유전자알고리즘을 이용한 동적 통행배정모형에 관한 연구

A Development of Dynamic Traffic Assignment Model Using Genetic Algorithm

박 경 철

(서울대학교 교통연구실 석사 2년)

목 차

Ⅰ. 서론

최근 ITS의 한 분야인 ATIS(Advanced Traffic Information Systems)는 실시간 교통정보를 활용하여 운전자에게 다양한 서비스를 제공한다. 이에 따라 실시간 교통상황에 의해서 통행배정을 예측할 수 있는 동적 통행배정 모형의 개발이 요구되고 있다.

그러나, 동적 통행배정모형은 시공간적인 변수들의 복잡성으로 인해서 그 최적해를 찾는데 많은 수학적인 어려움이 존재한다. 따라서, 여러 가지 수학적인 기법들이 연구되어 왔지만, 기존의 방법들은 제약조건과 목적함수가 convex한 경우에 있어서만 풀이가 가능하다는 단점을 가지고 있다.

본 연구에서는 최근 최적화 분야에 활발히 적용되고 있는 유전자 알고리즘(Genetic Algorithm)을 동적 통행배정모형에 도입하였다. 연구에 사용된 동적 통행배정모형은 제약조건이 convex하지 않은 Merchant와 Nemhauser의 모형(1978)이다.

유전자 알고리즘을 이용하여 이에 대한 해법을 제시하였고, 그 결과를 검증하기 위해서 기존의 비선형해법 프로그램을 이용해서 구한 결과와 비교하였다.

Ⅱ. 이론적 고찰

1. Merchant-Nemhauser 모형

1) 모형설명

Merchant와 Nemhauser는 1987년 다기점(Mutiple origins)과 단일종점(Single destination)을 가진 네트워크에서의 동적 통행배정을 위한 이산시간(Discrete time) 모형을 제시하였다.

분석시간을 적당한 길이의 작은 시간간격 $\{t \mid t=0,1,\cdots,T\}$로 분해하면, 기본적으로 다음의 두 가지 방정식을 만족한다.

■ 상태 방정식(State Equation)

$$x_a^{t+1} = x_a^t - g_a(x_a^t) + d_a^t \ , t=0,1,\cdots,T-1 \tag{1}$$

■ 교통량보존 방정식(Traffic Conservation Equation)

$$\sum_{a\in A(n)} d_a^t = F^t(n) + \sum_{a\in B(n)} g_a(x_a^t) \tag{2}$$

$A(n)$: 노드 n에서 진출하는 링크의 집합
$B(n)$: 노드 n으로 진입하는 링크의 집합
$F^t(n)$: t시간대에 노드 n에 부과되는 외부교통량(OD)

이와 같은 기본적인 제약조건에 혼잡을 고려한

연속하고 감소하지 않는 convex한 비용함수 h_a를 가정하면, 비용을 최소화시키는 문제 P는 다음과 같다.

$$P = \min \sum_{t=1}^{t=T} \sum_{a=1}^{a=A} h_a(x_a^t) \tag{3}$$

$$\text{s.t. } x_a^{t+1} = x_a^t - g_a(x_a^t) + d_a^t,$$
$$t = 0, 1, \cdots, T-1, \ \forall a \in A$$
$$\sum_{a \in A(n)} d_a^t = F^t(n) + \sum_{a \in B(n)} g_a(x_a^t),$$
$$t = 0, 1, \cdots, T-1, \ \forall n \in N - \{n\}$$
$$x_a^0 = R_a \geq 0, \quad \forall a \in A \tag{4}$$
$$d_a^t \geq 0, \quad t = 0, 1, \cdots T-1, \ \forall a \in A \tag{5}$$
$$x_a^t \geq 0, \quad t = 0, 1, \cdots T, \ \forall a \in A \tag{6}$$

2) Nonconvexity 문제점

Merchant-Nemhauser 모형은 유출량 함수 g_a가 선형함수가 아닌 경우에 목적함수 P에 대한 가능해 영역이 convex하지 않다는 문제를 가지고 있다.
이런 이유로 비록 비용함수 h_a가 선형함수라고 해도 P는 다수의 국지해(Local Optima)를 갖게 되고, 일반적인 방법으로는 최적해(Global Optima)를 찾을 수가 없게 된다.
본 연구에서는 목적함수와 제약조건 형태 즉, convex하다는 조건에 상관없이 적용이 가능한 다음의 유전자 알고리즘을 적용하였다.

2. 유전자 알고리즘

1) 유전자 알고리즘의 특징

유전자 알고리즘은 확률적인 알고리즘의 부류에 속해 있지만 방향성 있는 탐색과 확률탐색의 요소를 결합하였으므로 다른 알고리즘과는 상당히 다르다.
반면에 등반방법(Hill Climb Method)과 같은 경우, 시작점에 의해서 최적값이 좌우되고 국소 최적값만을 제공하게 되는 단점이 있다.
이런 단점을 유전자 알고리즘은 해결할 수 있어서 현존하는 방향성 있는 탐색방법보다 더 효율적이며 목적함수나 제약조건의 형태에도 제약을 받지 않기 때문에 더욱 광범위한 적용이 가능하다.

2) 유전자 알고리즘의 수행절차

① 개체집단 초기화
제약조건에 맞는 범위 내에서 정해진 세대수(pop_size)에 맞는 bit string(2진수형태)형태의 1세대 염색체 개체집단을 만든다.

② 개체선택
염색체의 적합도[1]에 비례하는 룰렛 선택을 이용하여 우수한 개체를 선택한다.

③ 교배연산
룰렛선택을 통해 선택된 뛰어난 개체들 간에 서로의 특징을 교환하여 새로운 세대의 개체를 만드는 과정이다.

④ 돌연변이
돌연변이 연산자는 비트별로 수행된다. 모든 비트의 string은 0에서 1로 또는 그 반대로 바뀌는 변이과정을 거친다.

⑤ 새로운 세대로 진화
선택, 교배, 그리고 돌연변이를 한 후에 새로운 개체집단은 평가함수에 의해서 적합도 평가를 받는다. 이 평가는 다음 선택을 위한 확률분포를 만드는데 사용된다. 나머지 단계는 위의 단계들을 주기적으로 반복하면 되고, 결국 최상의 염색체가 최적해로 선택되어 진다.

3) 제약조건처리방법

제약조건의 처리문제는 유전자 알고리즘을 응용하고자 할 때, 부딪히는 가장 중요한 문제이다. 이를 처리하는 방법으로는 벌점(Penalty) 함수방법이나 해독기(Decoder) 또는 복구(Repair) 알고리즘을 이용하는 방법이 있다.
그러나 이와 같은 방법들은 문제에 따라 적용방법이 달라지는 단점을 갖거나 다양한 문제를 다루기에는 역부족인 경우가 많다. 이런 이유로 일반적인 제약조건을 처리하기 위해 개발된 방

1) 목적함수에 대입하여 계산된 값

법이 GENOCOP(Genetic Algorithm for Numerical Optimization for Constrained Problems) 시스템이다. 이 방법의 핵심은 다음과 같다.
① 등식 구속조건의 소거.
② 모든 염색체가 적합한 영역 내에 계속 머무는 것을 보장할 수 있는 유전 연산자들의 고안.

III. 모형 구축

1. 모형입력함수 도출

기본적인 모형은 Merchant와 Nemhauser(1978)의 모형을 활용하였고, 수정된 Greenshield 모형을 도입하였다. 모형에 사용된 변수들은 다음과 같다.

변수기호	변수 설명
n	노드집합
a	링크집합
t	분석시간
x_a^t	t시간대에 링크 a상의 차량수
d_a^t	t시간대에 링크 a로의 유입량
$g_a(x_a^t)$	t시간대에 링크 a의 유출함수
l_a	링크 a의 길이(km)
u_a^t	t시간대에 링크 a의 통행속도(km/시)
u_{max}	링크 a의 최대 통행속도(km/시)
u_{min}	링크 a의 최소 통행속도(km/시)
k_a^t	t시간대에 링크 a의 밀도(대/km)
k_j	jam상태의 밀도(대/시)
$T_a(x_a^t)$	t시간대의 링크 a의 통행시간

1) 링크 통행시간 함수

전통적인 BPR 함수는 교통류를 정적상태로 보는 가정을 내포하고 있으며, 이러한 가정은 시간에 따라 교통량이 변하는 동적인 교통 네트워크에서는 타당하지 않다.
본 연구에서는 새로운 링크 통행시간 함수를 만들기 위해서 다음의 수정된 Greenshield 모형을 이용하였다.

$$u_a^t = u_{min} + (u_{max} - u_{min})\left(1 - \frac{k_a^t}{k_j}\right) \tag{7}$$

따라서, 링크 a의 통행시간은 다음과 같이 쓸 수 있다.

$$T_a(x_a^t) = \frac{l_a^2 k_j}{u_{max}\, k_j\, l_a - (u_{max} - u_{min})\, x_a^t} \tag{8}$$

이 함수는 혼잡에 대한 비효용을 의미하며, 어떤 링크의 통행비용을 나타내는 단조 증가하는 연속함수이다.

2) 링크 유출량 함수

링크에서의 유출량은 그 링크의 물리적인 특징에 의해서 결정된다고 가정하면, 어떤 시간대의 유출량은 그 링크 상의 교통량과 같게 된다.
이미 알려진 교통량-밀도-속도 방정식은

$$Q = u \cdot k \tag{9}$$

이고, 따라서 유출량은 다음과 같다.

$$g(x_a^t) = u_{max}\frac{x_a^t}{l_a} - (u_{max} - u_{min})\,\frac{(x_a^t)^2}{l_a^2\, k_j} \tag{10}$$

위의 식은 어떤 링크의 유출량은 링크 상의 차량수에 의해서 결정된다는 것을 나타내고 있다.
최대 유출량은 $\frac{dg_a}{dx_a} = 0$ 이 되는 1차 미분 조건에서 다음과 같이 구해진다.

$$g_a(x_a^*) = \frac{u_{max}^2\, k_j}{4(u_{max} - u_{min})} \tag{11}$$

3) 동적 사용자최적과 동적 체계최적 목적함수

일반적인 정적 통행배정모형에서 목적함수는 사용자 평형(User Equilibrium)과 체계최적해(System Optimal)의 두 가지로 구분된다. 그러나 동적 통행배정모형에서는 사용자들은 순간순간 변하는 환경에 대응하여 경로를 선택하게 된다. 그러므로 동적 통행배정에서는 더 이상

사용자 평형이라는 말을 사용하지 않고 동적 사용자최적(Dynamic User Optimal)이라는 말을 사용한다.
이를 Wardrop의 사용자 평형원리를 이용하면 다음과 같이 정의할 수 있다.

「고려대상이 되는 매 시간마다 각각의 기종점 교통량에 대해, 모든 노드에서 종점까지 통행에 이용되는 모든 경로의 통행시간이 동일하고 그 시간이 최소통행시간인 상태를 나타낸다.

또한 동적 체계최적(Dynamic System Optimal)은 매 순간마다 시스템의 총 통행비용을 최소화시키는 상태를 나타낸다.

■ 동적사용자최적해(DUO)의 목적함수

$$\min \sum_{a=1}^{A} \sum_{t=1}^{T} \int_{x_a^t} T_a(x)\, dx \qquad (12)$$

■ 동적체계최적해(DSO)의 목적함수

$$\min \sum_{a=1}^{A} \sum_{t=1}^{T} x_a^t \cdot T_a(x_a^t) \qquad (13)$$

4) 최소시간간격 결정

동적 통행배정모형에서는 어떤 시간대에 링크에 진입한 교통량은 동일 시간간격 동안에 그 링크를 통과할 수 없다는 것을 가정하고 있다. 따라서, 분석 시간간격은 다음 수식을 만족한다.

$$T < \min \left\{ \frac{l_a}{\text{링크 } a\text{의 } u_{\max}}, \ \forall a \right\} \qquad (14)$$

2. 모형 정식화

1) 목적함수

동적 사용자최적	$\min \sum_{a=1}^{A} \sum_{t=1}^{T} \int_{x_a^t} T_a(x)\, dx$
동적 체계최적	$\min \sum_{a=1}^{A} \sum_{t=1}^{T} x_a^t \cdot T_a(x_a^t)$

2) 제약조건

① 상태방정식(State Equation)

$$x_a^{t+1} = x_a^t - g_a(x_a^t) + d_a^t, \quad t = 0, 1, \cdots, T-1$$

② 교통량보존 방정식(Traffic Conservation Equation)

$$\sum_{a \in A(n)} d_a^t = F^t(n) + \sum_{a \in B(n)} g_a(x_a^t)$$
$$t = 0, 1, \cdots, T-1, \ \forall n \in N - \{n\}$$

$A(n)$: 노드 n에서 진출하는 링크의 집합
$B(n)$: 노드 n으로 진입하는 링크의 집합
$F^t(n)$: t시간대에 노드 n에 부과되는 외부교통량(OD)

③ 링크 차량수 조건
링크에서의 최대 차량수는 그 링크의 임계밀도(k_j)상태에서의 교통량이 된다.

$$0 \le x_a \le k_j \times l_a \qquad (16)$$

분석시간 $t=0$에 대한 초기 조건은 모든 링크에 대해서 주어진다.

$$x_a^0 = R_a \ge 0, \quad \forall a \in A \qquad (17)$$

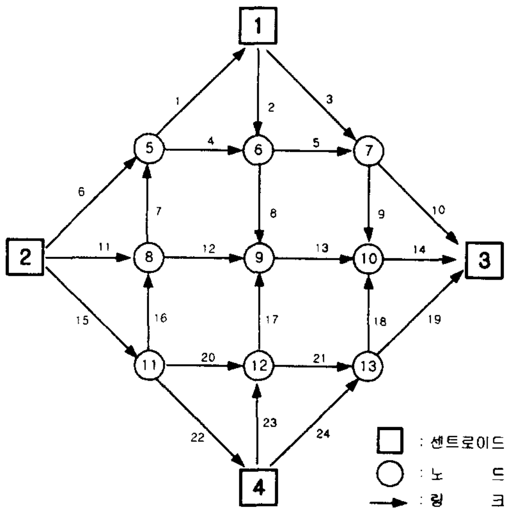
④ 유출량 제약조건
유출량은 앞에서 구한 최대 유출량 조건과 비음조건으로 이루어진다.

$$0 \le d_a \le \frac{u_{\max}^2 \ k_j}{4(u_{\max} - u_{\min})} \qquad (18)$$

Ⅳ. 모형적용 및 결과분석

1. 가상 네트워크 구성

본 연구에 쓰인 분석 대상 네트워크는 <그림 1>과 같다.
이 네트워크는 4개의 기·종점 노드와 9개의 일반 노드로 이루어졌다. 그리고 이를 24개의 링크로 연결하고 있다.
링크는 일방향이며, 고속도로급과 국도급 두 가지로 구분하였다.



<그림 1> 분석 대상 네트워크

각 링크에 대한 물리적인 특징은 다음 표와 같다. 최저속도와 임계밀도는 고속도로나 국도에 모두 같은 값을 가정하였다.

<표 1> 링크의 물리적인 특징

링크 번호	속 성	길이 (km)	차선수	최대 속도 (km/시)	최저 속도 (km/시)	임계 밀도 (대/km)
1	국 도	12	1	80	10	80
2	국 도	8	1	60	10	80
3	국 도	12	1	80	10	80
4	국 도	8	1	60	10	80
5	국 도	8	1	60	10	80
6	국 로	12	1	80	10	80
7	국 도	8	1	60	10	80
8	국 도	8	1	60	10	80
9	국 도	8	1	60	10	80
10	국 도	12	1	80	10	80
11	고속도로	15	2	100	10	80
12	고속도로	15	2	100	10	80
13	고속도로	15	3	100	10	80
14	고속도로	15	4	100	10	80
15	국 도	12	1	80	10	80
16	국 도	8	1	60	10	80
17	국 도	8	1	60	10	80
18	국 도	8	1	60	10	80
19	국 도	12	1	80	10	80
20	국 도	8	1	60	10	80
21	국 도	8	1	60	10	80
22	국 도	12	1	80	10	80
23	국 도	8	1	60	10	80
24	국 도	12	1	80	10	80

1) 분석시간간격

가상 네트워크에서 가장 짧은 링크의 길이는 8km이고, 이 링크의 최대 통행속도는 60km/대이다. 이 경우 링크통행시간은 다음과 같다.

$$T \ < \ \frac{8km}{60km/hr} = 0.13hr = 8\min$$

본 연구에서는 8분보다 작은 5분을 분석 시간간격으로 선택하고 전체 분석시간은 50분으로 정하였다.

2) 최대 유출량

링크의 차선당 최대 유출량은 식 (11)에 나타나 있다. 고속도로와 국도의 링크 속성을 이용해서 각 링크의 차선당 최대 유출량을 구해 보면 다음과 같다.

<표 2> 링크별 최대 유출량

구 분	최대속도	차선당 유출량
국 도	80km/시	152 대/5분
	60km/시	120 대/5분
고속도로	100km/시	185 대/5분

3) 외부입력 교통량

<표 3> 외부입력 교통량 (대/5분)

기점노드 / 시간간격	노드1	노드2	노드3
1	120	390	100
2	130	450	150
3	135	510	185
4	130	480	150
5	115	560	180
6	130	590	140
7	100	615	130
8	110	620	150
9	130	590	130
10	105	570	140

4) 초기교통량조건

분석시작 시점($t=0$)에 대한 교통량은 국도에

대해서는 각 링크 최대 차량수의 40%로 가정하였고, 고속도로에 대해서는 30%로 가정하였다.

2. 결과분석

구축된 모형에 대한 해는 유전자 알고리즘과 비선형방정식을 풀 수 있는 Excel 프로그램을 이용하여 각각 구하였다.
Excel Solver는 비선형문제에 대한 선형근사와 반복계산 횟수가 적은 '준 뉴튼법'을 사용한다. '준 뉴튼법'은 계산과정에 초기값의 영향을 많이 받는다. 1차 수행과정에서는 결정변수의 초기값을 0으로 가정하여 수행하였고, 2차 수행과정에서는 1차 수행결과에서 구한 초기값 인접치로 보정한 후 계산하였다.
계산에 사용된 컴퓨터는 Pentium Ⅱ Processor (333MHz)이며 동적 사용자최적과 동적 체계최적 목적함수에 대한 적용결과는 다음과 같다.

<표 4> 유전자 알고리즘을 이용한 수행결과

구 분	목적함수값 (시)	계산시간 (초)
사용자 최적	16,385.92	10
체계 최적	19,505.46	9

<표 5> '준 뉴튼법'를 이용한 수행결과

구 분		목적함수값 (시)	계산시간 (초)
1차 수행	사용자 최적	16,385.91	209
	체계 최적	19,506.41	198
2차 수행	사용자 최적	16,385.98	63
	체계 최적	19,506.00	58

<표 4>와 <표 5>를 비교해 보면 목적함수 측면에 있어서는 기존의 비선형해법 알고리즘과 유전자 알고리즘이 거의 차이가 없다는 것을 알 수 있다.
그러나 계산시간 면에 있어서는 유전자 알고리즘이 기존의 비선형해법 알고리즘보다 효율적이란 것을 알 수 있다.
더욱이, '준 뉴튼법'에서 초기값을 최적해로 보정한 경우에 있어서도 유전자 알고리즘이 보다 효율적인 수행결과를 보여주고 있다.

Ⅴ. 결론 및 향후연구과제

본 연구에서는 교통분야에서 최근 활발히 연구되고 있는 동적 통행배정모형에 대해서 유전자 알고리즘을 이용한 해결 방법을 제시하고 있다. 특히, 제약조건이 convex하지 않기 때문에 기존의 비선형해법으로는 정확한 해결이 불가능한 문제에 대해서 그 적용 가능성을 검증해 보았다.
가상 네트워크의 적용결과 유전자 알고리즘이 목적함수 면에서는 기존의 알고리즘과 같은 해를 제시하였고, 계산시간 측면에서는 매우 효율적인 결과를 보여주었다. 따라서, 제약조건이나 목적함수가 convex하지 동적 통행배정문제에 있어서 유전자 알고리즘이 충분히 적용 가능하다는 것을 판단할 수 있다.

본 연구 과정에 있어서 향후연구과제는 다음과 같다.

첫째, 본 연구에 있어서 가장 기본이 되는 것은 통행시간 함수이다. 따라서, 시간에 따라 변하는 교통상황을 묘사할 수 있는 정확한 링크 통행시간 함수에 대한 연구가 필요하다.

둘째, 본 연구에 사용된 동적 통행배정모형은 다기점·단일종점에 관한 것인데, 보다 현실을 반영할 수 있는 다기점·다종점 모형에 관한 연구가 필요하다.