

# 최단경로문제에서 k-치명호를 결정하는 진화 알고리즘의 개발<sup>†</sup>

정호연\* · 김여근\*\*

## Development of Evolutionary Algorithms for Determining the k most Vital Arcs in Shortest Path Problem

Ho Yeon Chung\*\* · Yeo Keun Kim\*\*\*

### ■ Abstract ■

The purpose of this study is to present methods for determining the k most vital arcs (*k*-MVAs) in shortest path problem (SPP) using evolutionary algorithms. The problem of finding the *k*-MVAs in SPP is to find a set of k arcs whose simultaneous removal from the network causes the greatest increase in the shortest distance between two specified nodes. Generally, the problem of determining the *k*-MVAs in SPP has been known as NP-hard. Therefore, to deal with problems of the real world, heuristic algorithms are needed. In this study we present three kinds of evolutionary algorithms for finding the *k*-MVAs in SPP, and then to evaluate the performance of proposed algorithms.

Keyword : k most vital arcs, evolutionary algorithm, shortest path problem

논문접수일 : 2000년 12월 29일 논문게재확정일 : 2001년 5월 1일

† 본 연구는 한국과학재단 특정기초연구(과제번호 : 98-0200-09-01-3)지원으로 수행되었음.

\* 전주대학교 산업공학과

\*\* 전남대학교 산업공학과

## 1. 서 론

최단경로문제의 네트워크에서 가장 치명적인 영향을 미치는 호를 최단경로문제에서의 치명호(most vital arc : MVA)라고 한다.[2, 5-7, 11]. 최단경로문제에서  $k$ 개의 치명호를 찾는 문제는 최단경로문제에서  $k$ 개의 호의 제거로 시발점에서 종착점까지의 최단거리를 가장 크게 증가시키는  $k$ 개의 호 집합을 찾는 문제이다[2, 7, 11]. 이에 대한 문제는 Corley & Sha[6], Malik et.al.[11], Bar-Noy et.al.[6], 안재근 등[2]에 의해서 연구가 수행되었다. Corley & Sha[7]는  $k$ 번째 최단경로( $k$ -shortest path)를 구하는 해법을 이용하여, 최단경로문제에서 1개의 치명호(1-MVA)를 찾는 해법을 제시하였고,  $k$ 개의 치명호를 찾는 문제( $k$ -MVA)에 대해서는 충분조건을 제시하였다. Malik et.al.[11]은 Corley & Sha의 방법을 개선하여 무방향 네트워크에서 1-MVA를 찾는 해법을 제시하였고,  $k$ -MVA에 대해서도 비다항시간의 계산시간(non-polynomial time complexity)을 가지는 해법을 제시하였다. 그러나 이 문제는  $k$ -MVA를 찾는 문제가 NP-hard라는 사실을 증명한 Bar-Noy et.al.[6]에 의해 반례가 제시되어 해법에 오류가 있음이 판명되었다[2]. 안재근 등[2]은 최근 호와 경로와의 관계를 이용한  $k$ -MVA에 대한 수리모형식과 이에 대한 해법을 제시하였다. 이들은  $k$ -MVA가 제거된 상태에서 구한 최단경로의 길이를 이용하여 이 길이보다 짧은 경로들에는 적어도 하나 이상의 치명호가 포함되지만 그렇지 않는 경우의 경로에는 치명호가 포함되지 않음을 보였다. 그러나 이 해법은 실험결과가 제시되어 있지 않아 해법의 효율성을 파악하기 어려운 문제점을 안고 있다. 따라서 현재 최단경로문제에서  $k$ 개의 치명호를 찾는 문제에 대한 해법은 Corley & Sha가 제시한 충분조건과 이 문제가 NP-hard 라는 사실만이 알려져 있으며, 실제 최단경로문제를 대상으로 한 실험결과는 아직 어느 문헌상에서도 찾아볼 수 없는 실정이다. 이 문제는 그러나 Bar-Noy[6] 등에 의해 NP

-hard의 계산시간(complexity)을 갖는다는 사실이 밝혀졌기 때문에 최적화 알고리즘이 존재하더라도 문제의 크기가 커짐에 따라 계산시간이 기하급수적으로 증가하는 한계를 갖게 된다. 이러한 유형에 속한 문제는 최적해를 구하기 보다는 발견적 해를 구하는 것이 여러 면에서 경제적이다[1, 3]. 따라서 본 연구에서는 발견적 해법 중에서도 짧은 시간 안에 양질의 해를 제공하는 것으로 알려져 있는 진화 알고리즘을 적용하여 최단경로문제에서  $k$ 개의 치명호를 결정하는 3가지 종류의 해법을 개발하고, DIMACS[8]의 표준문제 생성기에서 생성된 20개의 문제에 대하여 실험을 통해 알고리즘의 성능을 평가해 보고자 한다

## 2. $k$ -MVA 결정을 위한 유전 알고리즘

$k$ -MVA를 구하는 가장 원시적인(naive) 방법은 주어진 네트워크에서 호의 수가  $m$ 일 때  $\binom{m}{k}$  =  $\frac{m!}{k!(m-k)!}$  번의 최단경로문제를 풀어서 해를 구하는 방법이다. 그러나  $m$ 의 수가 클 경우에 이러한 계산횟수를 수행한다는 것은 현실적으로 불가능하기 때문에 그 동안 학자들은 효율적으로  $k$ -MVA를 결정하는 방법에 관하여 연구를 수행하여 왔다. 그러나 현재까지 이에 대한 명확한 해법이 제시되어 있지 못하기 때문에 본 연구에서는 단순 유전알고리즘(Simple Genetic Algorithm ; 이하 SGA), 안정상태 유전알고리즘(Steady State Genetic Algorithm ; 이하 SSSGA), 생태계 유전알고리즘(ECOsystem Genetic Algorithm ; 이하 ECOGA) 3종류의 유전알고리즘을 사용하여 최단경로문제에서  $k$ 개의 치명호를 찾는 방법을 제시하고자 한다.

### 2.1 개체표현 방법 및 모집단 구성

유전알고리즘을 적용하기 위해서는 먼저 주어진

문제의 특성을 반영하는 개체 표현방법이 필요하다. 전통적인 유전알고리즘에서는 이진표현을 많이 사용하고 있으나 본 연구에서 다루는 네트워크문제는 해를 이진수로 표현하기 어렵고, 표현한다고 하더라도 이진표현에서 유전연산이 가능해를 유지하기 어렵기 때문에 적합하지 않다[1]. 본 연구에서는 이진표현 대신 주어진 네트워크의 호를 표현하는 두 가지 방법을 검토하였다. 먼저 호를 표현할 때 마디(node)를 사용하여 호를 표현하는 방법과 호에 일련번호를 부여하여 호를 표현하는 방법을 고려했다. 첫 번째에 고려한 마디를 사용하여 호를 표현하는 방법은 표현방법이 쉽고 해석이 용이한 장점이 있지만 해의 탐색능력이 떨어지고 유전 연산시에 비가능호가 발생할 수 있기 때문에 추가적인 보수(repair)방법이 요구되는 문제점이 있다. 반면에 호에 일련번호를 부여하여 호를 표현하는 방법은 마디를 사용하는 방법보다 해의 탐색능력이 우수하고 비가능호가 발생되지는 않지만 대신 중복호가 발생할 수 있다. 따라서 이에 대한 문제점이 보완된다면 호에 일련번호를 부여하여 호를 표현하는 방법이 더 좋은 방법이라 할 수 있다. 본 연구에서는 호에 일련번호를 부여하여 호를 표현하는 방법을 사용한다. 이 표현을 사용하면 하나의 개체는 치명호에 해당하는 k개의 호로 구성되는 호 인자들으로써 개체가 표현될 수 있으며, 그렇게 함으로써 k-MVA에 대한 잠재해를 표현할 수 있게 된다.

SGA와 SSGA의 경우에 있어서 초기 모집단의 크기(population size)는 임의로 생성된 k개의 호 집합으로 구성되는 개체 100~500개의 범위 내에서 적절히 선택할 수 있게 하였으며, ECOGA의 경우에는 모집단을 2차원 격자구조의 토러스 형태로 구성하고  $r \times r$ 의 정사각형 구조를 갖는 이웃을 사용하여 진화되게 하였다.

## 2.2 적응도 평가 및 계산

본 연구에서 정의한 개체는 치명호에 해당하는 k

개의 호로 구성된다. 이에 대한 적응도는 개체를 구성하는 k개의 호를 제거한 다음 최단경로의 길이를 구한 값과 원문제에서의 최단경로의 길이의 차로써 계산된다. 이를 수식으로 표현하기 위해 먼저 용어를 정의하면 다음과 같다.

- $G = (N, A) : |N| = n$  개의 마디와  $|A| = m$  개의 호로 구성된 최단경로문제
- $k : G = (N, A)$  에서 구하고자 하는 치명호의 갯수
- $L :$  주어진  $G = (N, A)$  에 대한 최단경로의 길이
- $L' : G = (N, A \setminus k)$  에 대한 최단경로의 길이

위의 수식을 사용하면 다루는 문제의 목적함수는 다음과 같이 표현된다.

$$\begin{aligned} \max \quad & |L - L'| \\ \text{s.t. given} \quad & G = (N, A \setminus k) \end{aligned}$$

따라서 개체에 대한 적응도 값은 그 값이 클수록 좀 더 치명적(vital)이라고 말할 수 있다. 최단경로를 구할 때 사용하는 알고리즘은 Label setting algorithm에 속하는 Dijkstra 알고리즘을 사용한다.

## 2.3 선별 방법(selection method)

SGA의 경우 현재 모집단의 우수한 개체를 지속적으로 유지하기 위해 Goldberg에 의해 제안된 우수개체 보호전략(elitism)을 사용한다[8]. 이를 위해 먼저 모집단  $P(t)$ 의 적응도 값을 내림차순으로 정렬한 후, 최상위 개체 하나를 선택하여 복사(copy)해 놓는다. 현재의 모집단  $P(t)$ 를 대상으로 토너먼트 선별을 실시하여 다음 세대  $P(t+1)$ 를 생성시킨 다음 적응도 값에 따라 내림차순으로 정렬한다. 이 때의 최상위 개체 값과 그 이전 세대에서 복사해 놓은 최상위 개체 값과의 비교를 통해 더 우수한 개체를  $P(t+1)$ 에 그대로 유전시킨다. 이 방법을 사용하면 세대가 진행됨에 따라 모집단 내의 우수개체는 계속 유지되게 되고, 토너먼트 선별

을 통해 전 세대에서 구한 우수개체보다 더 우수한 개체를 지속적으로 발생하게 함으로써 좋은 해를 발생할 가능성을 높이게 된다.

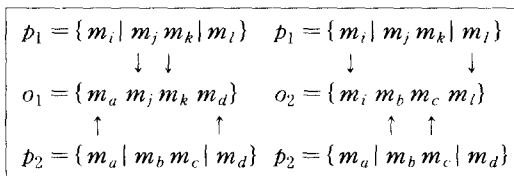
SSGA의 경우에는 모집단 전체가 세대 단위로 진화하는 표준 유전알고리즘과 달리 일부의 개체만을 반복 재생산하여 이들 개체를 적응도가 낮은 개체와 대체하는 대체전략을 사용한다. 이렇게 하면 좋은 자손이 생성되는 즉시 이 개체가 갖는 유전 정보를 이용하여 좋은 해의 이용을 높일 수 있게 된다.

ECOGA의 경우는 2차원 격자구조를 갖는 토러스 형태에서 임의의 위치  $(i, j)$ 에 존재하는 이웃  $N_{ij}$ 를 정의한 다음,  $N_{ij}$ 에서 적응도를 기준으로 두 개체를 확률적으로 선별하게 된다.

### 2.4 유전 연산자

유전 연산자는 교차와 돌연변이로 구성되는데, 본 논문에서는 교차나 돌연변이를 실시할 때 발생할 수 있는 중복을 방지하기 위하여 순서교차(order crossover)를 사용한다. 순서교차는 한 부모로부터 부분적인 순서(인자)를 받아들이고, 다른 부모로부터 상대적인 순서를 유지하여 받아들이는 교차방법이다[1]. 본 연구에서는 상대적인 순서를 받아들이는 개념을 수정하여 중복이 발생할 경우에 한해 해당하는 부모개체를 그대로 상속하게 하는 개념으로 변형하여 적용하였다. 순서교차에서 절단점은  $k=2$ 일 때에는 일점(one-point) 절단점을 사용하고,  $k=3$  이상일 때는 이점(two-point) 절단점을 임의로 선택하여 교차를 실시하였다.

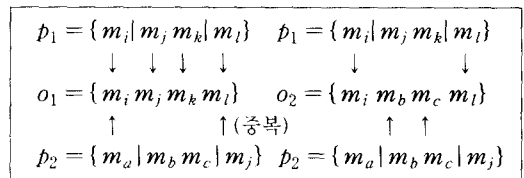
$k=4$ 인 경우의 순서교차 예가 [그림 1]에 나타나 있다. [그림 1]은 부모  $p_1$ 과  $p_2$ 로부터 절단점



[그림 1] 순서교차 방법 (중복 미 발생 경우)

2개가 임의로 지정되면 그에 따라 자손  $o_1$ 과  $o_2$ 가 두 개의 절단점을 기준으로 하여 부모  $p_1$ 으로부터 가운데 인자를 상속받고, 부모  $p_2$ 로부터 가운데 인자를 제외한 나머지 인자를 상속받아 발생됨을 보여주고 있다.

[그림 2]의 경우에는 부모  $p_1$ 으로부터  $m_j$ 와  $m_k$ 를 상속받고  $p_2$ 로부터  $m_a$ 와  $m_j$ 를 상속받아 자손  $o_1 = \{m_a m_j m_k m_l\}$ 가 생성되나 인자  $m_j$ 가 중복이 되기 때문에 이를 방지하기 위해 부모  $p_1$ 의 인자를 그대로 상속받아 자손  $o_1 = \{m_i m_j m_k m_l\}$ 이 생성되는 과정을 보여주고 있다.



[그림 2] 순서교차 방법 (중복이 발생될 경우)

돌연변이(mutation)는 개체에 완전히 새로운 유전자를 생성시키는 단계로써 해 공간을 다양하게 탐색하게 해주는 역할을 한다. 본 연구에서 사용한 돌연변이는 원소 단위로 행하여지며, 중복이 발생할 경우에는 다른 원소를 발생시켜 중복이 되지 않는 원소로 돌연변이 시킨다. 그 절차는 다음과 같다.

#### 단계 1 : (돌연변이 될 인자 선택)

- (1.1) 모집단을 구성하고 있는 개체 중에서 돌연변이율에 따라 임의의 개체  $v_c$ 를 돌연변이 개체로 선택한다.
- (1.2)  $v_c$ 를 구성하는 원소 각각에 난수를 발생시켜 돌연변이 되는 원소  $m_j$ 를 선택한다.

#### 단계 2 : (돌연변이 인자 생성)

- 네트워크를 구성하고 있는 호 중에서 임의의 호  $m_b$ 를 생성한다.

#### 단계 3 : (중복 인자 발견)

- (3.1)  $v_c$  중에서  $m_j$ 를 제외한 나머지 인자 중 어

는 인자도  $m_p$ 와 같지 않으면 단계 4로 간다.

(3.2) 그렇지 않으면 단계 2로 간다.

**단계 4 :** (돌연변이)

돌연변이 개체  $v_c$  중에서  $m_j$ 를  $m_p$ 로 돌연변이 시킨다.

### 2.5 종료조건

본 연구에서는 새로 생성된 개체수가 1000개 이상이면 더 이상 해를 개선시킬 수 없다고 보고 실험을 종료하도록 하는 종료조건을 사용한다.

#### □ k-MVA를 결정하기 위한 단순 유전알고리즘(SGA)

**단계 1 :** (초기 모집단 구성)

임의 생성법을 사용하여 초기 모집단  $P(t)$ 를 구성한다.

**단계 2 :** (적응도 평가)

$P(t)$ 에 있는 모든 개체의 적응도를 평가한다.

**단계 3 :** (선별)

$P(t)$ 로부터  $P(t+1)$ 를 선별한다. 이를 위해  $P(t)$ 의 적응도 값을 내림차순으로 정렬했을 때 나타나는 최상위 개체 값과 그 이전 세대에서 복사해 놓은 최상위 개체 값과의 비교를 통해 더 우수한 개체를  $P(t+1)$ 에 유전시키고, 나머지 개체들은 토너먼트 선별을 통해 세대를 구성한다.

**단계 4 :** (교차)

현재의 모집단에서 임의로 선정된 두 개체로 부터 순서교차를 사용하여 자손 개체를 생성한다. 순서교차에서 절단점은  $k=2$ 일 때에는 일점 절단점을 사용하고,  $k=3$  이상일 때에는 이점 절단점을 임의로 발생시켜 교차를 실시한다.

**단계 5 :** (돌연변이)

모집단을 구성하고 있는 개체 중에서 돌연변이율에 따라 임의의 개체를 돌연변이

개체로 선택하고, 선택된 개체 중에서 임의의 원소를 선택해 중복이 되지 않는 임의의 원소로 돌연변이 시킨다.

**단계 6 :** (종료조건 검사)

종료조건을 만족하면 끝내고, 그렇지 않으면  $t = t+1$ 로 두고 단계 2로 간다.

#### □ k-MVA를 결정하기 위한 안정상태 유전알고리즘(SSGA)

표준 유전알고리즘은 모집단 전체가 세대 단위로 진화하는 방법이다. 이에 반하여 SSGA은 단지 몇 개의 개체(흔히 하나 또는 두 개의 개체)만을 반복하여 재생산하여 이들 개체를 적응도가 낮은 개체와 대체하는 전략을 사용한다. 이 알고리즘의 장점은 적응도가 낮은 개체와 대체함으로써 자연스럽게 가장 좋은 개체를 보존할 수 있다는 것이다. 그리고 하나의 좋은 개체가 생산되면 이 개체가 바로 재생산에 참여할 수 있다. 즉, 좋은 자손이 생성되는 즉시 이 개체가 갖는 유전 정보를 이용하여 좋은 해의 이용(exploitation)을 높일 수 있다는 것이다. 알고리즘의 절차는 다음과 같다.

**단계 1 :** (초기모집단)

$t \leftarrow 0$ 으로 두고  $P(t)$ 를 생성한다. 즉, 초기모집단을 만든다.

**단계 2 :** (적응도 평가)

$P(t)$ 에 있는 모든 개체의 적응도를 평가한다.

**단계 3 :** (선별)

$t \leftarrow t+1$ 로 두고  $P(t-1)$ 로 부터 소수의 개체들을 선별한다. 선별된 개체의 집합을  $sub-P(t)$ 로 둔다.

**단계 4 :** (교차와 돌연변이)

$sub-P(t)$ 의 개체에 대해 교차와 돌연변이를 행한다.

**단계 5 :** (대체)

대체전략에 따라  $P(t)$ 에서 대체시킬 개체를 선택하여 이들을  $sub-P(t)$ 의 개체로 대체한다.

**단계 6 :** (적응도 평가)

새로 생성된 자손 개체들 즉,  $sub-P(t)$ 의 개체들의 적응도를 평가한다.

**단계 7 :** (종료조건)

종료조건을 만족하면 끝낸다. 그렇지 않으면 **단계 3**으로 간다.

SSGA에서는 (모집단의 크기 $\times$ 대체율)에 의해 대체 모집단의 크기가 정의되며, 대체 모집단의 크기 만큼만 대체 된다. 대체 방법으로는 다음 세 가지가 있다. 프로그램에서는 확정적 대체가 사용되었다.

**(1) 확정적 대체**

적응도가 가장 낮은 개체부터 재생산된 자손개체와 동일한 수의 개체를 선택하여 자손개체와 이를 대체한다.

**(2) 임의 대체**

임의의 개체를 선택하여 재생산된 자손개체와 대체한다.

**(3) 확률적 대체**

대체확률에 따라 자손개체와 대체될 개체를 선택한다. 여기에서 대체확률은 적응도의 역에 기초한 확률바퀴(roulette wheel) 선별을 이용하여 구해진다. 대체확률은 적응도가 낮은 개체일수록 높다.

### □ $k$ -MVA를 결정하기 위한 생태계 유전 알고리즘(ECOGA)

생태계 유전알고리즘은 한정된 공간 즉, 이웃에서 짝짓기를 하면서 진화하는 알고리즘이다. 모집단을 2차원 격자구조의 토러스 형태로 구성하고  $r \times r$ 의 정사각형 구조를 갖는 이웃을 사용한다. 이웃에 있는 개체들은 안정상태 유전알고리즘에 기초하여 진화시킨다. 여기서 이웃  $N_{ij}$ 는 모집단에서 위치  $(i, j)$ 에 존재하는 개체를 중심으로 한 정사각형 구조를 갖는다. ECOGA는 초 우수개체에 의해 모집단이 국부 최적으로 조기 수렴하는 것을 방지하며, 다양한 해를 탐색할 수 있다는 장점을 갖는다. 생태계 유전 알고리즘의 절차는 다음과 같다.

**단계 1 :** (초기모집단)

$t \leftarrow 0$ 으로 두고  $P(t)$ 를 생성한다. 즉, 초기모집단을 만든다.

**단계 2 :** (적응도 평가)

$P(t)$ 에 있는 모든 개체의 적응도를 평가한다.

**단계 3 :** (이웃 설정)

임의의 위치  $(i, j)$ 를 선택하고, 그 이웃  $N_{ij}$ 를 정의한다.

**단계 4 :** (진화)

(4.1)  $N_{ij}$ 에서 적응도를 기준으로 두 개체를 확률적으로 선별한다.

(4.2) 선별된 두 개체를 유전연산(교차와 돌연변이)하여 두 자손을 생산한다.

**단계 5 :** (대체)

생산된 두 자손을 대체전략에 의해  $N_{ij}$ 에 있는 두 개체와 교체한다.

**단계 6 :** (적응도 평가)

새로이 생산된 자손개체의 적응도를 평가한다.

**단계 7 :** (종료조건)

종료조건을 만족하면 끝낸다. 그렇지 않으면 **단계 3**으로 간다.

## 3. 실험 및 성능평가

최단경로문제에서  $k$ -MVA를 결정하기 위한 유전 알고리즘의 실험은 DIMACS[8]의 표준문제 생성기에서 생성된 20개의 문제를 대상으로 실험하였다. 제안한 유전알고리즘은 Visual C++ 컴퓨터 언어를 사용하였으며, 128M RAM을 가진 350MHz의 Pentium-II CPU를 장착한 IBM PC 호환기종에서 수행되었다.

### 3.1 유전파라미터 값 설정을 위한 예비실험

제안된 유전 알고리즘으로 좋은 해를 찾기 위해서는 유전 알고리즘에 민감하게 영향을 미치는 유

<표 1> 예비 실험계획

유전 파라미터	수준		
	SGA	SSGA	ECOGA
모집단의 크기	100/200/300/400/500	100/200/300/400/500	-
교차율	0.5/0.6/0.7/0.8/0.9	0.5/0.6/0.7/0.8/0.9	0.5/0.6/0.7/0.8/0.9
돌연변이율	0.1/0.2/0.3	0.1/0.2/0.3	0.1/0.2/0.3
대체율	-	0.1/0.15/0.2	-
생태계크기	-	-	10/15/20/25/30
종료조건	새로 삽입되는 개체가 1000개 이상이면 종료	새로 삽입되는 개체가 1000개 이상이면 종료	새로 삽입되는 개체가 1000개 이상이면 종료

전 파라미터의 값을 최적으로 설정하여야 한다. 최적 유전 파라미터의 값을 결정하기 위하여 본 연구에서는 DIMACS의 표준문제 생성기에서 생성된 spagrid3 문제를 대상으로 예비실험을 수행하였다. spagrid3 문제는 마디(node)의 수가 661개이고 호의 수가 1980개인 최단경로문제이며, 실험에서는 치명호의 수가 3인 3-MVA를 구하기로 하였다. 예비실험은 먼저 모집단의 크기를 100으로 설정한 후 교차율과 돌연변이율을 여러 단계로 값을 변경해 가면서 각 조합당 20회씩의 반복실험을 통해 최적 교차율과 돌연변이율을 탐색하였다. 이 때 각각의 교호작용은 무시하였으며, 20회의 반복실험 중에서 가장 높은 적응도의 평균값을 기준으로 최적 유전 파라미터의 값을 설정하였다. 각 조합 조건에

대한 실험의 종료조건으로는 실험을 통해 새롭게 생성된 개체의 수가 1000개 이상이면 더 이상 해가 개선될 가능성이 없기 때문에 새로 생성된 개체의 수가 1000개 이상일 때 실험을 종료하는 종료조건을 사용하였다. 아래 <표 1>에 각 알고리즘별로 최적 교차율과 돌연변이율을 탐색하기 위한 예비 실험계획이 나타나 있다.

위의 <표 1>에 나타나 있는 예비 실험계획에 따라 SGA와 SSGA의 경우 모집단의 크기를 100으로 고정된 상태에서 5가지의 교차율과 3가지의 돌연변이율에 따른 15가지의 조합에 대하여 각각 20회씩 반복 실험을 수행하였고, ECOGA의 경우 생태계의 크기를 10으로 고정된 상태에서 5가지의 교차율과 3가지의 돌연변이율에 따른 15가지의 조

<표 2> 예비실험 결과

순서	파라미터 조합	SGA		SSGA(대체율 = 0.15)		ECOGA	
		best fitness	average fitness	best fitness	average fitness	best fitness	average fitness
1	(0.5/0.1)	1915	1374	1542	912	1780	1108
2	(0.5/0.2)	2138	1392	1676	855	2036	1168
3	(0.5/0.3)	2109	1313	1484	973	1632	1043
4	(0.6/0.1)	2080	1319	1542	1009	1535	1014
5	(0.6/0.2)	2081	1319	1464	925	1749	970
6	(0.6/0.3)	1915	1216	2037	902	1828	1150
7	(0.7/0.1)	2190	1283	1846	962	1750	1033
8	(0.7/0.2)	1863	1407	1542	1019	1846	1091
9	(0.7/0.3)	2116	<b>1430</b>	1484	899	1594	1142
10	(0.8/0.1)	2180	1401	1846	1008	1487	1013
11	(0.8/0.2)	1846	1270	2215	989	1908	<b>1200</b>
12	(0.8/0.3)	1968	1412	1269	934	1508	991
13	(0.9/0.1)	1924	1390	1799	1043	1808	1024
14	(0.9/0.2)	1951	1341	1565	<b>1063</b>	1915	1066
15	(0.9/0.3)	1862	1216	1476	902	1915	1089

〈표 3〉 추가 예비실험결과

순서	SGA			SSGA			ECOGA		
	(교차율/ 돌연변이율)	best fitness	average fitness	(교차율/돌연변이율/ 대체율)	best fitness	average fitness	파라미터조합 (교차율/돌연변이율)	best fitness	average fitness
1	(0.7/0.3)	2116	1394	(0.6/0.3/0.1)	1542	1000	(0.8/0.2)	2633	1164
2	(0.8/0.3)	2138	1307	(0.9/0.2/0.15)	1565	1016	(0.5/0.2)	1686	1058
3	(0.7/0.2)	1863	1334	(0.8/0.1/0.1)	2037	993	(0.6/0.3)	1828	1125

〈표 4〉 모집단의 크기나 생태계의 크기에 따른 평균적응도 값의 변화

모집단 크기	SGA		SSGA		생태계크기	ECOGA	
	best fitness	average fitness	best fitness	average fitness		best fitness	average fitness
100	2116	1430	1565	1063	10	1908	1200
200	4631	1516	1380	1028	15	2079	1286
300	1796	1086	4006	1389	20	2059	1198
400	1589	973	1492	1052	25	1376	1108
500	2037	1179	1846	1072	30	1484	1098

함에 대하여 각각 20회씩 반복 실험을 수행하였다. 그 결과가 아래 <표 2>에 정리되어 있다.

아래의 <표 2>는 교차율과 돌연변이율의 각 조합에 대하여 20회씩의 반복실험을 통해 구해진 가장 좋은 적응도(best fitness)값과 평균 적응도(average fitness)의 값을 보여준다.

본 연구에서는 위의 예비실험 결과에서 좋은 알고리즘의 성능을 보인 파라미터 조합에 대하여 우수한 성능에 대한 재현성을 확인하기 위하여 적응도 평균값 상위 3개의 조합에 대하여 추가실험을 실시하였다. 최고의 적응도(best fitness) 보다 평균 적응도(average fitness) 값이 높은 파라미터의 조합을 선택한 이유는 우연성을 배제하기 위함이며 안정적으로 해를 산출할 수 있는 파라미터의 조합을 구하기 위함이다. 다음 <표 3>은 SGA의 경우(교차율/돌연변이율)이 (0.7/0.3), (0.8/0.3), (0.7/0.2)의 조건에 대하여, SSGA의 경우(교차율/돌연변이율/대체율)이 (0.6/0.3/0.1), (0.9/0.2/0.15), (0.8/0.1/0.1)에 대하여, ECOGA의 경우(교차율/돌연변이율)이 (0.8/0.2), (0.5/0.2), (0.6/0.3)에 대하여 20회씩 반복실험을 통해 구한 추가 예비실험의 결과 및 그 값들의 평균값을 나타낸다.

위의 <표 3>은 추가 실험결과 SGA의 경우(교차율/돌연변이율)이 (0.7/0.3)일 때, SSGA의 경우

(교차율/돌연변이율/대체율)이 (0.9/0.2/0.15)일 때, ECOGA의 경우(교차율/돌연변이율)이 (0.8/0.2)일 때 가장 좋은 평균 적응도 값을 나타내고 있음을 보여 주고 있다. 이는 기존 예비실험의 결과와도 일치되는 것으로써 실험 결과에 대한 재현성을 나타내고 있다.

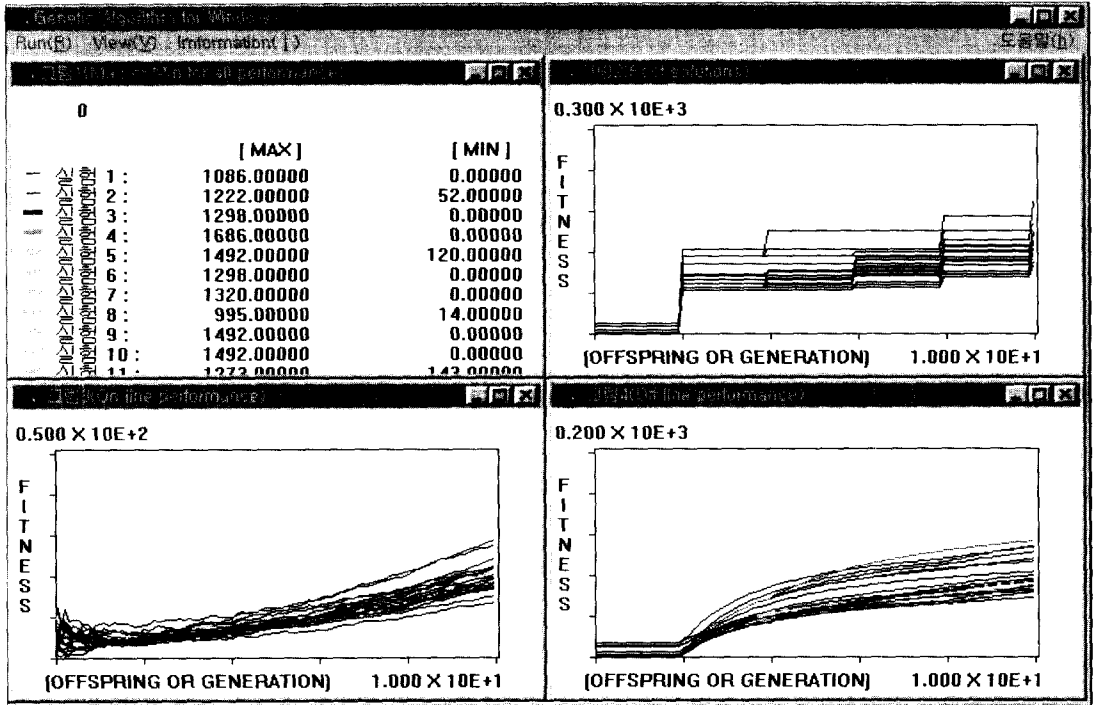
최적 모집단의 크기나 최적 생태계의 크기는 앞의 예비실험에서 구했던 최적 파라미터 값을 고정시켜 놓은 상태에서 모집단의 크기를 몇 단계로 변경시켜 가면서 20회씩의 반복실험을 통해 탐색하였다. 모집단의 크기에 따른 평균 적응도 값의 변화가 다음 <표 4>에 나타나 있다.

위의 <표 4>에서 보는 바와 같이, SGA의 경우에는 모집단의 크기가 200일 때, SSGA의 경우에는 300일 때 평균적응도 값이 가장 높게 나타났

〈표 5〉 각 알고리즘별로 결정된 최적 유전파라미터값

파라미터	수준		
	SGA	SSGA	ECOGA
모집단의 크기	200	300	-
생태계 크기	-	-	15
교차율	0.7	0.9	0.8
돌연변이율	0.3	0.2	0.2
대체율	-	0.15	-
종료조건	새로 생성되는 개체가 1000개 이상		





[그림 3] 프로그램 실행화면

으며, ECOGA의 경우에는 생태계의 크기가 15일 때 평균 적응도의 값이 가장 우수하게 나타났다. 이상의 예비실험 결과, 가장 좋은 알고리즘의 성능을 보인 최적 파라미터의 조합은 다음 <표 5>와 같이 최종 결정되었다.

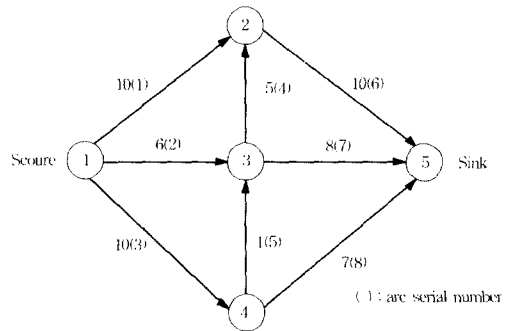
<표 5>의 최적 유전 파라미터 값을 사용하여 spagrid3 문제에 대한 실험을 실시한 결과는 <표 7>에 나타나 있다. 다음 [그림 3]은 SGA로 spagrid3 문제를 풀 때 볼 수 있는 프로그램의 실행 화면을 나타내고 있다.

위의 [그림 3]에서 왼편상단에 나타난 그림은 각 실험에서 생성된 전체 모집단 개체들의 최대 적응도 값과 최소 적응도 값을 나타내고, 오른편 상단에 나타난 그림은 모집단에 삽입되는 개체들에 대하여 10개체 단위별로 탐색된 전체 모집단의 Best Fitness의 개선상태를 보여준다. 왼편 하단에 나타난 그림은 모집단에 삽입되는 개체들에 대하여 10개체 단위별로 모집단 전체 개체들의 평균값의 변

화를 나타내 주고, 오른편 하단의 그림은 모집단에 삽입되는 개체들에 대하여 10개체 단위별로 Best Fitness의 평균값의 변화를 보여주고 있다.

### 3.2 실험결과 및 분석

본 연구에서는 먼저 DIMACS의 표준문제 생성



[그림 4] test 2의 최단경로문제

〈표 6〉 작은 규모의 문제에 대한 실험결과(복수 최적해 포함)

문제명 (마디수, 호의수)	best fitness	average fitness	2 - MVA	근사해/최적해 여부
test 1 (5,7)	$\infty$	$\infty$	{3,6},{7,6}	최적해
test 2 (5,8)	6	6	{3,2},{3,7},{7,8}	최적해
test 3 (6,10)	$\infty$	$\infty$	{1,2},{7,5},{7,10},{8,5},{10,8}	최적해
test 4 (7,10)	$\infty$	$\infty$	{1,2},{9,8},{9,10}	최적해
test 5 (7,11)	$\infty$	$\infty$	{1,2},{11,10}	최적해
test 6 (8,13)	7	7	{1,7},{6,7}	최적해
test 7 (9,13)	$\infty$	$\infty$	{1,2},{12,13}	최적해
test 8 (7,15)	$\infty$	$\infty$	{1,2},{14,15}	최적해
test 9 (10,15)	19	19	{14,15}	최적해
test 10 (10,18)	$\infty$	$\infty$	{17,18}	최적해

기에서 생성된 문제를 대상으로 실험하기에 앞서 최적해를 알고 있는 10개의 test 문제에 대하여 본 연구에서 제안한 알고리즘의 성능을 평가하였다. 10개의 test 문제는 호의 수가 5~18개인 작은 규모 (small size)의 최단경로문제로 구성되었다. [그림 4]는 마디 수가 5개이고, 호의 수가 8개인 최단경로문제(test 2)를 나타낸다.

이 문제에 대하여 제안한 알고리즘 SGA, SSGA, ECOGA를 각각 적용하여 2-MVA를 찾는 결과 세 알고리즘 모두에서 {3,2}, {3,7}, {7,8}의 복수 최적해가 동일하게 산출되었다(〈표 5〉의 test 2 참조). 10개의 test 문제에 대한 실험결과는 다음 〈표 6〉에 나타나 있다

실험 결과 10개의 test 문제 모두에서 정확히 최적해를 찾아내 제안한 3가지의 유전 알고리즘의 정확도가 매우 우수함을 확인할 수 있었다.

본 실험은 앞 절의 예비실험을 통해서 구했던 〈표 5〉의 최적 유전 파라미터 값을 사용하여 DIMACS의 표준문제 생성기에서 생성된 20개의 문제를 대상으로 실험을 실시하였다. 그 결과가 다음 〈표 7〉에 정리되어 있다.

〈표 7〉의 실험 결과를 보면 ECOGA는 20개의 문제중 10개의 문제에서 평균 적용도 값이 다른 알고리즘 보다 더 높게 나타나 SGA나 SSGA 보다 우수하였다. 이는 모집단을 2차원 격자구조의 토러스 형태로 구성하고  $r \times r$ 의 정사각형 구조를 갖는 이웃을 사용하면서, 이웃간에는 안정상태 유전

알고리즘에 기초하여 진화시키는 ECOGA의 기본적인 특성이 문제에 잘 반영되고 있기 때문인 것으로 파악된다. SGA는 20개의 문제 중 6개에서, SSGA는 4개에서 다른 알고리즘 보다 평균 적용도 값이 더 높게 나타나 SGA가 SSGA보다 더 좋은 결과를 보여 주었다. 이는 SGA 개발시 표준 유전 알고리즘의 단점을 피하기 위해 우수개체 보호전략(elitism)을 사용한 때문으로 풀이된다.

〈표 7〉의 실험문제에 대한 소요시간은 대상문제에 따라 대략적으로 3분에서부터 40분 정도까지 소요되었다. 실시간내에 해를 찾지 못하는 이유는 한 세대당 200~300번 정도의 최단경로문제를 반복적으로 풀면서 세대를 진행하기 때문이다.

## 4. 결 론

본 논문에서는 최단경로문제에서  $k$ -MVA를 결정하는 진화 알고리즘으로 단순 유전 알고리즘(SGA)과 안전상태 유전 알고리즘(SSGA), 그리고 생태체계 유전 알고리즘(ECOGA)을 제시하였다. 최단경로문제에서  $k$ 개의 치명호를 찾는 문제에 대한 해법은 Corley & Sha가 제시한 충분조건과 이 문제가 NP-hard 라는 사실만이 알려져 있으며, 실제 최단경로문제를 대상으로 한 실험결과는 아직 문헌상에 나타나 있지 않다. 그러나 본 논문에서 제시한 해법은 DIMACS의 표준문제 생성기에서 생성된 20개의 문제에 적용해 본 결과 호의 수

〈표 7〉 중·대형 문제에 대한 실험 결과

problem	SGA			SSGA			ECOGA		
	best fitness	average fitness	3-MVA	best fitness	average fitness	3-MVA	best fitness	average fitness	3-MVA
Spagrid 3 (661,1980)	4631	1641	{1317,1365,1320}	4006	1389	{1320,1317,735}	2079	1286	{1476,1666,1676}
Spagrid 4 (1289,3864)	1882	1230	{362,2197,3397}	1604	1057	{348,2201,1525}	1665	1153	{1602,112,3675}
Spagrid 5 (1036,3105)	2869	2131	{2650,2172,2808}	2126	1530	{863,2213,1506}	2735	1898	{2808,2322,237}
Spagrid 6 (1601,4800)	3445	2239	{5,2124,3359}	2492	1716	{4004,1,2118}	3553	2290	{3245,2508,4416}
Spagrid 7 (1701,5100)	1556	938	{1156,2349,3045}	1068	675	{3643,4897,350}	1638	798	{2335,4477,3643}
Spagrid 10 (2001,6000)	9935	5400	{436,907,4109}	7155	5212	{4109,53,2209}	8973	4149	{909,4476,4109}
Sprand 3 (409,2000)	2746	1326	{1510,1354,1799}, {1510,1469,308}	2746	2766	{1510,1030,414}, {1510,110,1534}	2746	1713	{440,332,1510}, {730,369,1510}
Sprand 4 (508,3000)	7357	5376	{1969,1249,1711}, {742,1711,298}	7357	7373	{2894,2140,1711}, {2991,710,1771}	7846	6311	{377,2978,1711}, {565,583,1771}
Sprand 5 (890,5000)	3958	2714	{222,997,2802}	2641	2626	{1487,1503,212}, {3465,212,4458}	3958	2703	{1314,222,212}
Sprand 6 (999,5003)	2142	2142	{3755,2914,1671}, {3080,1,4950}	2142	2142	{1,462,3562}, {3051,3524,1}	4508	2378	{3525,1,4868}
Sprand 7 (1200,6000)	812	812	{189,5844,169}, {5844,388,2663}	812	812	{1901,104,5350}, {3374,5844,5665}	2246	955	{5726,4774,4353}
Sprand 8 (1367,7645)	6082	5821	{5357,7234,5974}, {759,4412,1227}	6082	5908	{4840,1680,7234}, {2272,201,3880}	6082	5970	{2831,7234,5187}, {7260,5009,201}
Sprand 9 (1789,3837)	14835	7713	{1042,3694,533}, {1660,3694,533}	14835	9675	{40,3694,1692}, {490,734,3694}	27074	9443	{2083,3694,2945}
Sprand 10 (2000,8999)	3904	3396	{7096,1439,8988}, {1293,2675,6791}	3904	3395	{8988,4749,8418}, {2675,240,5485}	3904	3457	{8988,5557,1108}
Sprand 11 (2789,10000)	3531	3531	{3355,5974,3918}, {4266,1508,3910}	7639	4215	{278,150,2788}, {2788,9882,3753}	infinity	6300	{3082,2789,2492}, {2789,164,1698}
Spacyc 5 (530,2450)	12567	4800	{1,633,1404}	7517	5351	{1,1278,11321}, {1,788,13151}	7517	4458	{1,274,294}, {330,545,1}
Spacyc 6 (770,3050)	5799	5799	{2398,1827,2184}, {2885,1315,459}	5799	5799	{1831,459,9921}, {594,459,1508}	8014	5983	{1477,2184,3017}
Spacyc 7 (1120,7025)	2359	1718	{2061,5180,3720}	2359	1658	{15,3720,3853}	2359	1825	{1532,3720,6026}, {3245,3720,5954}
Spacyc 8 (1597,14364)	7135	7135	{4676,1076,6393}	7135	4575	{4676,13598,887}	7135	4941	{4676,11604,5925}, {4676,8120,12245}
Spacyc 10 (1956,20140)	2232	2232	{4169,6574,1598}, {1299,19032,6574}	5799	1204	{8028,7127,3760}, {3483,3160,7127}	5799	3421	{3765,7127,13625}

가 20,000개 이상인 대형 네트워크문제에서도 다수 개의 치명호를 잘 찾을 수 있었으며, 특히 호의 수가 5~18개인 작은 규모(small size)의 최단경로문제에서는 제안한 3종류의 알고리즘 모두에서 100%

정확도로 최적해를 찾을 수 있었다. DIMACS의 표준문제 생성기에서 생성된 20개의 문제에 대하여 본 연구에서 제안한 알고리즘을 적용한 결과는 ECOGA가 가장 우수하였고, 다음으로 SGA, 그리

고 SSGA의 순으로 나타났다.

향후 제한한 유전알고리즘의 정확한 성능평가를 위해서 최단경로문제에서  $k$ 개의 치명호를 구하는 최적해법이 추후로 연구될 필요가 있으며, 네트워크 문제에 대한 개체 표현 방법 등이 좀 더 연구될 필요가 있다.

## 참 고 문 헌

- [1] 김여근, 윤복식, 이상복, 「메타휴리스틱 - 유전알고리즘, 시뮬레이티드 어닐링, 타부서치」, 초판, 영지문화사, 1997.
- [2] 안재근, 정호연, 박순달, “최단경로문제에서  $k$ 개의 치명호를 찾는 방법”, 「한국경영과학회지」 제23권, 제4호(1998), pp.11-20.
- [3] 정호연, “최대유통문제에서  $k$ -MVA를 결정하는 방법”, 「한국군사운영분석학회지」, 제25권, 제2호(1999), pp.106-116.
- [4] Ahuja R.J., T.L. Magnanti, J.B. Orin, *Network Flows : Theory, Algorithms, and Applications*, Prentice-Hall, 1992.
- [5] Ball M.O., B.L. Golden and R.V. Vohra, “Finding the Most Vital Arcs in a Networks,” *Operations Research Letters*, Vol.8 (1989), pp.73-76.
- [6] Bar-Noy A., S. Khuller, B. Schieber, “The Complexity of Finding Most Vital Arcs and Nodes,” *Univ. of Maryland Technical Reports*, CS-TR-3539 (1995), pp.1-7.
- [7] Corley JR. H.W., D.Y. Sha, “Most Vital Links and Nodes in Weighted Networks,” *Operations Research Letters*, Vol.1 No.4 (1982), pp.157-160.
- [8] DIMACS, <http://dimacs.rutgers.edu>.
- [9] Goldberg A.V., <http://www.neci.nj.nec.com/homepages/avg>.
- [10] Lubore S.H., H.D. Ratliff, G.T. Sicilia, “Determining the Most Vital Link in a Flow Network,” *Naval Research Logistic Quarterly*, Vol.18, No.4 (1971), pp.497-502.
- [11] Malik K., A.K. Mittal, S.K. Gupta, “The  $k$  Most Vital Arcs in the Shortest Path Problem,” *Operations Research Letters*, Vol.8 (1989), pp.223-227.
- [12] Munakata and D.J. Hashier, “A Genetic Algorithm Applied to the Maximum Flow Problem,” *ICGA* (1993), pp.488-493.
- [13] Ratliff H.D., S.H. Lubore, G.T. Sicilia, “Finding the  $n$  Most Vital Links in a Flow Network,” *Management Science*, Vol.21, No.5 (1975), pp.531-539.