

## 유전자 알고리즘과 GIS를 이용한 대중교통 경로선택에 관한 연구

### Route Selection in the Network of Public Transportation using the GA and the GIS

전철민<sup>1)</sup>

Jun, Chulmin

#### Abstract

GIS-based applications for route guidance are increasingly developed recently, but most of them are for self-driven cars. Although some of them are intended for public transportation, they show limitations in dealing with time constraints problems taken place in transfer areas. Developing a public transportation guidance system requires the following aspects: (i) people may change transportation means not only within the same type but also among different modes such as between buses and subways, and (ii) the system should take into account the time taken in transfer from one mode to the other. This study suggests the framework for developing a public transportation guidance system that generates optimized paths in the transportation network of mixed means including buses, subways and other modes. For this study, the Genetic Algorithms are used to find the best routes that take into account transfer time and other service-time constraints. The method for constructing the data structure in the GIS was also suggested.

Keywords : Genetic Algorithm, GIS, Shortest path, Public transportation, Time window

#### 요 지

최근 개발되고 있는 GIS 기반 교통경로 안내 시스템은 자가 운전자를 위한 시스템이 대부분이다. 대중 교통에 적용된 몇몇 사례들은 환승시에 고려해야 할 시간적 제약 요소들을 반영하고 있지 않아 한계점을 지니고 있다. 대중 교통을 위한 시스템 개발에는 — (1) 이용객들이 목적지까지 이동하면서 서로 다른 교통수단간에 환승할 수 있다는 사실과 (2) 환승 시에는 이동시간, 대기 시간 등의 환승소요 시간이 있다는 것 —이 반영되어야만 한다. 본 연구에서는 다수의 교통수단이 혼재된 대중교통 네트워크에서 최적의 경로를 산출하는 방안을 제시하였다. 시간 제약과 기타 서비스 가용시간 등을 고려한 최적해 또는 이에 근사한 해를 찾기 위해 유전자 알고리즘(Genetic Algorithm)을 사용하였으며, 이를 GIS에서 구현하기 위한 데이터 구조를 제시하였다.

핵심용어 : 유전자알고리즘, GIS, 최적경로, 대중교통, 시간창

#### 1. 서 론

GIS의 응용분야가 확대되면서 인터넷 또는 모바일 기반의 교통경로안내 시스템이 등장하고 있다. 그러나 이들은 대부분 도로를 자유롭게 운행할 수 있는 자가운전자만을 위한 시스템들이다. 최근 대중교통을 위한 경로안내시스템이 소개되고 있으나 수단간 환승문제가 고려되지 못하

고 있다는 한계점을 가지고 있다. 환승문제는 주로 환승시의 시간적 제약을 의미하는 것으로서, 이동시간, 대기시간 및 서비스 가용시간 등에 의해 환승이 제한되어지는 것을 말한다. 여러 수단이 혼재되어 있는 복합적인 대중교통 네트워크에서 현실적인 경로안내 시스템을 구축하기 위해서는 환승시의 시간제약 문제가 고려되어야 한다. 본 연구에서는 이러한 시간제약 요소들을 염두한 대안적인 대중교통

1) 정회원 · 서울시립대학교 지적정보학과 부교수(E-mail:cmjun@uos.ac.kr)

안내시스템을 구축하는 방안을 제시하고자 한다. 본 연구에서는 교통 수단들의 복합적인 조합으로 경로가 구성되는 상황에서 최적의 경로를 찾기 위한 방법으로 유전자 알고리즘(Genetic Algorithm, 이하 GA)을 사용하였다. GA는 생태계의 진화과정, 즉 자연선택(natural selection)과 유전법칙을 모방한 확률적 최적해 탐색 기법이다. 본 연구에서는, 일반적 최적화(optimization)가 불가능한 대중교통망에서 최적해를 찾기 위해 GA를 변형하여 사용하였다. GA의 경로탐색과정을 실험하기 위해 GIS를 이용하여 소규모의 가상의 네트워크를 구축하고 GA과정을 프로그램하였다.

## 2. 최적경로 관련 기존 연구 고찰

최근에 ITS(지능형 교통체계) 및 GIS를 응용한 연구, 개발이 활발히 이루어지면서 현실성을 반영한 최단경로 탐색 알고리즘에 대한 중요성이 부각되고 있다. 최단경로 탐색 알고리즘은 교통 수요 분석이나 경로 안내 시스템에서 사용되는 중요한 핵심기술로서, 출발지에서 목적지까지의 경로 중에서 가장 경제적인 경로를 탐색하는 기법이다. 이와 같은 최단경로 탐색문제는 이미 1950년대에 활발히 연구되어 왔으며(Bellman 1958, Moore 1959, Dijkstra 1959), 이 중에서 Dijkstra의 알고리즘과 이에 기반한 변형된 알고리즘이 가장 널리 연구, 개발되었다.

Dijkstra 알고리즘은 이전 노드(predecessor node)를 검색하여 순차적으로 경로를 탐색해 가는 수형망 알고리즘(tree building algorithm)의 대표적인 예이다. 이러한 순차적 탐색기법은 — i) 동일한 노드가 최단경로상에 두 번 이상 포함될 수 없다는 것과, ii) 기점으로부터 각 노드까지의 최단경로를 단계적으로 산출하고, 최종적으로 이들을 연결하게 되면 기-종점간 최단경로가 구축된다 —는 가정에 기초하고 있다.

그러나 첫번째 가정에 의하면, U-Turn 및 P-turn 등과 같은 경우에서의 동일 노드의 방문을 허용치 않기 때문에 현실적인 최단경로를 탐색하지 못한다는 제약을 가진다. 그래서 이들 알고리즘을 사용하기 위해 기존 가로망에 가상의 노드와 링크를 추가하는 방법을 써서 Turn의 문제를 해결한 연구도 다수 소개되었다(Gallo and Pallottino 1984). 그러나 이와 같이 자료를 구조적으로 변형하는 방법은 노드와 링크 수의 현저한 증가로 컴퓨터 처리 속도가 저하되는 문제와 더불어 구조 변경에 드는 시간과 노력 또한 매우 크다는 점이 문제로 지적될 수 있다.

또한 위의 두번째 가정에 의하면, 기점으로부터 최단경

로상의 어떤 노드까지의 경로도 최단경로의 부분 경로가 된다는 것을 의미하지만, 이러한 가정은 다음 절에 소개될 환승노드가 있는 대중교통망에서는 성립이 되지 않는다. 이러한 환승문제를 고려하기 위해서 Turn 문제를 해결하기 위한 방안과 유사한 방법으로 기존 교통망에 환승노드와 환승링크를 추가하는 방법을 사용한 연구도 소개되었다(Spiess and Florian 1989). 이 역시 기존의 순차적 탐색 알고리즘을 그대로 사용할 수 있다는 장점을 지니지만 교통망의 확장에 많은 시간이 소비되는 단점을 가진다.

이와 같이 노드기반에서 순차적 탐색법을 적용하기 위해서는 교통망을 변형함으로써 수단간 환승행태를 교통망에 표현해야 한다는 점이 경로안내 시스템을 구현하는데 있어서 비현실적인 요소로 작용한다. 또한 기존의 순차적 탐색법을 사용하여 버스, 마을버스, 지하철 등 다수의 수단이 혼재된 네트워크에서의 환승을 고려한다는 것은 매우 복잡할 뿐만 아니라 모든 노드까지의 단계별 계산과정으로 인해 수행속도도 저하되는 문제가 있다.

본 연구에서는 교통망을 구조적으로 변형하지 않고도 대중교통망에서 최적경로를 산출할 수 있는 대안적인 방안으로써 유전자 알고리즘(Genetic Algorithm)을 적용하였다. GA가 교통네트워크에 적용된 예는 우선 그래프이론에서 전통적으로 다루어지고 있는 순회판매원문제(traveling salesman problem)에서 찾을 수 있는데(Michalewicz 1994), 여기에서는 경유점들의 연결을 하나의 해, 즉 염색체로 표현하고 있다. 이 밖에도 노드가 아닌 링크들의 연결을 하나의 염색체로 표현한 연구도 찾아 볼 수 있다(Choi et al. 1997, Chang 2000). 이러한 링크 기반의 GA 표현은 각 염색체의 평가를 위해 네트워크 상의 모든 노드와 링크를 고려해야 하므로 계산수행시간이 커진다는 단점이 있다. 그래서 본 연구에서는 노드 기반의 GA 표현방법을 채택하였으며, 이후에 소개되는 바와 같이 ‘연속적’이어야 한다는 교통 네트워크의 특성으로 인해 몇 가지 연산자들을 변형하여 구현하였다.

이제까지 순차적 탐색 또는 GA관련 경로탐색 연구는, 대부분이 알고리즘의 분석, 개발에 초점을 두고 있기 때문에 실험을 위해 가상의 네트워크를 구성하는 경우가 대부분이었다. 본 연구에서는 GA 기법에 기반한 최단경로 탐색시에 GIS 데이터를 사용하는 방안을 제시하였으며, 기존의 교통관련 연구에서 일부 수행되오던 GA를 GIS 데이터 활용에 맞도록 대폭 수정하였다. 이렇게 구축되어 있는 GIS데이터를 사용함으로써 데이터구축시의 시간 및 노력을 대폭 절감할 수 있고, 안내시스템을 구현할 경우에

필요에 따라서는 건물, 필지 등의 기타 레이어를 간단하게 중첩시켜 표현할 수도 있다는 장점이 있다. 본 연구에서는 GIS와 함께 관계형 데이터베이스를 이용함으로써 다수의 교통수단과 정류장, 환승지역 간에 존재하는 복잡한 관련성을 효과적으로 표현하는 방안을 소개하였다.

### 3. 시간적 제약문제

교통 네트워크에서 최단경로를 다루는 연구 또는 개발 사례들은 대부분 네트워크를 자유 운행할 수 있는 자가운전자들을 대상으로 하고 있다. 이러한 경우에는 시간적 제약문제, 즉 특정 노드에서 출발가능한 시간에 대한 제약을 고려할 필요가 없다. 그러나 네트워크가 버스, 지하철, 기차, 배, 비행기 등 스케줄링된 다수의 수단으로 이루어져 있고, 그들 간에 환승이 발생한다면 환승시의 시간적 제약요소가 반영되어야만 한다. 교통분야에서 시간제약문제는 “시간창(time window)”이란 용어로 언급되어 왔으며, 이는 하나의 노드에는 정해져 있는 출발시간들이 존재하며, 이 노드에서의 출발은 이들 시간 중에만 허용된다는 것을 가정한다(Desrochers and Soumis 1988). 환승이 발생하는 이러한 노드에서는 각 가능 모드별로 출발시간이 제한되므로 최소시간 경로를 탐색하기 위해서는 이들 출발시간을 비교해야 한다.

그림 1은 시간제약 문제를 예시하고 있다. 여행객이 Mode A를 이용하여 노드 2에 도달했다면 그는 Mode B 또는 Mode C로의 환승이 가능하다. 그림 1 하단에는 노드

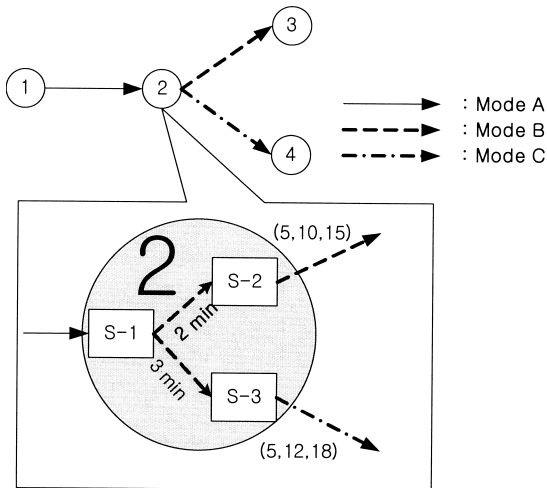


그림 1. 환승지역에서의 시간제약 문제 예시

2의 확대된 형태를 보여준다. 대체로 환승지역에서 정류장 간 이동에는 시간이 소요된다. 예를 들어, 정류장 S-1에서 S-2로, S-1에서 S-3로의 이동시간은 각각 2분과 3분이라고 가정한다. 그리고 S-2, S-3 정류장에서의 출발 가능시간은 각각 (5분, 10분, 15분)과 (5분, 12분, 18분)으로 스케줄되어 있다고 가정한다. 만약 여행객이 정류장 S-1에 4분 시점에 도달한다면 S-2에는 6분에 도착할 수 있을 것이다. 이때 10분에 출발하는 Mode B를 타기 위해서 4분의 대기시간이 필요해진다. 만약 Mode C로 환승하고자 한다면 S-3까지는 7분에 도착할 수 있으며 Mode C의 12분에 출발하는 차를 타기 위해서 5분을 기다려야 한다.

한편, S-1에 14분에 도달하는 경우를 생각해 보자. 이 때 여행객은 Mode B로의 환승은 불가능하다. 왜냐하면, 16분에 S-2에 도착할 수 있으며, 이 때는 이미 마지막 출발시간인 15분을 초과하기 때문이다. 하지만 S-3에서 1분을 기다린 후 Mode C의 18분 출발차로의 환승이 가능하다. 이처럼 시간창(time window)은 특정 노드에서의 가장 빠른 출발시간 및 가장 늦은 출발시간에 대한 제약을 포함한다.

### 4. 유전자 알고리즘

유전자 알고리즘(GA)은 적자생존과 유전학의 법칙을 모방한 탐색 알고리즘으로서 유전학에서 사용하는 용어를 사용한다. GA에서는 해결하고자 하는 문제에 대한 후보해들을 문자의 배열로 구성되는, 문자열(string) 또는 염색체(chromosome)로 표현한다(그림 2). 유전자 알고리즘은 다수의 문헌에 소개되고 있으므로 그 원리에 대한 언급은 생략하도록 하겠다.

본 연구에서는 다수의 교통수단이 존재하는 교통 네트워크에서 유전자 알고리즘을 적용하여 최적 해를 탐색하는 방안을 제시하기 위해 그림 3과 같은 간단한 네트워크를 구성하였으며, 이를 기반으로 초기화, 적응도 평가, 개체선택,

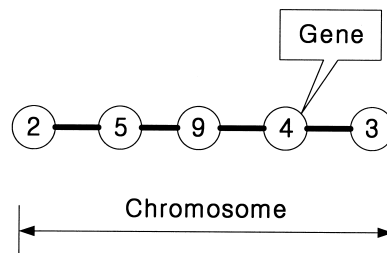


그림 2. GA에서 후보해의 표현

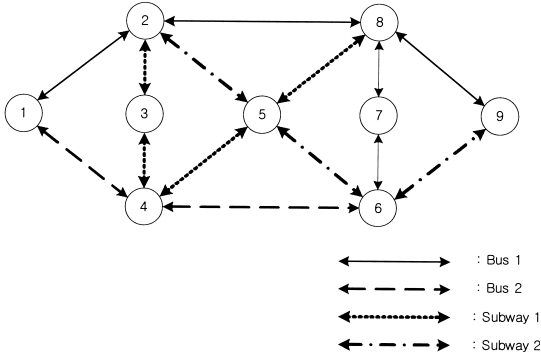


그림 3. 다수의 모드가 존재하는 네트워크의 예

유전연산자(돌연변이, 교배) 등에 대한 설명을 하고자 한다.

### 개체표현(Representation)

하나의 염색체는, 여행자의 시점에서 목적지까지 이르는 노드들을 연결하여 표현할 수 있다. 예를 들어, 시점이 ①이고 종점이 ⑨인 경우, 하나의 후보해 또는 염색체는 ①부터 ⑨까지 연결된 노드들의 배열이 된다.

### 초기화(Initialization)

미리 정해 놓은 모집단 크기만큼 염색체들의 최초 모집단이 생성된다. 하나의 노드에서 무작위로 다음 노드를 선택하는 방식으로 하나의 염색체가 형성된다. 예를 들면, 염색체들은 다음과 같은 방식으로 생성된다.

$C_1 = (1, 2, 8, 9)$   
 $C_2 = (1, 4, 5, 6, 9)$   
 $C_3 = (1, 2, 5, 6, 7, 8, 9)$   
 ...

### 적응도 평가(Evaluation)

평가함수는 후보해들의 적응도를 평가하는 역할을 한다. 염색체 C에 대한 적응도(fitness) 함수  $eval$ 은 다음과 같이 염색체 C에 있어 시점에서 종점까지의 총소요시간으로 설정하였다.

$$eval(C) = gene\_travel\_time(x),$$

k개의 노드로 이루어진 i번째 염색체(경로)를  $C_i = (n_1, n_2, \dots, n_k)$ 라 할 때,  $gene\_travel\_time(C_i)$ 를 계산하기 위해 다음과 같은 변수를 정의하였다.

- $ARR(MD_m(n_{ij}))$ : m번째 Mode(교통수단)의 j번째 노드 도착시간
- $DEP(MD_m(n_{ij}))$ : m번째 Mode의 j번째 노드 출발시간
- $MD_m(n_{ij}, n_{(j+1)})$ : m번째 Mode의 j번째 노드와 j+1번째 노드 사이를 운행하는 교통수단
- $T(MD_m(n_{ij}, n_{(j+1)}))$ : m번째 Mode의 j번째 노드에서 j+1번째 노드까지의 차량이동 시간
- $WALK_{ij}(MD_{m1}, MD_{m2})$ : j가 환승이 가능한 노드이고 이 노드에서 Mode m1에서 Mode m2로 환승이 이루어졌을 경우의 도보이동 소요시간
- $WAITING_{G(i,t)}(MD_{m2})$ : j가 환승이 가능한 노드이고 이 노드에서 시간 t에 Mode m2를 타기 위해 기다리는 시간
- $TRANS_{ij}(MD_{m1}, MD_{m2}) = WALK_{ij}(MD_{m1}, MD_{m2}) + WAITING_{G(i,t)}(MD_{m2})$ : j가 환승이 가능한 노드일 경우, 이 노드에서 Mode m1에서 Mode m2로 환승 시 소요되는 시간
- $TR_s$ : 환승노드 s
- $S(MD(TR_s)) = \{MD_1, MD_2, \dots, MD_r\}$ : 환승노드 s에 존재하는 Mode의 집합
- $TS(n_{ij}, MD_m) = \{ts_1(n_{ij}, MD_m), ts_2(n_{ij}, MD_m), \dots, tsp(n_{ij}, MD_m)\}$ : 환승 가능 노드 j에서 환승하고자 하는 Mode m의 서비스 가능 시간대의 집합

이 때, 시점에서 종점까지의 총 소요시간은 다음과 같이 산출된다. 기점에서  $j=0$ ,  $DEP(MD_1(n_0))=0$ 으로 초기화하고, 이 이후 마지막 노드  $n_k$ 까지 도달할 때까지의 총 소요시간의 산출을 다음과 같이 의사코드(pseudo code)로 나타낼 수 있다.

```

j++;
s=1;
WHILE(j <= k AND s > 0)
  ARR(MD_m(n_j))=DEP(MD_m(n_{j-1}))+T(MD_m(n_{j-1}, n_j));
  IF(j≠k)
    IF(n_j=환승가능노드)
      WHILE(S(MD(TR_s))이 존재 AND 환승미발생)
        s++;
        ARR(MD_m(n_j))+= TRANS_{ij}(MD_{m-1}, MD_m);
        IF(ARR(MD_m(n_j))>ts_p(n_j, MD_m))
          s--;
          S(MD(TR_s))-= {MD_m};
        ELSE
          환승발생;
          DEP(MD_m(n_j))=ARR(MD_m(n_j));
      j++;

```

## 개체선택(Selection)

개체선택은 현재 모집단을 변화시켜 새로운 모집단을 형성하는데 필요한 준비 과정이다. “좋은” 염색체를 보존하기 위해, 적응도 함수에 의한 평가에서 좋은 성적을 받은 상위 일정 부분은 “선택”되어 다음 세대로 복사되어지며 이들은 교배나 돌연변이 과정을 거치지 않게 된다. 이렇게 함으로써 엘리트(elite)로 지칭되는 “좋은” 해들이 탐색 과정 중에 삭제되는 것을 방지할 수 있게 된다. 개체선택은 엘리트만을 선택하는 것이 아니라, 교배나 돌연변이 과정을 겪게 될 염색체들을 선택하는 과정도 포함한다.

## 유전 연산자(Genetic Operators)

각 세대에서 보존 대상을 제외한 나머지 중 일정 부분의 염색체들은 두 가지 유전 연산자, 즉, 교배(crossover)와 돌연변이(mutation)를 통해 변형된 자손을 생성한다. 교배는 두 개의 부모 염색체 특성을 결합하여 양 측의 염색체 일부분을 교환함으로써 부모를 닮은 두 개의 자식 염색체를 만드는 연산자이다. 예를 들어, 부모 염색체가  $C_2$ 와  $C_3$ 라고 가정한다면, 먼저 이들의 공통된 노드(예를 들어, Node ⑤)가 선택된다. 그리고 이 공통 노드들의 뒷부분이 서로 교환되어 새로운 자손을 만든다. 이렇게 공통노드를 선택하는 이유는, 부모 염색체 양측의 임의의 부분을 상호 교환할 경우, 노드의 연속성이 깨질 가능성이 크기 때문에 본 문제에 맞도록 변형시킨 것이다.

$C_2 = (1, 4, \underline{5}, 6, 9)$   
 $C_3 = (1, 2, \underline{5}, 6, 7, 8, 9)$   
 $\downarrow$   
 $C'_2 = (1, 4, \underline{5}, 6, 7, 8, 9)$   
 $C'_3 = (1, 2, \underline{5}, 6, 9)$

돌연변이는 유전자들을 무작위로 바꾸는 과정이다. 교배에서와 마찬가지로, 본 교통문제에서 유전자, 즉, 노드를 무작위로 바꿀 경우 연속이 깨진 노드들의 배열이 나올 가능성이 높다. 그래서 본 문제에 맞춰 돌연변이 과정도 변형하였다. 돌연변이의 대상으로 선택된 염색체에서 어떤 특정 노드가 역시 돌연변이의 대상으로 선택되는 경우, 이 노드를 임시의 “시점”으로 생각할 수 있다. 그래서 이 노드를 시점으로 하여 종점까지 다시 부분적인 염색체를 생성하는 것이다. 예를 들어,  $C_2$ 가 선택되고, 노드 ⑤가 돌연변이 대상 노드로 선택된다면 이 노드를 기점으로 하여 노드 ⑨까지 이르는 부분적인 해를 도출할 수 있다. 이렇게 생성된 부분 염색체를 노드 ⑤다음에 대체해 넣는 것이다. 다음

에는 염색체  $C_2$ 에서 노드 ⑤번 이후가 돌연변이 된(재 생산된) 경우를 보여 준다.

$C_2 = (1, 4, \underline{5}, 6, 9)$   
 $\downarrow$   
 $C'_2 = (1, 4, \underline{5}, 2, 8, 7, 6, 9)$

돌연변이의 결과 염색체의 평가값이 좋아지기도 하지만, 이 예에서 본 바와 같이, 더 나빠지기도 한다. 다음의 그림 4는 전체 GA 과정을 의사코드(pseudo code)를 이용하여 요약한 것이다.

```

t : 세대
Tmax : 목표 총 세대수
P(t) : t 세대의 모집단

GeneticAlgorithm
{
    t = 0;
    initialize P(t);           // 모집단 생성 및 초기화
    evaluate P(t);             // 평가

    while( t <= Tmax )        // 목표 세대수가 될 때까지
    {
        t = t + 1;
        create P(t) from P(t-1); // 새로운 세대 형성
        {
            select P(t)e       // 우수개체 보존
            alter P(t)c & P(t)m; // 유전연산에 의한 변형
                                   // (교차, 돌연변이)
        }
        evaluate P(t);         // 적응도 평가
    }
}

```

그림 4. 유전자 알고리즘의 흐름 요약

## 5. GIS에서의 데이터 구축문제

다수의 교통수단이 존재하는 대중교통망을 GIS로 구축하는 문제는 자가운전자 또는 단일 수단만을 가정했을 경우보다 매우 복잡하다. 이를 위해서는 다음과 같은 사항들이 고려되어야 한다.

- 여행객은 시점에서 목적지까지 다양한 수단의 조합을 이용할 수 있다.
- 환승지역에서는 시간제약문제가 있다(이동시간, 출발 가능시간, 대기시간 등).
- 네트워크의 특정 부분은 다수의 모드(수단)가 공유할 수 있다.
- 버스정류장이나 지하철정류장은 교차로 이외의 지역에 위치할 수 있다.

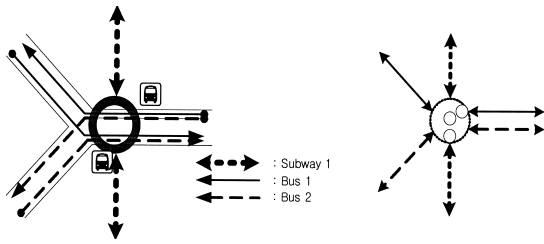


그림 5. 환승지역의 실제상황과 모델링

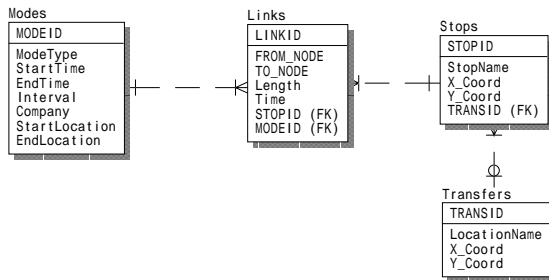


그림 6. GIS 데이터 구축을 위한 개체 관계도

- 노드와 링크간에는 위상관계(topological relationship)가 존재한다. 이 때 노드는 정류장을 의미하며, 링크는 버스나 지하철 등의 루트를 말한다.

- 버스의 경우, 한 대 이상의 버스가 정류장을 공유할 수 있다.
- 지하철정류장이나 버스정류장들이 도보로 이동 가능할 정도로 근접해 있을 경우, 이러한 지역을 환승지역이라고 할 수 있다.

그림 5는 단순화시킨 환승지역의 예를 보여준다. 환승지역을 하나의 노드로 취급하도록 모델링하였으나, 실제로는 이러한 환승노드는 다수의 정류장과 역을 포함하게 된다. GIS에서 데이터를 구축할 때, 환승노드와 루트간의 위상관계가 적절히 구현되어야 한다.

그림 6은 모드, 링크, 정류장, 환승지역간의 관계를 관계형 DB로 모델링한 예를 보여준다. 이들은 GIS내에서 속성 테이블로 구축되며, 상호 관계가 주키(primary key)와 외래키(foreign key)를 이용하여 유지되게 된다.

본 연구에서는, 이와 같은 테이블간의 위상관계를 GIS로 구축하고 테스트 하기 위해 ArcINFO의 Route System의 구조를 이용하였다. Route System을 이용하게 되면, 하나의 네트워크 세그먼트 상에 다수의 교통 모드를 구축할 수 있으며, 버스정류장들을 이벤트 테이블로 나타낼 수 있다. 그림 7은 ArcINFO내의 관련 테이블들 중에서 주된 몇 개의 테이블들을 보여준다. 루트속성테이블에는 각 노선별로

ROUTE ATTRIBUTE TABLE

TR#	TR-ID	MODE-ID	MODENAME	START	END	INTERVAL	VELOCITY	FARE
1	1	1	subway-1	300.00	1380.00	2.00	70.00	600
2	2	1	subway-2	300.00	1380.00	2.00	70.00	600
3	3	1	subway-3	300.00	1380.00	2.00	70.00	600
4	4	2	bus-10	300.00	1380.00	5.00	50.00	500
5	5	2	bus-20	300.00	1380.00	5.00	50.00	500
6	6	2	bus-30	300.00	1380.00	5.00	50.00	500
...	...	...	...	...	...	...	...	...

STOPS.EVE

tr-id	location	stop	trans-id
1	0	1	11
1	2000	2	12
1	4000	3	8
1	6000	4	9
1	8000	5	10
2	0	1	1
2	2000	2	2
2	4000	3	0
2	6000	4	0
...	...	...	...

TRANSTIME.DAT

trans-id	frt	frt-st	trt	trt-st	time
1	1	1	5	1	3
1	1	13	5	1	3
2	2	2	6	8	3
7	2	12	3	1	3
8	1	3	3	2	3
8	6	6	1	3	3
8	6	6	3	2	3
...	...	...	...	...	...

그림 7. ArcINFO에서의 데이터 구축의 예

서비스 가능 시간과, 배차간격, 평균속도, 요금 등을 넣을 수 있으며, 경로들에 대한 적응도를 산출할 때 이들을 이용하여 된다. 버스정류장 이벤트 테이블에서는 이들의 위치와 소속된 환승 지역에 관한 정보를 담게 되고, 또한 외부 속성 테이블로서 환승지역 테이블에는 특정 환승 지역에서 가능한 환승의 조합과 도보이동시간 등이 저장될 수 있다.

## 6. GIS를 이용한 구현사례

GA탐색과정을 테스트하기 위해 그림 8과 같은 가상의 네트워크를 구성하였다. 단순한 네트워크이지만 추후 실제 데이터를 이용한 구현을 염두하여 앞 절에서 고려한 중요한 사항들이 반영되도록 구축하였다. 우선, 다수의 bus와 지하철이 상호 환승이 가능하도록 하였다(버스 3개, 지하철 3개). 또한 동일 모드가 이용하는 노드들은 환승이 발생하지 않으나, 서로 다른 모드가 공유하는 노드들은 환승지역에 포함되도록 구성하였다. GIS데이터는 ArcInfo 커버리지로 구축하였고, 속성테이블은 앞 절의 개체 관계도에서 언급한 바와 같이 상호 관계를 가지도록 구성하였다. GA탐색과정은 ArcInfo에 포함된 매크로 언어인 AML을 사용하였다. AML에서는 ArcINFO의 테이블들을 커서를 통해 접근할 수 있으므로, 프로그램의 구현시 별도의 변환작업이 필요하지 않고 수행과정을 같은 프로그램 내에서 시각적으로 표현할 수도 있어서 본 연구와 같은 실험적 구축 목적으로는 적당하다고 할 수 있다. 그러나 커서를 이용한 데이터 접근

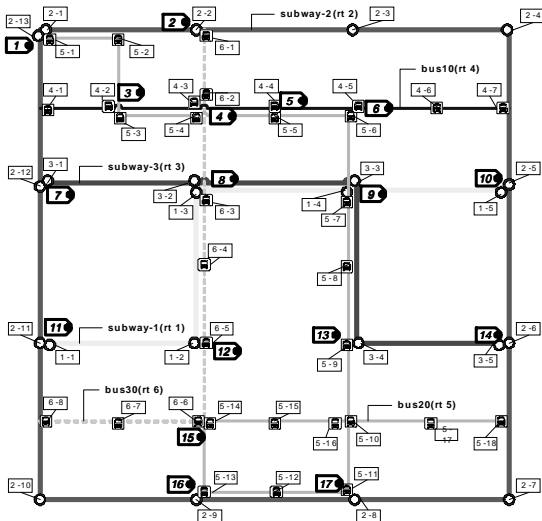


그림 8. 다수의 모드가 포함된 가상 네트워크의 구축

이 매우 느린데다가 AML 자체가 인터프리터 방식의 언어이므로 이를 현실적으로 그대로 이용하기는 어렵고, 실제 시스템의 구축 시에는 C++와 같은 컴파일 방식의 언어에서 데이터를 배열로 저장하여 처리함이 바람직 할 것이다.

그림 9는 염색체, 즉 후보해가 생성되어 화면에 출력되는 과정을 예시하고 있다. 사용자는 우선 시작위치와 목적지를 넣게 된다. 그 다음으로는 여행의 시작시간을 제공하도록 되어 있다. 이 예시는 좌측 상단 코너와 우측 하단 코너를 시작점과 목적지로 각각 입력했다고 가정하고, 출발시간은 12:30pm으로 입력했다고 했을 때 산출된 경로의 예를 보여준다. 그 밖에 GA과정에 필요한 모집단 크기, 반복의 수(목표 세대수), 기타 적용 확률들과 같은 파라미터들은 적절히 가정되어 프로그램 내에서 표현되었으며, 버스의 배차간격, 정류장간 거리, 서비스 시작 및 종료시간과 같은 시간제약 요소들도 역시 그림 7에서 예시된 바와 같이 적절히 가정하여 테이블 내에 저장되었다. 각 염색체들을 평가할 적응도 함수는 시작점에서 종점까지 이르는 소요시간을 평가하도록 구성하였다. 그림 9 하단에는 결과로 나타난 몇 가지 값들이 표시되어 있다.

Pentium PC의 800MHz CPU 및 512MB RAM을 장착한 IBM-PC에서 수행하였을 경우, 모집단을 초기에 10개를 생

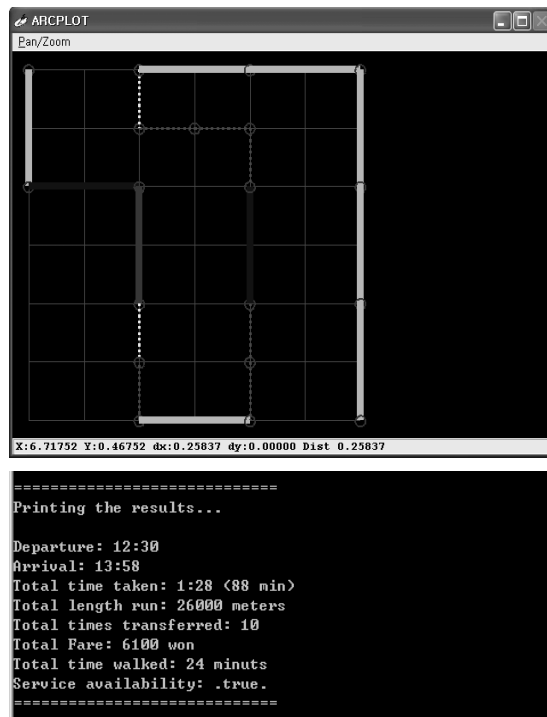


그림 9. 염색체(후보해) 생성과정의 예

성하는 데 걸리는 시간이 약 22초 가량 소요되어 하나의 경로를 평균적으로 2.2초만에 산출하는 것을 알 수 있었다. 10개 경로 중 10%, 즉, 상위 1개 정도만 보존하고 나머지 가운데 교배와 돌연변이를 각각 5%에서 시작하여 5% 단위로 50%까지 변화시키면서 실험한 결과, 20% 전후에서 가장 우수한 수행결과를 보여주었으며, 50%에서 적응도함수값이 급격히 증가하였다. 그래서 교배와 돌연변이 비율을 각각 20%씩 거치도록 하여 다음 세대를 생성하도록 하고, 이렇게 20세대를 생성하는 데 걸리는 시간은 대체로 5~8분 정도가 소요되었다. 이를 20여회 실시한 결과는 모두 오류 없이 최적해가 얻어져서 알고리즘 상의 문제는 없는 것으로 분석되었다. 그러나 앞서 언급된 바와 같이 이를 그대로 이용하기에는 AML 자체의 데이터 접근방법에 근본적인 문제가 있는 것으로 파악되었으며, 현 연구단계에서는 GIS구조를 이용한 GA 알고리즘 구현에 있어서 성공적인 것으로 분석되었다.

## 7. 결 론

본 연구에서는 다수의 수단간 환승이 발생하는 대중교통망에서 최적루트를 탐색하는 방안을 제시하였다. 현실적인 탐색과정이 되려면 환승지역에서의 몇 가지 시간제약요소가 반영되어야 한다. 이러한 점이 대중교통 네트워크에서의 최적 경로 탐색을 복잡하게 만드는 요인이 된다. 대중교통에서의 최적루트는 최단거리만으로 결정되는 것이 아니라 환승소요 시간을 고려한 최단소요시간 또는 환승의 최소횟수, 총 소요요금 등이 반영되어야 한다. 이용객이 선호하는 요소는 처음에 반영되도록 프로그램을 구성하는 것이 바람직하다.

이렇게 시작점에서 종점까지 한번에 최적해가 구해질 수 없는 상황에서 유전자 알고리즘(GA)이 효과적으로 사용될 수 있음을 알았다. 시-종점간에 여러 가지 해가 병렬적으로 존재할 때, 이들을 결합하고 변경하면서 우수한 형질을 가진 염색체들을 추출해 가게 된다.

이론을 정립하고 프로그램을 통한 초기 테스트를 목표로 한 이번 연구단계에서는 GA 탐색과정을 자동화하기 위해 ArcInfo에 임베드 되어있는 언어인 AML을 이용하였다. AML은 컴파일되는 언어가 아니라 인터프리트 방식의 언어인데다가 속성데이터를 읽는 과정에서 데이터 테이블을 커서를 사용해서 읽고, 배열을 사용할 수 없어서, 전반적으로 기대한 것보다 시간이 많이 소요되었다. 그러나 GA는 병렬처리(parallel processing)가 적용될 수 있는 좋은 예로 알려져 있다. 세대와 세대간은 순차적인 과정이지만 각 세대에서의 염색체들의 생성과정은 상호 의존관계가 적어 이

들 각각을 병렬처리를 함으로써 소요시간을 매우 향상시킬 수 있게 된다. 실제로 웹을 통해 서비스를 할 경우, 사용자에게 요청에 반응하는 시간이 10초 정도를 초과하게 되면 현실적이 못할 것으로 판단된다. 병렬처리시스템을 사용할 경우, 현재의 AML의 성능이 그대로 유지된다고 가정하더라도 10개의 CPU를 사용하게 되면 산출 속도를 30초 정도로 단축할 수 있다는 계산이 나온다. 즉, 컴파일 언어를 사용하고 배열처리 방법을 사용한다면 기대한 만큼의 성능을 얻을 수 있을 것으로 판단된다. 본 연구의 성과를 기반으로 실제 데이터를 이용하여 웹을 통한 경로안내시스템 구축에 중점을 두고 향후 연구를 발전시키는 것을 기대하고 있다.

## 감사의 글

본 연구는 2002년도 한국학술진흥재단 신진교수연구과제 지원(KRF-2002-003-B00362)에 의하여 수행된 연구로서 재단측에 감사 드립니다.

## 참고문헌

- Bellman, R. (1958), On a routing problem, *Quarterly Applied Mathematics*, Vol. 16, pp. 87-90.
- Choi, G-S., Park, J-J., and Seo, K-S. (1997), Round route search algorithm for commercial vehicle using evolution program, *Journal of Korean Society of Transportation*, Vol. 15, No. 4, pp. 125-141.
- Deo, N. and Pang, C. (1984), Shortest path algorithms: taxonomy and annotation, *Networks*, Vol. 14, pp. 275-323.
- Desrochers, M. and Soumis, F. (1988), A Reoptimization Algorithm for the Shortest Path Problem with Time Windows, *European Journal of Operational Research*, Vol. 35, pp. 242-254.
- Dijkstra, E. W. (1959), A note on two problems in connection with graphs, *Numerische Mathematik I*, pp. 269-271.
- Gallo, C. and Pallottino, S. (1984), Shortest path methods in transportation models, Florian, E. (eds.) *Transportation Planning Model*, Elsevier, pp. 227-256.
- Michalewicz, Z. (1994), *Genetic Algorithm + Data Structure = Evolution Programs*, Springer-Verlag.
- Moore, E. F. (1959), The shortest path through a maze, *In Proceedings of the International Symposium on the Theory of Switching, Part II*, Harvard University, Cambridge, MA, pp. 285-292.
- Spiess, H. and Florian, M. (1989), A new assignment model for transit networks, *Transportation Research*, Vol. 23B, pp. 83-102.