

실시간 멀티캐스트 라우팅을 위한 유전자 알고리즘

서용만* · 한치근**

A Genetic Algorithm for Real-Time Multicast Routing

Yong Man Suh* · Chi-Geun Han**

■ Abstract ■

The real-time multicast problem is to construct a multicast tree starting from a source node and including multiple destination nodes and that has minimum network cost with delay constraints. It is known that to find a tree of the minimum network cost is the Steiner Tree problem which is NP-complete. In this paper, we propose a genetic algorithm to solve the multicast problem, in which we want to find a multicast tree with minimum network cost and the delay constraints. The computational results obtained by comparing an existing algorithm, Kompella algorithm, and the proposed algorithm show that our algorithm tends to find lower network cost on the average than Kompella algorithm does.

1. 서 론

최근에 네트워크와 통신기술의 발전과 더불어, 멀티미디어 자료들을 선택된 사람들에게 분배하는 멀티캐스팅 서비스들이 필요하게 되었다. 비디오 브로드캐스팅, 멀티미디어 화상회의, 집단 메일링 등이 멀티캐스팅 서비스의 예이며, 이들은 하나의 시작노드로부터 다수의 목적지노드로 메시지를 전달하는 서비스이다. 메시지를 목적지노드로 전달할 때, 필

요한 전체 네트워크 비용을 최소화하고, 주어진 시간(Delay Constraint)안에 각각의 목적지노드로 메시지 전달을 해야하는 제약조건을 갖고 있다. 즉, 두 가지 제한요소는 홉(Hop)수 또는 실제 거리(Distance)등을 나타내는 비용(Cost)과 전송(Transmission) 또는 교환(Switching) 지연 등을 의미하는 지연(Delay)이다. 멀티캐스트 통신에서 어려운 점은 최소의 네트워크 지연을 가지면서 전체적인 네트워크 비용을 최소화시키는 연결 패스를 결정하는 것

* 포스텍이다

** 경희대학교 전자정보학부

인데, 그것을 위해서 일반적으로 라우팅 트리를 구성하는 방법이 사용된다. 라우팅 트리에 포함된 에지에 할당된 비용의 총합이 최소가 되는 트리를 Steiner Tree라고 하고, Steiner Tree를 찾는 문제는 NP-Complete[8]로 알려져 있다

이러한 멀티캐스트 라우팅 트리를 얻기 위해 최소비용 신장트리와 최단경로 트리를 혼합시킨 Modified Dijkstra 알고리즘, Kompella 알고리즘 [9], 그리고 Modified Kumar 알고리즘[2]과 같은 여러 가지 휴리스틱 방법들이 제안되었다. 이러한 휴리스틱 방법들은 최적해에 수렴한다는 보장이 없고, 최소네트워크에는 좋은 성능을 나타내지만 밀집네트워크에는 좋은 성능을 나타내지 못하는 단점이 있다. 이러한 휴리스틱 방법의 문제점을 해결하기 위해 유전자 알고리즘을 이용한 멀티캐스트 라우팅문제의 해법에 관한 연구가 진행되어 왔지만, 실시간 제약을 고려하지 않고 네트워크의 최소비용만을 고려한 연구가 대부분이었다[4, 7]. 최근에서야 네트워크 비용과 실시간 제약을 동시에 고려한 해법에 대한 연구들이 진행되고 있다[11].

본 논문에서는 실시간 멀티캐스트 라우팅 문제를 해결하기 위해 유전자알고리즘을 적용하였으며, 비용과 실시간 제약을 모두 고려한 새로운 적합함수(Fitness Function)를 이용해서 실시간 제약이 있는 멀티캐스트 라우팅 문제의 해법을 제시하였다. 2장에서는 문제 정의와 기존의 해법에 대해 살펴보고, 3장에서는 새로운 알고리즘을 제안하였다. 4장에서는 실험계산과정과 결과를 설명하고, 5장에서 결론 및 추후 연구방향을 제시하였다.

2. 실시간 멀티캐스트 라우팅문제

2.1 문제 정의

실시간 멀티캐스트 라우팅문제 해법을 위한 그래프 $G = (V, E, s, D, \Delta)$ 는 노드의 집합 V 와 에지의 집합 E , 시작노드 $s \in V$, 목적지노드 집합 $D \subset V$ 로 이루어진 무향, 연결, 가중 그래프이다. 여기서, s 와 D 는 미리 정해진다. 각각의 에지는 링크

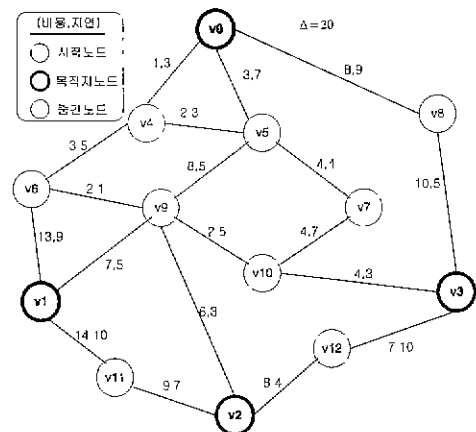
의 비용과 지연시간이라는 두 개의 매개변수를 가진다. $c(e)$ 는 에지 $e \in E$ 에 할당된 비용을 나타내는 함수이고, $d(e)$ 는 에지 $e \in E$ 에 할당된 지연을 나타내는 함수이다. 그리고 Δ 는 각 목적지 노드가 만족하여야 하는 값으로, 라우팅 트리에서 시작노드로부터 각 목적지까지의 경로에 포함된 모든 에지에 할당된 지연시간의 합은 Δ 이하여야 한다. 따라서 실시간 멀티캐스트 라우팅 문제는 $\{s\}UD$ 를 포함하는 서브트리 T 를 찾는 문제로 정의된다.

$$\text{Minimize } \sum_{e \in T} c(e)$$

T 는 지연제약 Δ 를 만족한다

T 에 포함된 노드 중에서 목적지가 아닌 노드를 중간노드라고 한다. 실시간 멀티캐스트 라우팅문제에서는 중간노드를 알면 최종적인 해는 시작노드, 목적지노드 집합, 중간노드로 이루어지는 실시간 제약조건 Δ 를 만족하는 최소비용 신장트리(CMST: Constrained Minimum Spanning Tree)를 구함으로써 얻을 수 있다.

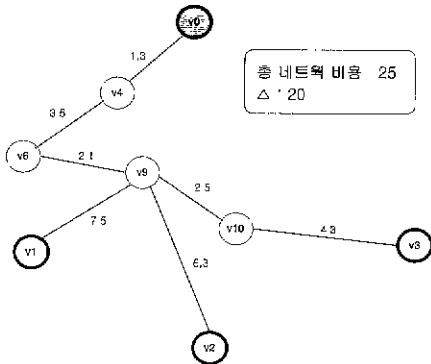
[그림 1]은 13개의 노드를 가진 네트워크의 예를 보여준다. 각각의 에지에 부여된 매개변수들은 (비용, 지연)을 나타낸다.



[그림 1] 네트워크 예

[그림 2]는 [그림 1]의 네트워크에서의 실시간 멀티캐스트 라우팅 트리(CMST)의 예를 보여준다.

v0을 시작노드(s), 목적지노드의 집합(D)은 {v1, v2, v3}이고 {v4, v6, v9, v10}은 중간노드가 된다. 각 목적지노드의 허용지연 시간 $\Delta = 20$ 이라고 했을 때, 네트워크의 총비용은 25이다.



[그림 2] 멀티캐스트 라우팅 트리 예

2.2 기존해법

실시간 멀티캐스트 라우팅을 위한 기존의 해법으로는 최소비용 신장트리와 최단경로 트리를 혼합시킨 Modified Dijkstra 알고리즘, Kompella 알고리즘[9], 그리고 Modified Kumar 알고리즘[2]과 같은 여러 가지 휴리스틱 방법들이 제안되었다. Modified Kumar 알고리즘은 간단하고 효과적으로 네트워크 비용과 허용지연을 절충할 수 있지만 최적의 결과는 얻지 못한다. Kompella 알고리즘은 특정한 지연시간을 갖는 지연제약 최소신장 트리인 CST(Constrained Steiner Tree)를 구함으로 실시간 멀티캐스트 라우팅 문제를 해결하였다. 라우팅 트리를 생성하기 위해 에지를 선택하는 과정에서 두 가지의 그리디 정책을 사용하였다. 첫째는 delay-cost 휴리스틱(CST_{CD})으로서, 에지를 선택할 때 비용과 지연을 모두 사용하여 에지를 선택하는 방법으로 비용과 지연 두 가지를 모두 최적화시킨다. 두 번째는 cost 휴리스틱(CST_C)으로, 일단 지연제약이 만족되면 과도하게 지연을 최소화하지 않고 비용을 최소화시키는 정책을 이용하는 방법이다.

본 논문에서는 이와 같은 실시간 멀티캐스트 라

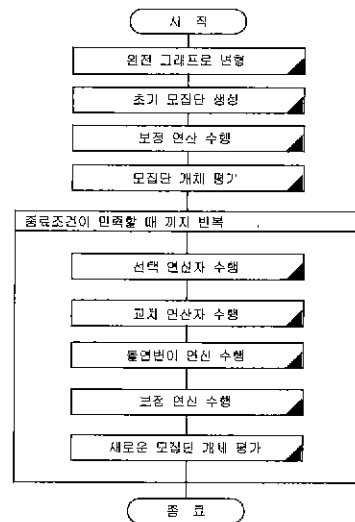
우팅문제의 해법을 위해 네트워크 비용과 지연시간을 고려한 GA를 사용함으로써 휴리스틱 방법에서의 문제점인 지역해에 빠지기 쉬운 단점을 보완하고, 또한 대규모 네트워크에 적합하며 실시간 지연을 만족하고 네트워크 비용을 최소화하는 새로운 유전자 알고리즘을 제안하였다.

3. 실시간 멀티캐스트 라우팅을 위한 제안된 유전자 알고리즘

3.1 유전자 알고리즘(GA)

GA는 자연적인 진화, 즉 적자생존의 원리에 기반을 둔 계산적인 모델이고, 주로 최적화문제를 해결하는데 널리 사용되었다[5, 6]. 또한, NP-hard인 조합 최적화문제를 공략하는데 효과적으로 적용되었다[12]. GA의 장점은 가능한 해 공간에서 전체 모집단을 갱신할 때 유전연산자를 사용하는 것이다.

GA는 다음 세 가지의 기본적인 부분으로 구성된다. 첫째, 개체의 인코딩, 둘째, 적합함수 수행, 셋째, 교차 연산자, 돌연변이 연산자 설계 및 적용으로 이루어진다. [그림 3]은 본 논문에서 제안한 실시간 멀티캐스트 라우팅 알고리즘에 대한 전체적인 흐름도를 나타낸다.



[그림 3] 제안된 GA의 흐름도

3.2 염색체 구성

제안된 알고리즘의 목표는 시작노드(s)와 목적지노드 집합(D)을 반드시 포함하고 지연제약 조건을 만족하면서 전체네트워크 비용이 최소가 되는 라우팅 트리(CMST : Constrained MST)를 구하는 것이다. 그러므로, 기본적인 그래프 G 에서 임의의 CMST를 구함으로써 초기 모집단을 구성할 수 있다. GA를 사용하여 문제의 해 공간을 탐색하기 위해서는 기본적으로 유전자 연산을 효과적으로 적용할 수 있도록 개체에 대한 염색체를 구성해야 한다. 해 공간은 그래프 G 에서의 CMST의 집합으로 제한할 수 있다. 다음은 CMST를 염색체로 구성하기 위한 방법이다.

먼저, 그래프 G 를 모든 두개의 노드사이에는 에지가 항상 존재하는 완전그래프로 변형한다. 이는 어떠한 스트링을 갖는 염색체에도 대응되는 가능해를 보장한다. 만약, 두 노드사이에 직접적인 에지가 없다면, 두 노드사이의 최단경로를 가상 에지로 나타낸다. 이 때 완전그래프의 각각의 에지에 할당되는 비용과 지연에 관한 정보는 디코딩과 적합도를 측정할 때 사용하기 위해 임시 테이블에 저장한다.

그래프 G 에서 $n=|V|$ 은 전체 그래프의 노드 개수이고, m 은 시작노드와 목적지노드 집합의 원소 개수의 합이다. 그래프 G 의 노드들을 $\{u_0, u_1, u_2, \dots, u_{n-1}\}$ 라고 하고, u_0 을 시작노드라고 하면 CMST는 $n-m$ 개의 중간노드를 이진 스트링, 즉 $a = (a_1, a_2, \dots, a_{n-m}) \in \{0,1\}^{n-m}$ 으로 나타낼 수 있다. 여기서 중간노드가 CMST에 포함되면 1, 포함되지 않으면 0을 부여한다. 트리 $T(a)$ 는 시작노드, 목적지노드 집합, 그리고 a 에서 1로 대응된 중간노드로 유도된 서브그래프 G' 의 모든 CMST의 집합을 나타낸다. [그림 4]는 [그림 2]의 CMST에 대한 염색체구성을 나타낸다.

v_1	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
1	0	1	0	0	1	1	0	0

$$a = (1,0,1,0,0,1,1,0,0)$$

[그림 4] [그림 2]의 CMST의 염색체 표현

3.3 디코딩

염색체를 평가하기 위해서는 염색체를 다시 트리로 나타내어야 한다. 본 논문에서는 염색체의 원소 중에서 1로 할당된 중간 노드와 $\{s\}UD$ 를 포함하는 CMST를 생성한다. 이를 위해 트리를 구성하는 에지를 선택할 때 비용과 지연에 가중치를 부여하는 변형된 Prim 알고리즘을 사용한다. [그림 5]는 변형된 Weighted Prim 알고리즘을 나타낸다[3].

```

Weighted_Prim
 $G' = (V', E')$ ;
 $T \leftarrow \{s\}$ ;
while  $V' - T \neq \emptyset$  do
     $(v, u) \leftarrow \text{Min2T}()$ ;
        where  $v \in T, u \in V' - T$ ;
    add edge  $(v, u)$  to the tree;
     $T \leftarrow T \cup \{u\}$ ;
End Weighted_Prim
    
```

[그림 5] Weighted Prim 알고리즘

위의 알고리즘에서, G' 는 $\{s\}UD$, 그리고, 중간노드 중에서 일부를 포함하는 V' 와 E' 로 이루어진 서브그래프이다. T 는 지금까지 만들어진 트리에 포함되어 있는 노드들의 집합을 나타낸다 $\text{Min2T}()$ 는 T 에 속한 노드와 $V'-T$ 에 속한 노드 사이에서 $[C_w \times c(e) + D_w \times d(e)]$ 의 값이 최소가 되는 에지를 찾는다[10]. 단, C_w 와 D_w 는 각각 비용가중치와 지연가중치를 나타낸다. 본 논문에서는 $C_w = D_w = 1$ 로 동일하게 부여한다.

3.4 적합함수

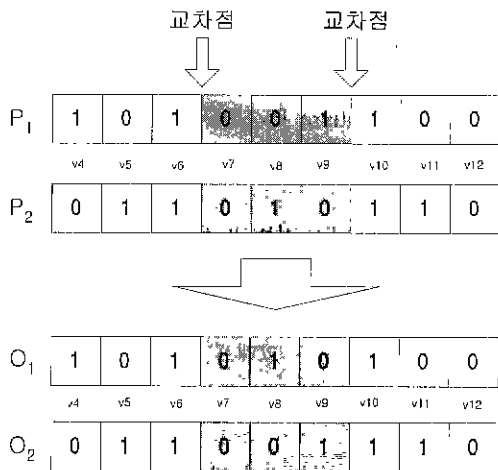
일반적으로 GA에서의 적합함수는 문제에서 최적화 시키고자하는 목적함수를 사용하는데, 모집단에서의 각 개체의 적합도를 측정하기 위해 사용된다. 본 논문에서는 실시간 지연제약을 고려한 CMST를 찾는 것이 목적이기 때문에, 비용과 지연을 모두 고려한 평가함수를 다음과 같이 정의한다.

$$F(a) = \frac{\beta}{\sum_{e \in T} CD(e)} \times \delta_T$$

여기서, $CD(e) = [Cw \times c(e) + Dw \times d(e)]$ 를 나타내고, β 는 비례상수, δ_T 는 t 가 지연제약을 만족하면 1. 그렇지 않으면 1/2의 값을 주어서 선별 연산에서 지연제약을 만족하는 개체가 선택될 수 있는 확률을 높여준다. 즉, 세대를 진화해 가면서 지연제약을 만족하는 개체는 다음세대로 진화하고 시간제약을 만족하지 못하는 개체는 배제된다. 선별 연산에서는 모집단의 각 개체에 적합도에 비례하는 확률을 부과하고 확률분포에 따라 개체를 선별하는 확률바퀴(Roulette Wheel) 방법을 사용한다.

3.5 교차 연산

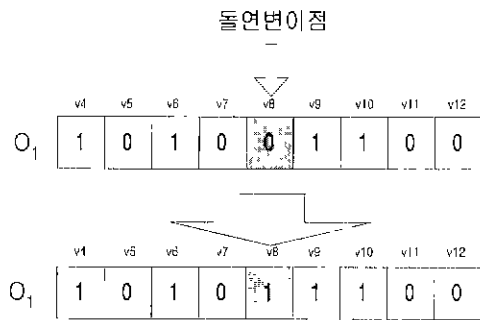
두 개의 부모 염색체 P_1, P_2 가 주어지고, 교차 연산에 의해 두 개의 자식 염색체 O_1, O_2 가 생성된다. 본 논문에서는 이점 교차 연산을 적용하였다. 염색체 사이의 두 교차점을 임의의 잡고, 두 교차점에 의해서 나누어진 염색체의 앞부분, 중간부분, 뒷부분 중에서 앞부분과 뒷부분은 한 부모로부터 상속받고 중간부분의 원소는 다른 한 부모로부터 상속받는다. 실험적으로 이점 교차연산이 일점 교차 연산보다 더 좋은 해를 유도하는 것으로 밝혀졌다[1]. [그림 6]은 이점 교차 연산의 예를 나타낸다.



[그림 6] 이점 교차연산의 예

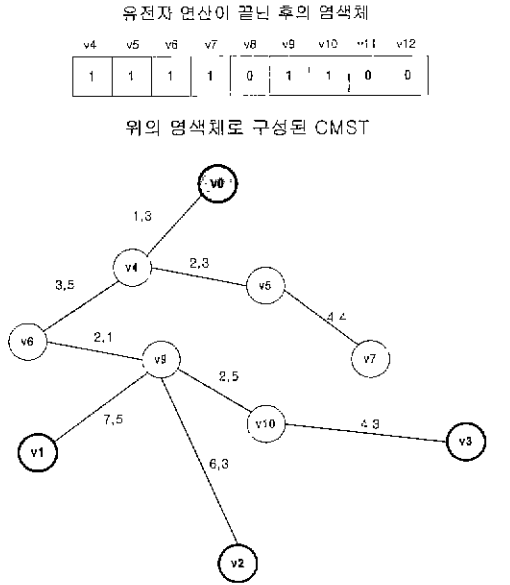
3.6 돌연변이 연산 및 보정 연산

돌연변이 연산은 교차 연산과는 달리 2개의 스트링을 이용해서 새로운 2개의 스트링을 만드는 것이 아니라 하나의 스트링을 부분적으로 조작하여 새로운 스트링을 만들어 내는 조작이다. 즉, 돌연변이 연산은 교차연산이 실질적인 진화 동작을 하는 것과는 달리 하나의 스트링 상에서 변화를 줌으로써 교차연산 수행 시 지나칠 수도 있는 상태를 만들어 내는 연산자이다. [그림 7]은 돌연변이 연산에 대한 예를 보여준다.

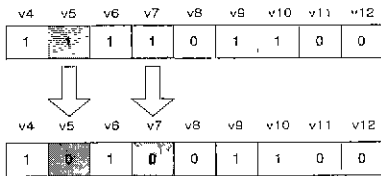


[그림 7] 돌연변이 연산의 예

본 논문에서 다루는 문제에서는, $\{s\}UD$ 와 스트링에서 1로 표시된 중간노드들로 구성된 CMST를 구할 때, 1로 표시된 중간노드가 CMST의 단말노드가 된다면 문제의 조건에 위배되므로 스트링 내에서 보정되어야 한다. 그러므로 초기모집단 구성과 유전자 연산을 통해서 얻어진 개체에 대해서 평가를 하기 전에 문제의 조건에 맞도록 염색체를 보정해 주어야 한다. 보정 연산에서는 염색체의 원소 중에서 문제의 정의에 위배되는 중간노드의 원소를 0으로 바꾸어 주는 방법을 사용한다. [그림 8]은 유전자 연산이 끝난 후의 염색체와 그에 대응되는 CMST를 나타낸다. [그림 9]는 [그림 8]의 CMST에서 단말노드가 된 중간노드를 보정하는 과정이다.



[그림 8] 유전자 연산과정이 끝난 후의 염색체와 CMST



[그림 9] 염색체 보정 과정

4. 실험 및 성능평가

4.1 실험 환경

본 논문에서는 유전자 알고리즘을 이용한 실시간 멀티캐스트 라우팅 문제 해법의 성능을 평가하기 위해 기존의 휴리스틱 알고리즘 중에서 가장 좋은 성능을 갖는 Kompella 알고리즘[9]과 비교 분석을 하였다. Kompella 알고리즘은 휴리스틱 방법이므로 제안한 유전자 알고리즘의 반복적으로 해를 찾는 방법과는 해를 찾는 시간의 비교는 중요한 의미를 갖지 않으므로 시간의 비교는 하지 않았다. [11]에서 제안한 방법에는 염색체의 구성에 관한 구체적인 설명이 없으므로 본 논문에서 제안한 방

법과의 직접적인 비교가 불가능하였다.

실험에 사용된 문제의 그래프는 전체 노드 개수와 에지의 개수, 그리고 그래프의 평균 연결도가 < 표 1>과 같은 값을 갖는 그래프를 무작위로 구성하였다. 또한 그래프를 완전그래프가 되도록 생성을 하여 멀티캐스트 라우팅 문제에 맞도록 하였다. 두 노드사이의 링크는 평균 연결도에 준하여 임의의 두 노드를 연결하였으며, 각 에지에 할당되는 비용과 지연은 각각 1부터 20사이의 정수 값을 무작위로 발생시켰다 그리고 각 문제마다 허용지연 Δ는 임의로 정의를 하였다.

<표 1> 문제로 사용된 그래프 구성

문제	노드 수	에지 수	노드 degree
1	10	15	3
2	20	30	3
3	30	45	3
4	40	100	5
5	50	125	5
6	60	150	5
7	70	175	5
8	80	280	7
9	90	315	7
10	100	750	15

유전자 알고리즘의 성능평가를 위한 실험환경은 한 세대의 개체수를 20, 전체 세대수는 100세대, 교차연산의 발생 비율은 0.7에서 0.9 사이의 값을 사용하였고, 각 돌연변이의 발생 비율은 0.02로 설정하여 실험하였다. 실험계산을 위한 하드웨어 환경은 IBM PC Pentium Pro(166 MHz)의 시스템 환경에서 실험하였다.

4.2 기존 휴리스틱 방법과 유전자 알고리즘의 성능 비교

본 실험에서 사용된 그래프 문제는 임의로 구성된 것이므로 최적해는 알려져 있지 않다. 따라서 허용지연을 만족할 때의 생성될 수 있는 여러 가지 라우팅 트리 중에서 최소가 되는 총 네트워크 비용의 값으로 성능 평가를 비교하였다. 유전자 알고리즘

의 경우 모든 세대를 거친 후 세대 내에서 최소 네트워크 비용을 갖는 라우팅 트리를 성능 평가 기준으로 삼았다.

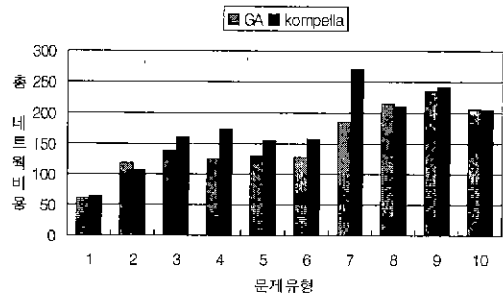
<표 2>는 <표 1>의 10개의 문제에 임의의 목적지 노드의 개수를 설정한 그래프에 대해 기존의 휴리스틱 방법과 본 논문에서 제안한 유전자 알고리즘을 적용한 후의 네트워크 비용과 최적해가 구해질 때까지의 시간을 초 단위로 나타낸다.

<표 2> GA와 기존 방법의 실험결과 값

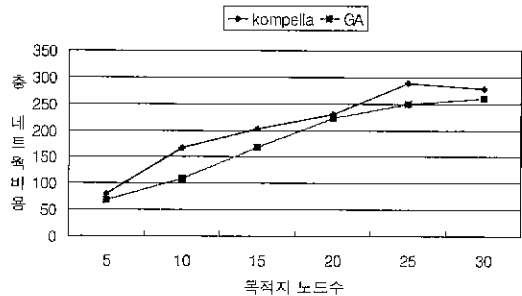
문제	목적지개수	Kompella(비용)	GA	
			비용	시간(초)
1	5	65	61	0.110
2	5	107	119	0.110
3	10	161	138	0.430
4	10	174	125	0.210
5	10	156	130	0.270
6	15	158	128	0.220
7	15	270	185	1.430
8	20	210	215	2.030
9	20	242	235	0.220
10	30	205	206	0.380

[그림 10]은 <표 2>의 결과값을 그래프로 나타낸 것이다. 그림에서 보면 알 수 있듯이 대부분의 문제에서 기존의 휴리스틱 방법보다 제안한 유전자 알고리즘이 더 좋은 성능을 보이고 있다. 그리고 전체 노드수가 작은 소규모의 네트워크에서는 휴리스틱 방법이 다소 좋은 성능을 나타내지만, 네트워크를 구성하는 노드수가 많은 대규모의 네트워크에서 유전자 알고리즘은 기존의 휴리스틱 방법보다 더 좋은 성능을 나타낸다는 것을 알 수 있다.

[그림 11]은 노드 개수가 50, 에지의 개수가 125, 네트워크의 평균 연결도가 5이고, 허용지연 $\Delta = 70$ 으로 동일하게 설정한 네트워크에 대해 목적지 노드 수를 증가시킬 때 기존의 휴리스틱 방법과 유전자 알고리즘의 총 네트워크 비용을 비교한 것이다. 목적지 노드수가 증가할수록 두 방법 모두 총 네트워크 비용은 증가하지만, 휴리스틱 방법에 비해 유전자 알고리즘이 효율적이라는 것을 알 수 있다.



[그림 10] 기존 방법과 GA 방법의 네트워크 비용 비교

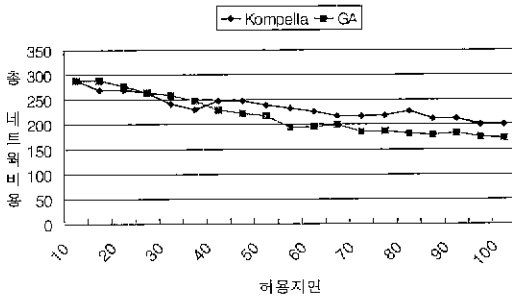


[그림 11] 목적지 노드수 증가에 따른 총 네트워크 비용

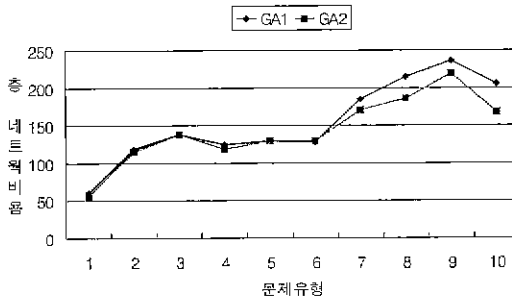
[그림 12]는 허용지연의 증가에 따른 휴리스틱 방법과 제안한 유전자 알고리즘의 성능을 비교한 것이다. 위 실험에 적용된 문제는 전체 노드의 개수가 70개, 에지의 평균 연결도가 7, 에지의 개수가 175개로 이루어진 네트워크에서 목적지 노드의 개수가 15로 주어진 경우에 대한 실험 결과이다. 그림에서 보면 허용지연 값이 작을 경우에는 라우팅 트리에 포함되는 에지를 선택함에 있어서 비용만을 고려한 휴리스틱 알고리즘이 비용과 지연을 모두 고려한 제안된 유전자 알고리즘보다 더 작은 네트워크비용을 갖지만, 허용지연의 값이 증가하면 제안된 유전자 알고리즘이 더 좋은 성능을 나타낸다.

[그림 13]은 에지를 트리에 추가 할 때 본 논문에서 제안한 Weighted Prim 알고리즘을 사용한 경우(GA1)와, 에지의 비용만을 고려하여 최소비용의 에지를 트리에 추가시키는 Prim 알고리즘을 사용한 경우(GA2)의 총 네트워크비용을 나타내고 있

다. GA2의 실험 결과가 GA1보다 좋은 이유는 라우팅 트리를 생성할 때 비용만을 최적화시키므로 비용과 허용지연을 모두 최적화시키는 GA1보다 다소 좋은 결과를 나타낼 수 있었다.



[그림 12] 허용지연 증가에 따른 총 네트워크 비용



[그림 13] Weighted Prim 기반 GA와 Prim 기반 GA의 총 네트워크 비용

5. 결론 및 추후 연구 방안

본 논문에서는 실시간 지연제약을 만족하는 멀티캐스트 라우팅문제를 해결하기 위하여 새로운 유전자 알고리즘을 제안하였다. 문제의 네트워크를 완전그래프 형태의 네트워크로 변형시켜서 고정길이 이진 스트링과 문제의 가능해와의 대응관계를 정의하였다. 디코딩 과정에서 트리를 구성하는 링크를 비용과 지연시간 모두를 고려할 수 있도록 새로운 알고리즘을 적용하였고, 개체의 염색체 구성을 효율적으로 이용한 유전 연산자가 사용되었다. 또한 휴리스틱 방법인 Kompella 알고리즘과 성능 비교를 통해서 유전자 알고리즘에 의한 접근이 더 좋은

최적해에 근접하는 경향이 있다는 것을 보였다. 또한 유전자 알고리즘이 밀집 네트워크와 희소 네트워크에 모두 효과적이라는 것도 보였다. 앞으로 디코딩 과정에서 비용가중치(Cw)와 지연가중치(Dw)를 다양하게 적용함에 따른 성능변화와 다중 최적화 기법인 파레토-최적집합의 개념을 적용해 볼 필요가 있다.

참고 문헌

- [1] 김여근 외, 「메타휴리스틱」, 영지출판사, 1997.
- [2] Bharah-Kumar K. and J. M. Jaffe., "Routing to Multiple Destinations in Computer Networks", *IEEE Trans. Commun.*, Vol.31 (1983), pp.343-351.
- [3] De Jong, K.A. and W.M. Spears, "Using Genetic Algorithms to Solve NP-complete Problems", in *Proc. 3rd Int. Conf. on Genetic Algorithms and Their Application*, J.D. chaffer, Ed. San Mateo, CA, Morgan Kaufmann, pp.124-132, 1989.
- [4] Esbensen, H., "Computing Near-Optimal solutions to the Steiner Problem in a Graph Using a Genetic Algorithm", *Networks*, Vol. 26(1995), pp.173-185.
- [5] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, Addison Wesley, 1989.
- [6] Holland, J.H., *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI, Univ. of Michigan Press, 1975.
- [7] Kapsalis, A., V.J. Rayward-Smith, and G.D. Smith, "Solving the Graphical Steiner Tree Problem Using Genetic Algorithms", *J. Opr. Res. Soc.*, Vol.44, No.4,(1993) pp.397-406.
- [8] Karp, R.M., "Reducibility Among Combinatorial Problems", in *Complexity of Computer Computations*, R.E. Miller and J.W. Thatcher. Eds. New York, NY : Plenum, pp.85-103,

- 1972.
- [9] Kompella, V.P., J.C. Pasquale and G.C. Polyzos, "Multicast Routing for Multimedia Communications", *IEEE/ACM Trans. on Networking*, Vol.1, No.3(1993), pp.286-292.
- [10] Ng, J.M. and P.K. Ng. "Cost-Delay Path Selection Function for Real-Time Multicast Routing", *Proc. 6th Int. Symp. on Modeling Analysis and Simulation of Computer and Telecommunication Systems*, pp.56-62, 1998.
- [11] Ravikumar C.P., and R. Bajpai, "Source-Based Delay-Bounded Multicasting in Multimedia Networks," *Computer Communications*, Vol.21, No.2(1998). pp.126-132.
- [12] Sedgewick, R., *Algorithms in C++*, Reading, MA, Addison Wesley, 1992.