

교통제약을 고려할 수 있는 네트워크재구성과 다수최적경로들을 탐색할 수 있는 진화프로그램의 개발

Development of Network Reconfiguration for Traffic Restrictions and Evolution Program for Multiple Shortest Paths

김 성 수* 민 승 기** 홍 순 정***
Kim, Sung-Soo Min, Seung-Ki Hong, Soon-Jung

Abstract

This paper is to develop a network reorganization method to consider the constraints which are turn restrictions, U-turns, and P-turns in two way traffic network for Intelligent Transport System(ITS). Several kinds of traffic constraints can be represented in reorganized network using virtual nodes and arc values. The evolution program is also developed to search the multiple shortest paths in this reorganized traffic network within limited time.

키워드 : 최적경로, 네트워크 재구성, 지능형교통시스템

Keywords : *optimal path, network reconfiguration, Intelligent Transport System*

1. 연구의 목적과 필요성

지능형교통시스템(ITS)은 도로, 차량, 신호시스템 등 기존의 교통체계 구성요소와 전자제어통신 등 첨단기술을 접목하여, 교통정보의 흐름을 원활히 하고, 구성 요소들이 상호 유기적으로 작동하게 함으로써 기존 교통시설을 효율적으로 이용하고 운전자의 편의성과 안정성을 극대화하는 차세대 교통정보시스템이다. 이러한 지능형 교통시스템의 여러 분야중 한 부분이 첨단교통정보시스템(ATIS, Advanced Traveler Information System)이며 그 시스템의 가장 핵심이 되는 기술이 바로 최단경로

탐색이다. 교통네트워크에서 최단경로를 구하는 문제는 이미 1950년대 말부터 여러 가지 알고리즘들이 개발되었으며, 잘 알려진 알고리즘으로는 Dijkstra, Ford-Bellman, Yen, Shier, Floyd-Warshall 알고리즘 등이 있으나, 첨단교통정보시스템에 사용되기 위해서는 기존 최단 경로 탐색 알고리즘으로는 현실적인 교통네트워크의 복잡한 각종 제약(회전금지, U-turn금지, U-turn, P-turn 등)과 동적 교통상황을 고려하여 다수최적 경로들을 일정한 시간내에 도출해낸다는 것은 쉽지 않다.

교통네트워크에서 제약상황을 고려하기 위한 연구는 크게 2가지 접근 방법에 의해 분류 될 수 있다. 첫 번째 방법으로는 네트워크를 재구성하여 교차로 제약이 암묵적으로 포함되어 있는 네트워크로 만들어 기존의 최단경로 알고리즘을 적용하는 방법이다. 이러한 네트워크의 재구성 방법은 제약이 암묵적으로 표현되어 있는 네트워크로 재구성

* 강원대학교 산업공학과 부교수, 공학박사, 책임
저자

** 풀무원(주) 대리

*** 대한통운(주) 대리

해주기 때문에 어떠한 알고리즘이라도 적용할 수 있다는 특징이 있다. 두 번째 접근 방법은 기존의 최단 경로 알고리즘을 수정하여 교차로 제약을 고려할 수 있도록 수정하는 방법이다. 하지만 기존의 알고리즘을 수정하는 방법은 근본적으로 실질적 교통네트워크의 최단경로를 제한된 시간 내에 구하는 것은 쉬운 일이 아니다.

본 논문의 목적은 네트워크 재구성성을 통하여 실제 교통네트워크의 복잡한 대부분의 제약상황을 고려하며, 이 방법을 사용함으로써 발생하는 복잡도의 상승의 문제점을 해결하고 차후 대규모 네트워크에서 다수 최적경로를 사용자가 제시하는 제한 시간내에 산출할 수 있도록 진화프로그램을 개발하는 것이다. 또한, 실제 사용자가 서비스 받을 수 있도록 사용 가능한 최적경로안내시스템의 프로토타입을 Java(JFC)를 이용하여 설계하고, 실제 네트워크에 적용하여 본 논문에서 제시하는 방법의 적합성과 효율성을 검증하는 것이다.

2절에서는 네트워크에 적용하여 최적경로를 찾고자 시도한 기존 논문들을 평가 분석함으로써 각 논문이 제시하는 방법들의 장 단점을 서술하였다. 3절과 4절에서는 이러한 방법들의 장점을 공유하고 단점을 보완하여 최적경로를 제시할 수 있는 새로운 방법론을 제시하고자 한다. 특히, 대부분의 교통 제약 요인들을 고려할 수 있는 네트워크 재구성 방법을 설명하고, 이 방법의 단점인 복잡도 증가를 극복하고 사용자가 제시한 제한된 시간내에 다수 최적경로들을 효율적으로 탐색할 수 있는 진화프로그램에 대하여 서술하였다. 5절에서는 새로 제안한 방법의 적용과 실험결과에 대하여 서술하고, 6절에서는 새로 제안한 방법과 기존 알고리즘의 성능을 비교 분석하였다.

2. 기존 연구의 조사 및 분석

교통네트워크의 제약상황들을 고려하고자 하는 기존연구는 다음과 같은 시도들이 있었다. 박찬규, 박순달, 진희재[4]의 논문에서는 기존의 Dijkstra 알고리즘에 교차로의 제약상황이 주어졌을때 이를 고려하기 위해 네트워크를 변형시킴으로써 해결하고자 하였다. 이 방법은 보다 정밀한 네트워크의 방향성을 나타낼 수 있는 장점이 있는 반면, 노드와 호의 수가 불필요하게 증가하게 되어 계산량의 급증을 야기시킬 수 있고, 고려할 수 있는 사항이 한계가 있다.

최기주[6]의 논문에서는 회전금지를 표현하기 위해 각 노드의 전번 노드와 전전번 노드를 기억하고, 이를 알고리즘 자료로 사용하는 방법을 제시하였다. 이는 회전금지가 연속해서 발생할 경우 기존의 알고리즘에 한계가 발생되며, 잘못된 정보를 가지게 될 수도 있다. 또한, 제약상황이 늘어남에

따라 필요한 기억용량과 계산량이 큰 장애 요소로 작용하게 된다.

김성수, 차영민[2]논문에서는 회전제약을 고려하기 위하여 회전제약이 일어나는 호를 무한대(∞)로 치환하여 구하였으며, 다른 제약상황은 가상노드를 만들어 고려하였다. 그러나 회전금지 고려 시 회전금지가 하나 늘어날 때마다

2^n (n : 회전금지 수)배로 계산량이 급증하게 된다.

김성수, 전영주, 차영민[3] 논문에서는 회전계약 상황 데이터를 기억하여, 선택된 경로에 계약상황을 제외하는 방법을 사용하였으며, 다른 계약상황들은 가상의 호를 만들어 고려하였다. 그러나, 교차로에서의 P-turn만을 고려하였다고 가정하였기 때문에 사실적인 계약상황의 표현에 적당하지 못할 수 있다.

유전자 알고리즘 또는, 진화 프로그램을 사용하여 최적경로를 찾고자 하는 기존 연구는 다음과 같은 시도들이 있었다. 최규석과 우정방[5]의 논문에서는 초기염색체 경로생성방법으로 기점 노드에서 시작하여 종점노드에 이를 때까지 현재 노드와 그 이웃 노드를 확률적 랜덤선택에 의해 경로를 생성하는 방법을 사용했다. 이 방법은 네트워크의 규모가 커지면, 개체생성 가능성이 적으며 비효율적이다.

Gen, et al[8]의 논문에서는 모든 노드에 우선순위를 중복되지 않게 랜덤 하게 설정하여 하나의 개체를 만드는 방법을 사용하였다. 경로생성(염색체생성)방법은 한 노드에서 이웃된 노드들 중 우선순위가 가장 높은 것을 선택하는 과정을 반복하여 도착노드에 다다를 때까지 경로를 연결하게 된다. 개체생성에 있어서 네트워크의 규모가 커지면 불완전한 경로의 경우의 수가 커질 것이라 생각된다.

Voicu와 Myler[9]의 논문에서 경로를 찾아가는 방법은 시작노드와 도착노드에서 동시에 랜덤하게 인접한 노드들을 연결하여 서로 만나게 함으로써 경로를 생성해 간다. 이 논문의 특징은 초기 개체 생성부터 유전자알고리즘을 사용하여 교배와 돌연변이를 적용하여 초기 개체 생성에서 보다 다양한 개체 집단을 생성하였다고 서술하고 있으나 개체 생성을 위한 유전자알고리즘 적용과정이 불명확하다.

Seo와 Choi [10]의 논문에서 유전자 알고리즘을 이용하여 최적경로로 수렴을 시키면서 공통노드, 호의 중복을 피함으로써 다양성을 갖는 최적경로를 제시한다고 서술하였으나 방법론이 불명확하다.

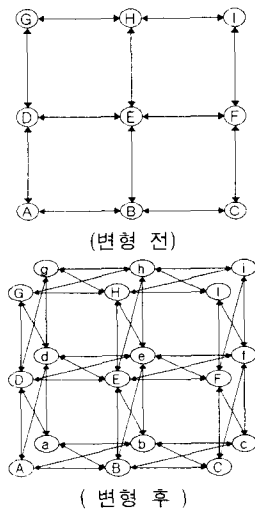
2절에서는 기존방법들의 장단점을 서술하였다. 이를 바탕으로 3, 4절에서는 새로운 방법론을 제시하고자 한다.

교통제약을 고려할 수 있는 네트워크재구성과 다수최적경로들을 탐색할 수 있는 진화프로그램의 개발

3. 네트워크 재구성 방법

네트워크 재구성의 목적은 실제교통 제약요인을 고려할 수 있도록 네트워크를 변형하는 것이다. 즉, 교통제약요인을 암묵적으로 네트워크에 포함될 수 있도록 하는 것이다. 본 절에서는 홍순정, 김병하[7]가 제안한 네트워크 재구성 방법을 개선하여 보다 효율적으로 사용하기 위해 평가 분석하였다.

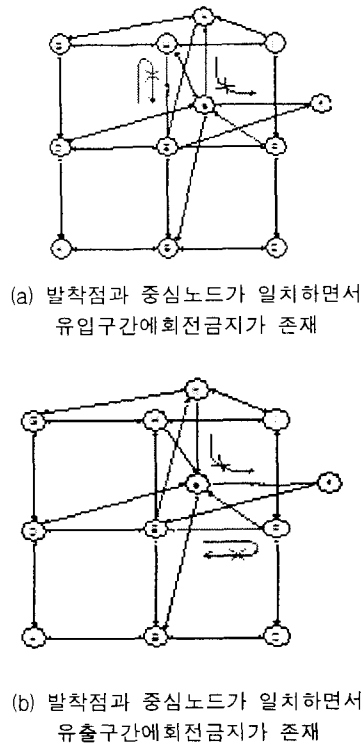
네트워크 재구성방법은 제약상황이 어떻게 주어지느냐에 따라 달라진다. 한 교차로에 2개의 회전금지 구간이 존재하는 경우(2중 회전금지)와 단일 회전금지 구간이 있는 경우(단일 회전금지)의 경우는 데이터 변환 방식이 서로 다르며, 한 교차로에 두 개의 U-turn 금지가 있는 것과 한 개만 존재하는 것 또한 데이터 변환 방식이 다르게 이루어진다. 이렇게 입력된 제약상황의 데이터 분류가 끝난 다음은 네트워크를 재구성 한다. 이 네트워크 재구성은 기본적으로 각 노드마다 가상노드를 추가 시키는 것을 전제로 이루어지며, 실제노드와 가상노드와의 연결은 존재하지 않는다. 이러한 가상노드의 추가가 끝나면 데이터 변환 과정이 이루어지게 된다. 이 과정은 실질적으로 재구성된 네트워크가 제약상황을 고려할 수 있도록 노드와 노드사이의 호 값들을 무한대(∞)로 치환하는 단계이다. 즉, 입력된 제약상황의 경로는 배제하면서 다른 경로들을 모두 고려할 수 있게 데이터를 다시 재구성하는 것이다.



[그림 3.1] 네트워크의 가상노드 추가

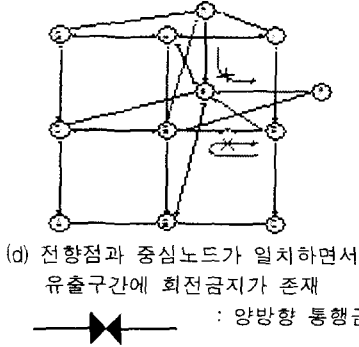
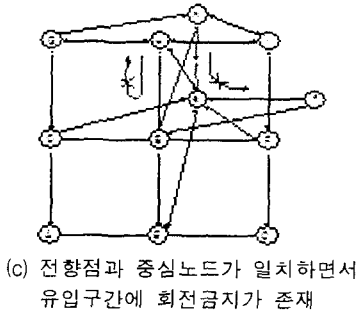
[그림 3.1]의 가상노드를 추가시킨 네트워크에서 회전금지 구간이 B→E→D인 단일 회전금지일 경우, 실제노드 B와 가상노드 b로부터 회전금지 구간 중심노드 E로 진입하는 것을 막기 위해

$\pi_{BE} = \infty$, $\pi_{bf} = \infty$ 로 치환한다. 이번에는 중심노드의 가상노드인 e에서 회전금지 구간 종착점인 실제노드 D와 가상노드 d로의 진입을 막기 위해 $\pi_{ed} = \infty$, $\pi_{ed} = \infty$ 로 치환한다. 이렇게 치환함으로써 B→E→D의 회전금지 구간 제외된 나머지 모든 경로를 검색할 수 있게 된다. 2중 회전금지(B→E→D, H→E→F)의 경우도 비슷한 개념으로써 우선 첫 번째 회전금지 구간의 데이터 변환은 단일 회전금지 구간의 경우와 동일하다. 그러나, 두 번째 회전금지 구간은 반대로 회전금지 구간의 출발점의 실제노드 H와 가상노드 h에서 가상노드 e로의 진입을 막는다. ($\pi_{he} = \infty$, $\pi_{he} = \infty$) 또한, 중심점의 실제노드 E에서 회전금지 구간 종착점(노드 F, f)으로의 진입을 차단한다. 이렇게 조항을 바꿔 줌으로써 회전금지 구간이 제외된 나머지 모든 경로를 고려할 수 있게 된다.

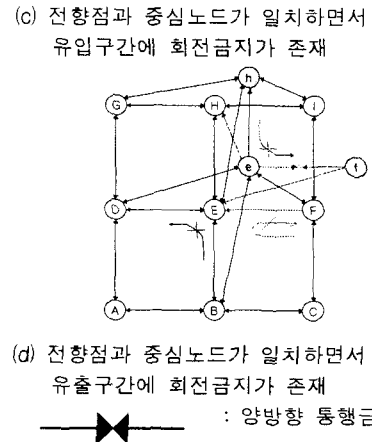
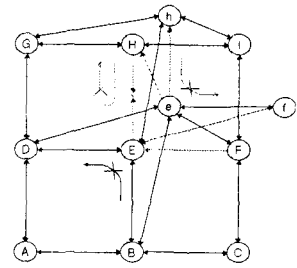


(a) 발착점과 중심노드가 일치하면서 유입구간에 회전금지가 존재

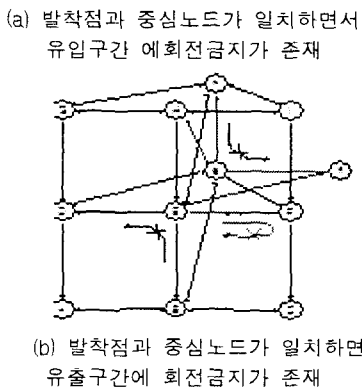
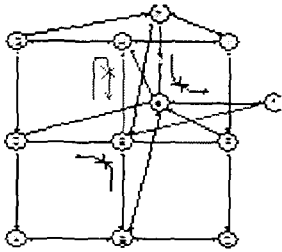
(b) 발착점과 중심노드가 일치하면서 유출구간에 회전금지가 존재



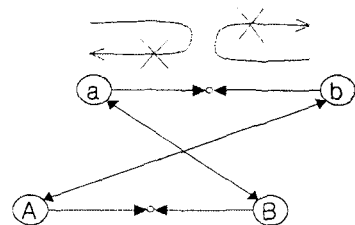
[그림 3.2] U-turn 금지를 고려하기 위한 데이터 변환 (단일 회전 금지인 경우)



[그림 3.3] U-turn 금지를 고려하기 위한 데이터 변환 (이중 회전 금지인 경우)



[그림 3.2]과 [그림 3.3]를 이용하여 데이터를 변환을 사용하면 어떠한 상황에서도 U-turn 금지를 고려할 수 있게 되며, 이러한 경우는 총 8가지로 분류된다. 이러한 각 경우들은 데이터 변환 방법에 따라 [표 3.1] 과 같이 구분되며, 각자 다른 데이터 변환 방법을 따른다. 이 8가지 경우 이외에 한 구간내에 U-turn 금지가 중복되어 있는 경우는 [그림 3.4]와 같이 U-turn 금지구간(A)의 발착점A는 또 다른 U-turn 금지구간(B)의 전향점B와 일치하고, 발착점B는 전향점A와 일치할 경우 앞에서 언급한 내용을 사용할 수 없게 된다. 이 경우에는 [그림 3.4]와 같이 데이터 변환을 한다. 실제노드간, 가상노드간 호의 연결을 차단한다.



[그림 3.4] 2개의 U-turn 금지가 한 구간내에 중복되어 있는 경우 데이터 변환방법

교통제약을 고려할 수 있는 네트워크재구성과 다수최적경로들을 탐색할 수 있는 진화프로그램의 개발

[표 3.1] 데이터 변환방법에 따른 4가지 구분

그림번호	데이터 변환 방법
[그림 3.3]-(a)	발착(실제) → 전향(실제) => ∞
[그림 3.3]-(b)	전향(가상) → 발착(가상) => ∞
[그림 3.4]-(a)	전향(실제) → 발착(가상) => ∞
[그림 3.4]-(b)	발착(가상) → 전향(가상) => ∞
[그림 3.3]-(c)	전향(가상) ↔ 발착(실제) => ∞
[그림 3.4]-(d)	전향(실제) ↔ 발착(실제) => ∞
[그림 3.3]-(d)	전향(실제) ↔ 발착(실제) => ∞
[그림 3.4]-(c)	전향(실제) ↔ 발착(실제) => ∞

앞에서 설명한 데이터 변환으로 인한 네트워크 재구성 방법은 실제 교통 네트워크에서 일어나는 복잡한 제약상황 대부분을 고려할 수 있으며, 다양하고 복잡한 경로를 쉽게 고려할 수 있다는 특징이 있다. 또한, 재구성된 네트워크는 이미 실제 교통 네트워크의 제약요인을 암묵적으로 표현하고 있기 때문에 기존최단경로 알고리즘을 그대로 사용할 수 있다. 이처럼 알고리즘을 필요에 따라 다양하게 선택할 수 있다는 것은 네트워크 재구성방법의 큰 특징이다. 그러나 이 방법은 노드수를 2배로 증가시키게 되고, 이로 인하여 복잡도를 증가시키는 결과를 가져온다. 이로 인한 계산시간과 계산량의 증가는 기존의 알고리즘으로 다수최적경로서비스를 제공하는데 있어 장애가 된다. 이 문제점을 해결하기 위해 네트워크 재구성방법을 바탕으로 하여 다음과 같이 진화프로그램을 개발하였다.

4. 진화 프로그램의 개발

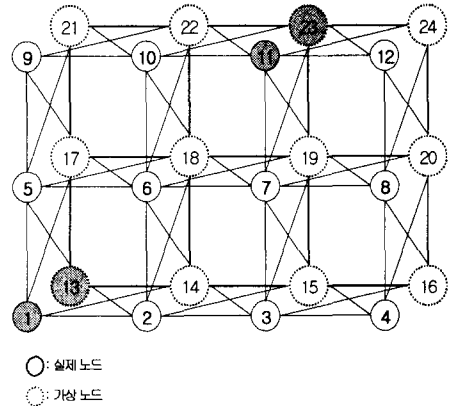
본 절에서는 3절에서 설명한 네트워크 재구성 단계 후 복잡도가 증가하게 되는 네트워크에서 다수개의 최적경로를 사용자가 제시하는 제한된 시간 내에 탐색할 수 있도록 진화프로그램을 제시하고자 한다.

4.1 경로생성 방법

3절에서 제안한 네트워크 재구성은 복잡한 교통 제약사항들을 고려하기 위하여 각 노드마다 가상노드를 추가하게 된다. 이러한 이유로 진화프로그램 내에서는 다음과 같은 사항을 고려해야만 한다.

첫 번째는 가상노드와 실제노드는 같은 노드를 의미하지만, 알고리즘 내에서는 다르게 인식된다. 두 번째는 한 경로에서 중복된 노드는 나타날 수 없도록 조정하였다. 이는 네트워크 재구성으로 인하여 생성된 가상노드가 실제노드를 대신하는 역할을 하여 제약 사항들을 이미 고려하기 때문이다.

[그림 4.1]에 주어진 교통네트워크에서 제안된 출발점 1과 도착점 11 사이의 경로 생성 방법은 [그림 4.2]에 나타나 있는 바와 같이 출발점(출발점의 가상노드)와 도착점(도착점의 가상노드) 두 곳에서부터 시작하여 각 노드의 인접 노드를 확률적 랜덤선택에 의해 노드를 계속적으로 연결해 나간다.

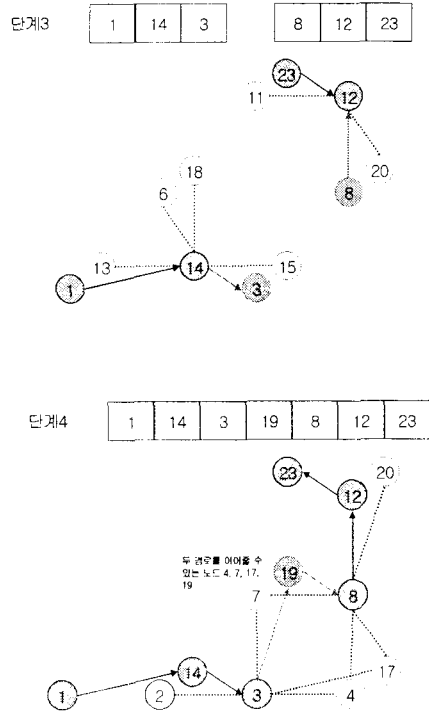
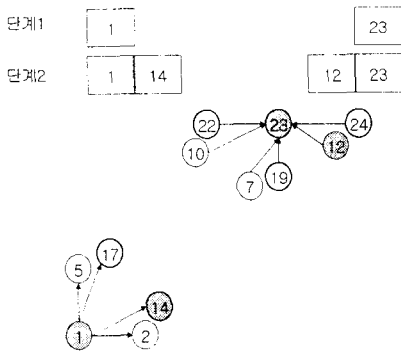


[그림 4.1]교통네트워크의 예

다시 말해서, [그림 4.2]의 단계1에서는 출발점과 도착점을 결정하는 단계이다. [그림 4.2]의 단계1의 두 노드 1과 23이 결정되고 나면, 노드 1과 23을 중간에서 연결해 줄 수 있는 노드가 있는지의 여부를 판단하게 된다. 즉, 단계 2와 같이 이웃된 연결 노드 중 하나씩을 랜덤 하게 선택한다. 여기서 중요한 한가지는 출발점에서 경로를 생성해 나갈 때는 출발점에서 나아가는 방향으로 가는 새로운 노드를 생성하지만, 도착점에서 경로를 생성할 때는 도착점으로 들어오는 방향으로 오는 새로운 노드를 생성하게 된다. 이러한 두 조건을 모두 만족하는 새로운 경로들 중 같은 노드가 발견되면, 이를 선택하여 출발점에서 생성된 경로와 도착점에서 생성된 경로를 연결할 수 있게 되는 것이다. 기존의 방법은 한 끝점의 연결 노드정보에서 다른 끝점을 찾는 방법이었으나, 여기서 사용한 방법은 한 끝점의 연결 노드 정보와 다른 끝점의 연결 노드 정보를 비교하여 찾는 방법이다. 이 때문에 또 다른 경로를 생성할 확률을 줄이고, 빠르게 염색체 생성을 할 수 있을 것이다.

노드 14는 1에서 갈 수 있는 노드들 [2, 5, 14, 17] 중 하나가 랜덤 하게 선택된 것이며, 노드12는 23으로 갈 수 있는 노드들 [7, 10, 12, 19, 22, 24] 중에서 선택된 것이다. 단계3에서도 마찬가지로 노드 14와 노드 12가 연결되어 있지 않다. 14에서 연결된 노드들

1, 3, 6, 13, 15, 18 을 구한다. 여기서 1은 이미 단계1에서 경로로 포함되었기 때문에 중복경로를 만들 수 없는 것에 의해 제외시켜야 한다. 또한, 도착점 생성경로의 끝점인 12로 연결될 수 있는 노드들 8, 11, 20, 23 중 23번 노드는 단계 1에서 경로에 포함되었으므로, 제외시킨다. 그리고 난 후, 이들 노드 정보들 중 같은 노드가 있다면, 두 경로는 이어지게 되는 것이고, 없다면, 이들 중 하나씩을 랜덤 하게 선택하여 경로를 생성해 나가는 것이다. 14에서의 연결 노드 정보는 3, 6, 13, 15, 18 이며, 12에서의 연결 노드 정보는 8, 11, 20 이다. 공통되는 노드가 없으므로 연결은 이루어지지 않으며, 새로운 경로를 하나씩 생성하게 된다. 14에서 연결될 노드는 3이, 12에서 연결될 노드는 8이 랜덤하게 선택되었다. 단계4에서는 3에서 연결할 수 있는 노드들 2, 4, 7, 14, 16, 19 을 구한다. 여기서도 단계 3에서 선택된 노드 14는 제외시킨다. 8로 연결할 수 있는 노드 4, 7, 12, 16, 19, 24 중 마지막으로 12는 제외되며, 이들 중 같은 노드 정보는 다음과 같이 4, 7, 16, 19 가 나타나게 되며, 이들 중 19가 랜덤 하게 선택되어, 두 경로를 이어 줌으로써 하나의 탐색체 생성을 마치게 된다.



[그림 4.2] 단계별 탐색체 생성 과정

4.2 표본선택 방법 및 적합도 평가

표본 선택과 적합도 평가과정은 (식1)(식2)(식3)(식4)를 사용하여 룰렛 휠 선택 방법(공성근외 4명,1996)을 적용한다. 각 탐색체 G_i ($i = 1 \dots Pop_size$ (탐색체수))에 대한 평가함수값(경로의 총 소요시간 또는 거리) $Z(G_i)$ 를 계산한다.

- Total_bit : i 번째 탐색체의 총 비트 수
- Node_i : i 번째 비트의 노드
- Dist_{ij} : i와 j사이의 소요시간 또는 거리

$$Z(G_i) = \sum_{k=2}^{Total_bit} Dist_{(Node_{k-1})(Node_k)} \dots(1)$$

최소값을 갖는 $Z(G_i)$ 가 좋은 값이므로 최소화 문제이다. 이는 룰렛 휠 방법을 적용하기 위해

$$F_i = \frac{1}{Z(G_i)}$$

로 바꾸어 최대화 문제로 접근한다.

- 개체 집단의 총 적합도를 계산한다.

교통제약을 고려할 수 있는 네트워크재구성과 다수최적경로들을 탐색할 수 있는 진화프로그램의 개발

$$F = \sum_{i=1}^{Pop_size} F_i \dots\dots\dots(2)$$

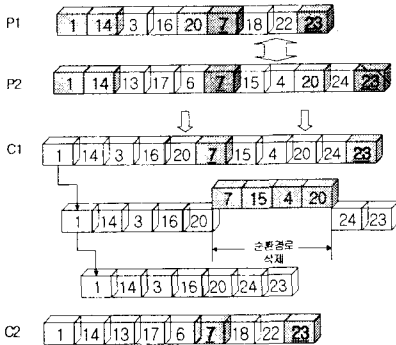
각 염색체에 대한 선택 확률 P_i 와 누적확률 q_i 를 계산 한다.

$$P_i = \frac{F_i}{F} \dots\dots\dots(3)$$

$$q_i = \sum_{j=1}^{Pop_size} P_j \dots\dots\dots(4)$$

4.3 교배

두 노드 교배연산자는 두 염색체간 공통 노드가 2개 이상이 되면 가능하며, 그 이상일 때는 노드를 랜덤 하게 선택하여 두 교배 노드를 선택하게 된다. [그림 4.3]을 보면 두 부모 염색체(P1, P2)의 공통노드가 1, 14, 20, 7, 23 이다. 랜덤 하게 결정된 두 교배노드 (7과 23) 사이의 서브경로를 상호 교환함으로써 새로운 두 자식 개체(C1, C2)를 생성할 수 있게 된다. 교배 후에 교배되는 서브경로의 노드와 교배되지 않는 서브경로와 겹치는 노드 즉, 순환경로가 발생될 수 있다.



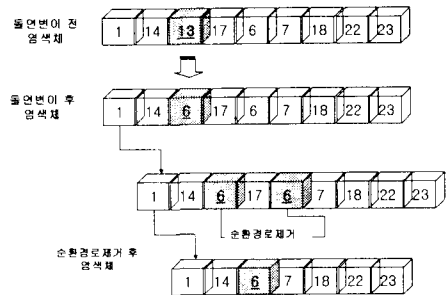
[그림 4.3] 교배와 순환경로의 제거

[그림 4.3]의 C1을 보면 앞의 교배되지 않은 부분 중 노드20이 교배되는 서브경로의 노드20과 겹치게 된다. 이러한 때는 순환노드 사이의 경로를 삭제한다. 이미 모든 제약사항은 네트워크 재구성된 데이터에서 고려해주었기 때문에 순환경로가 생기는 것은 의미가 없다.

4.4 돌연변이

돌연변이는 염색체의 각 비트마다 난수(R)를 발생시켜 돌연변이율 보다 작은 비트가 돌연변이 된다. 돌연변이 되는 방법에는 크게 두 가지가 있다. 첫 번째 방법은 어느 임의의 노드가 돌연변이 될 비트라면 그 노드 앞과 뒤에 있는 노드들을 연결할 수 있는 노드들 중 한 노드를 랜덤 하게 선택하여 돌연변이 시키는 방법이다. 두 번째 방법은

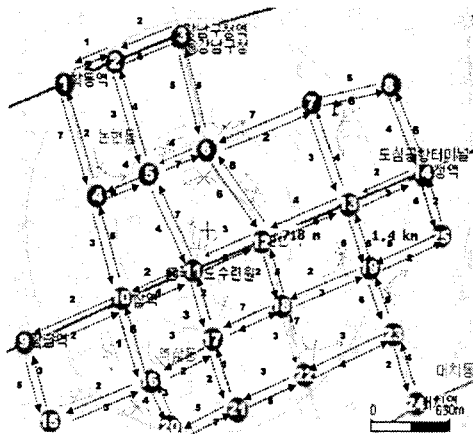
돌연변이 될 노드를 다른 경로 노드들로 대체시킴으로써 새로운 경로 개체를 생성하도록 하는 방법이다. 이때 돌연변이 될 특정노드 의 앞 노드를 출발점으로 하고 다음노드를 도착점으로 하여 IV.1절에서 설명한 경로 생성방법으로 새로운 개체를 생성하여 삽입하게 된다. 이러한 과정이 끝나면 순환경로가 생기는지 확인하여야 한다. [그림 4.4]를 보면 3번째 비트 노드 13의 전 노드 14 와 다음노드 17을 연결할 수 있는 노드 6으로 돌연변이 시킨 것이다. 돌연변이 후에 6과 6이 경로에 두 번들어감으로써 순환경로를 만들게 된다. 이때는 교배와 마찬가지로 역시 순환경로는 의미가 없으므로 이를 삭제해 버린다. 돌연변이 될 비트를 경로로 삽입할 때도 마찬가지로 이러한 순환경로가 생기면 공통노드 사이의 경로는 삭제해 버린다.



[그림 4.4] 돌연변이 와 순환경로 제거

5. 새로운 방법의 적용과 실험 결과

본 절에서는 3절과 4절에서 제안한 새로운 방법을 실제 네트워크에 적용해 보기로 한다. [그림 5.1]은 서울시 강남구 교통네트워크를 표시한 것인데, 각 교차로를 노드번호로 표시하였다. 회전 금지, U턴, P턴, U턴금지 등은 임의적으로 결정하여 표시하였으며, U턴은 각 교차로에서만 발생한다고 가정하였다. 교차로가 아닌 곳에서 U턴이 발생되도록 하려면 U턴이 발생하는 곳을 하나의 교차로로 인식하여 네트워크를 구성하도록 한다.



[그림 5.1] 서울시 강남구 양방향 교통망의 예

- 회전금지 발생하는 구간은 10개 지역 ([1,4,5], [3,6,7], [8,7,13], [12,11,5], [10,16,17], [13,19,18], [18,22,23], [20,16,15], [12,6,5], [23,18,25])
- U-턴 금지가 발생하는 구간은 6개 지역 ([6,5,6], [6,12,6], [11,17,11], [6,15,16], [16,10,16], [16,17,16])

또한, 각 호의 비용 값(시간 또는 거리)은 입력 데이터를 노드가 25개 이므로 25X25행렬로 표현하여 사용하며, 이 데이터는 네트워크 재구성과정을 거쳐 가상노드를 포함한 50X50행렬로 변화된다. 진화프로그램 알고리즘의 매개변수는 여러 실험에 따라 구해진 적절한 값으로 아래와 같이 초기화시킨 후 재구성된 네트워크 데이터를 가지고 진화 프로그램을 모의 실험하였다.

- 염색체 초기 개체군의 개체수 : 10
- 교배율 (P_c) : 0.3
- 돌연변이율 (P_m) : 0.2

진화프로그램을 이용하여 교통네트워크에서 다수 최적경로를 찾아내기 위해 4.1절에서 설명한 초기화 과정으로 다음과 같이 초기 개체군을 생성한다. 출발노드는 학동(노드1), 도착노드는 휘문중고(노드25)로 가정한다. 실제노드는 노드1에서 노드 25까지이며, 4절에서 설명하였듯이 노드1의 가상노드 26에서부터 노드 25의 가상노드 50까지 교통제약요소를 고려하는 네트워크 재구성을 위하여 추가된다.

[표 5.1] 초기 생성된 염색체군

개체 번호	생성 경로	F_i
-------	-------	-------

1	26 2 28 6 37 13 19 38 39 50	0.0250
2	1 2 26 29 5 36 30 11 42 46 22 48 49 23 19 38 14 13 44 25	0.0137
3	26 27 28 2 3 6 37 38 14 25	0.0294
4	26 29 35 41 42 16 20 21 45 46 47 48 44 13 19 43 37 38 14 50	0.0120
5	1 27 30 11 12 18 42 46 17 43 44 50	0.0313
6	1 2 30 29 26 4 10 9 35 11 5 27 28 31 37 38 14 25	0.0159
7	26 4 10 9 40 41 20 16 42 21 17 43 22 18 44 23 49 48 19 38 14 13 39 25	0.0119
8	1 4 10 34 35 11 37 38 39 13 19 25	0.0294
9	1 4 35 34 15 9 10 41 42 11 37 38 14 13 19 48 24 23 44 25	0.0158
10	26 2 30 29 5 11 35 36 37 18 17 46 22 48 44 38 14 50	0.0166
F (총 적합도)		0.2012

※ (각 염색체(i)는 연결되는 노드 번호로 이루어져 있으며, 25보다 큰 노드 번호는 [노드번호-25]의 가상노드이다)

4.2절의 (식1),(식2),(식3),(식4)를 적용하여 각 개체의 경로와 적합도 함수값을 계산한다. [표5.1]을 보면 가장 큰 F_i 를 가지는 5번째 염색체가 선택될 수 있는 확률이 다른 염색체 보다 크다. [표 5.2]에는 룰렛 휠 방법을 이용한 새로운 염색체 군이 나타나있다. 첫번째 난수 0.2184은 q_2 (0.0923)와 q_3 (0.338)사이에서 포함되므로 세번째 염색체가 선택이 되어 새로운 개체군의 첫번째 염색체로 재배치 된다. 이와 같은 방법으로 개체수 만큼 반복하여 새로운 개체군을 형성한다.

[표 5.2] 룰렛 휠 방식을 이용한 재배치 후 염색체

개체 번호	재배치 된 염색체 경로
1	26 27 28 2 3 6 37 38 14 25
2	1 4 35 34 15 9 10 41 42 11 37 38 14 13 19 48 24 23 44 25
3	1 4 10 34 35 11 37 38 39 13 19 25
4	26 2 28 6 37 13 19 38 39 50
5	26 27 28 2 3 6 37 38 14 25
6	1 2 30 29 26 4 10 9 35 11 5 27 28 31 37 38 14 25
7	26 29 35 41 42 16 20 21 45 46 47 48 44 13 19 43 37 38 14 50

교통제약을 고려할 수 있는 네트워크재구성과 다수최적경로들을 탐색할 수 있는 진화프로그램의 개발

8	1 27 30 11 12 18 42 46 17 43 44 50
9	1 2 26 29 5 36 30 11 42 46 22 48 49 23 19 38 14 13 44 25
10	1 4 35 34 15 9 10 41 42 11 37 38 14 13 19 48 24 23 44 25

재구성된 개체 집단은 교배연산을 행하며, 각각의 염색체에 대하여 범위[0과 1]사이의 난수(R)을 발생하여 만일 $R < 0.3$ (교배율)이면, 그 염색체를 교배에 참여시킨다. 난수를 통해 교배 연산에 참가하는 염색체는 개체6,7,8,10 이다. (선택 개체의 수가 홀수일 경우에는 하나를 제거하고, 두개(짝수 개)의 개체로 교배 쌍을 임의로 선택한다.) 첫 번째 쌍은 개체6과 개체7의 공통노드 29, 26, 35, 37, 38, 14중 노드 35와 38이 랜덤하게 선택되었으며, 첫 번째 쌍의 노드35와 노드38 사이의 경로를 서로 바꿈으로써 한쌍의 자손이 된다.

새 개체6 : 1 2 30 29 26 4 10 9 35 41 42 16

20 21 45 46 47 48 44 13 19 43 37

38 14 25

새 개체7 : 26 29 35 11 5 27 28 31 37 38 14
25

두 번째 쌍도 마찬가지로 방법으로 교배가 이루어진다. 교배를 마친 개체군은 돌연변이 연산자에 적용이 되며, 돌연변이는 각 염색체 모든 비트마다 난수(R)를 발생시켜, $R < 0.2$ 이면, 해당 비트를 돌연변이 연산 시킨다. 돌연변이 될 비트들이 결정되면 그 비트 앞뒤의 노드와 연결시킬 수 있는 여러노드들 중 하나를 랜덤 하게 선택하여 돌연변이 시키거나 돌연변이 비트의 앞과 뒤를 기점과 종점으로 하는 경로를 4.1절에서 서술한 방법으로 생성하여 돌연변이 시킨다. 첫 번째 염색체에서 돌연변이에 참가하는 비트 번호는 난수에 의해

2, 4, 8 로 결정되었으며, 첫 번째 염색체는 아래와 같은 돌연변이 연산이 된다.

새로운 염색체1 : 26 2 28 27 3 6 37 13 14 25

다른 염색체들도 첫 번째 염색체와 같은 방법으로 돌연변이를 하게 된다. 교배와 돌연변이를 마친 후 개체군은 [표 5.3]에 나타나 있다. 지금까지가 진화 프로그램의 한 세대가 진행된 것이다. 한 세대가 끝난후의 개체 집단은 다시 4.2절에서 설명한 식(1),식(2),식(3),식(4)에 의하여 평가되며, 재배치된다. 이는 총적합도(F)가 0.2505로 전세대의

총 적합도(F) 0.2012 보다 좋은 값을 얻을 수 있었다.

[표 5.3] 한세대 이후의 개체집단

개체 번호	경로	F_i
1	26 2 28 27 3 6 37 13 14 25	0.0294
2	1 4 10 41 42 11 37 38 14 13 19 48 24 23 44 25	0.0196
3	1 4 10 34 35 11 37 38 39 50 19 25	0.0285
4	26 2 28 6 37 13 19 38 39 50	0.0250
5	26 2 28 6 37 38 19 25	0.0344
6	2 30 29 10 9 35 41 42 16 45 46 47 48 19 25 44 43 37 38 14 25	0.0126
7	26 29 10 11 5 27 3 31 37 38 14 50	0.0208
8	1 27 30 11 37 18 42 46 17 43 44 50	0.0312
9	1 27 26 4 30 11 12 38 14 13 44 25	0.0270
10	1 4 35 34 15 9 10 41 42 11 37 38 14 13 44 25	0.0217
F (총 적합도)		0.2505

이제 선택과정을 다시 거치고, 유전 연산자를 적용하여 세대를 계속 진행해 나가면서 평가한다. [표 5.4] 에는 진화 프로그램 종료 후 마지막 개체군에 대한 평가 값 및 적합도 값이 나타나 있다.

[표 5.4] 진화 프로그램 종료 후 적합도 값과 최적경로

개체 번호	경로	F_i
1	1 4 35 36 12 13 14 25	0.417
2	1 27 30 36 37 18 44 25	0.1266
3	1 29 35 36 37 18 19 25	0.417
4	1 27 30 36 37 18 44 25	0.1266
5	1 4 10 36 12 18 19 25	0.417
6	1 4 10 11 12 18 44 25	0.417
7	1 27 30 36 37 18 44 25	0.1266
8	1 4 10 11 37 18 19 25	0.417
9	1 4 10 11 37 18 19 25	0.417
10	1 27 30 36 37 18 44 25	0.1266
F (총 적합도)		0.7563

마지막 개체군은 총 적합도(F)가 0.7563으로 처음세대와 확연한 차이를 보인다. 새로운 개체집단의 총 경로 길이의 합은 196 이고, 가장 좋은 값을 가지는 진화 프로그램의 염색체는

Dijkstra 알고리즘에 의한 최적경로 1 -> 2 -> 5

-> 11 -> 12 -> 18 -> 19 -> 25 (최적 값:18)와

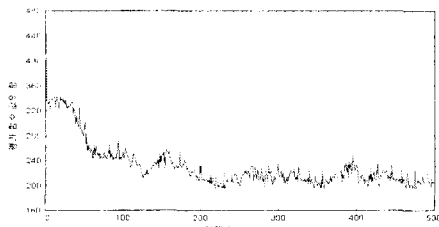
같은 값을 가진다. [그림5.1]에서 양방향 네트워크의 적합성을 검증하기 위해 노드25를 기점으로 하고 1을 종점으로 하는 최적경로

25->14->13->12->11->10->4->1 (최적 값:20)

는 노드1을 기점으로 하고 25를 종점으로 하는 최

적경로 1->2->5->11->12->18->19->25 (최적 값:18)와 다르게 구해진다. 이는 양방향 네트워크에 적용 할 수 있음을 알 수 있다. [그림5.2]에는

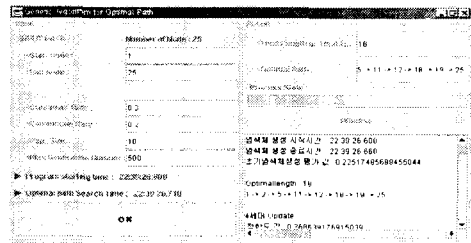
매 세대마다의 $\sum_{i=1}^{size} Z(G_i)$ 값을 표시 한 것이다. 이 그림을 보면 초기에는 세대의 변화에 따라 적합도 함수의 값이 민감하게 변화를 하는 것을 볼 수 있으며, 최적값에 가까이 도달한 뒤의 세대의 변화는 적합도 함수 값에 큰 변화를 주지 않고, 미세한 변화를 가지며, 개체집단을 형성하게 된다 는 것을 알 수 있다.



[그림 5.2] 세대별 각 탐색체가 가지는 평가함수 값 (시간 또는 거리)에 합의 변화

또한 서로 다른 다수개의 최적 경로를 얻을 수 있다. 이는 작은 네트워크를 통하여 본 논문에서 제시한 진화프로그램에 대하여 검증하였기 때문에 혼잡도가 큰 양방향 대규모 네트워크의 경우에도 본 연구에서 제시한 최적 경로 알고리즘이 제 역할을 할 수 있음을 유추할 수 있다.

아래 [그림 5.3]는 실제 사용자들이 서비스 받을 수 있도록 사용 가능한 최적경로안내시스템의 프로토타입을 Java(JFC)를 이용하여 설계한 것이다. [그림 5.1]에서 출발점(노드1)과 도착점(노드25) 그리고 진화프로그램에 관련된 매개변수를 입력하게 되면 최적경로에 관련된 정보가 출력된다.

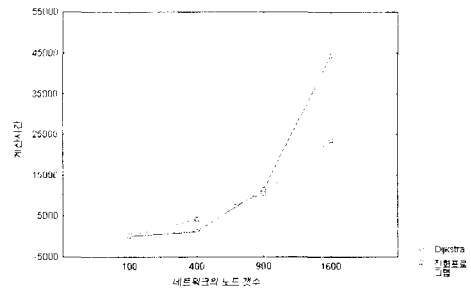


[그림 5.3] GUI를 이용한 최적경로안내시스템

최적경로는 사용자에게 색으로 표시 할 수 있으며, 유전자 알고리즘이 진행되면서, 알고자 하는 각종정보(초기개체 생성 시간, 최적경로, Update 정보)가 표시된다.

6. 기존 방법과의 비교분석

본 절에서는 네트워크에서 기존의 알고리즘인 Dijkstra의 최적 경로를 탐색하는데 걸리는 시간과 본 논문에서 제시하는 방법의 최적경로를 탐색하는데 걸리는 시간을 실제 노드수(N)를 변화시켜 가면서 비교 분석하여 진화프로그램에 대한 적합성과 효율성을 검증한다. 네트워크에 관한 데이터는 직접 노드생성기 프로그램을 만들어서 사용하였다. 시간측정은 각 알고리즘마다 약20회 정도 시간을 측정하여 평균시간으로 평가하였다.



[그림 6.1] 노드 수의 증가에 따른 Dijkstra 와 진화프로그램의 계산시간 변화

[그림 6.1]에서는 노드수(N)의 증가에 따른 각 알고리즘의 계산시간을 비교 한 것이다. Dijkstra를 이용하여 최적경로를 구하는데 소요되는 계산수행시간은 노드수가 증가함에 따라 지수 함수 형태로 증가하지만, 진화프로그램을 이용한 계산수행시간은 선형함수와 비슷한 형태로 증가함을 알 수 있다. 계산시간을 보면 노드가 100개 일때와 400개 일때는 Dijkstra가 최적경로 찾는데 더 빠르다는 것을 알 수 있다. 그러나 900개 일때부터 진화 프로그램이 Dijkstra보다 빠르게 최적 경로를 구하는 것을 볼 수 있다. 이는 교통네트워크의 노드수가 약 900개 이상이면 진화프로그램이

교통제약을 고려할 수 있는 네트워크재구성과 다수최적경로들을 탐색할 수 있는 진화프로그램의 개발

Dijkstra 알고리즘 보다 더 효율적이라는 것을 의미한다. 따라서, 교통네트워크의 노드수가 증가되고 복잡도가 증가할수록 본 논문에서 제시하는 방법의 효율성이 증대된다.

7. 결론

본 논문에서는 양방향 대규모 네트워크에서 네트워크 재구성과 진화프로그램을 사용하여 여러 가지 교통 제약들을 네트워크에 암묵적으로 고려하면서 사용자가 제시하는 제한된 시간 내에 다수 최적경로를 산출해 낼 수 있는 새로운 방법을 제안하였다.

본 논문에서 제안하는 최단경로 탐색방법의 특징은 네트워크 재구성을 통하여 대부분의 교통제약 상황을 고려하였다. 네트워크 재구성 방법을 사용하여 대부분의 교통제약 상황을 고려할 수 있음에도 불구하고 이 방법을 사용함으로써 노드 수 증가로 복잡도가 증가하였다. 본 논문에서는 이 문제점을 개선하기 위해서 네트워크 재구성 방법하에서 사용자가 제시하는 제한 시간내에 다수 최적 경로들을 탐색할 수 있는 진화프로그램을 개발하였다. 본 논문에서 제안하는 진화프로그램의 특징은 새로운 경로 생성 방법을 사용하여, 경로 생성 과정에 있어서 확률적으로 빠르게 생성 할 수 있도록 조정하였으며, 경로의 표현을 경로 노드 자체를 사용함으로써, 알고리즘 내에서의 수행시간과 기억용량을 효율화하였다. 또한 실제 교통네트워크처럼 양방향 데이터를 사용할 수 있어, 실시간으로 교통정보를 수집할 수 있다면 이는 실제 교통네트워크에 적용 할 수 있다는 장점이 있다. 본 논문에서 제안하는 방법은 기존의 알고리즘을 사용하여 얻은 최적경로와 계산시간을 비교 확인함으로써 복잡성이 높은 실제 교통네트워크에 보다 효율적으로 적용할 수 있음을 검증하였다.

위한 유전 알고리즘에 근거한 최적 경로 탐색 알고리즘”, *제어 자동화 시스템공학 논문지*, 1997.

- [6] 최기주, “U-turn을 포함한 가로망 표현 및 최단경로의 구현”, *대한교통학회지* 제13권 제3호, 1995.
- [7] 홍순정, 김병하, “네트워크 재구성을 통한 개선된 최단경로탐색 방법과 실제 교통 네트워크에서의 적용”, *제5회 SK Telecom, M&M 논문집*, 2000.
- [8] Gen, M., Cheng, R. and Wang, D., “Genetic Algorithms for Solving Shortest Path Problems”, *Proceedings of the 1997 IEEE International 97 Magnetics Conference*, 1997.
- [9] Voicu, L.I. and Myler, H.R., “Cloning operator and its applications”, *Proceeding of the Applications and Science of Computational Intelligence*, 1998.
- [10] Seo, K. and Choi, G., “The Genetic Algorithm Based Route Finding Method for Alternative Paths”, *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, 1998.

참 고 문 헌

- [1] 공성관 외 4명, “유전자 알고리즘”, 그린, 1996.
- [2] 김성수, 차영민, “첨단 교통 정보 시스템의 최적경로 알고리즘 개발”, 산업기술 연구소, 강원대학교, 2001.
- [3] 김성수, 전영주, 차영민, “교차로에서의 좌회전금지, U-turn, P-turn을 고려한 개선된 Dijkstra Algorithm의 관한 연구”, 산업기술 연구소, 강원대학교, 2001.
- [4] 박찬규, 박순달, 진희재, “교차로 제약과 지연이 있는 네트워크에서 최단경로 탐색”, *한국경영과학회지*, 1998.
- [5] 최규석, 우광방, “지능형 주행 안내 시스템을