가 경로 돌연변이 연산자 보다 탐색성능에 더 영향을 줌을 알 수 있었다.

가상 도시교통망은 좌회전 금지, U턴 금지 등의 회전제약 및 부분적인 교통정체, 정점 노드(vertex node)를 제외한 모든 노드에서 신호등에 의한 통행지연이 존재하는 가상 도로망으로서 정의하였으며, 100개의 노드와 328개의 링크(양방향 포함)로서 구성하였다. (그림 11) 여기서, 정점 노드는 2개의 이웃 노드를 갖는 노드로서 외곽 순환도로와 같이 도로구간 길이가 긴 경우에 그 중간 정점으로 삼입되는 노드이다. 또한 각 교차로에서의 신호시간 스케줄은 실제 도로망의 경우를 반영하여 적절히 가정하였다. 반복적인 시뮬레이션을 통해 교배 확률 0.9, 돌연변이 확률 0.2, 초기 개체수 30, 최대 개체수 50, 최대 세대수 30, 보조 개체군 생성율 0.3 에서 최대 성능을 보임을 확인하였으며, 이들 값을 적용해 최적경로 탐색을 실행하였다. 링크거리 데이터는 800m - 1,000m에서 랜덤발생하여 전체 링크에 분포 시켰으며, 10km/h - 80 km/h(굵은 선 부분은 공사, 사고, 혼잡 등으로 인한 교통지체 지역으로 30km/h 이하)의 교통 상황을 가정한 링크 통행속도 데이터를 랜덤발생하여 전체 링크에 분포 시켰다. 또한, 교차로 통행 지체를 증가시키는 한계 속도(THRESHOLD_SPEED)는 40km/h로 가정하였다. 이러한 가상 도로 교통망상에서 기점 6번 노드에서 종점 81번 노드에 이르는 최적 경로탐색 시뮬레이션을 수행한바, 그림 10 및 그림 11(a)와 같은 결과를 얻었다. 그림 10에 나타난 바와 같이 각 세대에는 다양한 경로 개체들이 존재하며 세대가 진전됨에 따라 경로 비용이 적은 보다 더 우수한 개체들이 생성되었음을 알 수 있다. 그림 11(a)에 나타난 경로는 마지막 세대(그림 10)에 나타난 경로 개체 중 가장 작은 경로 비용을 갖는 경로를 최적 경로로서 출력한 것으로서 회전 금지 및 교통혼잡을 회피하면서 신호 부담이 적은 외곽 순환도로를 일부 이용하여 경로가 형성되었음을 알 수 있다.

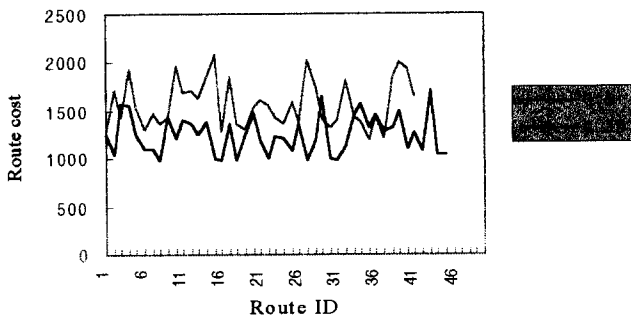
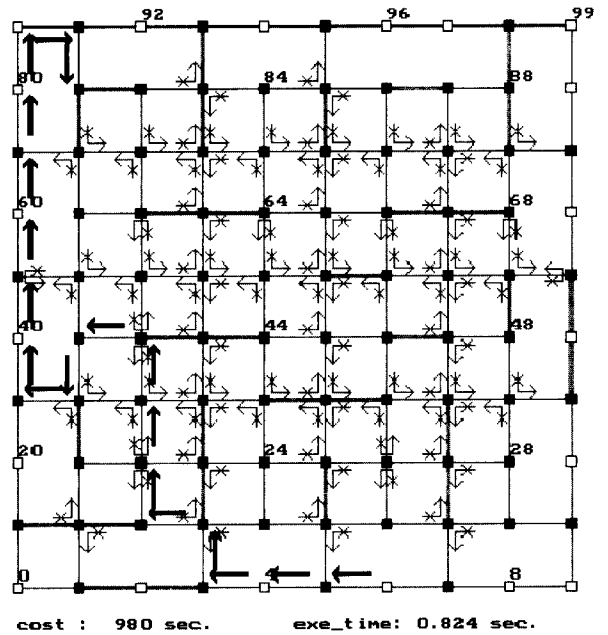
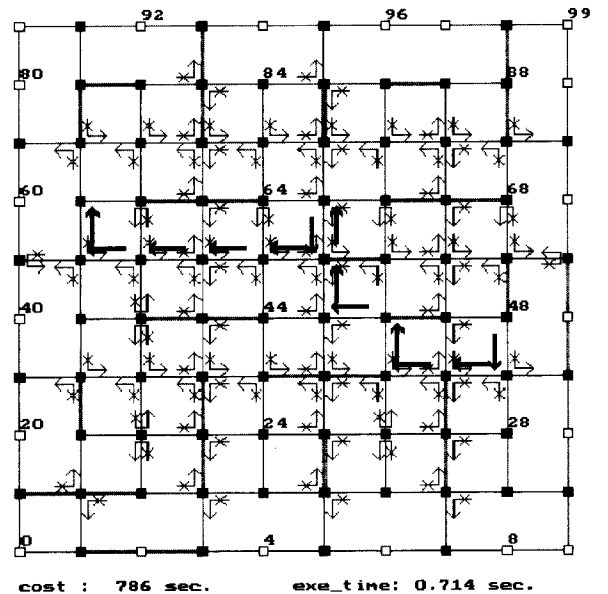


그림 10. 5세대와 30세대의 경로개체 비용값 분포.
Fig. 10. Route cost distribution in generation 5,30.

48번 노드를 기점으로 하고 61번 노드를 종점으로 하는 최적 경로를 탐색한 결과 그림 11(b)에 나타난 바와 같은 최적 경로가 도출되었다. 여기서, U턴을 통해 교통혼잡이 가중된 도로 및 좌회전 제약을 피하고 소통이 원활한 도로 구간으로 최적 경로가 생성되었음을 알 수 있다.



(a) 탐색결과 1.



(b) 탐색결과 2 (U턴 실행예).

그림 11. 제안 알고리즘에 의한 최적경로 탐색결과.
Fig. 11. Result of optimum route search by the proposed algorithm.

V. 결론

본 논문에서는 향후 출현할 지능형 주행안내 시스템의 적용이 가능하도록 유전 알고리즘의 개념에서 힌트를 얻어 신호 교차로에 의한 통행 지연, 회전 제약사항, 교통량에 근거한 링크 통행속도 정보 등을 효과적으로 고려하여 주행시간 측면에서 목적지까지의 실질적인 최소 비용 경로를 산출해내는 시작점과 종점간의 최적 경로 탐색 알고리즘을 개발하였다. 탐색성능을 높이기 위해 경로 염색체에 연령 및 수명 개념을 도입하여 매 세대에 존재하는 경로 개체 집단(해집합)의 크기를 변화시켰으며, 반복적 시뮬레이션에 통해 알고리즘에서 사용하는 여러가지 파라미터의 특성 분석 및 이의 최적값을 도출하여 문제 영역에 적용하였다. 동적 교통상황 및 각종 회전 제약을 갖는 100개 노드로 구성된 가상 도시교통망을 대상 도로망으로 하여 제안된 알고리즘을

적용한 시뮬레이션을 수행하였는 바, 회전 제약 및 교통혼잡을 회피하면서 최소 비용(시간 비용)을 갖는 실질적인 의미의 최적 경로 및 추정 주행시간을 효과적으로 신속하게 도출함을 확인하였다.

참고문헌

[1] D. J. Jeffery, "Route guidance and vehicle location systems," *Mobile Information Systems*, pp. 177-202, ARTECH HOUSE, 1990.
 [2] R. Sedgewick, *Algorithms in C++*, Addison-Wesley Publishing Co., Inc. 1992.
 [3] S. D. Chi, Y. K. Kim, B. G. Yu and G. S. Choi, "A role of discrete event modeling for advanced traveller information systems," *Proc. of conference on AI, Simulation, and Planning*, pp. 37-43, SanDiago, USA, 1996.
 [4] G. S. Choi, S. P. and K. B. Woo, "Using the paging

network to develop the advanced traveler information systems," *Proc. of The Second World Congress on Intelligent Transport Systems 95 YOKOHAMA*, pp. 640-645, 1995.

[5] T. Kitamura, M. Kobayashi, K. Takeuchi, "The dynamic route guidance systems of UTMS," *Proc. of The Second World Congress on Intelligent Transport Systems 95 YOKOHAMA*, pp. 610-615, 1995.
 [6] Z. Michalewicz, *Genetic Algorithms + Data Structures=Evolution Programs*, Berlin: Springer-Verlag, 1992.
 [7] J. A. McHugh, *Algorithmic Graph Theory*, Prentice-Hall International, Inc., 1990.
 [8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Co.Inc., 1989.



최규석

1960년 2월 4일생. 1982년 연세대학교 전기공학과 졸업. 동대학원 석사(1987), 동대학원 박사(1997). 1987년 1월 ~ 1990년 12월 (주) 데이콤 정보통신연구소 근무. 1990년 12월 ~ 1997년 1월 한국이동통신 중앙연구원(책임연구원).

1997년 2월 ~ 현재 충남산업대학교 인공지능학과 교수. 관심분야는 지능시스템 및 컴퓨터 응용, 지능형 교통시스템(ITS), 이동통신 및 이동데이터 시스템 등



우광방

1957년 연세대 전기과 졸업. 동대학원석사(1959). (미) 오레곤주립대 전기공학박사(1964). 1997년 ~ 현재 연세대학교 전기공학과 교수. 관심분야 플랜트 제어 및 응용, 반도체 공정제어, 수처리공정제어, 지능형 모델링 및 제어.