
스타이너 트리를 구하기 위한 부동소수점 표현을 이용한 유전자 알고리즘

김채주* · 성길영** · 우종호***

Genetic Algorithm Using-Floating Point Representation for Steiner Tree

Chae-joo Kim* · Kil-Young Sung** · Chong-ho Woo***

요 약

주어진 네트워크에서 최적의 스타이너 트리를 구하는 문제는 NP-hard이며, 최적에 가까운 스타이너 트리를 구하기 위하여 유전자 알고리즘을 이용한다. 본 논문에서는 이 문제를 해결하기 위하여 유전자 알고리즘에서 염색체를 기존의 이진스트링 대신 부동소수점으로 표현하였다. 먼저 주어진 네트워크에 Prim의 알고리즘을 적용하여 스패닝 트리를 구하고, 부동소수점 표현을 갖는 유전자 알고리즘을 사용하여 새로운 스타이너 점을 트리에 추가하는 과정을 반복함으로써 최적에 가까운 스타이너 트리를 구했다. 이 방법을 사용하면 이진스트링을 사용하는 기존의 방법에 비해서 트리가 보다 빠르고 정확하게 최적에 가까운 스타이너 트리에 접근했다.

ABSTRACT

The genetic algorithms have been used to take a near optimal solution because The generation of the optimal Steiner tree from a given network is NP-hard problem,. The chromosomes in genetic algorithm are represented with the floating point representation instead of the existing binary string for solving this problem. A spanning tree was obtained from a given network using Prim's algorithm. Then, the new Steiner point was computed using genetic algorithm with the chromosomes in the floating point representation, and it was added to the tree for approaching the result. After repeating these evolving steps, the near optimal Steiner tree was obtained. Using this method, the tree is quickly and exactly approached to the near optimal Steiner tree compared with the existing genetic algorithms using binary string.

키워드

Steiner tree, NP-hard, Genetic algorithm, Prim's algorithm, Floating point representation

1. 서 론

네트워크의 최단경로 설정에 대한 문제는 오랫동안 연구되고 있는 분야이다. 이에 대한 연구는

최소 스패닝 트리(minimal spanning tree) 방식과 최소 스타이너 트리(minimal steiner tree) 방식을 중심으로 이루어졌다. Pawel Winter 등은 스타이너 트리 문제의 기하학적 특성을 이용하여 직선거

*부경대학교 대학원 컴퓨터공학과

**부경대학교 전자컴퓨터정보통신공학부

**경상대학교 정보통신공학과, 해양산업연구소, 교신저자,

접수일자 : 2004. 5. 15

리 문제의 최적해를 구하는 해법들을 제시하였다. 그러나 이 연구들에서는 주어진 계산 시간내에 무작위로 발생하는 해를 풀지 못하여 노드의 최대 크기를 30개 이하로 제한했다[1-4]. David와 Pulin[5]은 유전자 알고리즘과 수학적 프로그래밍을 이용해서 스타이너 점을 찾는 방법을 제안했다. Barreiros[6]는 스타이너 트리의 계산 최적화를 위해 계층구조 유전자 알고리즘을 제안했다. 하위단계는 스타이너 트리 문제를 해결하기 위해 각 해를 스타이너 점들에 대응하도록 표현하며, 상위단계는 노드집합을 분할하여 표현하였다. 이 분할은 알고리즘을 수행하는 동안 동적으로 만들어진다. 이러한 방법은 좋은 결과와 효율적인 해를 구하는 대신에 극단적으로 민감한 변화에도 해의 값이 바뀔 수 있다. Z. Michalewicz[7]는 이진염색체를 실수 표현에 적용할 경우 발생할 수 있는 문제점에 대하여 기술하였다. 일반적으로 이진염색체의 표현은 실수 값을 이진 벡터로 변환하여 염색체에 표현하는데, 고정밀도를 요구하는 수치문제에서 이진염색체를 사용하여 실수값을 표현할 경우 이진염색체의 길이가 길어져 유전자 알고리즘이 잘 동작하지 않을 수 있다. 따라서 수치최적화를 위한 유전자 알고리즘에서는 부동소수점의 벡터로 염색체를 표현한다. Jianhua Jiang[8]은 수치최적화 문제와 관련하여 세대수를 5개 사용한 micro-GA(μ GA)에서 이진염색체와 실수염색체를 비교 및 실험하였다. 이 문제에서 실수염색체가 이진염색체보다 수치최적화에 효과적임을 확인하였다. 또한 실수염색체에 사용되는 산술교차 연산, 휴리스틱 연산, 그리고 둘을 조합한 연산에 대한 비교에서는 산술교차연산의 목적값이 적은 세대에서 빨리 수렴하여 가장 좋은 해를 구하는 것으로 나타났다.

본 연구에서는 최소경로에 대한 거리의 정밀도를 높이고 거리기준 응용에 주로 사용하며 실수 표현을 적용할 수 있도록 유전자 알고리즘을 구성하였다. 이 알고리즘은 최단거리 문제에 널리 이용하는 Prim 알고리즘으로 최소 스패닝 트리를 구성하고 스타이너 트리의 형태를 유지하기 위한 염색체를 실수형태로 표현한다. 따라서 최적해 또는 최적해에 근접한 근사해를 구하여 유전자 알고리즘을 노드간 거리에 따라 최소 스패닝 트리와 비교하여 최소거리값을 평가하였다.

II. 스타이너 트리

2.1 스타이너 트리 문제

스타이너 트리는 삼각형에서 거리의 합이 최소

가 되게 하는 단일점을 찾는 문제이다. 이 단일점을 스타이너 점(steiner point)이라고 부른다. Ronald Graham 등은 스타이너 트리의 길이를 계산하는 문제가 NP-hard임을 증명하였다[9]. 따라서 이 문제의 해를 구하기 위한 최적화 알고리즘을 찾아내는 것이 어렵고, 최적화 알고리즘이 존재해도 문제의 크기가 증가함에 따라 전체 최적점을 찾는 것은 높은 계산복잡도와 공간복잡도를 요구하여 실용성이 없게 된다.

2.2 스타이너 트리의 구성방법

스타이너 트리 구성에 대한 방안으로 Prim 알고리즘을 이용하여 스타이너 트리와 유사하도록 트리를 구성한다. 주어진 점 $Q_1 \dots Q_m$ 에 적당한 개수의 점 $P_1 \dots P_n$ 을 추가하여 최소 스패닝 트리를 만든다. 점 $P_1 \dots P_n$ 을 조금씩 움직여 최소 스패닝 트리의 길이를 줄여가는 방법으로 스타이너 트리 형태를 만들어 간다. 이렇게 구성된 스타이너 트리를 개체로 표현하고 유전자 알고리즘을 이용하여 개체를 진화시킨다. 즉, 각 세대의 개체들은 스타이너 트리가 되고 유전자 알고리즘을 적용하여 세대를 반복할수록 이전 세대의 것보다 우수한 개체를 형성해 나가는 방법을 이용한다. 이 과정을 효율적으로 수행하기 위해 스타이너 트리의 표현을 트리구조로 프로그래밍하여 알고리즘의 구현을 효율적으로 수행할 수 있도록 한다.

III. 유전자 알고리즘에서 염색체의 표현

본 장에서는 유전자 알고리즘을 이용하여 노드 상 즉, 통신망의 연결이나 설비배치 문제 등의 최단거리를 구하기 위한 과정과 방법에 대하여 기술한다. 알고리즘에 입력한 자료를 대상으로 공간상의 좌표값을 직접 실수로 표현하여 염색체로 나타내었으며, 이 염색체는 Prim 알고리즘을 사용하여 트리구조로 표현하고 부동소수점 표현의 산술교차 연산을 사용한 유전자 알고리즘을 이용하여 최단 거리를 구한다.

3.1 고전적인 이진염색체의 표현

지금까지의 유전자 알고리즘은 비트열을 중심으로 한 이진표현에 의존하고 있다. 이와 같은 일차원 표현은 해석하기 쉬운 이점이 있지만, 실수로 표현한 노드를 다루기 위해 그림 1과 같이 이진수를 정수로 변환하고 정수를 다시 실수로 변환해야 하는 코드화 및 디코드화의 기술을 필요로 한다[7].

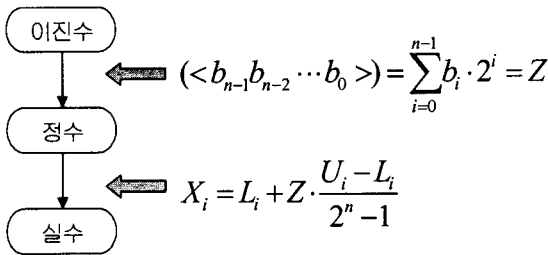


그림 1. 이진수를 실수로 변환

Fig 1. Conversion of binary number to real number

3.2 탐색체의 부동소수점 표현

하나의 실수를 한 유전자로 택하여 인자들의 속성이 실수이면 그림 2와 같이 유전자들과 일대일로 대응될 수 있게 하는 실수연산을 사용한다. 이러한 경우 동일한 위치에 있는 두 부모의 유전자 값을 평균해서 자손의 동일한 위치 유전자 값으로 결정하는 산술교차 연산을 사용한다.

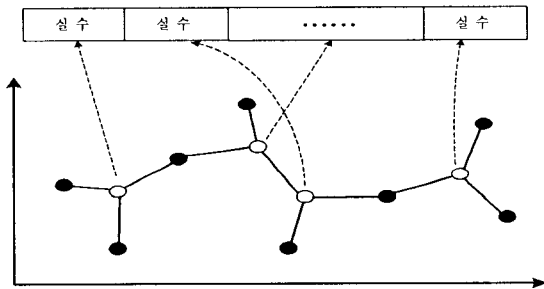


그림 2. 실수와 유전자의 일대일 대응

Fig 2. A match real number to gene by one to one

3.3 유전자 알고리즘의 적용

유전자 알고리즘의 적용은 노드 정보를 위한 기본 데이터를 무작위로 입력하고, 이 데이터를 기준으로 초기 개체군에 대한 하나의 실수를 부호화함으로써 새로운 스타이너 점을 생성하게 된다. 새로운 스타이너 점의 생성은 Prim 알고리즘을 이용하여 최소 스패닝 트리를 순환하도록 하며, 이렇게 생성된 초기 개체군은 각 트리에서 선분길이에 대한 가중치로 계산한다. 구현할 때에는 실험을 쉽게 하기 위하여 사용자가 임의로 노드를 입력하여 트리구조를 생성할 수 있도록 설계하여 실행하였다.

본 연구에 사용한 유전자 알고리즘의 적용 예는 그림 3의 실행과정을 통하여 나타내고, 각 절차를 제시하였다.

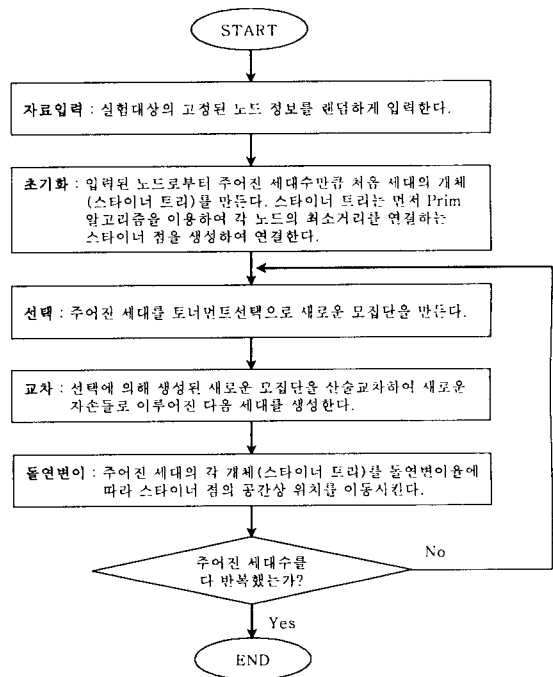


그림 3. 구현한 프로그램의 실행과정

Fig 3. Execution process of program

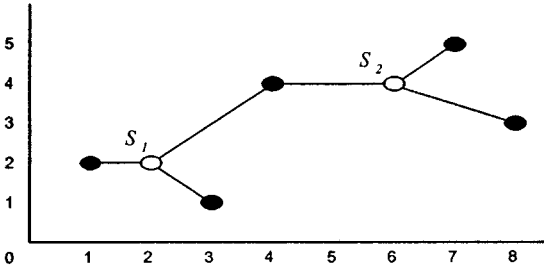
3.3.1 부호화와 초기화

입력한 데이터를 대상으로 실수로 표현한 공간상의 좌표값을 탐색체로 구성하였다. 실수를 직접 탐색체에 표현하기 위하여 유전자 하나에 하나의 실수값을 직접 표현하는 방법을 이용하여 탐색체를 선택한다. 이 방법은 실수를 이진수로 표현하거나 이진수를 실수로 변환하는 과정이 필요 없기 때문에 값을 변환하기 위한 오버헤드가 없는 장점이 있다. 탐색체의 유전자와 해 벡터의 요소는 일대일로 대응하게 하였으며 탐색체의 구성은 스타이너 점의 좌표를 하나씩 지정하도록 하였다.

본 실험은 임의로 데이터를 입력하여 랜덤하게 만들어진 최소 스패닝 트리를 순환하게 함으로써 스타이너 트리 형태를 만들어 간다. 한 개체에서 스타이너 트리 형태를 만드는 방법은 다음과 같다. 먼저, 입력된 노드 정보로부터 주어진 개체군의 크기만큼 모든 노드에 이르는 최소 스패닝 트리를 랜덤하게 만든다. 스타이너 점들의 위치를 유전자 알고리즘의 개체로 표현하고, 최소 스패닝 트리에 의해 적합도를 판정하도록 하였다. 다음으로 최소 스패닝 트리 상에 존재하는 인접한 두 쌍의 기본점들의 선분들에 새로운 점을 추가하여 얻은 최소 스패닝 트리를 순환함으로써 스타이너 트리 형태를 유지하게 하였다.

3.3.2 토너먼트선택 연산

주어진 한 염색체의 가중치는 스타이너 점에 연결되어 있는 선분길이의 합으로 계산한다. 선택되는 가중치는 스타이너 점과 각 노드들을 연결하기 위한 최적 함수값을 구하기 위해 사용한다. 그림 4는 트리에서 계산한 가중치의 간단한 예를 보여준다.



S1의 가중치 = 5.24, S2의 가중치 = 5.65
 그림 4. 트리의 가중치
 Fig 4. Weight of tree

최단경로는 최소거리를 찾는 문제이므로 가중치가 작을수록 적합함수(fitness function)값은 커야 한다. 이 때 적합함수값이 큰 염색체가 작은 염색체보다 선택될 확률이 높아지게 되며, 이렇게 구한 적합함수값을 이용하여 선택연산을 수행한다. 본 연구에서는 선택연산의 수행시간을 고려하여 수행시간이 짧고 선택압(selection pressure)으로 교차연산의 질을 조절할 수 있는 토너먼트선택을 이용하였다.

3.3.3 산술교차 연산

교차연산은 염색체가 실수표현으로 부호화되어 있고 자손염색체 값의 범위를 두 부모염색체가 표현하는 점으로 구축되는 공간상에 존재하게 제한할 필요가 있다. 따라서 일반적으로 실수표현에 사용하고 교차율로 범위를 한정지을 수 있는 교차연산이 필요하다. 본 연구에서는 두 부모 P1, P2의 선형조합에 의해 두 자손 O1과 O2를 생산하는데 널리 사용되는 산술교차법을 이용한다. 그 과정은 다음과 같다.

- 단계1: 교차할 두 부모를 임의로 선택한다.
- 단계2: 각 부모에서 유전자를 교환할 트리를 임의로 선택한다.
- 단계3: 선택한 각각의 트리에서 선분의 위치를 근거로 교환하여 자손들을 구성한다.
- 단계4: 단계1~3 과정을 교차율×세대수 만큼 반복 수행한다.

즉, 두 부모가 P1, P2 이고, 교차율을 a라고 할

때, 생성되는 자손은 다음과 같다.

$$\begin{aligned}
 P1 &= (P11, P12, \dots, P1m) \\
 P2 &= (P21, P22, \dots, P2n) \\
 O1 &= aP1 + (1 - a)P2 \\
 O2 &= (1 - a)P1 + aP2
 \end{aligned}$$

여기서, 해들이 탐색영역에 계속 머물기 위해서 $a \in [0, 1]$ 을 만족하는 변수를 사용한다.

자손 모집단의 모든 개체에 대하여 0에서 1사이의 랜덤함수 값을 생성시켜 그 값이 산술교차 파라메타 값보다 작은 개체들을 선택하고 개체들에 대하여 산술교차를 수행한다.

3.3.4 돌연변이 연산

돌연변이 연산은 스타이너 점의 공간상 위치를 이동시키기 위해 사용한다. 선택한 개체의 스타이너 점의 좌표를 자신의 좌표값에서 돌연변이율 10% 이하의 변화를 준 좌표와 대체하도록 하였다. 돌연변이 연산을 적용함으로써 보다 다양한 해를 발생시켜 좋은 해의 값은 상승시키고 나쁜 해를 제거하여 지역 최적해에서 벗어날 수 있다. 돌연변이는 해 벡터의 한 좌표값만을 변화시킨다.

그림 5에서와 같이 연결점 중 노드사이의 연결거리를 줄이기 위해 돌연변이 연산이 임의로 S1을 선택하게 되면, S1에 해당하는 스타이너 점의 위치는 돌연변이율과 곱해져서 보다 가까운 스타이너 점인 S2로 이동하게 된다. 이렇게 생성된 자손염색체의 적합도가 부모염색체보다 나은 경우 그 값을 상승시켜서 진화하게 된다.

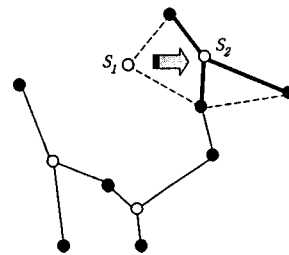


그림 5. 돌연변이 연산 후의 위치이동
 Fig 5. Position transfer after mutation operation

IV. 결과 및 고찰

유전자 알고리즘을 실행하기 위한 교차율과 돌연변이율의 적정한 파라미터를 선택하도록 하였다. 노드간의 길이에 대해 교차율과 돌연변이율을 각각 변경하면서 모의실험을 수행하였다. 교차율

의 변화는 0.1~0.5, 돌연변이율의 변화는 0.05~0.2 사이의 범위에서 10개의 노드에 대해 최소 스패닝 트리의 거리값과 비교하여 유전자 알고리즘의 적정 파라메타를 구하였다. 실험결과로 교차율은 0.3의 범위에서, 돌연변이율은 0.1의 범위에서 좋은 결과가 나왔다.

실험은 구현한 프로그램의 좌표상에 무작위로 점들을 생성하는 방법을 이용하였다. 알고리즘의 성능을 평가하기 위해 유전환경 인자로서 세대수는 100세대, 교차확률 $P_c = 0.3$, 돌연변이 확률 $P_m = 0.1$ 로 하여 수행하였다. 실험에 사용한 시스템은 AMD 1600+, 메모리는 256M이고, 프로그램은 JDK1.4.1을 사용하여 Java Applet으로 작성하였다.

그림 6은 Prim 알고리즘을 이용해 우리나라 100개 도시들을 서로 연결하여 최소거리를 구한 후, 유전자 알고리즘을 반복 적용하여 최적의 스타이너 트리에 근접한 결과를 나타낸 것이다.

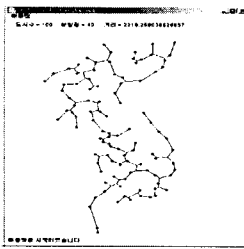


그림 6. 100개의 도시를 연결한 근접 최적 스타이너 트리

Fig 6. The near optimal Steiner tree connected 100 cities in Korea

그림 7은 그림 6에서와 같은 방법으로 최소 스패닝 트리와 본 논문의 유전자 알고리즘을 100개의 노드에서 수렴한 결과를 나타내었다.

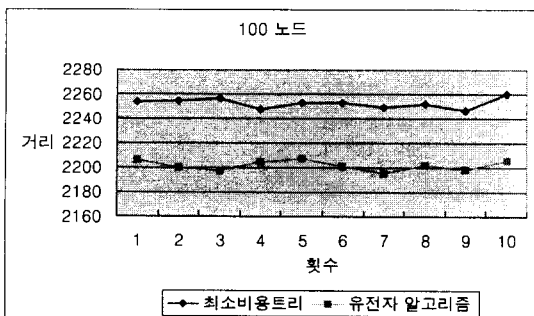


그림 7. 유전자 알고리즘의 구현 결과

Fig 7. Implementation results of Genetic Algorithm

프로그램의 특성상 입력의 정확성이 떨어지는 경우가 나타날 수도 있으므로, 실험을 각각 10회 실시하여 거리값을 비교하였다. 최소 스패닝 트리로 연결한 노드의 선분들보다 유전자 알고리즘으로 연결한 선분들의 거리값이 평균 2~3%정도 절약되었다.

표 1과 그림 8은 각 노드 수에 따른 최소거리를 본 논문의 유전자 알고리즘과 최소 스패닝 트리를 비교 및 실험하여 연결한 각 노드 수에 대한 거리비(%)를 나타내었다. 본 논문의 유전자 알고리즘을 이용하여 각 노드를 연결하면 최소 스패닝 트리를 이용한 것보다 2~4.3%의 거리단축 효과를 기대할 수 있었다.

표 1. 연결 노드 수에 따른 결과
Table 1. Result by connecting node number

노드수	최소비용 트리	본 논문의 유전자 알고리즘	거리절감율 (%)
10	920.63	888.22	3.52
20	1117.89	1095.10	2.04
30	1325.91	1280.84	3.40
40	1513.59	1448.80	4.29
50	1708.47	1640.75	3.97
60	1805.29	1744.16	3.39
70	1958.97	1890.03	3.52
80	2085.60	2021.35	3.09
90	2170.36	2106.95	2.93
100	2252.39	2201.82	2.32

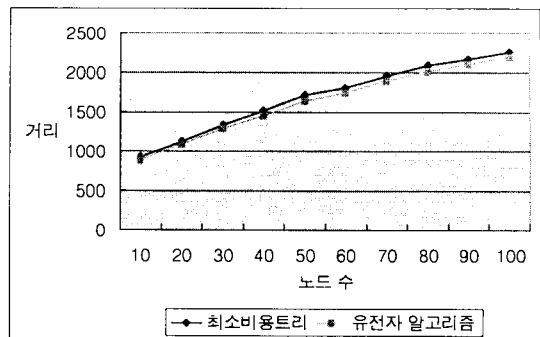


그림 8. 각 노드별 실행결과
Fig 8. Execution result from each node

표 2는 Joseph[10]가 제안한 스타이너 최소트리 문제를 위한 유전자 알고리즘과 본 논문의 유전자 알고리즘을 비교한 것이다. 실험방법은 최소 스패닝

트리의 거리를 100%로 하여 비교 및 분석하였다.

표 2. 최소거리의 비교
Table 2. Comparison of minimum distance

	16개의 Grid 노드	100개의 랜덤노드
Joseph의 유전자 알고리즘	95 %	98.8 %
본 논문의 유전자 알고리즘	94.3 %	97.7 %

Joseph의 유전자 알고리즘에서는 트리의 각 노드를 이진스트링 형태로 표현하고, 룰렛 휠 선택방법과 이점교차를 사용하였고, 본 논문에서는 트리의 좌표값을 그대로 이용할 수 있도록 부동소수점 표현방법을 사용하는 산술교차를 이용하였다. 성능비교는 컴퓨터 환경 등이 서로 다른 상태에서 시행되었고, Joseph의 논문에서 제시한 결과를 나타내었다. 본 논문의 유전자 알고리즘은 Joseph의 유전자 알고리즘보다 거리를 0.7~1.1% 정도 줄이는 것으로 나타났다.

V. 결 론

네트워크 망에서 최단거리를 구하기 위한 문제는 스타이너 트리를 이용할 수 있다. 그러나 스타이너 트리는 NP-hard로 알려져 있어 그 해를 구하기 위한 실용성에 대한 문제가 제기되고 있다.

본 연구에서는 Prim의 알고리즘으로 최소 스패닝트리의 선분을 분할하고 트리 형태의 구조를 유지하기 위해 부동소수점 표현에 의한 유전자 알고리즘을 사용하였다. 공간상의 좌표를 트리 형식으로 나타내고, 직접 염색체에 표현하기 위하여 부동소수점 표현법을 선택하여 스타이너 트리에 근접하도록 하였다.

실험은 이진스트링 염색체를 사용한 Joseph의 유전자 알고리즘과 최소거리를 비교하였다. 그 결과, 이진스트링으로 표현된 염색체보다 부동소수점 표현의 유전자 알고리즘이 거리를 0.7~1.1% 정도 단축시킬 수 있었고, 최소 스패닝 트리와 비교하여 노드 사이의 총 거리가 2~4.3% 단축되는 것을 확인하였다. 또한 트리 형식의 염색체 표현과 부동소수점 표현으로 교차하는 산술교차법을 사용한 접근이 최적해에 더 근접함을 알 수 있었다. 따라서 부동소수점 표현을 이용하면 보다 정확한 실

제 네트워크망의 연결거리를 계산하는 데 효율적이다.

본 연구에서는 각 네트워크 망에 대한 최소 거리에 중점을 두고 실험하였다. 향후 연구에서는 최소 거리를 찾는 데 필요한 알고리즘의 연산 시간을 분석하면 요구 조건에 보다 효율적으로 근접하는 최적의 스타이너 트리를 구할 수 있을 것으로 판단된다.

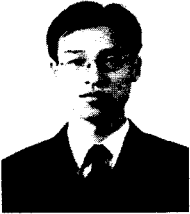
참고문헌

- [1] G.R.Raidl and B.A.Julstrom, "Edge-Sets: An Effective Evolutionary Coding of Spanning Trees", Technishe Univ., Technical Report, 2002.
- [2] Pawel Winter and Martin Zachariasen, "Euclidean Steiner Minimum Trees: An Improved Exact Algorithm", Networks, Vol.30, pp.149-166, 1997.
- [3] Pawel Winter, "An algorithm for the Steiner problem in the Euclidean plan", Networks, Vol.15, pp.323-345, 1985.
- [4] Cockayne and D.E.Hewgill, "Improved Computation of Plane Steiner Topology", Algorithmica, Vol.7, pp.219-229, 1992.
- [5] David J. Thuente and Pulin Sampat, "Mathematical Programming in A Hybrid Genetic Algorithm for Steiner point Problems", Proc.of the ACM symposium, 1995.
- [6] Jorge Barreiros, "An Hierarchic Genetic Algorithm for Comuting (near)Optimal Euclidean Steiner Trees", GECCO2003, pp.56-65, 2003.
- [7] Zbigniew Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Third Extended Edition, Springer-Verlag, 1995.
- [8] Jianhua Jiang, "Rigorous Analysis and Design of Diffractive Optical Elements", A Dissertation, Univ. Alabama, pp.74-89, 2000.
- [9] Richard Courant and Herbert Robbins, What is Mathematics?, Second Edition, Oxford University Press, 1996.
- [10] Joseph Jones, Frederick C and Harris,Jr, "A Genetic Algorithm for the Steiner Minimal Tree Problem", ISCA's Int. Conf. on

Intelligent Systems, 1996.

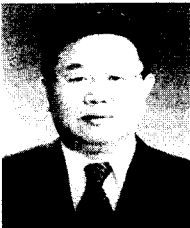
저자소개

김채주(Chae-Joo Kim)



1999년 동서대학교 컴퓨터공학과
공학사
2004년 부경대학교 대학원 컴퓨터공학과
공학석사
※관심분야 : 유전자 알고리즘, P2P통신, Network Security

성길영(Kil-Young Sung)



1980년 경북대학교 전자공학과
공학사
1985년 건국대학교 대학원 전자공학과
공학석사
2000년 부경대학교 대학원 전자공학과
공학박사

1995년~현재 : 경상대학교 정보통신공학과 교수,
해양산업연구소 연구원
※관심분야 : VLSI 어레이, 컴퓨터구조, 영상압축

우종호(Chong-Ho Woo)



1978년 경북대학교 컴퓨터공학과
공학사
1981년 경북대학교 대학원 전자공학과
전산공학전공 공학석사

1990년 경북대학교 대학원 전자공학과 전산공학전공
공학박사
1981년~1996년 부산수산대학교 컴퓨터공학과
교수
1996년~현재 부경대학교 전자컴퓨터정보통신
공학부 교수
1987.8~1988.8 미국 뉴욕, RPI CS, Visiting
professor
2001.8~2002.8 미국 루이지애나, LSU at Baton
Rouge ECE, Visiting professor
※관심분야 : Embedded System, Network
Evaluation, e-Learning System, VLSI Array
Design