

■ 論 文 ■

도로선형최적화를 위한 유전자 연산자의 적용

Incorporating Genetic Operators into Optimizing Highway Alignments

김 응 철

(인천대학교 토목환경시스템공학과 전임강사)

목 차

- I. 서론
 - II. 도로선형 최적화모형과 탐색 알고리즘
 - III. 도로선형 최적화를 위한 유전자 알고리즘의 적용
 - 1. 도로선형 최적화를 위한 매개변수 묘사법
 - 2. 도로선형최적화 모형의 결정변수(Decision Variable)와 목적함수 및 유전자 암호화(Genetic encoding)
 - 3. 유전자 알고리즘의 도로선형최적화 모형으로의 적용
 - IV. 도로선형 최적화를 위한 유전자 연산자
 - 1. 돌연변이 연산자
 - 2. 교차 연산자(Crossover Operator)
 - V. 유전자 연산자를 적용한 도로선형 최적화 사례
 - VI. 결론
- 참고문헌

Key Words : 유전자 연산자, 유전자 알고리즘, 도로선형최적화, 돌연변이 연산자, 교차 연산자

요 약

본 연구에서는 인공지능(Artificial Intelligence)방법 중의 하나인 유전자 알고리즘(Genetic Algorithm)을 도로선형최적화 모형개발의 탐색엔진으로 활용하기 위한 핵심도구인 유전자 연산자(Genetic Operator)의 개발과 적용과정을 통해 그 특징과 유용성을 제시하였다. 균일돌연변이 연산자, 직선돌연변이 연산자, 비균일 돌연변이 연산자, 전체 비균일 돌연변이 연산자 등 4개의 돌연변이 연산자가 탐색영역(Search space)의 가능한 모든 부분을 탐험(Exploration)하기 위해 적용되었으며, 단순교차 연산자, 두 개의 점을 이용한 교차 연산자, 산술 교차 연산자, 학습교차 연산자 등 4개의 교차 연산자가 노선대안의 우수한 유전형질을 다음세대에 효과적으로 전달(Exploitation)하기위해 시험되었다.

사례연구와 민감도 분석과정을 통해 유전자 알고리즘 및 개발·적용된 8개 유전자 연산자의 도로선형최적화 과정 도입이 우수한 노선대안을 빠르고 효과적으로 탐색함을 알 수 있었으며, 돌연변이 연산자와 교차 연산자의 효과적 조합이 상호보완기능을 통해 탐색능력의 향상에 큰 영향을 끼치는 것으로 파악되었다. 또한, 개발·적용된 연산자 이외에도 새로운 연산자의 개발 가능성이 무한하며, 이는 도로선형최적화에 유전자 알고리즘의 적용이 타당함을 반증함도 주목할 만하다.

I. 서론

도로선형최적화는 이론적으로 무한한 대안노선들 중 비용을 최소화(혹은 편익을 최대화)하는 동시에 다양한 설계제약요소들을 만족시키는 최적노선을 탐색하는 작업이다. 그러나 도로선형최적화는 많은 요소들을 고려해야 하기 때문에 쉽게 접근하기 어려운 문제이다.

고려해야 할 많은 요소들 중에는 최소곡선반경 등 기하구조적 요인들과 더불어 터널, 교량, 교차로, 인터체인지와 같은 구조물들, 지형, 사회경제적 영향, 생태, 지질, 토지이용, 환경 및 지역정서까지 정량적 및 정성적 요소들을 망라한다.

이러한 요소들은 노선선정과정의 각 단계에서 다양한 형태로 영향을 미치고 있으며 관련 부서, 도로설계 담당자, 계획가 및 주민간의 이해관계 상충 등으로 인해 상당한 시간과 노력을 요구하고 있다. 또한, 도로선형 최적화 과정은 다양한 비용항목, 계량화하기 어려운 여러 항목들의 존재, 초기계획 단계에서 인지되지 못하는 여러가지 돌발요소들 등으로 인하여 모형화하는데 어려움이 있었다.

그러나, 선형 최적화 과정을 수학적 모형의 개발과 컴퓨터를 이용하여 상당한 시간과 비용을 절감할 수 있음이 확인되고 있고(OECD, 1973; Shaw and Howard, 1982; Fwa, 1989), 최근 다양한 비용항목을 고려한 도로선형의 3차원 최적화 모형 등이 개발

되고 있다(Jong, 1998; Jha, 2000; Kim, 2001).

도로선형최적화를 위한 수학적 모형 구축과 자동화된 컴퓨터 모형의 개발이 풀고자 하는 문제, 즉, 목적함수가 미분 불가능(Non-differentiable)하고 국소 최적점이 많이 존재(Noisy)하며 함축적(Implicit)임에도 불구하고 최근 들어 가능해진 이유는 인공지능(AI:Artificial Intelligence) 방법 중의 하나인 유전자 알고리즘(GA:Genetic Algorithm)이 개발되고 그 응용이 속속 진행되고 있기 때문이다.

본 연구는 도로선형최적화를 위해 개발된 유전자 연산자를 소개하고 각각의 특징을 살펴보고 그 유용성을 검증하고자 한다. 본 연구의 대상도로는 양방향 2차로 도로로 정하였으나 다차로 도로에도 쉽게 확장 적용이 가능하다.

하나의 복잡한 가상 연구대상지를 대상으로 개발된 유전자 연산자의 유용성을 검증하고자 하며, 유전자 알고리즘의 도로선형최적화 탐색 알고리즘으로서의 장점과 적용타당성을 제시하고자 한다.

II. 도로선형 최적화모형과 탐색 알고리즘

<표 1>은 최근까지 개발된 도로선형최적화와 관련된 모형들 및 최적화 방법들을 소개하고 있다.

최적화 대상으로는 평면선형만을 대상으로 한 모형, 종단선형만을 대상으로 한 모형, 평면 및 종단선형을 3차원적으로 동시에 대상으로 한 모형 등 3가지 유형

<표 1> 도로선형최적화 모형들

최적화 대상	탐색 알고리즘	연구사례
평면선형	Calculus of variations	Wan(1995), Howard et al.(1968), Thomson and Sykes(1988), Shaw and Howard(1981 and 1982)
	Network optimization	OECD(1973), Turner and Miles(1971), Athsanassoulis and Calogero (1973), Parker(1977), Trietsch (1987)
	Dynamic programming	Hogan(1973) and Nicholson et al.(1976)
	Genetic algorithm	Jong(1998)
종단선형	Enumeration	Easa(1988)
	Dynamic programming	Puy Huarte(1973), Murchland(1973), Goh et al.(1988) and Fwa(1989)
	Linear programming	Reville et al.(1997) and Chapra and Canale(1988)
	Numerical search	Hayman(1970), Goh et al.(1988), Robinson(1973), Fwa(1989) and MINERVA (OECD, 1973)
	Genetic algorithm	Jong(1998)
평면 및 종단선형	Dynamic programming	Hogan(1973) and Nicholson et al.(1976)
	Numerical search	Chew at al.(1989)
	Two-stage optimization	Parker(1977) And Trietsch(1987)
	Genetic algorithm	Jong(1998), Jha(2000) and Kim(2001)

으로 분류할 수 있다. 모형개발 초기인 60~70년대에는 주로 2차원적인 평면선형이나 중단선형만을 대상으로, 하나 혹은 두 가지 비용항목(예로, 토공비용) 및 단순한 연구대상지를 대상으로 모형개발이 이루어졌으나, 최근 유전자 알고리즘을 이용한 3차원 최적화 모형의 개발이 이루어지고 있음을 알 수 있다.

한편, 모형에 사용된 탐색알고리즘은 Calculus of variations, Network optimization, Dynamic programming, Enumeration, Linear programming, Numerical search 및 유전자알고리즘(Genetic algorithm) 등 7가지 방법이 사용되었다.

도로선형최적화 모형내에서 빠르고 효과적인 탐색 알고리즘의 개발 및 적용은 필수적인데 이는, 평가해야 할 대안이 무수히 존재하고, 비용함수가 복잡한 양상을 나타내기 때문이다. 도로선형최적화 문제의 비용함수는 일차방정식이나, 이차방정식처럼 미분가능하지 않다(Non-differentiable). 즉, 문제를 단순화 하지 않는 한 전통적인 수학적 접근방법으로는 해를 도출하기가 매우 어렵다는 것이다. 더불어, 선형최적화 문제는 국소 최적점이 많이 존재한다(Noisy)는 것이다. 즉, 어느 특정 노선대안이 여타 노선대안에 비해 상대적으로 뛰어나다고 평가되었다 하더라도 더 좋은 노선대안이 어딘가에 존재할 가능성을 배제할 수 없다는 점이다.

또 다른 어려운 점은 노선대안의 비용함수가 명확(Explicit)하지 않고 함축적(Implicit)이라는 점이다. 예를 들어, 포장비나 사용자 통행시간비용 등은 노선대안의 가·중점이 최종적으로 연결되지 않는 한 계산해 낼 수 없다. 즉, 노선을 부분적으로 최적화 할 수 없다는 점이 문제해결을 더욱 어렵게 한다는 점이다.

최근에 많은 연구자들이 고민하고 있는 최적화 문제에는 분야에 관계없이 위에서 제기한 세가지 어려움이 존재하고 있다. 이러한 최근 경향은 많은 연구자들이 인공지능방법(AI: Artificial Intelligence)에 관심을 가지게 했고, 그 결과로 유전자 알고리즘(Genetic Algorithm), 퍼지이론(Fuzzy Theory), 인공신경망이론(Neural Network Theory) 및 다중대리탐색(Multi-Agent Search) 등의 방법이 개발 중에 있으며, 많은 부분에서 그 뛰어난(평가의 신속성 및 최적해의 높은 도출가능성)과 적용성(미분 불가능한 목적함수에의 적용가능성, 특정부분이 아닌 해가 존재할 수 있는 전체영역의 탐색) 등이 증명되고 있다.

물론 어떤 방법도 최적해(Global Optimum)를

도출한다는 보장을 하지는 못한다. 어떤 문제에도 적용할 수 있고 동시에 최적해를 도출해내는 알고리즘은 현재로서는 개발되지 않고 있다. 그러나 위에서 열거한 최신의 탐색알고리즘 중 유전자 알고리즘은 도로선형최적화 문제에 충분히 적용할 만한 가치가 있으며 최근의 적용사례가 존재한다.

Goldberg(1989)는 기존의 탐색방법들이 가지지 못한 유전자 알고리즘의 특징을 다음의 4가지로 요약한다:

- (1) 유전자 알고리즘은 파라미터가 아닌, 그들의 조합을 이용한다(GAs work with a coding of the parameter set, not the parameters themselves).
- (2) 유전자 알고리즘은 특정 해가 아닌 해들의 집합으로부터 탐색한다(GAs search from a population rather than a single point).
- (3) 유전자 알고리즘은 목적함수 자체를 이용하며, 미분치나 다른 정보를 필요하지 않는다(GAs use objective function information, not derivatives or other auxiliary knowledge).
- (4) 유전자 알고리즘은 확정적이 아닌 확률적 탐색을 사용한다(GAs use probabilistic transition rules, not deterministic rules).

〈표 2〉 기존 도로선형최적화 모형에 사용된 방법들의 문제점

방법	문제점
Calculus of variations	-미분 가능한 목적함수 요구 -도출된 해(도로선형)가 국소최적점일 우려가 높음
Network optimization	-도출된 해(도로선형)가 부드러운 곡선이 아님(not smooth) -탐색영역(search space)이 연속적(continuous)인 경우 적용이 불가함
Dynamic programming	-도출된 해(도로선형)가 부드러운 곡선이 아님(not smooth) -탐색영역(search space)이 연속적(continuous)인 경우 적용이 불가함 -함축적(implicit)인 목적함수에 적용이 불가함 -부문제(subproblems)간 독립성을 가정함
Enumeration	-탐색영역(search space)이 연속적(continuous)인 경우 적용이 불가함 -비효율적임(inefficient)
Linear programming	-비선형(non-linear) 목적함수에 적용이 어려움 -기울기(gradient)와 곡률(curvature) 제약 요소의 일부 제한된 점들만을 평가함
Numerical search	-도출된 해(도로선형)가 국소최적점일 우려가 높음 -모형이 복잡함 -불연속 비용항목들을 모형화하기 어려움

〈표 2〉는 도로선형 최적화에 사용된 기존 방법들 중 유전자 알고리즘을 제외한 여타 방법론들에 대한 단점을 분석한 결과이다.

III. 도로선형 최적화를 위한 유전자 알고리즘의 적용

1. 도로선형 최적화를 위한 매개변수 묘사법

최근 도로선형의 도로중심선 좌표와 지리정보시스템의 자료들을 수학적 모형 및 컴퓨터를 이용하여 위에서 제시한 건설비 즉, 토공비, 지가보상비 뿐만 아니라 포장비를 포함하고 사용자 비용(통행시간비용, 연료소모비 및 사고비용)까지 함께 평가하며 최적화 할 수 있는 모형이 개발되고 있다(Jong, 1998; Jha, 2000).

이러한 모형들은 기본적으로, 수백 혹은 수천개의 노선대안을 무작위로 발생시켜 각각의 노선대안을 개발된 목적함수에 근거하여 평가한 후 최적의 노선을 찾아가는 과정을 거친다.

한편, 개발된 모형들은 위치벡터(Position Vector)를 이용하여 선형의 중심선을 나타내는 3차원 좌표를 획득한 후 노선의 평가에 필요한 각종 비용을 GIS 등의 데이터베이스(DB)를 이용하여 도출해 낸다. 즉, 도로선형 최적화 모형을 통한 도로선형 대안을 획득함은 도로 중앙선을 나타내는 점들의 3차원 좌표를 구함을 의미한다.

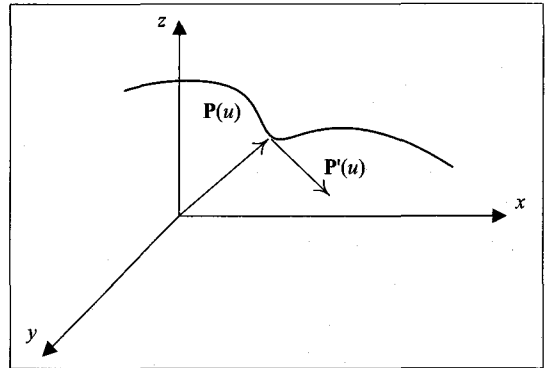
도로선형을 수학적으로 묘사하기 위해서는 매개변수 묘사법(Parametric Representation)이 유용하며(Lovell, 1999; Mortenson, 1997) 〈그림 1〉은 공간상에서의 도로선형을 묘사하는 예를 보여준다(진한 대문자는 공간에서의 벡터를 표현한다).

정의(Definition): $\mathbf{P}(u)$ 를 도로선형, L , 상의 위치벡터로 가정하자. 여기서, $u = \frac{\int_0^u \|\mathbf{P}'(t)\| dt}{\int_0^1 \|\mathbf{P}'(t)\| dt}$ 그리

고 $\|\mathbf{P}'(t)\| = \sqrt{(x'(u))^2 + (y'(u))^2 + (z'(u))^2}$.

기본적으로 \mathbf{P} 는 u 에 의해 매개변수화되며 u 는 \mathbf{P} 의 위치까지의 곡률경로를 나타내는 비율로 표시된다. 만약, 선형 L 이 도로 출발점($\mathbf{S} = [x_s, y_s, z_s]^T$)과 종점($\mathbf{E} = [x_E, y_E, z_E]^T$)를 연결한다면, 위치벡터

$\mathbf{P}(u)$ 는 $\mathbf{P}(0) = \mathbf{S}$ 와 $\mathbf{P}(1) = \mathbf{E}$ 를 만족해야 한다. 또한 $\mathbf{P}(u)$ 는 간격 0과 1사이에서 ($u \in [0, 1]$) 연속하여야 하는 동시에 연속적으로 미분 가능하여야 한다.



〈그림 1〉 공간상 곡선의 매개변수적 묘사

즉, 위치벡터를 이용하여, 모형에서는 선형의 어떤 점도 그 3차원 좌표를 획득할 수 있을 뿐만 아니라, 노선의 전체 길이, 직선부 및 곡선부 등의 파악이 가능하게 됨으로써 관련된 비용들을 평가할 수 있는 것이다.

더불어 도로상의 구조물들 즉, 교차로, 교량, 터널 및 인터체인지들도 함께 고려한 모델링이 노선선정 단계 즉, 타당성조사나 기본계획 단계에서 가능한 것도 사실이다(Kim, 2001). 이러한 수학적 모형의 개발은 기본적으로 전산화를 통한 평가 및 최적화의 필요조건인 것이다.

2. 도로선형최적화 모형의 결정변수(Decision Variable)와 목적함수 및 유전자 암호화(Genetic encoding)

도로선형최적화 모형의 결정변수는 3.1에서 언급하였듯이 교차점(points of intersections, P_i)이며 이들의 한 조합(a set of P_i)이 노선대안이 된다.

각각의 교차점(P_i)은 3차원 좌표, 즉, x, y, z 좌표로 표현되며, n 개의 교차점(P_i)으로 이루어진 노선대안(alignment alternative)은 $3n$ 개의 유전원질(gene)로 이루어진 염색체(chromosome)로 나타낼 수 있다. 즉, 하나의 특정 노선대안(alignment alternative 혹은 염색체(chromosome))은 식(1)과 같이 표현된다.

$$\begin{aligned} A &= [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}] \\ &= [x_{P_1}, y_{P_1}, z_{P_1}, \dots, x_{P_n}, y_{P_n}, z_{P_n}] \end{aligned} \quad (1)$$

여기서,

A : 염색체(chromosome 혹은 특정 노선대안)

$\lambda_i = i^{th}$ 유전원질 (gene),
 for all $i = 1, \dots, 3n$
 $(x_{P_i}, y_{P_i}, z_{P_i}) = i^{th}$ 교차점 (point of intersection)
 의 3차원 좌표, for all $i = 1, \dots, n$

따라서 도로선형최적화모형은 식(2)와 같이 제시될 수 있다(Kim, 2001: Jha, 2000: Jong, 1998).

$$\begin{aligned} & \text{Minimize} && C_T \\ & x_{P_1}, y_{P_1}, z_{P_1}, \dots, x_{P_n}, y_{P_n}, z_{P_n} \\ & = C_U + C_R + C_P + C_E + C_S \end{aligned} \quad (2)$$

단, 다음조건에 부합할 경우

$$\begin{aligned} x_0 \leq x_{P_i} \leq x_{\max}, \quad \forall i = 1, \dots, n \\ y_0 \leq y_{P_i} \leq y_{\max}, \quad \forall i = 1, \dots, n \end{aligned}$$

여기서,

x_0, y_0 : 연구대상지역의 왼쪽 하단 코너의 x, y 좌표
 x_{P_i}, y_{P_i} : 교차점 P_i 의 x, y 좌표

x_{\max}, y_{\max} : 연구대상지역의 오른쪽 상단코너의 x, y 좌표
 C_T : 노선대안의 총비용
 C_U : 노선대안의 사용자비용(주행시간비용, 차량 운행비용, 사고비용의 합)
 C_R : 노선대안의 보상비용
 C_P : 노선대안의 포장비용
 C_E : 노선대안의 토공비용
 C_S : 노선대안의 구조물 비용(터널 및 교량)

3. 유전자 알고리즘의 도로선형최적화 모형으로의 적용

〈표 3〉은 유전자 알고리즘의 용어를 설명하고 있으며 도로선형최적화 모형으로 접목 예를 설명해준다. 이러한 접목은 도로선형최적화에 유전자 알고리즘을 적용 가능함을 알려준다.

한편, 본 연구에서는 암호화된 해(염색체, 노선대안)의 GA코딩은 binary(2진수)가 아닌 실제 3차원 좌표값(즉, real value)을 사용하였다.

한편, 〈표 4〉는 자연진화 과정의 유전자 알고리즘

〈표 3〉 유전자 알고리즘 관련 용어의 정립

유전자 알고리즘	해설	도로선형 최적화
Chromosome (genotype, individual, string)	암호화된 해(염색체)	하나의 특정노선대안(교차점(P_i))의 조합
Phenotype	해독된 해	노선대안의 3차원 좌표
Gene	염색체의 일부(유전원질)	특정 교차점(P_i)
Locus	유전원질의 염색체내에서의 위치	특정교차점(P_i)의 위치
Alleles	유전원질의 가치(Value)	특정교차점(P_i)의 3차원 좌표, 즉 x,y,z의 값
Gene Pool	가능한 유전원질의 조합	가능한 x,y,z 값들의 조합
Population	해들의 집합	노선대안들의 집합

〈표 4〉 자연진화 과정의 유전자 알고리즘으로의 사상(Mapping) 및 도로선형최적화와의 관계

자연	유전자 알고리즘	도로선형 최적화
환경	문제	3차원 도로 선형 최적화
자연환경에서의 얼룩말 집단	문제의 해들의 집합	노선대안들의 집합
뚝뚝하고 경계심이 많은 얼룩말이 오래 생존하는 경향	적절한 해가 문제내에서 계속 남아있는 경향	저비용 노선대안이 계속 남아있는 경향
느리고 경계심이 적은 얼룩말이 사냥 당할 가능성이 높은 경향	부적절한 해가 문제내에서 빨리 사라지는 경향	고비용 노선대안이 계속 사라지는 경향
교배를 통한 2세의 배출	유전자연산자를 이용한 새로운 해의 도출	유전자연산자를 통해 새로운 노선대안의 도출
여러세대를 거치면서 얼룩말 집단의 환경 적응력 향상	여러세대를 거치면서 해가 최적해 또는 국소최적해에 도달하는 경향	여러세대를 거치면서 최적(최저비용)노선대안을 찾아가는 경향

으로의 사상(Mapping)과정 및 도로선형최적화 모형과의 관계를 나타낸다. <표 4>는 <표 3>에서 제시된 용어의 접목 예와 더불어 왜 유전자 알고리즘이 도로선형최적화모형의 구축에 적합한 탐색 알고리즘인지 다시 한번 확인해 준다.

또한 <그림 2>는 유전자 알고리즘의 기본적인 흐름틀을 제시해 주고 있으며 그 기본적인 과정이 도로선형최적화 모형의 구축과 어떤 관계가 있는지를 보여 준다. 전통적인 유전자 알고리즘의 틀은 매우 적절히 도로선형최적화 모형의 구축에 도입될 수 있을 뿐 아니라 그 구조적 틀을 자연스럽게 받아들일 수 있다.

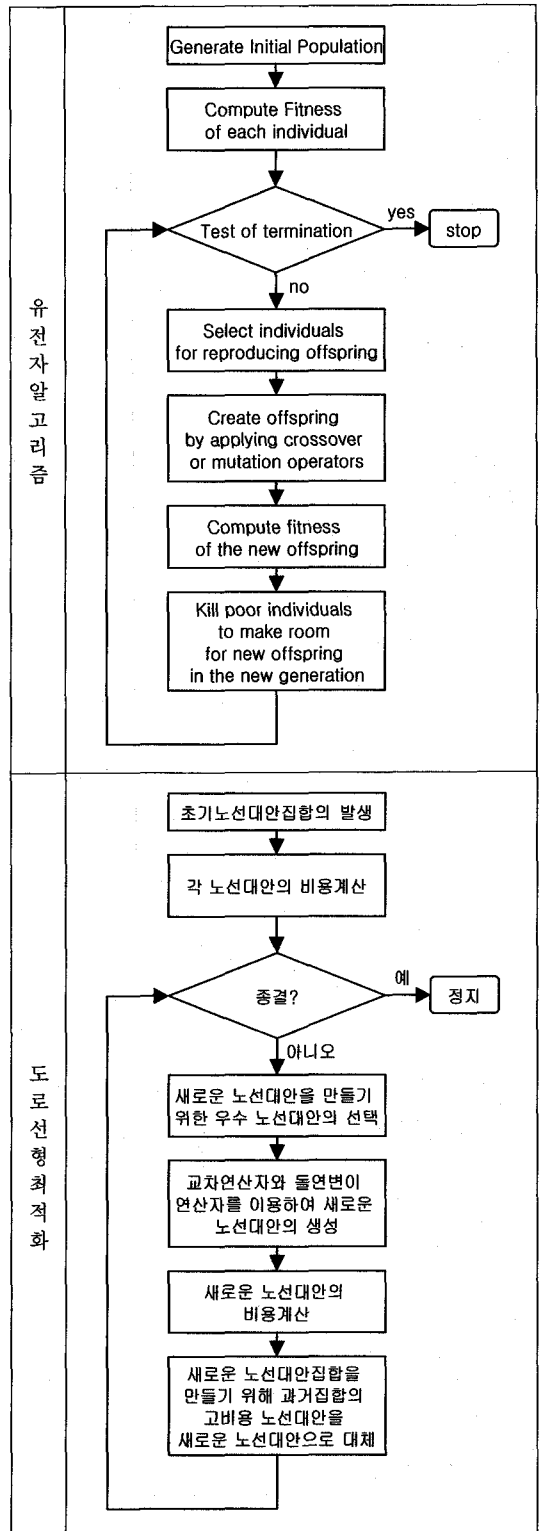
<그림 2>의 과정 중에 제시된 선택 메카니즘을 위해 본 연구에서는 엘리트보존방법(Elitism)과 랭킹방법(Ranking method)을 결합하여 사용하였다. 이는 목적함수 값이 가장 우수한 개체(노선대안)와 우수한 상위개체들이 다음세대에 선택될 확률을 높이는 것이 최적화를 하는데 효율적이기 때문이다.

IV. 도로선형 최적화를 위한 유전자 연산자

유전자 알고리즘에서 제시된 연산자는 크게 교차 연산자(crossover operator)와 돌연변이 연산자(mutation operator)로 대별된다. 교차 연산자(crossover operator)는 좋은 해(저비용 노선대안)의 특성을 반영하여 새로운 해(새로운 노선대안)를 만들어 나가기 위한, 즉, 좋은 노선대안의 유전적 특징을 담은 해를 만들기 위한 연산자로서 탐색과정의 신속성을 보장해 주는 반면, 돌연변이 연산자(mutation operator)는 탐색과정이 국소 최적점(local optimum)에 교착되지 않도록 하는, 즉, 최적해(global optimum)가 전혀 다른 곳에 있을 가능성을 고려하고 전체 탐색영역을 고루 확인하기 위한 연산자라고 할 수 있다.

즉, 교차연산자(crossover operator)는 최적해(최적노선대안) 탐색과정의 촉진(exploitation)을 보장하며, 돌연변이 연산자(mutation operator)는 전체 탐색영역(search space)의 고루 탐사(exploration)를 보장한다. 이 두 특징은 유전자 알고리즘의 우수성을 설명해 준다.

교차 연산자(crossover operator)와 돌연변이 연산자(mutation operator)의 기본적 개념은 위에서 설명한 바와 같으나 특정 문제에 적용하기 위해서는 연구자가 위의 기본 개념을 이해한 후 문제의 특징에



<그림 2> 유전자 알고리즘의 기본적 흐름 틀과 도로선형 최적화에의 적용

부합토록 연산자를 개발하여야 한다.

본 연구에서는 네 개의 교차 연산자(crossover operator)와 네 개의 돌연변이 연산자(mutation operator)를 소개하고자 한다(Jong, 1998)

1. 돌연변이 연산자

1) 균일돌연변이 연산자(Uniform Mutation Operator)

$\Lambda = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}]$ 를 돌연변이 시킬 염색체(노선대안)라고 가정할 때 이 연산자는 노선대안 중의 특정 교차점, P_i (k^{th} point of intersections)를 이산난수(discrete random number, $k = r_d[1, n]$)를 사용하여 돌연변이 시킬 교차점으로 선택한 후 관련된 x, y 좌표, 즉, λ_{3k-2} 와 λ_{3k-1} 을 연속난수(continuous random number)를 발생시켜 변환시키는 연산자이다.

$$\lambda'_{3k-2} = r_c[x_0, x_{max}] \quad (3a)$$

$$\lambda'_{3k-1} = r_c[y_0, y_{max}] \quad (3b)$$

2) 직선돌연변이 연산자(Straight Mutation Operator)

$\Lambda = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}]$ 를 돌연변이 시킬 염색체(노선대안)라고 가정할 때 이 연산자는 두 개의 독립적인 이산난수 i 와 j ($i = r_d[0, n+1]$, $j = r_d[0, n+1]$, $i \neq j$, 그리고 $i < j$)를 발생시킨 후 그 사이에 있는, 즉, $(3i)$ 번째와 $(3j-2)$ 번째 사이의 유전원질(x, y, z 좌표)을 직선으로 연결하는 연산자이다.

$$\lambda'_{3l-2} = \lambda_{3i-1} + (l-i) \left[\frac{\lambda_{3j-2} - \lambda_{3i-2}}{j-i} \right],$$

for all $l = i+1, \dots, j-1$ (4a)

$$\lambda'_{3l-1} = \lambda_{3i-1} + (l-i) \left[\frac{\lambda_{3j-1} - \lambda_{3i-1}}{j-i} \right],$$

for all $l = i+1, \dots, j-1$ (4b)

$$\lambda'_{3l} = \lambda_{3i} + (l-i) \left[\frac{\lambda_{3j} - \lambda_{3i}}{j-i} \right],$$

for all $l = i+1, \dots, j-1$ (4c)

3) 비균일 돌연변이 연산자(Non-Uniform Mutation Operator)

$\Lambda = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{3n-2}, \lambda_{3n-1}, \lambda_{3n}]$ 를 k 번째 ($k = r_d[1, n]$) 교차점에서 돌연변이 시킬 염색체(노

선대안)라고 가정할 때 이 연산자는 0 혹은 1의 난수 ($r_d[0,1]$)를 두 번 순차적으로 발생시킨 후, λ'_{3k-2} 과 λ'_{3k-1} 을 아래의 규칙에 의해 생성한 후 새로운 최종 염색체(새로운 노선대안)을 만든다.

CASE (1)

만약 첫 번째 난수($r_d[0, 1]=0$)가 0이면

$$\lambda'_{3k-2} = \lambda_{3k-2} - f(t, \lambda_{3k-2} - x_0) \quad (5a)$$

만약 첫 번째 난수($r_d[0, 1]=1$)가 1이면,

$$\lambda'_{3k-2} = \lambda_{3k-2} + f(t, x_{max} - \lambda_{3k-2}) \quad (5b)$$

여기서,

$$f = \lambda'_l = \lambda_i + (l-i) \left[\frac{\lambda_j - \lambda_i}{j-i} \right]$$

t : 현재 유전자 알고리즘의 세대번호(current generation number)

CASE (2)

만약 두 번째 난수($r_d[0, 1]=0$)가 0이면,

$$\lambda'_{3k-1} = \lambda_{3k-1} - f(t, \lambda_{3k-1} - y_0) \quad (5c)$$

만약 두 번째 난수($r_d[0, 1]=1$)가 1이면,

$$\lambda'_{3k-1} = \lambda_{3k-1} + f(t, y_{max} - \lambda_{3k-1}) \quad (5d)$$

한편, 이 연산자는 현재의 세대번호(t)와 돌연변이 폭(y)을 독립변수로 하는 식(6)과 같은 확률밀도 함수에 의해 작동한다.(Michalewicz, 1996).

$$f(t, y) = y \times r_c \times \left(1 - \frac{t}{n_T} \right) \quad (6)$$

여기서,

n_T : 최대 세대수

ξ : 비균일도를 결정하는 매개변수

$r_c[0, 1]$: 균일연속분포에서 발생하는 난수

식(6)에서, 사용자정의 매개변수, ξ 의 경우 작은 값이 지정된 경우는 결정변수의 변화폭을 크게 하는 기능을 가지며 큰 값이 적용된 경우는 그와 반대로 결정변수의 변화폭을 작게 한다. 본 연구는 면밀한 검토결과 "6"을 적용하였다.

4) 전체 비균일 돌연변이 연산자(Whole Non-Uniform Mutation Operator)

이 연산자는 비균일 돌연변이 연산자(Non-uniform Mutation Operator)를 모든 교차점에 적용하는 연산자로 노선대안의 전체 선형을 바꾸는 연산자이다.

$$A'_i = [\lambda_{i1}, \lambda_{i2}, \lambda_{i3}, \dots, \lambda_{i(3k)}, \lambda_{i(3k+1)}, \dots, \lambda_{i(3l)}, \lambda_{i(3l+1)}, \dots, \lambda_{i(3n)}]$$

$$A'_j = [\lambda_{j1}, \lambda_{j2}, \lambda_{j3}, \dots, \lambda_{j(3k)}, \lambda_{j(3k+1)}, \dots, \lambda_{j(3l)}, \lambda_{j(3l+1)}, \dots, \lambda_{j(3n)}]$$

2. 교차 연산자(Crossover Operator)

1) 단순교차 연산자(Simple Crossover Operator)

$A_i = [\lambda_{i1}, \lambda_{i2}, \lambda_{i3}, \dots, \lambda_{i(3n-2)}, \lambda_{i(3n-1)}, \lambda_{i(3n)}]$ 와 $A_j = [\lambda_{j1}, \lambda_{j2}, \lambda_{j3}, \dots, \lambda_{j(3n-2)}, \lambda_{j(3n-1)}, \lambda_{j(3n)}]$ 를 교차시킬 두 개의 염색체(노선대안)라고 할 때 우선 이산 난수 $k = r_d[1, n]$ 을 발생시킨 후 교차할 포인트 $3k$ 를 선정한다. 이후 두 개의 새로운 노선대안(offspring)은 아래와 같이 생성된다.

$$A'_i = [\lambda_{i1}, \lambda_{i2}, \lambda_{i3}, \dots, \lambda_{i(3k)}, \lambda_{j(3k+1)}, \dots, \lambda_{j(3n-2)}, \lambda_{j(3n-1)}, \lambda_{j(3n)}]$$

$$A'_j = [\lambda_{j1}, \lambda_{j2}, \lambda_{j3}, \dots, \lambda_{j(3k)}, \lambda_{i(3k+1)}, \dots, \lambda_{i(3n-2)}, \lambda_{i(3n-1)}, \lambda_{i(3n)}]$$

2) 두 개의 점을 이용한 교차연산자(Two-point Crossover Operator)

$A_i = [\lambda_{i1}, \lambda_{i2}, \lambda_{i3}, \dots, \lambda_{i(3n-2)}, \lambda_{i(3n-1)}, \lambda_{i(3n)}]$ 와 $A_j = [\lambda_{j1}, \lambda_{j2}, \lambda_{j3}, \dots, \lambda_{j(3n-2)}, \lambda_{j(3n-1)}, \lambda_{j(3n)}]$ 가 두 개의 점에서 교차될 염색체(노선대안)라고 할 때, 두 개의 점, $3k$ 와 $3l$ 이 난수 ($k = r_d[1, n], l = r_d[1, n], k \neq l, k < l$)를 이용하여 결정된다. 이후 두 개의 새로운 노선대안(offspring)은 아래와 같이 결정된다.

3) 산술교차 연산자(Arithmetic Crossover Operator)

$A_i = [\lambda_{i1}, \lambda_{i2}, \lambda_{i3}, \dots, \lambda_{i(3n-2)}, \lambda_{i(3n-1)}, \lambda_{i(3n)}]$ 와 $A_j = [\lambda_{j1}, \lambda_{j2}, \lambda_{j3}, \dots, \lambda_{j(3n-2)}, \lambda_{j(3n-1)}, \lambda_{j(3n)}]$ 를 산술교차 연산자를 이용하여 교차시킬 염색체(노선대안)라고 할 때 새로운 노선대안은(offspring)은 식(7)과 같이 생성된다.

$$A'_i = \omega A_i + (1 - \omega)A_j \tag{7a}$$

$$A'_j = \omega A_j + (1 - \omega)A_i \tag{7b}$$

여기서, $\omega = r_c[0, 1]$

4) 학습교차 연산자(Heuristic Crossover Operator)

$A_i = [\lambda_{i1}, \lambda_{i2}, \lambda_{i3}, \dots, \lambda_{i(3n-2)}, \lambda_{i(3n-1)}, \lambda_{i(3n)}]$ 와 $A_j = [\lambda_{j1}, \lambda_{j2}, \lambda_{j3}, \dots, \lambda_{j(3n-2)}, \lambda_{j(3n-1)}, \lambda_{j(3n)}]$ 를 학습 교차연산자를 통해 교차시킬 염색체(노선대안)라고 할 때 하나의 새로운 노선대안(offspring)은 식(8)과 생성된다. 단, 이 연산자의 경우 A_i 가 A_j 보다 우수한 노선대안임을 가정한다. 즉, $C_T(A_i) \leq C_T(A_j)$

$$A'_i = \omega(A_i - A_j) + A_i \tag{8}$$

여기서, $\omega = r_c[0, 1]$

제시된 8개의 연산자에 대한 특징을 요약하면 <표 5>와 같다.

<표 5> 8개 연산자의 특징요약

연산자 종류		특징	
돌연변이연산자	균일돌연변이	하나의 특정교차점(P_i)의 x, y 좌표를 무작위로 변화	- 탐색영역(search space)의 전체 공간 탐색(exploration)을 가능하게 함 - 각각의 연산자는 하나의 노선대안(offspring)을 생성
	직선돌연변이	두 개의 특정교차점(P_i, P_j)간을 직선으로 연결	
	비균일 돌연변이	하나의 특정교차점(P_i)의 x, y 좌표를 초기에는 무작위에 가깝게, 그리고 점차 변화폭을 줄여 변화	
	전체 비균일 돌연변이	모든 교차점의 x, y 좌표를 초기에는 무작위에 가깝게, 그리고 점차 변화폭을 줄여 변화	
교차연산자	단순교차	하나의 교차점을 선정하여 두 개의 노선대안을 교체	- 뛰어난 노선대안의 형질을 다음 노선대안(offspring)의 생성시 전이 - 각각의 연산자는 두 개의 노선대안(offspring)을 생성. 단, 학습교차 연산자는 하나의 노선대안(offspring)만 생성
	두 개의 점을 이용	두 개의 교차점을 선정하여 두 개의 노선대안을 교체	
	산술교차	두 개의 노선대안의 중간에 두 개의 새로운 노선대안(offspring)을 생성	
	학습교차	두 개의 노선대안 중 우수한 노선대안의 형질에 가깝게 새로운 노선대안(Offspring)을 생성	

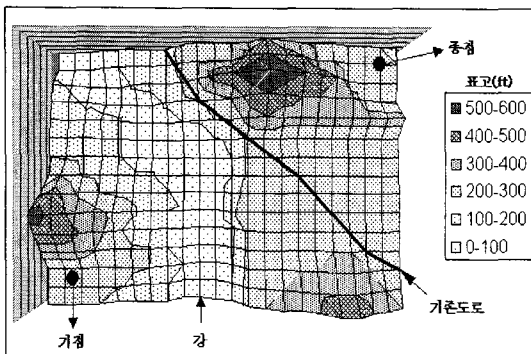
V. 유전자 연산자를 적용한 도로선형 최적화 사례

본 장에서는 하나의 인공연구 대상지를 작성하여 개발된 유전자 연산자를 적용한 도로선형최적화 사례를 제시하고자 한다. 사용된 프로그래머는 Visual C++이다.

〈그림 3〉은 상당히 복잡한 지형, 기존도로 및 강등을 포함하는 가상의 사례지역을 보여준다. 대상지역내의 기존도로는 북쪽에서 남서쪽으로 걸쳐 있으며, 3개의 산악 봉우리가 있고, 하나의 강이 북동쪽 모서리에서 남쪽으로 흐른다. 음영이 짙은 셀일수록 높은 표고를 의미하며 각각의 셀은 서로 다른 지가보상단가를 가지고 있다. 모형의 목표는 그림에 표시된 기·종점을 연결하는 새로운 도로의 최적선형을 찾는 것이다. 새로운 도로가 기존의 도로와 교차할 시 모형은 자동적으로 가능한 교차형태 즉, 교차로, 단순한 분리 및 인터체인지 중에서 가장 효과적인 구조물을 선정하게 구축되었으나, 본 글에서는 교차형태를 사용자가 교차로로 기 지정한 것으로 가정하였다.

한편, 비적합한(infeasible) 노선대안(예:최소곡선반경을 위반한 노선)이 생성되었을 경우는 벌점함수(penalty function)를 사용하여 탐색이 진행됨에 따라 탈락하도록 모형이 설계되었다.

〈그림 3〉에서 제시된 기·종점을 잇는 최적노선을 탐색하기 위해 본 연구에서는 〈표 6〉과 같이 각 2개의 돌연변이 연산자와 각 4개의 교차연산자를 사용하였다. 교차점(P_i)의 개수는 15개를 사용하였으며 150개의 population size(한세대내에서의 노선대안수),



〈그림 3〉 가상의 사례지역

〈표 6〉 선형최적화를 위한 시나리오

연산자와 기타 매개변수		적용
돌연변이 연산자	균일돌연변이 연산자	2
	직선돌연변이 연산자	2
	비균일 돌연변이 연산자	2
	전체 비균일 돌연변이 연산자	2
교차 연산자	단순교차 연산자	4
	두 개의 점을 이용한 교차 연산자	4
	산술교차 연산자	4
	학습교차 연산자	4
교차점(P_i)의 갯수		15
Population size		150
No. of generation		2000

〈표 7〉 최적노선 도출결과

Seed	노선총비용 (\$ million)	계산소요시간 (시간/분/초)	컴퓨터사양
20035	17.29	2/27/38	1GHz CPU Speed 261MB RAM

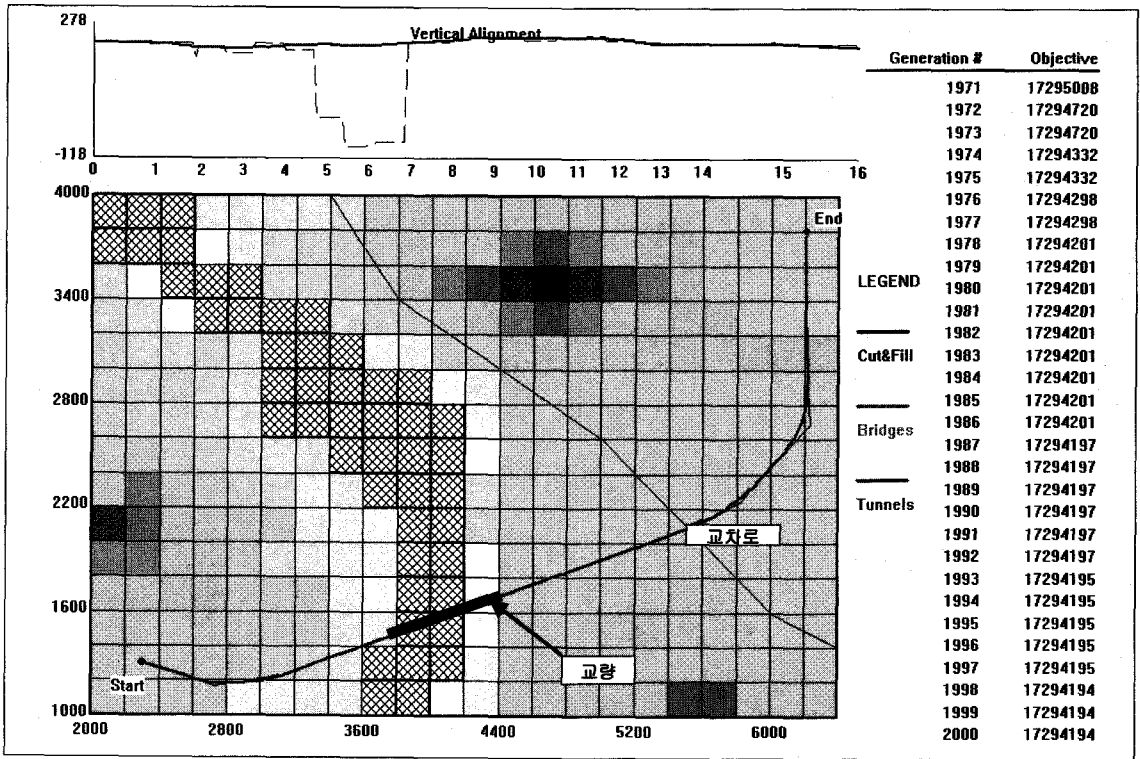
2000번의 세대수를 적용하였다. 유전자알고리즘의 확률특성을 반영하여 세 번의 최적화 탐색과정을 시도하였다. 세 번의 탐색시도는 서로 다른 Seed를 사용하였다(첫 번째 시도 : 21088, 두 번째 시도 : 20035, 세 번째 시도 : 21276).

〈표 7〉과 〈그림 4〉는 세 번의 시도에 의한 결과와 최적해를 보여준다. 〈그림 4〉는 최적해의 평면선형, 종단선형, 세대번호 및 목적함수 즉, 현 대안의 비용을 보여준다. 최적해는 두 번째 시도에서 도출되었으며 최적화된 노선은 하나의 교량과 기존도로를 교차하는 교차로를 가지고 있다.

유전자 연산자의 적용결과, 매우 복잡한 사례연구 대상지, 어려운 목적함수(비용최소화 함수), 다양한 설계제약요소, 교량·터널·교차로 등 구조물들의 고려 등에도 불구하고 도출된 노선은 매우 타당하며 탐색속도도 매우 빠름을 알 수 있다. 사실 탐색속도를 기존 모형과 비교하는 것은 기존 모형들이 탐색속도를 제시하지 않아 불가능하였다.

〈표 8〉은 개발된 연산자와 교차점(P_i)의 개수를 각각 다르게 적용한 민감도 분석 시나리오를 보여준다.

〈표 8〉에 제시된 시나리오들은 각각의 연산자와 교차점(P_i)의 개수를 다르게 적용하였을 시 어떤 결



〈그림 4〉 최적노선 도출결과

〈표 8〉 유전자 연산자 및 교차점 (P_i)을 다르게 적용한 시나리오의 작성

시나리오	I	II	III	IV	
돌연변이 연산자	(1)	4	4	2	2
	(2)	4	4	2	2
	(3)	4	4	2	2
	(4)	4	4	2	2
교차 연산자	(5)	2	2	4	4
	(6)	2	2	4	4
	(7)	2	2	4	4
	(8)	2	2	4	4
교차점 (P_i)의 갯수	15	10	15	10	
No. of generation	2000				

주) (1)균일돌연변이 연산자 (2)작선훈돌연변이 연산자 (3)비균일 돌연변이 연산자 (4)전체 비균일 돌연변이 연산자 (5)단순교차 연산자 (6)두 개의 점을 이용한 교차 연산자 (7)산술교차 연산자 (8)확습 교차 연산자

과가 도출되는지를 파악하기 위함이며, 기본적으로는 돌연변이 연산자와 교차연산자의 기능과 역할을 비교해보기 위함이다.

〈표 9〉 네 개의 시나리오별 최적해 도출결과

시나리오	노선총비용 (\$ million)	계산소요시간(초)	최적해가 찾아진 세대번호
I	23.795	6,473	1998
II	22.522	3,856	1990
III	21.057	1,846	2000
IV	20.862	2,479	1999

〈표 9〉는 네 개의 시나리오에 대한 결과를 보여준다.

예상한 바와 같이 돌연변이 연산자를 많이 사용한 시나리오 I과 II의 경우가 교차연산자를 사용한 시나리오 III, IV보다는 계산 소요시간이 더 많이 소요된 것을 알 수 있다. 이는 교차연산자가 전세대 염색체의 우수한 유전형질을 다음세대에도 전달하는 기능을 하고 있고, 그런 연유로 특정해에 빠르게 도달할 수 있는 개연성을 내포하고 있기 때문이다.

한편 가장 좋은(비용이 가장 적은) 노선대안은 시나리오 IV의 경우에 획득이 되었는데, 이는 예상을 조금 빗나간 결과이다. 이론적으로는 돌연변이 연산자를 많이 사용할 시 더 좋은 해를 찾는 경향이 있으며

그 이유로는 돌연변이 연산자의 큰 특징, 즉, 탐색영역(Search space)의 탐험(exploration)기능 때문이다.

그러나, 이러한 결과는 오히려 유전자 알고리즘의 확률적 특징을 잘 나타내주는 결과로 분석가능하다. 즉, 각각의 시나리오의 경우마다 서로 다른 해를 보여주며 시간과 비용이 주어지는 한 더 많은 최적화 시도를 할수록 더 좋은 해를 구할수 있다는 가능성이 있는 것이다. 실제로 위의 시나리오에서 다른 Seed를 이용하여 열 번의 추가적 최적화 시도를 수행한 결과 최적노선은 첫 번째 시나리오의 경우에서 나타났으며 그 총비용은 약 18.1 \$ million으로 나타났고 계산 소요시간은 3,886초로 나타났다(〈표 10〉).

한편, 세대가 진행될수록 즉, 탐색과정이 진행될수록 어떻게 더 좋은 노선대안이 찾아지는가를 목적함수의 변화를 통해 알 수 있다. 〈그림 5〉는 〈표 10〉의 3번째 경우에 대한 목적함수 값의 변화를 보여주는 데, 총 비용함수 값은 21번째 세대이내에서 급속히

개선되며, 이후 29번째와, 91번째 및 137번째 세대에서 약간의 개선이 이루어졌음을 알 수 있다. 비록 최적노선대안이 1948번째 세대에서 찾아졌지만 137번째 세대이후의 변화는 매우 미미했던 것으로 분석된다.

VI. 결론

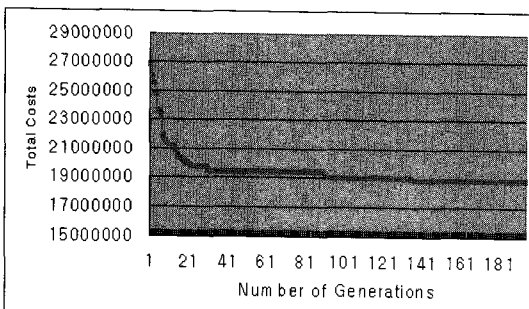
본 연구에서는 유전자 알고리즘을 적용한 도로선형 최적화 모형의 핵심탐색도구인 유전자 연산자에 대해 그 특성, 종류, 적용사례 등을 검증해 보았다. 지금까지 개발된 도로선형 최적화 모형내에 사용된 탐색알고리즘은 Calculus of variations, Network optimization, Dynamic programming, Enumeration, Linear programming, Numerical search 및 유전자알고리즘(Genetic algorithm) 등 7가지가 적용되었다. 이중 유전자 알고리즘은 도로선형 최적화 문제처럼 목적함수가 미분 불가능(Non-differentiable)하고, 국소최적점이 많이 존재(Noisy)하며 함축적(Implicit)인 경우에 매우 유용함이 알려져 있다.

유전자 알고리즘내에서 유전자 연산자는 크게 돌연변이 연산자(Mutation Operator)와 교차연산자(Crossover Operator)로 구분되어지는데, 그 이론적 토대를 기초로 풀고자하는 문제의 특성에 맞게 재제작·설계되어야 한다.

본 연구에서는 4개의 돌연변이 연산자(균일 돌연변이 연산자, 직선 돌연변이 연산자, 비균일 돌연변이 연산자, 전체 비균일 돌연변이 연산자)와 4개의 교차연산자(단순교차연산자, 두 개의 점을 이용한 교차연산자, 산술 교차연산자, 학습교차연산자)가 도로선형 최적화에 효과적으로 적용될 수 있도록 개발되었다. 4개의 돌연변이 연산자는 탐색영역(Search space)의 가능한 모든 부분을 탐험하도록 하는 장점(Exploration)이 있는 반면에 탐색영역이 커질수록 해를 찾는 소요시간이 길어지는 단점이 있다. 이와 반대로 4개의 교차연산자는 전 세대 염색체(Chromosome, 노선대안)의 우수한 유전형질을 다음세대에 효과적으로 전달하는 기능(Exploitation)을 통해 빠르게 해에 도달할 가능성과 이른 성숙(Prematurity)을 초래할 수 있는 단점을 가지고 있다. 따라서, 돌연변이 연산자와 교차연산자의 효과적 조합이 매우 중요한 과제임을 알 수 있다. 이들 연산자의 효과적 조합은 모든 문제

〈표 10〉 시나리오 I의 반복수행에 따른 최적해 도출결과

시나리오	Replication	노선 총비용 (\$million)	계산소요 시간(초)	최적해가 찾아진 세대번호
I	I-1	23.795	6,473	1,998
	I-2	21.084	5,980	1,996
	I-3	18.142	3,886	1,948
	I-4	20.481	6,215	1,972
	I-5	19.176	4,992	1,988
	I-6	20.124	6,192	1,999
	I-7	19.875	6,652	2,000
	I-8	20.576	5,762	1,989
	I-9	21.596	5,943	1,994
	I-10	22.382	6,014	1,969



〈그림 5〉 최적노선이 찾아진 시도(Replication)의 세대 별 목적함수변화

에 통일되게 적용되는 방법이 있는 것은 아니며, 개개의 문제특성에 맞게 고려·검토되어야 한다.

도로선형최적화 모형 내에 탐색알고리즘으로서의 유전자 알고리즘의 도입 및 탐색의 직접적인 도구로서의 유전자 연산자의 개발과 적용은 본 연구를 통해 매우 타당함이 제시되었다. 한편, 개발된 8개의 연산자 이외에도 도로선형 최적화를 보다 효과적으로 수행할 수 있는 연산자의 개발 여지는 이론적으로 무한하다고 할 수 있다. 또한 본 연구에서는 일일이 적시하지 않았지만 개개의 연산자의 적용에는 많은 파라미터(Parameter)가 관계한다. 이러한 많은 파라미터의 민감도 분석은 추후의 연구과제이다.

참고문헌

1. OECD(1973), "Optimisation of Road Alignment by the Use of Computers", Organisation of Economic Co-operation and Development, Paris.
2. Shaw, J. F. B. and Howard, B. E.(1982), "Expressway Route Optimization by OCP", Transportation Engineering Journal of ASCE, Proceedings of the American Society of Civil Engineers, ASCE, Vol.108, No.TE3, pp.227~243.
3. Fwa, T. F.(1989), "Highway Vertical Alignment Analysis by Dynamic Programming", Transportation Research Record 1239, TRB, National Research Council, Washington, D.C., pp.1~9.
4. Jha, Manoj K.(2000), "A Geographic Information Systems-Based Model for Highway Design Optimization", Ph. D. Dissertation, University of Maryland, College Park.
5. Jong, J-C(1998), "Optimizing Highway Alignments with Genetic Algorithms", Ph. D. Dissertation, University of Maryland, College Park.
6. 김응철(2001), "Modeling Intersections and Other Structures in Highway Alignment Optimization", 박사학위논문, University of Maryland, College Park.
7. Goldberg, D. E.(1989), "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Publishing Company, Inc., Massachusetts.
8. Lovell, D. J.(1999), "Automated Calculation of Sight Distance from Horizontal Geometry", Journal of Transportation Engineering, Vol. 125, No.4, pp.297~30
9. Mortenson, M. E.(1997), "Geometric Modeling", 2nd Edition, John Wiley & Sons, Inc., New York.
10. Michalewicz, Z.(1996), "Genetic Algorithms +Data Structures=Evolution Programs", Third Edition, Springer-Verlag, New York.

✉ 주 작 성 자 : 김응철

✉ 논문투고일 : 2003. 7. 15

논문심사일 : 2003. 8. 12 (1차)

2003. 8. 28 (2차)

2004. 3. 19 (3차)

심사관정일 : 2004. 3. 19

✉ 반론접수기한 : 2004. 8. 31