

Continuous Network Design Problem Solver with EMME/2 Macro Implementation Equilibrium Decomposed Optimization Algorithm vs. Genetic Algorithm

Jaeyoung Kim*
Seoul National University

Abstract

The objective of this study is to develop efficient solution algorithm with EMME/2 macro language for continuous network design problem so as to help transport engineers and planners use it easily without external program. The network design problem considers both supply side and demand side. In supply side, the operator can charge tolls at specific sections, change ramp metering rates at freeway entrance or change signal timing at the junctions, which are all user-regulating methods. On the other hand, network users select their routes (or modes) to minimize (or maximize) their individual travel costs (or benefits). We know this problem may have many multiple local solutions due to its inherent characteristics – Nonlinear Objective function and Nonlinear, Nonconvex constraints. Hence, it is difficult to solve for a globally optimal solution.

So far, many solution algorithms are developed to find solutions to this problem. In this study, both EDO(Equilibrium Decomposed Optimization), which is most efficient algorithm in computation time, and GA(genetic algorithm), which is well-known global solution algorithm, are implemented with EMME/2 Macro.

In 16-link example, both algorithms have converged into stable point. As expected, EDO algorithm is efficient in computation time, but it failed to achieve global solution.

GA is much computationally demanding, but its solution is superior to EDO's Solution.

Although it is desirable to use GA for solving the network design problem, it's hard to implement in real due to its computational burden. The EDO appears to be promising as an efficient algorithm for finding approximate solution to the equilibrium network design problem. But, GA with EMME/2 Macro can be applied to any kind of transportation optimization problem, it is still noteworthy.

EMME/2 매크로를 이용한 연속형 변수의 가로망 설계 문제의 해석 - EDO 알고리즘 vs 유전자 알고리즘 -

Continuous Network Design Problem Solver with EMME/2 Macro Implementation - Equilibrium Decomposed Optimization Algorithm vs. Genetic Algorithm -

김재영(Kim, Jaeyoung)*

서울대학교 환경대학원 환경계획연구소(Seoul National University)

목 차

I. Introduction	
II. Continuous Equilibrium Network Design Model	IV. EMME/2 Implementation
III. Two Types of Solution Algorithms for NDP	1. EMME/2 Macro using EDO
1. Equilibrium Decomposed Optimization Algorithms	2. EMME/2 Macro using GA
2. Genetic Algorithms	V. Experimental Results
	VI. Conclusions
	References

Key Words : EMME/2 macro, Network Design Model, Equilibrium Decomposed Optimization, Genetic Algorithm,

요 약

이 연구의 목적은 교통계획분야에서 흔히 발생하는 가로망 설계 문제를 효율적으로 풀기 위한 알고리즘을 EMME/2 플랫폼에서 구동 가능하게끔 구현하여, 별도의 외부 프로그램을 이용하지 않고 실무현장에서 사용할 수 있도록 하는 것이다. 가로망 설계 문제는 교통체계 내에서 운영자가 수요와 공급 측면을 동시에 고려하는 문제로, 도심 진입 통행료의 결정, 교차로 신호시간의 결정, 가로의 용량 결정 등 광범위한 주제를 가지고 있다. 이 문제는 Stackelberg 게임으로도 알려져 있으며, 비선형 비분리 목적함수와 비선형제약 및 비볼록집합을 갖는 특성으로 인해 다수의 국지해가 존재할 가능성이 있다. 이러한 문제를 해결하기 위하여, 그간 여러 가지 해석 알고리즘이 개발되어 왔는데 본 연구에서는 실행시간이 가장 짧은 Equilibrium Decomposed Optimization 알고리즘과 확률적 최적화 방법의 하나인 유전자 알고리즘을 EMME/2 Macro로 구현하였다. 16개의 링크와 설계변수를 갖는 소규모 가로망에서 구현된 알고리즘을 실행한 결과 두 알고리즘 모두 안정적인 수렴성을 보여주었는데, EDO알고리즘이 계산시간측면에서 효율적이나 전역해에 수렴하지 못하였으며, GA는 계산시간이 오래 걸리긴 하였으나 EDO알고리즘보다 우수한 결과를 도출할 수 있었다. 이론적으로는 GA를 사용하는 것이 바람직하나 대규모 교통망에서 GA를 적용하는 것은 계산량에 대한 부담으로 현실적으로 어렵기 때문이다. EDO알고리즘이 유사해를 찾는데 효율적인 알고리즘인 것으로 판단된다. 그러나, 여기서 개발된 EMME/2기반의 GA는 가로망설계모형뿐 아니라 다른 곳에도 적용가능하므로, EMME/2 사용자들이 관심을 가질 필요가 있을 것으로 생각된다.

I. Introduction

가로망 설계 문제(Network Design Problem; NDP) 연구는 교통체계 내에서 운영자가 수요와 공급 측면을 동시에 고려하는 문제이며, 광범위한 주제를 다루고 있는 분야이다. 운영자가 교통망의 효율성을 높이기 위해 새로운 설계 변수를 체계 내에 도입하면 경로통행비용의 변화가 발생하고, 교통체계를 이용하는 사용자들의 노선선택번호도 바뀐다. 따라서 운영자는 운영자의 결정에 따라 이용자들이 체계에 적응하는 패턴을 예측해 교통체계가 최적의 상태로 유지될 수 있는 설계변수를 결정하여야 한다. 이 문제는 단기의 고정수요하에서는 주로 사회적 비용(일반적으로 총 교통비용)의 최소화, 장기의 탄력수요하에서는 사용자 편익의 최대화가 운영자의 목적이 되며, 공급측면에서 정책자의 비선형 목적함수와 비선형(혹은 선형) 제약식, 수요측면에서 비선형 목적함수와 비선형, 비볼록한 제약식을 갖는 Stackelberg 게임 혹은 Leader-Follower 게임 문제(Fisk, 1984)로 알려져 있으며 이러한 특성으로 인해 다수의 국지해가 존재할 가능성이 있다.

이 문제를 해석하기 위하여 그간 여러 가지 알고리즘이 개발되어왔으며, 각각의 해석 알고리즘은 장단점을 가지고 있다. 여기에선 NDP의 해석 알고리즘 중 실행시간면에서 가장 우수한 Equilibrium Decomposed Optimization 알고리즘과 전역해를 얻을 수 있는 유전자 알고리즘을 EMME/2에서 실행할 수 있는 매크로를 개발하였으며, 그 결과를 실행시간과 목적함수를 이용하여 비교하였다.

II. Continuous Equilibrium Network Design Model

f_a = 링크 a의 균형상태 교통량

$$\mathbf{f} = (\dots, f_a, \dots)$$

c_a = 링크 a의 통행비용

y_a = 링크 a의 추가 용량

$$\mathbf{y} = (\dots, y_a, \dots)$$

β_a = 링크 a의 용량추가에 드는 비용

θ = 사용자 정의의 weighting factor

일반적으로 연속형 설계변수의 가로망 설계 모형은 다음과 같은 bi-level형태의 수리계획모형으로 표현한다.

$$Z(\mathbf{y}) = \sum_a c_a(f_a(\mathbf{y}), y_a) f_a(\mathbf{y}) + \theta \beta_a(y_a) \quad (1)$$

subject to

$$y_a \geq 0 \quad \forall a \quad (a=1, 2, \dots, n)$$

$\mathbf{f}(\mathbf{y})$ = 균형상태의 교통량

균형상태의 교통량은 통행배분모형에 따라 다르나 가장 간단한 확정적 균형배분 모형의 경우 다음의 수리계획 문제를 풀면 해를 구할 수 있다.

$$\min P(\mathbf{f}) = \sum_a \int_0^{f_a} c_a(\omega, y_a) d\omega \quad (2)$$

현재까지 가로망 설계 문제(Network Design Problem; 이하 NDP)를 풀기위한 해석법은 반복

적이고 경험적인 알고리즘을 채택하고 있다. 이러한 알고리즘은 통행배분모형의 형태와 교통망에의 투자가 이산적인지 연속형인지, 투자에 대한 제약식이 목적함수에 포함되었는지 부가적인 제약식으로 고려되는지의 여부에 따라 여러 계층으로 구분할 수 있다.

대체적으로 대부분의 경험적인 알고리즘은 아래와 같은 방법론을 채택하고 있다.

【STEP 1】

\mathbf{y}^0 초기화(대개 'do-nothing' 경우임)

【STEP 2】

교통망에 OD를 배분하여 균형 교통량을 얻고, 총 교통비용(혹은 총 편익) Z^0 를 구한다.

【STEP 3】

알고리즘에 따라 정의된 계단값 δ 를 이용하여 \mathbf{y}^0 를 다음과 같이 갱신한다.

$$\mathbf{y}^1 = \mathbf{y}^0 + \delta$$

【STEP 4】

교통망에 OD를 재배분하여 새로운 균형 교통량을 얻고, 새로운 총 교통비용(혹은 총 편익) Z^1 을 얻도록 한다.

IF $Z^1 < Z^0$, set $Z^0 = Z^1$, $\mathbf{y}^0 = \mathbf{y}^1$ and goto STEP 3.

ELSE IF δ 가 정해진 수렴조건보다 크면, $\delta = \delta_{new}$ 로 갱신한 후 **【STEP 3】**으로 간다

ELSE STOP and keep \mathbf{y}^0

각기의 알고리즘은 δ 를 선택하는 방법과 한번의 반복에서 하나의 링크만을 업데이트할 것인지 모든 링크를 업데이트할 것인지를 선택하는 방법을 달리하고 있다. 많은 알고리즘이 급경사법을 이용하고 있으나, NDP는 non-convex 문제이므로 최적해에 수렴한다는 보장이 없으며, 최적해를 얻기 위해서는 상당히 많은 수의 프로그램 실행이 필요하다.

III. Two Types of Solution Algorithms for NDP

1. Equilibrium Decomposed Optimization Algorithms

Suwansirikul et al.(1987)에 의해 제안된 Equilibrium Decomposed Optimization 알고리즘은 가로망의 설계 과정을 링크 단위로 분해하여 각각의 sub-problem을 푸는 경험적인 알고리즘이다. 이들은 목적함수로 교통망상의 총 통행비용과 투자에 따른 투자비용을 합산한 비용 최소화를 선택하였으며, 식()의 상위 레벨 문제를 링크 단위로 분해한 sub-problem을 ()과 같이 사용하였다.

【STEP 0】 조절변수 u 의 상한벡터 \mathbf{U}^0 와 하한벡터 \mathbf{L}^0 를 읽어 들인 후, 각각에 대한 하위 레벨의 User Equilibrium상태의 링크 교통량 벡터 $\mathbf{f}(\mathbf{U}^0)$ 와 $\mathbf{f}(\mathbf{L}^0)$ 를 구한다. 또한, 링크 각각에 대해 $Z_a(\mathbf{U}^0)$ 와 $Z_a(\mathbf{L}^0)$ 를 구한다. $j=1$ 로 한 후 Step j 로 간다.

【STEP j 】 링크 각각에 대해, 일차원 탐색 알고리즘(golden section , Fibonacci, Bolzano search등)을 통해 새로운 점 I_a^j 를 구간 $[L_a^{j-1}, U_a^{j-1}]$ 사이에서 구한 후, 이 값에 상응하는 교통량 벡터 $\mathbf{f}(I^j)$ 를

구한다. 모든 링크에 대해 $Z_a(\vec{F})$ 를 계산한 후, 이 값에 근거해 새로운 구간 $[L_a^j, U_a^j]$ 를 찾는다. 만약 모든 링크에 대해서 $U_a^j - L_a^j < \epsilon$ 이면 $y_a^j = (U_a^j + L_a^j)/2$ 로 한 후 알고리즘을 끝낸다. 아니라면 $j=j+1$ 로 한 후 step j를 반복한다.

EDO 알고리즘은 상위문제와 하위문제를 한번씩 반복해서 풀어 해를 구하는 Iterative Optimization Assignment 알고리즘과 유사한 형태를 띠고 있지만, 링크 단위로 분해하여 각 링크에서의 최적 설계 변수를 찾는다는 점이 다르다. 또, IOA 알고리즘에서는 전 단계의 균형 교통량을 고정시킨 후 현 시점에서의 최적 설계 변수를 구하지만 EDO 알고리즘은 현 시점의 교통량을 가지고 설계 변수가 바뀌는데 따른 교통량의 변화를 재 평가하는 구조를 가지고 있기 때문에 근사적으로 Stackelberg 게임을 모사할 수 있다.

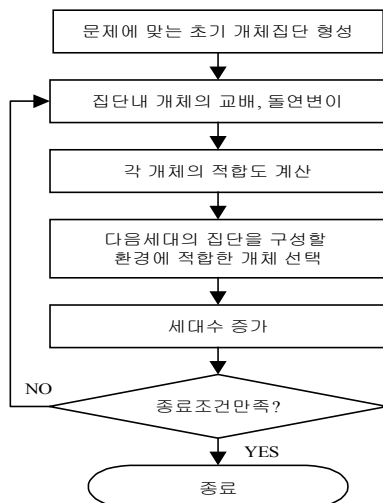
EDO 알고리즘은 Stackelberg 게임의 근사해로 수렴하는 것으로 알려져 있다.

2. Genetic Algorithms

유전자 알고리즘은 1970년대 초에 John Holland가 각각의 종(種)이 세대를 거듭하면서 변화하는 환경에 적응한다는 적자생존(survival of the fittest)과 자연 진화의 개념을 받아들여 개발한 알고리즘으로 다윈의 진화론과 멘델의 유전 이론에 근거하고 다중 점 탐색 기법을 사용하여 세대를 거치면서 목적함수를 만족시키는 최적해로 수렴해가는 알고리즘이다. 이 알고리즘은 해집단이 동시에 병렬적으로 주어진 환경에 따라 최적의 상태로 진화하고 진화시 터득한 정보는 다음 세대로 전달되며 이론적으로는 탐색공간에 제한이 없으며, 기존의 최적화 알고리즘들이 겪었던 함수의 연속성, 미분성, 선형성 등의 문제를 겪지 않아도 되는 안정적인 직접탐색 알고리즘이다.

유전자 알고리즘은 유전학의 용어를 사용하는데 여러 개의 연속적인 유전자(gene)로 이루어진 개체(스트링, 염색체, individual)들이 하나의 집단(population)을 형성하게 된다. 또한 집단내 하나의 개체가 주어진 환경에 부합하는 정도를 나타내는 척도가 적합도(fitness)이다.

유전자 알고리즘의 실행 순서를 다이어그램으로 나타내면 <그림 1>과 같다(Michalewicz, 1997)



<그림 1> 유전자 알고리즘의 실행 순서

김재영 등(2000)은 유전자 알고리즘을 이용하여 연속형 변수의 가로망 설계모형을 구축하였으며, 그 실

행 과정은 아래와 같다.

【STEP 0】 알고리즘의 제어 파라미터 결정

M, G, P_m, P_c 를 결정

세대수 $t \leftarrow 0$.

(여기서 M: 집단내 개체의 수, G: 최대세대수, P_m : 돌연변이확률, P_c :교배확률)

【STEP 1】 초기개체집단 P_0 생성(식(3) 참조)

$$P_0 = (\mathbf{y}_0^1, \dots, \mathbf{y}_0^k, \dots, \mathbf{y}_0^M), \quad \mathbf{y}_a^k = (\dots, y_{a,o}^k, \dots)$$

$$y_{a,o}^k = L_a + (U_a - L_a) \cdot r \quad \forall a, k \quad (3)$$

여기서, r 은 0~1의 범위를 갖는 임의의 실수 난수, L_a 는 링크 a의 설계변수 하한, U_a 는 링크 a의 설계변수 상한

$y_{a,o}^k$ 는 0세대, k 번째 개체의 a 번째 유전자 (링크 a의 설계변수)

【STEP 2】 집단 P_t 내 각 개체(설계변수) \mathbf{y}_i^k 에 대하여 \mathbf{y}_i^k 를 고정시키고, 변동 부등식 제약을 만족시키는 균형교통량 \mathbf{f}_i^k 를 구한후 총 교통비용 Z_i^k 를 산정

$$Z_i^k = \sum_a C_a(\mathbf{f}_i^k, \mathbf{y}_i^k) + \theta \sum_a \beta_a(\mathbf{y}_i^k)$$

$$s.t. \quad \sum_a c_a(\mathbf{f}_i^k, \mathbf{y}_i^k) \cdot (\hat{f}_a - f_a) \geq 0 \quad \forall \hat{\mathbf{f}} \in F$$

$$\mathbf{f}_i^k \in F, \mathbf{y}_i^k \in Y$$

【STEP 3】 다음 세대를 구성할 개체를 적합도와 정의된 선택법을 이용하여 구성

3-1. 엘리트 보존 전략을 사용하여 집단내 최우수 개체 \mathbf{y}_i^{best} 는 다음 세대에도 살아남으며, t세대의 최우수 개체가 t-1세대의 최우수 개체보다 적합도가 낮을 경우 t세대의 가장 열등한 개체를 t-1세대의 최우수 개체로 대체.

3-2. 토너먼트 크기가 2인 토너먼트 선택법을 사용하여 t세대의 개체로부터 t+1세대의 개체들을 구성

【STEP 4】 새로 구성된 집단 P_{t+1} 에 대하여 교배 실행

4-1. set $k \leftarrow 1$

4-2. 난수 r 생성

4-3. if $r < p_c$, k 번째 개체를 교배의 대상으로 삼는다.

4-4. $k \leftarrow k+1$, 2개의 개체가 선택될 때까지 4-2 ~ 4-4를 반복. 2개의 개체가 선택되면 4-5로 간다

4-5. 균일 교배를 실시하여 새로운 개체를 생성

4-6. if $k < M$, then 교배절차 종료 else $k \leftarrow k+1$ and go 4-2

【STEP 5】 교배가 끝난 집단에 대하여 돌연변이 실행

5-1. set $k \leftarrow 1$

5-2. 난수 r 생성

5-3. if $r < p_m$, k 번째 유전자를 돌연변이의 대상으로 삼아 동적 돌연변이 실행

5-4. if $k < M \times L$, then 돌연변이절차 종료 else $k \leftarrow k+1$ and go 5-2

【STEP 6】 종료조건 검사

if $t < G$, then $t \leftarrow t+1$, $T(t)$ 갱신 and go **【STEP 2】**, else 알고리즘 종료

IV. EMME/2 Implementation

위에서 제시된 NDP의 해석 알고리즘은 주로 소규모 가로망을 대상으로 연구자들이 작성한 프로그램을 통하여 구현되어있다. 이 알고리즘을 이용하여 실제 프로젝트를 수행하기 위해서는 몇 가지의 문제점이 있을 수 있다.

우선, 실제 교통수요분석이나 가로망 설계를 위해서는 자동차 통행배분 모형뿐 아니라 대중교통통행배분모형이 필요하며, 때로는 수단선택모형을 필요로 하기도 한다. 즉, Bi-level형태의 문제에서 lower-level 문제가 여러 가지 형태를 가질 수가 있으나, 이를 개인이 일일이 모형화하기 위해 필요한 시간과 노력이 상당하다.

둘째, 각기 모형을 Customizing하였어도 다른 프로젝트를 수행하기 위해서는 자료구조의 변화, 목적함수의 형태 변화에 따라 많은 부분에 수정이 필요할 수 있으며 다른 연구자가 모형을 이해하기 어려울 수 있으므로 수정이 용이치 않다.

이러한 단점을 해결하기 위해서 2가지 방법을 생각해 볼 수 있는데, 그 중 하나는 Upper Level의 문제는 연구자가 임의로 작성한 프로그램을 이용하며, Lower Level 문제(통행배분모형이나 수단선택모형이 포함된 균형상태의 통행량을 찾는 문제)는 상용화된 패키지를 이용하면서 그 사이의 인터페이스를 개발하는 것이다. 이는 실제로 가장 쉽게 단점을 커버할 수 있는 방법이다. 또 다른 방법은 상용화된 패키지가 제공하는 기능만을 이용하여 모형을 개발하는 것으로 가장 바람직한 방법이지만, 대개 이러한 상용패키지는 강력한 프로그래밍 툴을 제공하지 않기 때문에 연구자가 필요로 하는 모든 기능을 프로그램화할 수 없는 경우가 발생할 수 있다

여기서는 연속형 변수의 NDP 해석알고리즘 중 running time면에서 가장 효율적인 EDO 알고리즘과 해를 찾는 능력 면에서 우수한 것으로 알려진 유전자 알고리즘을 EMME/2의 Macro 기능을 이용하여 구현하였다. EMME/2를 이용할 경우 다차종통행배분모형과 가변수요통행배분모형, 확률적 통행배분모형 등 사용자가 원하는 모형을 용이하게 실행시킬 수 있으므로 보다 넓은 범위에 가로망 설계 모형이 사용될 수 있을 것으로 기대할 수 있다.

아래에서 개발된 매크로에 공통적으로 적용되는 사항은 다음과 같다.¹⁾

항목	수식	EMME/2표현
링크비용 $c_a(f_a, y_a)$	$A_a + B_a \cdot (f_a / (K_a + y_a))^4$	fd1=length+ul2*(volau/(lanes*10+ul3))^4
건설비용 $\beta_a(y_a)$	$d_a \cdot y_a$	ul1*ul3
총 교통비용 $Z(y)$	$\sum_a (c_a(f_a, y_a) \cdot f_a + \theta \cdot \beta_a(y_a))$, $\theta=1$	$\sum \text{timau} \cdot \text{volau} + \text{ul1} \cdot \text{ul3}$

1. EMME/2 Macro using EDO

1) 따라서, 사용자가 새롭게 정의하는 가로망 설계 문제는 새로운 형태의 목적함수와 비용함수를 사용하여야 하며, 매크로의 일부 수정이 필요하게 된다.

(1) 필요한 변수

EDO알고리즘은 6개의 extra network attributes와 2개의 스칼라매트릭스를 필요로 한다. 스칼라매트릭스는 큰 용량을 차지하지 않으므로 그 개수에 크게 구애받지 않으나, extra network attributes는 교통망의 크기에 따라 word의 크기가 기하급수적으로 커질 수 있으므로 이런 측면에서 EDO는 효율적이다.

EMME/2 변수	설명
@lb	사용자가 정의한 설계변수의 하한값 L^0
@ub	사용자가 정의한 설계변수의 상한값 U^0
@nlb	다음 반복에서의 새로운 설계변수 하한 L^j
@nub	다음 반복에서의 새로운 설계변수 상한 U^j
@zlow	설계변수의 하한에서의 링크의 비용 $Z_a(L)$
@zuppe	설계변수의 상한에서의 링크의 비용 $Z_a(U)$
ms91	@zlow의 합 $\sum_a Z_a(L)$
ms92	@zuppe의 합 $\sum_a Z_a(U)$
t1	t1 레지스터는 sub-macro와 결과파일이 저장되는 디렉토리를 지정하는 것으로, 예를 들어 c:\emme2\etc\edondp 에 submacro와 결과파일을 놓고 싶다면 t1=edondp\ 와 같이 선언한다.

(2) 매크로 사용법

```
~<edondp.mac <tolerance> <gpq> <iter> <rgap> <ngap>
여기서,
<tolerance> : converge tolerance of Objective Function
<gpq> : demand matrix of auto
<iter> : no. of iteration assignment stopping criterion
<rgap> : relative gap assignment stopping criterion
<ngap> : normalized gap assignment stopping criterion
```

(3) EDONDP 매크로 구성요소

	usage	function
calcvar.mac	~<calcvar.mac <extra attributes> <value>	<extra attributes>에는 저장될 링크데이터명을, <value>란에는 직접 값을 입력하거나, 계산이 필요할 경우 수식을 입력한다
ass.mac	~<ass.mac <gpq> <iter> <rgap> <ngap>	승용차 OD매트릭스는 <gpq>, 최대반복회수를 <iter>로 제한하고, 수렴조건인 relative gap은 <rgap>, normalized gap은 <ngap>으로하여 통행배분을 실시한다. 여기서는 가장 간단한 모형인 fixed demand single auto assignment를 수행하도록 하였다

다양한 통행배분모형 혹은 수단선택과 통행배분이 결합된 형태의 모형이 필요할 경우 ass.mac 를 그에 합당한 매크로로 변경하여야 한다.

2. EMME/2 Macro using GA

(1) 필요한 변수

GA는 경험적인 알고리즘이라는 특성상 다수의 변수를 가질 수 밖에 없는데, 그 중 extra network attributes의 사용량이 48개, 스칼라매트릭스 30개를 필요로 하였다. 따라서, 사용자가 이 매크로를 사용하기 전에 EMME/2뱅크에 충분한 여유용량을 제공하여야 한다.

EMME/2 변수	설명
@lb	사용자가 정의한 설계변수의 하한값 L^0
@ub	사용자가 정의한 설계변수의 상한값 U^0
@dv0~@dv19	t-1세대의 설계변수의 개체들 y_{t-1}^k
@tmp0~@tmp19	t세대의 설계변수의 개체들 y_t^k
@dv20	t-1세대에서 적합도가 가장 높은 개체의 설계변수
@tmpx, @tmpy	균일교배를 위한 임시변수
@tmpmk	균일교배를 위한 마스크값
@ztmp	임시변수
@tmpbn	1 또는 0값을 갖는 임시 변수
ms70~ms89	각 설계변수에 따른 목적함수값 Z_k
ms90	최우수개체의 목적함수값 Z_{best}
ms91	교배확률 P_c
ms92	돌연변이확률 P_m
ms93~ms99	임시 스칼라 매트릭스
t1	t1 레지스터는 sub-macro와 결과파일이 저장되는 디렉토리를 지정하는 것으로, 예를 들어 c:\emme2\etc\gandp 에 submacro와 결과파일을 놓고 싶다면 t1=gandp\ 와 같이 선언한다.

(2) 매크로 사용법

```

~<gandp.mac <0> <gpq> <maxgen> <pxover> <pmutat> <iter> <rgap> <ngap>
<t_beg> <t_end>
여기서,
<0> : dummy variable
<gpq> : demand matrix of auto
<maxgen> : maximum generation to evolve
<pxover> : crossover probability
<pmutat> : mutation probability
<iter> : no. of iteration assignment stopping criterion
<rgap> : relative gap assignment stopping criterion
<ngap> : normalized gap assignment stopping criterion
<t_beg>: Temperature At Start for Tournament Selection
<t_end>: Temperature At End for Tournament Selection
    
```

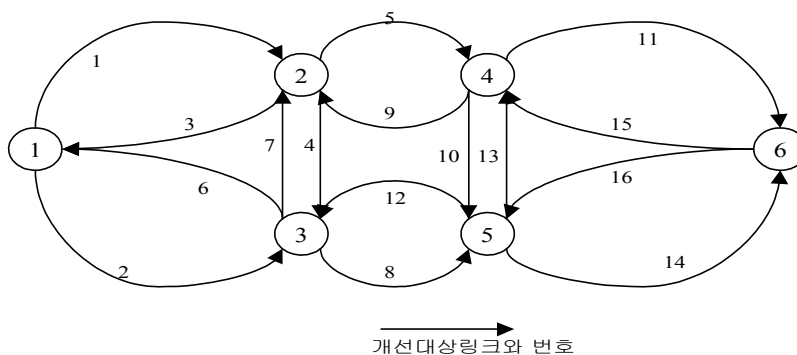
(3) 매크로 구성요소

gandp.mac는 다음과 같이 여러 가지의 sub-macro로 구성되어 있다.

sub-macro	usage	function
calcvar.mac	~<calcvar.mac <링크변수> <수식>	<링크변수>에 network calculation module을 이용하여 <수식>을 저장한다
mkext.mac	~<mkext.mac <var>	<var>개수만큼 extra attributes를 @dvxx형태로 만든다.
mkext2.mac	~<mkext2.mac <extra attribute name> <attribute description> <default value>	<attribute description>과 <default value>를 갖는 <extra attribute name>을 생성
mkext3.mac	~<mkext3.mac <var>	<var>개수만큼 extra attributes를 @tmpxx형태로 만든다.
rand.mac	~<rand.mac <var>	<var> 개수만큼(즉, @dv0 ~ @dv<var>)에 [0,1]사이의 난수를 발생시킨다
rand2.mac	~<rand2.mac	u13에 [0,1]사이의 난수를 발생시킨다
randms.mac	~<randms.mac	ms95에 [0,1]사이의 난수를 발생시킨다
intrand.mac	~<intrand.mac <var>	<var>에 0 또는 1의 난수를 발생시킨다
ass.mac	~<ass.mac <gpq> <iter> <rgap> <ngap>	승용차 OD는 <gpq>, 최대반복회수를 <iter>로 제한하고, 수렴조건인 relative gap은 <rgap>, normalized gap은 <ngap>으로하여 통행배분을 실시한다. 여기서는 가장 간단한 모형인 fixed demand single auto assignment를 수행하도록 하였다
tnselect.mac	~<tnselect.mac	Tournament Selection을 실행한다
elite.mac	~<elite.mac	Elite Selection을 실행한다
xover.mac	~<xover.mac <var>	교배를 실행한다 <var>이 1이면, whole arithmetical crossover를 <var>이 2이면 uniform crossover를 실행한다
mutate.mac	~<mutate.mac <var1> <var2> <var3>	돌연변이를 실행한다 <var1>이 1이면 단순돌연변이, <var1>이 2이면 동적돌연변이를 실행하는데, <var1>이 2일 경우 <var2>와 <var3>에 각각 현재 세대수와 최대 세대수를 대입해줘야 한다.

V. Experimental Results

두 개의 알고리즘을 비교하기 위하여 채택한 가상교통망은 6개의 노드와 16개의 링크로 이루어져 있으며, 기종점 수요는 1번 노드와 6번 노드사이의 왕복 통행만이 존재한다. 예제 가로망의 제원과 OD, 비용 함수는 <표 1>에 제시되어 있다.



<그림 2> 가상교통망

<표 1> 가상교통망의 제원

통행수요	$T_{1,6}$	$T_{6,1}$		
	10.0	20.0		
링크 a	자유류 통행시간 (A_a)	B_a	현재용량 (K_a)	단위건설비용 (d_a)
1	1.0	10.0	3.0	2.0
2	2.0	5.0	10.0	3.0
3	3.0	3.0	9.0	5.0
4	4.0	20.0	4.0	4.0
5	5.0	50.0	3.0	9.0
6	2.0	20.0	2.0	1.0
7	1.0	10.0	1.0	4.0
8	1.0	1.0	10.0	3.0
9	2.0	8.0	45.0	2.0
10	3.0	3.0	3.0	5.0
11	9.0	2.0	2.0	6.0
12	4.0	10.0	6.0	8.0
13	4.0	25.0	44.0	5.0
14	2.0	33.0	20.0	3.0
15	5.0	5.0	1.0	6.0
16	6.0	1.0	4.5	1.0

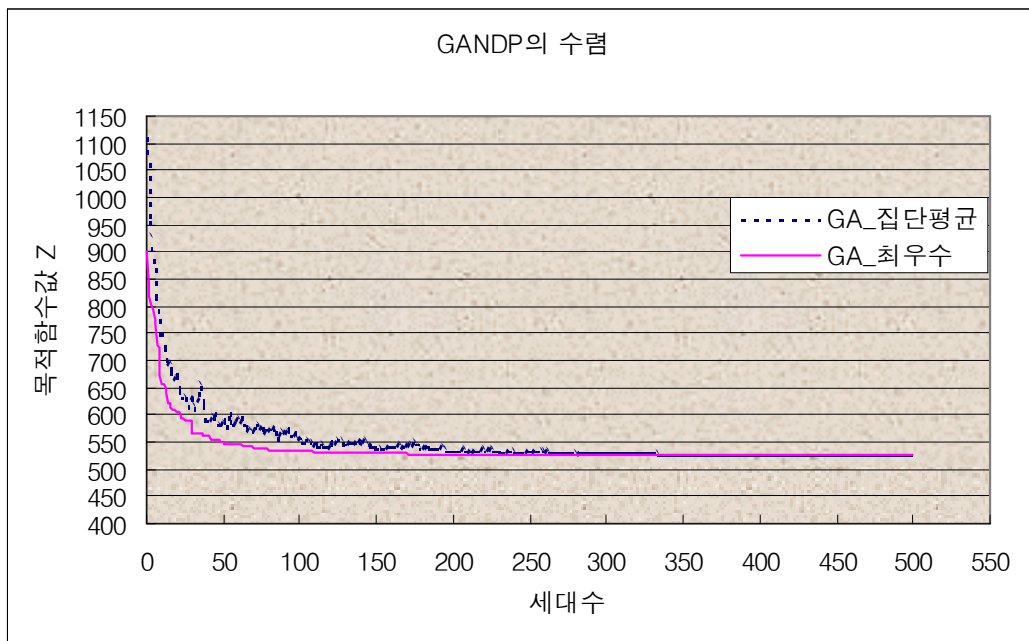
<표 2> GANDP에 사용된 모수

No. of Generation	500
개체집단의 크기	20
교배 확률	0.3
돌연변이 확률	0.1
Max. iterations	15
Max. rel. gap	0.01
Max. norm. gap	0.01
Temperature At Start	30
Temperature At End	10
교배 방법	균일교배
돌연변이 방법	동적돌연변이
선택법	토너먼트선택법 + 엘리트보존전략

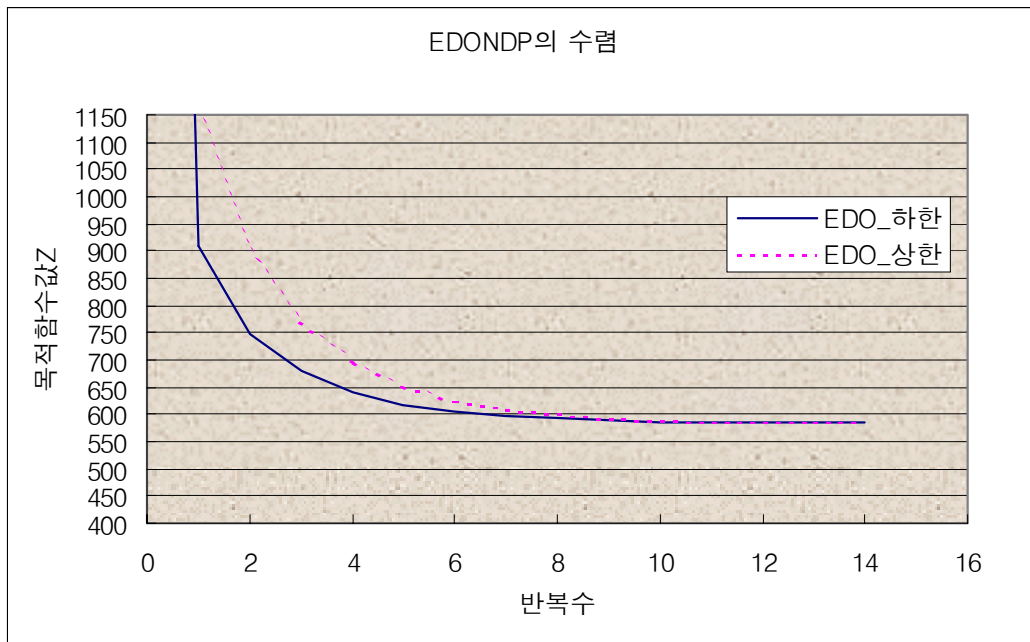
<표 3> EDONDP에 사용된 모수

수렴조건	0.5
Max. iterations	15
Max. rel. gap	0.01
Max. norm. gap	0.01

이때 최적의 설계변수 y 를 구하기 위하여, AMD 1GHz의 CPU가 장착된 컴퓨터 플랫폼에서 개발한 2개의 EMME/2매크로를 실행하였으며, 각 해석알고리즘에 사용된 모수는 <표 2>와 <표 3>에 제시되어 있다. 본래 유전자 알고리즘의 중요한 모수 중 하나인 한 세대 개체집단의 크기가 들어가나 개체집단의 크기를 늘리는 것은 메모리의 증가와 알고리즘 실행시간의 증가를 수반하므로, 여기서는 20개의 개체만으로 집단을 구성하였다.



<그림 3> GANDP의 알고리즘 수렴과정



<그림 4> EDONDP의 알고리즘 수렴과정

GA와 EDO모두 안정적인 수렴패턴을 보여주었으며, 그 과정은 <그림 3>과 <그림 4>에 도시되어 있다. GA의 경우 수렴조건은 최대세대수에 도달하는 것이므로 500세대가 지나서 종료되었으며, EDO의 수렴조건은 상한과 하한의 목적함수값의 절대적 차이가 0.5이하인 경우로 설정한 결과 15번 반복후 종료되었다. 알고리즘 종료 후 각각의 알고리즘이 찾아낸 설계변수와 목적함수값, 실행시간을 <표 4>에 나타내었다.

<표 4> 알고리즘 실행결과와 비교

알고리즘		EDO	GA
y의 상한		20.00	20.00
해	Y1	0.02	0.04
	Y2	4.62	4.64
	Y3	12.34	9.78
	Y4	0.02	0.06
	Y5	0.02	0.03
	Y6	7.66	7.62
	Y7	0.02	0.20
	Y8	0.58	0.76
	Y9	0.02	0.23
	Y10	0.02	0.10
	Y11	0.02	0.02
	Y12	0.02	0.21
	Y13	12.34	0.14
	Y14	1.32	1.76
	Y15	0.02	0.37
	Y16	19.98	18.66
총 교통비용(Z)		584.19	527.46
통행배분수행횟수 (No. of Assignment)		30	10,000
알고리즘실행시간(secs)		25	15,242

<표 4>의 결과에서 주목해야할 점은 총 교통비용과 알고리즘 실행시간이다. 기존의 연구결과와 유사하게 EDO알고리즘은 전역해 탐색에 실패하였으며, GA가 더 낮은 통행비용을 유발하는 설계변수를 찾아내었으며, 3번과 13번 링크에서 특히 차이를 알 수 있다.

알고리즘 실행시간 면에서도 기존연구결과와 유사하며, GA가 EDO의 약 600배에 달하였으며, 사용자균형배분문제를 수행한 횟수는 약 330배나 된다. 실제 대규모 가로망을 대상으로 이러한 알고리즘을 적용할 때는 실행시간이 가장 중요한 인자가 될 수 있다.

서울과 수도권을 대상으로 하는 교통망은 노드가 약 7,500개, 링크가 22,000개에 달하며, O/D 쌍도 1,000,000개가 넘는다. 이러한 교통망을 대상으로 한번의 EMME/2 실행에 필요한 시간을 약 10분으로 산정하면, EDO 알고리즘은 약 300분(5시간)이 GA는 약 100,000분(약 69.5일)이 소요되므로, 실제 규모의 가로망에서 확률적 최적화 방법을 적용하는 것은 어려운 것으로 판단된다. 이상에서 각 알고리즘의 장점과 단점을 다음 <표 5>에 비교하였다.

<표 5> EDO와 GA의 장점과 단점

구분 \ 해석알고리즘	Equilibrium Decomposed Optimization Algorithm	Genetic Algorithm
장 점	<ul style="list-style-type: none"> 실행시간이 짧다 알고리즘이 단순하며, 변수의 숫자가 적어 메모리 관리에 효과적이다 	<ul style="list-style-type: none"> 전역해에 도달할 확률이 높다 연속형 가로망 설계문제뿐만 아니라 이산형 가로망 설계문제(조합최적화문제)에도 쉽게 적용할 수 있다 수렴과정에서 개체집단의 크기만큼 후보해를 저장할 수 있으므로, 차선의 설계변수를 저장할 수 있다.
단 점	<ul style="list-style-type: none"> 지역해에 빠질 수 있는 가능성이 높다 연속형 가로망 설계문제에만 적용이 가능하다 제약식을 명시적으로 고려하기 어렵다 	<ul style="list-style-type: none"> 알고리즘 실행시간이 길다 사용자가 결정해야할 모수가 많다 상대적으로 개체의 수만큼 변수가 필요하므로 큰 메모리를 확보하여야한다 제약식을 명시적으로 고려하기 어렵다.

VI. Conclusions

연속형 변수의 가로망 설계 문제를 해석하기 위한 알고리즘 중 대표적인 2개의 해석 알고리즘을 EMME/2 매크로로 구현한 결과 유전자 알고리즘을 적용한 가로망 설계 모형은 EDO알고리즘에 비해 계산량에 대한 부담은 컸으나 더 좋은 해를 찾음을 알 수 있었다. 이 알고리즘은 초기 100세대 이내에 급격히 수렴하며 그 이후 거의 변화가 없는 패턴을 보여주는데, 알고리즘의 특성상 초기치가 임의로 생성되므로 안정적인 수렴 패턴을 가진다고 판단할 수 있다. EDO 알고리즘 역시 빠르면서도 안정적인 수렴 패턴을 보였으나, 전역해를 탐색하지 못하고 지역해에 도달하였음을 확인할 수 있었다. 이는 기존 연구에서도 알려졌다시피 가로망 설계 문제의 비선형, 비볼록의 특성으로 인해 다수의 국지해가 존재하고 전역해의 탐색을 어렵게 하며, 가로망의 형태와 목적함수에 따라 비볼록의 특성이 다양함을 의미한다. 그러나, EDO알고리즘은 알고리즘의 이해도 쉽고, 실행도 간단하며, 메모리도 적게 차지하므로 적용성이 좋다는 특성을 가졌다.

현실적으로 실무에서는 EDO 알고리즘이나 IOA 알고리즘 등을 사용하는 방법을 추천할 수 밖에 없을 것으로 판단되며, 가로망 설계 문제 중 특히 중요한 문제와 상대적으로 단순한 형태의 가로망에서는 GA를 사용하는 것이 바람직하다 할 수 있겠다.

여기서 개발한 모형은 현재 우리나라에서 가장 널리 쓰이는 교통패키지인 EMME/2의 기능

만을 이용하였으므로, 실무에서도 혼잡통행료나 교차로 신호시간, 램프미터링을 등 연속형 변수를 설계변수로 갖는 가로망 설계 모형에서 손쉽게 응용할 수 있을 것으로 판단되며, 추후 조금 더 일반화된 형태로 발전시킬 필요가 있다.

References

- 김재영, 임강원(2000), “유전자 알고리즘을 이용한 변동부등식제약하의 연속형 가로망 설계”, 대한교통학회지, 제18권 제1호, 61~73
- Fisk, C. S.(1984), “Game theory and transportation modeling”, Transportation Research, 18B, 301-313
- Michalewicz(1997), Genetic algorithm + data structures = evolution programs, Springer
- Suwansirikul, C., Friesz, T. L. and Tobin, R. L.(1987), “Equilibrium decomposed optimization: A heuristic for the continuous network design problem”, Transportation Science, 21, 254-264