

최단경로문제에서 k개의 치명호를 결정하는 유전알고리즘* (An Evolutionary Algorithm for Determining the k Most Vital Arcs in Shortest Path Problem)

정 호 연**

Abstract

The purpose of this study is to present a method for determining the k most vital arcs in shortest path problem using an evolutionary algorithm. The problem of finding the k most vital arcs in shortest path problem is to find a set of k arcs whose simultaneous removal from the network causes the greatest increase in the total length of shortest path. Generally, the problem determining the k most vital arcs in shortest path problem has known as NP-hard. Therefore, in order to deal with the problem of real world the heuristic algorithm is needed. In this study we propose to the method of finding the k most vital arcs in shortest path problem using an evolutionary algorithm which known as the most efficient algorithm among heuristics. The method presented in this study is developed using the library of the evolutionary algorithm framework and then the performance of algorithm is analyzed through the computer experiments.

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

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

** 전주대학교 산업공학과

1. 서 론

최단경로문제는 특정한 두 교점 사이의 경로 중에서 가장 짧은 길이의 경로를 찾는 문제이다. 이 문제는 Label setting algorithm이나 Label correcting algorithm을 사용하여 모두 짧은 시간 안에 최적해를 찾을 수 있다[4]. 그러나 이들 해법을 사용하여 최단경로에 관한 정보를 파악하는 것도 중요하지만 네트워크의 관리자 입장에서는 네트워크 상의 어떤 호(arc)에 문제가 발생했을 때 네트워크 전체 성능에 어떠한 영향을 미치는가를 파악하는 것도 매우 중요하다. 예를 들면 적과 대치하고 있는 통신망이나 수송보급로 등과 같은 네트워크에서 공격자의 입장에서는 적의 통신망이나 보급로에 가장 치명적인 손실을 줄 수 있는 통신회선이나 수송로를 알고 싶고, 방어자의 입장에서는 가장 타격이 클 것으로 예상되는 통신회선이나 수송로를 미리 파악하여 그곳에 여분의 통신회선을 확보하거나 경계를 강화하는 것과 같은 정보가 필요할 것이다[8,9,10,12]. 이와 같이 최단경로문제에서 최단경로에 가장 치명적인 영향을 미치는 호를 최단경로문제의 치명호(most vital arc ; MVA)라고 한다[2,5,6,9,10]. 최단경로문제에서 k 개의 치명호를 찾는 문제는 최단경로문제에서 k 개의 호의 제거로 시발점에서 종착점까지의 최단거리를 가장 크게 증가시키는 k 개의 호집합을 찾는 문제이다[2,6,9]. 이에 대한 연구는 Corley & Sha[6], Malik et.al.[9], Bar-Noy et.al.[5], 안재근 등[2]에 의해서 수행되었다. Corley & Sha[6]는 k 번째 최단경로(k -shortest path)를 구하는 해법을 이용하여, 최단경로문제에서 1개의 MVA(1-MVA)를 찾는 해법과, k 개의

MVA 찾는 문제(k -MVA)에 대한 충분조건을 제시하였다. Malik et.al. [9]은 Corley & Sha의 방법을 개선하여 무방향 네트워크에서 1-MVA를 찾는 해법을 제시하였고, k -MVA에 대해서는 k 개의 MVA를 찾는 비다항 시간의 복잡도(non-polynomial time complexity)를 가지는 해법을 제시하였다. 그러나 이 문제는 Bar-Noy et.al. [5]에 의해 반례가 제시됨으로써 해법에 오류가 있음이 판명됨과 동시에 k 개의 치명호를 찾는 문제가 NP-hard의 복잡도임이 증명되었다[2]. 따라서 현재 최단경로문제에서 k 개의 치명호를 찾는 문제에 대한 해법은 Corley & Sha가 제시한 충분조건과 이 문제가 NP-hard 라는 사실만이 알려져 있다. 안재근[2] 등은 호와 경로와의 관계를 이용한 k -치명호에 대한 수리모형식과 이에 대한 해법을 제시한 바 있으나 실용성에 다소 문제점을 안고 있다. 이 문제는 그러나 Bar-Noy[5] 등에 의해 NP-hard의 복잡도 임이 밝혀졌기 때문에 최적화 알고리즘이 존재하더라도 문제의 크기가 커짐에 따라 컴퓨터 용량과 계산시간의 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 알고리즘을 개발하고, 실험을 통해 알고리즘의 성능을 분석해 보고자 한다.

2. k -MVA 결정을 위한 유전 알고리즘

최단경로문제에서 가장 치명적인 영향을 미치는

k 개의 치명호(MVA)를 찾기 위해서는 주어진 네트워크에서 호의 수가 m 일 때 $\binom{m}{k} = \frac{m!}{k!(m-k)!}$ 번의 최단경로문제를 풀어야 해결할 수 있다. 그러나 m 의 수가 클 경우에 이러한 계산횟수를 수행한다는 것은 현실적으로 불가능하기 때문에 그 동안 학자들은 효율적으로 치명호를 결정하는 방법에 관하여 연구를 수행하여 왔다. 그러나 현재까지 이에 대한 명확한 해법이 제시되어 있지 못하기 때문에 본 연구에서는 Holland가 제안했던 유전알고리즘을 사용하여 최단경로문제에서 k 개의 치명호를 찾는 방법을 제시한다.

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 고려하였다. 첫 번째에 고려한 마디를 사용하여 호를 표현하는 방법은 표현방법이 쉽고 해석이 용이한 장점이 있지만 유전연산시에 비가능호가 발생할 수 있기 때문에 추가적인 보수(repair)방법이 요구되는 문제점이 있다. 반면에 호에 일련번호를 부여하여 호를 표현하는 방법

은 비가능호가 발생되지는 않지만 중복호가 발생할 수 있기 때문에 이에 대한 문제점만 보완된다면 호에 일련번호를 부여하여 호를 표현하는 방법이 좀 더 수월한 방법이라 할 수 있다. 본 연구에서는 호에 일련번호를 부여하여 호를 표현하는 방법을 사용한다. 이 표현을 사용하면 하나의 개체는 치명호에 해당하는 k 개의 호로 구성되는 호 인자들으로써 개체가 표현될 수 있으며, 그렇게 함으로써 k -MVA에 대한 잠재해를 표현할 수 있게 된다.

초기 모집단의 크기(population size)는 임의로 생성된 k 개의 호 집합으로 구성되는 개체 10~500개의 범위 내에서 적절히 선택할 수 있게 하였다.

2.2 적응도 평가 및 계산

본 연구에서 정의한 개체는 치명호에 해당하는 k 개의 호로 구성된다. 이에 대한 적응도는 개체를 구성하는 k 개의 호를 제거한 다음 최단경로의 길이를 구한 값과 원문제에서의 최단경로의 길이의 차로써 계산된다. 이를 수식으로 표현하기 위해 먼저 용어를 정의하면 다음과 같다.

- $G=(N, A)$: $|M|=n$ 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.

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

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 사용한 다.

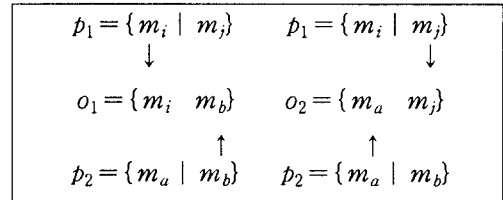
2.3 선별 방법(selection method)

본 연구에서는 현재 모집단의 우수한 개체를 지속적으로 유지하기 위해 Goldberg에 의해 제안된 우수개체 보호전략(elitism)을 사용한다[7]. 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 선별을 통해 전 세대에서 구한 우수개체보다 더 우수한 개체를 지속적으로 발생하게 함으로써 좋은 해를 발생할 가능성을 높게 된다.

2.4 유전 연산자

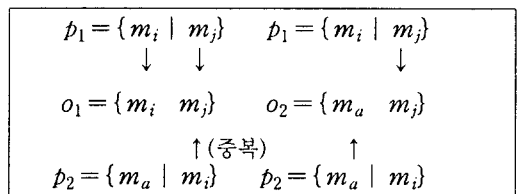
유전 연산자는 교차와 돌연변이로 구성되는데, 본 논문에서는 교차나 돌연변이를 실시할 때 발생할 수 있는 중복을 방지하기 위하여 순서교차(order crossover)를 사용한다. 순서교차는 한 부모로부터 부분적인 순서(인자)를 받아들이고, 다른 부모로부터 상대적인 순서를 유지하여 받아들이는 교차방법

이다[1]. 본 연구에서는 상대적인 순서를 받아들이는 개념을 수정하여 중복이 발생될 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. $k=2$ 인 경우 순서교차를 수행한 예가 다음 그림에 나타나 있다. [그림 1]을 보면 부모 p_1 과 p_2 로 부터 자손 o_1 과 o_2 가 하나의 절단점을 기준으로 하여 각 인자를 상속받아 발생되고 있음을 알 수 있다.



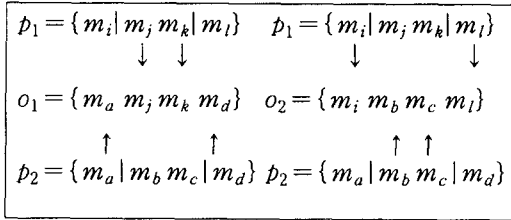
[그림 1] 순서교차 방법 (중복 미 발생 경우)

그러나 [그림 2]와 같이 부모 p_1 과 p_2 가 주어졌을 경우에는 부모 p_1 과 p_2 로 부터 각 인자를 상속받아 자손 o_1 과 o_2 가 생성되게 되는데 이 때 자손 $o_1 = \{m_i \mid m_j\}$ 가 되어 중복이 발생되게 된다. 이 때에는 부모 p_2 로 부터 인자 m_i 를 상속받지 않고 대신 부모 p_1 의 인자를 자손 o_1 에 그대로 상속되게 함으로써 중복을 방지한다.



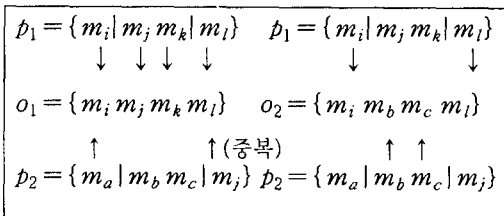
[그림 2] 순서교차 방법 (중복이 발생될 경우)

$k=4$ 인 경우의 순서교차 예는 [그림 3]에 나타나 있다. [그림 3]은 부모 p_1 과 p_2 로부터 절단점 2개가 임의로 지정되면 그에 따라 자손 o_1 과 o_2 가 두개의 절단점을 기준으로 하여 부모 p_1 으로부터 가운데 인자를 상속받고, 부모 p_2 로부터 가운데 인자를 제외한 나머지 인자를 상속받아 발생됨을 보여주고 있다.



[그림 3] 순서교차 방법 (중복 미 발생 경우)

[그림 4]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 4] 순서교차 방법 (중복이 발생될 경우)

돌연변이(mutation)는 개체에 완전히 새로운 유전자를 생성시키는 단계로써 해 공간을 다양하게 탐

색하게 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_p 를 생성한다.

[단계 3] (중복 인자 발견)

(3.1) v_c 중에서 m_j 를 제외한 나머지 인자 중 어느 인자도 m_p 와 같지 않으면 [단계 4]로 간다.

(3.2) 그렇지 않으면 [단계 2]로 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.

이상의 내용을 참고하여 최단경로문제에서 k -MVA를 결정하기 위한 유전 알고리즘을 체계적으로 정리하면 다음과 같다.

◇ k -MVA를 결정하기 위한 유전 알고리즘

[단계1] (초기 모집단 구성)

임의 생성법을 사용하여 초기 모집단 $P(t)$ 를 구성한다.

[단계2] (적응도 평가)

$P(t)$ 에 있는 모든 개체의 적응도를 평가한다.

[단계3] (선별)

$P(t)$ 로 부터 $P(t+1)$ 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.

[단계4] (교차)

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

[단계5] (돌연변이)

모집단을 구성하고 있는 개체 중에서 돌연변이율에 따라 임의의 개체를 돌연변이 개체로 선택하고, 선택된 개체 중에서 임의의 원소를 선택해 중복이 되지 않는 임의로 원소로 돌연변이 시킨다.

[단계6] (종료조건 검사)

종료조건을 만족하면 끝내고, 그렇지 않으면

$t=t+1$ 로 두고 단계2로 간다.

3. 실험 및 분석

최단경로문제에서 k -MVA를 결정하기 위한 유전 알고리즘의 실험은 DIMACS의 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 유전 파라미터 값 설정을 위한 예비실험

제안된 유전 알고리즘으로 좋은 해를 찾기 위해서는 유전 알고리즘에 민감하게 영향을 미치는 유전 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 최적 교차율과 돌연변이율을 탐색하기 위한 예비 실험계획이 나타나 있다.

[표 1] 예비 실험계획

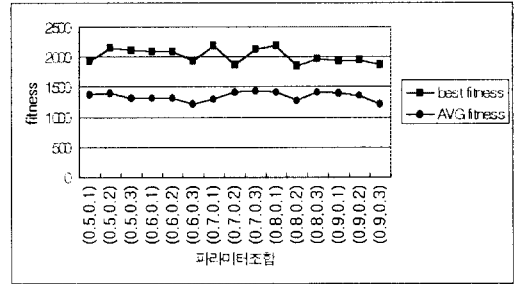
유전파라미터	수 준
모집단의 크기	100
교차율	0.5/0.6/0.7/0.8/0.9
돌연변이율	0.1/0.2/0.3
종료조건	새로 삽입되는 개체가 1000개이상이면 종료

위의 [표 1]에 나타나 있는 예비 실험계획에 따라 5가지의 교차율과 3가지의 돌연변이율에 따른 15가지의 조합에 대하여 각각 20회씩 반복 실험을 수행하였다. 그 결과가 아래 [표 2]에 정리되어 있다.

[표 2] 예비실험 결과

파라미터조합 (교차율/돌연변이율)	best fitness	average fitness
(0.5/0.1)	1915	1374
(0.5/0.2)	2138	1392
(0.5/0.3)	2109	1313
(0.6/0.1)	2080	1319
(0.6/0.2)	2081	1319
(0.6/0.3)	1915	1216
(0.7/0.1)	2190	1283
(0.7/0.2)	1863	1407
(0.7/0.3)	2116	1430
(0.8/0.1)	2180	1401
(0.8/0.2)	1846	1270
(0.8/0.3)	1968	1412
(0.9/0.1)	1924	1390
(0.9/0.2)	1951	1341
(0.9/0.3)	1862	1216

위의 [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. 이를 그림으로 나타내면 다음 [그림 5]와 같다.



[그림 5] 예비실험 결과

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 평균적응도(average fitness) 값이 높은 파라미터의 조합을 선택한 이유는 우연성을 배제하기 위함이며 안정적으로 해를 산출할 수 있는 파라미터의 조합을 구하기 위함이다. 다음 [표 3]은 (교차율/돌연변이율)이 (0.7/0.3), (0.8/0.3), (0.7/0.2)의 조건에 대하여 20회씩 반복실험을 통해 구한 추가 예비실험의 결과 및 그 값들의 평균값을 나타낸다.

[표 3] 추가 예비실험 결과

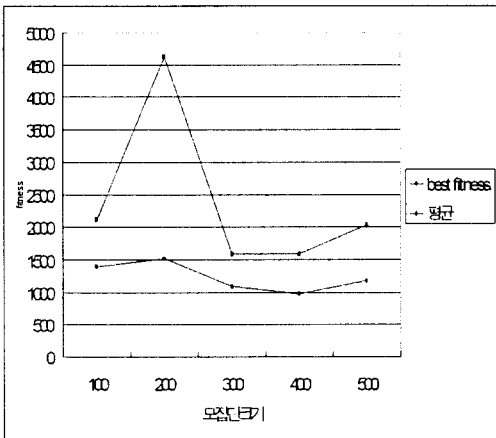
파라미터조합 (교차율/돌연변이율)	추가 예비실험		평균	
	best fitness	average fitness	best fitness	average fitness
(0.7/0.3)	2109	1359	2112	1394
(0.8/0.3)	2138	1328	2053	1370
(0.7/0.2)	1663	1262	1763	1334

위의 [표 3]은 추가 실험결과 (교차율/돌연변이율)이 (0.7/0.3)일 때 가장 좋은 평균 적응도 값을 나타내고 있음을 보여주고 있다. 이는 기존 예비실험의 결과와도 일치되는 것으로써 실험 결과에 대한 재현성을 나타내고 있다.

최적 모집단의 크기는 앞의 예비실험에서 구했던 (교차율/돌연변이율)을 (0.7/0.3)으로 설정한 상태에서 모집단의 크기를 몇 단계로 변경시켜 가면서 20 회씩의 반복실험을 통해 탐색하였다. 모집단의 크기에 따른 평균 적응도 값의 변화가 다음 [표 4]와 [그림 6]에 나타나 있다.

[표 4] 모집단의 크기에 따른 평균 적응도 값의 변화

모집단크기	best fitness	average fitness
100	2116	1394
200	4631	1516
300	1589	1086
400	1589	973
500	2037	1179



[그림 6] 추가 예비실험 결과

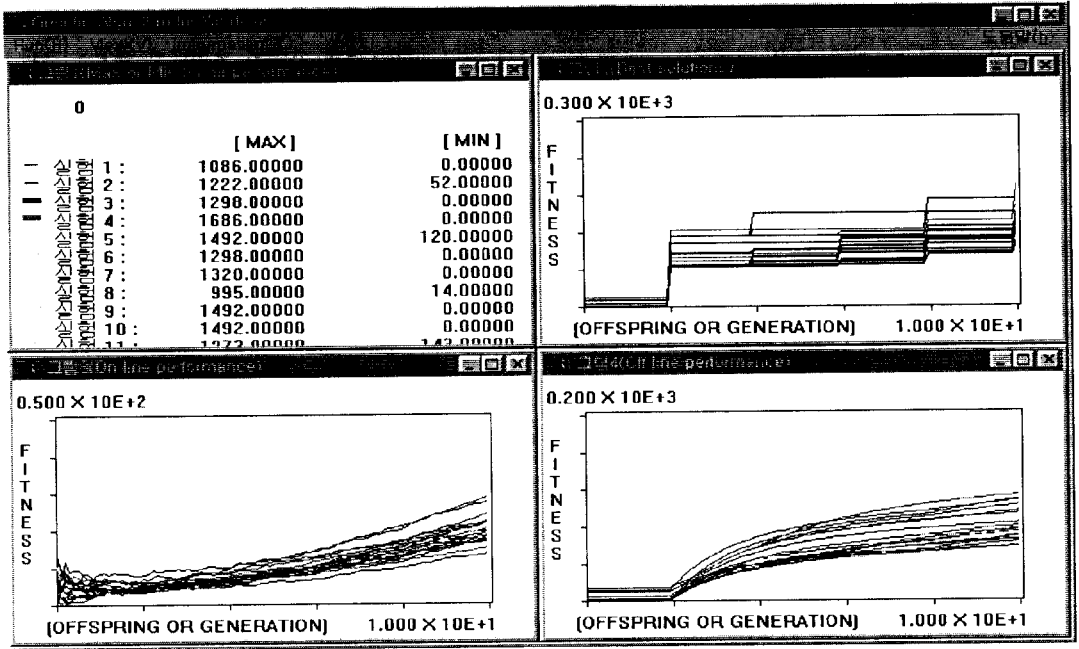
위의 [그림 6]에서 보는 바와 같이 모집단의 크기가 200일 때 평균적응도 값이 가장 높게 나타났다.

따라서 예비실험 결과 가장 좋은 알고리즘의 성능을 보인 최적 파라미터의 조합은 모집단의 크기는 200, 교차율은 0.7, 그리고 돌연변이율은 0.3으로 최종 결정되었다. 이러한 유전 파라미터의 값으로 spagrid3 문제에 대한 실험 결과 3-MVA는 (1317,1365,1320)로 결정되었다. 이 문제에 대한 프로그램의 실행 화면이 다음 [그림 7]에 나타나 있다.

[그림 7]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 변화를 나타내 주고, 오른편 하단의 그림은 모집단에 삽입되는 개체들에 대하여 10개체 단위별로 Best Fitness의 평균값의 변화를 보여주고 있다.

3.2 실험결과 및 분석

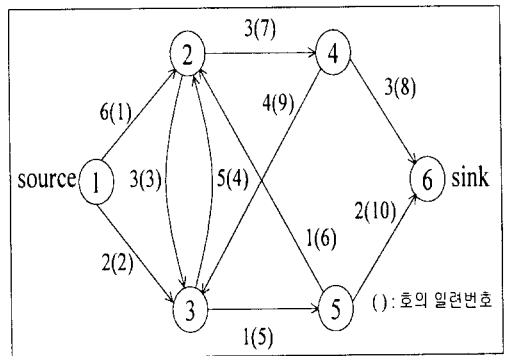
최적 유전 파라미터값(모집단의 크기=200, 교차율=0.7, 돌연변이율=0.3)을 사용하여 DIMACS의 표준문제 생성기에서 생성된 문제를 대상으로 실험하기 앞서 최적해를 알고 있는 10개의 Test문제에 대하여 본 연구에서 제안한 알고리즘의 성능을 평가하였다. 10개의 Test문제는 호의 수가 5~18개인 작은 규모(small size)의 최단경로문제로 구성되었다. [그림 8]은 마디 수가 6개이고, 호의 수가 10개인 최단경로문제(Test 2)를 나타낸다. 이 문제에 대하여 제안한 알고리즘을 적용한 결과 최적해



[그림 7] 프로그램 실행화면
 (3-MVA)는 (6,10,8)외에 10개의 복수 최적해로 계 실험 결과 10개의 Test 문제 모두에서 최적해를 찾
 산되었다.([표 5]의 Test 2 참조).10개의 Test 문제 아내 제안한 알고리즘의 정확도가 매우 우수함을 확
 대한 실험결과가 다음 [표 5]에 나타나 있다. 인할 수 있었다.

[표 5] 작은 규모의 문제 적용결과

문제명 (마디수, 호의 수)	best fitness	3- MVA	최적해 여부
Test1 (5,7)	∞	{7,4,6},{3,6,1},{3,7,6},{2,1,3},{6,1,7},{3,1,6},{5,6,7},{7,6,2},{3,4,2}	최적해
Test2 (6,10)	∞	{6,10,8},{10,7,8},{2,8,10},{5,1,2},{6,8,5},{3,8,5},{8,10,4},{5,8,4},{10,6,7},{5,7,10},{2,7,3}	최적해
Test3 (7,10)	∞	{2,1,9},{9,5,1},{10,1,9},{10,3,9},{4,2,3},{10,9,6},{4,5,9},{9,5,10},{9,3,8},{10,9,2},{4,9,10},{9,10,4},{8,9,5},{9,7,8},{5,6,1}	최적해
Test4 (8,13)	∞	{7,11,1},{7,1,4},{6,9,7},{7,6,2},{7,6,10},{3,1,7},{11,7,10},{1,7,12},{7,5,1},{2,7,1},{6,2,7},{13,7,1},{1,7,10},{3,6,7},{5,6,7}	최적해
Test5 (10,18)	∞	{3,18,17},{18,9,17},{2,3,1},{18,17,15},{1,17,18},{17,10,18},{17,7,18},{18,4,17},{17,8,18},{2,18,17},{14,18,17}	최적해



[그림 8] Test 2 문제

앞 절의 예비실험을 통해서 구했던 최적 유전 파라미터값(모집단의 크기=200, 교차율=0.7, 돌연변이율=0.3)을 사용하여 DIMACS의 표준문제 생성기에서 생성된 20개의 문제에 대하여 실험을 실시한 결과는 다음 [표 6]에 정리되어 있다.

[표 6] 문제적용결과

적용문제 (마디수, 호의수)	best fitness	average fitness	3- MVA
Spagrid3_1980	4631	1516	{1317,1365,1320}
Spagrid4_3864	1882	1106	{362,2197,3397}
Spagrid5_3105	2869	1860	{2650,2172,2808}
Spagrid6_4800	3445	2073	{5,2124,3359}
Spagrid7_5100	1556	949	{1156,2349,3045}
Spagrid10_6000	9935	4935	{436,907,4109}
Sprand3_2000	2746	995	{1510,1354,1799},{1510,1469,308},{1731,25,1510}
Sprand4_3000	7357	3494	{1969,1249,1711},{742,1711,298},{1711,1110,2544}, {727,1711,1577},{1711,1640,2836}
Sprand5_5000	3958	1900	{222,997,2802}
Sprand6_5003	2142	749	{3755,2914,1671},{3080,1,4950},{1223,4719,3605},{3755,906,3086},{2 543,4537,1223},{1582,37,1223},{1547,3982,3755}
Sprand7_6000	812	365	{189,5844,169},{5844,388,2663},{2220,5844,4596}, {5614,5057,5844},{4043,5726,5294},{1901,5175,1385}, {1480,5248,5711},{5844,4814,1891},{1480,2326,3258}
Sprand8_7645	6082	2619	{5357,7234,5974},{759,4412,1227},{1951,5765,4412} {142,5244,201},{4412,4409,5798},{4101,2045,4412}
Sprand9_3837	14835	6171	{1042,3694,149},{1660,3694,533},{865,3694,3510}
Sprand10_8999	3904	2277	{7096,1439,8988},{1293,2675,6791},{4303,4160,8988}
Sprand11_10000	3531	1412	{3355,5974,3918},{4266,1508,3910},{7493,2953,7432}, {7432,3457,7528},{4284,6363,1577},{7432,9865,6056}, {5252,8485,5974},{2335,4266,3221}
Spacyc5_2450	12567	4307	{1,633,1404}
Spacyc6_3050	5799	2899	{2398,1827,2184},{2885,1315,459},{2184,1405,2034}, {459,1411,36},{2184,188,2646},{1507,373,459},{3020,2535,823}, {2301,1768,459},{290,918,2184},{2184,625,1018}
Spacyc7_7025	2359	429	{2061,5180,3720}
Spacyc8_14364	7135	356	{4676,1076,6393}
Spacyc10_20140	2232	223	{4169,6574,1598},{1299,19032,6574}

4. 결 론

본 논문은 유전알고리즘을 사용하여 최단경로문제에서 k -MVA를 결정하는 방법을 다루고 있다. 최단경로문제에서 k -MVA를 결정하는 문제는 조합 최적화 문제로서 문제의 크기가 커지면 시간과 기억 용량이 지수적으로 증가하는 NP-hard 문제에 속한다. 이 문제에 대한 해법은 현재까지 알려져 있지 않으며, 최근에 제시된 방법도 현실적으로 실용화되

기 어려운 단점을 갖고 있다. 그러나 본 논문에서 제시한 해법은 여러 문제에 적용해 본 결과 호의 수가 20,000개 이상인 대형네트워크문제에서도 3-MVA를 잘 찾을 수 있었다.

대형문제에 대한 최적해가 정확히 알려져 있지 않기 때문에 제한한 알고리즘의 성능을 정확히 평가하기는 쉽지 않지만, 최적해를 알 수 있는 작은 규모의 문제에 대하여 본 연구에서 제안한 방법을 적용한 결과 100%의 해의 정확도를 보여 양질의 좋은 해를 구할 수 있음을 확인할 수 있었다.

향후 제안한 유전알고리즘의 정확성이나 성능평가를 위해서 최단경로문제에서 k 개의 치명호를 구하는 최적해법이나 발전적 해법이 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, 한국경영과학회, (1998.12), pp.11-20
- [3] 정호연, "최대유통문제에서 k -MVA를 결정하는 방법", 한국군사운영분석학회지, 제25권 제2호, 한국군사운영분석학회, (1999.12), pp.106-116
- [4] Ahuja.R.J., Magnanti.T.L., Orin.J.B., Network Flows : Theory, Algorithms, and Applications, Prentice-Hall, 1992,
- [5] 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.
- [6] 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.
- [7] Goldberg.A.V.,[http:// www.neci.nj.nec.com / homepages / avg](http://www.neci.nj.nec.com/homepages/avg)
- [8] 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
- [9] 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.
- [10] M.O. Ball, B.L. Golden, and R.V. Vohra. "Finding the Most Vital Arcs in a Networks", Operations Research Letters, Vol.8 (1989), pp 73-76.
- [11] Munakata and Hashier.D.J., "A Genetic Algorithm Applied to the Maximum Flow Problem", ICGA (1993), pp 488-493
- [12] 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