



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

경영학석사 학위논문

타부리스트가 결합된 유전자 알고리즘을
이용한 트리형 네트워크의 경제적 설계



2011년 2월

부경대학교 경영대학원

경영학과(e-비즈니스전공)

이 성 환

경영학석사 학위논문

타부리스트가 결합된 유전자 알고리즘을
이용한 트리형 네트워크의 경제적 설계



지도교수 염 창 선

이 논문을 경영학석사 학위논문으로 제출함

2011년 2월

부경대학교 경영대학원

경영학과(e-비즈니스전공)

이 성 환

이성환의 경영학석사 학위논문을 인준함

2011년 2월



주 심 경영학박사 이 현 규 (인)

위 원 공 학 박 사 한 재 호 (인)

위 원 공 학 박 사 염 창 선 (인)

< 목 차 >

표 목차	iii
그림 목차	iv
Abstract	vi
I. 서 론	1
1. 연구의 배경 및 목적	1
2. 연구의 방법 및 구성	4
II. 네트워크, 유전자 알고리즘 및 타부서치	5
1. 네트워크	5
가. 네트워크의 종류	5
나. 네트워크의 설계	7
2. 유전자 알고리즘	9
가. 유전자 알고리즘의 개요	9
나. 염색체	12
다. 재생산	16
라. 교배(crossover)	19
마. 돌연변이(mutation)	23
3. 타부서치	25
III. 타부리스트가 결합된 유전자 알고리즘을 이용한 트리형 네트워크 설계	26
1. 트리형 네트워크 설계 문제의 기술	26
가. 트리형 네트워크 설계에 대한 가정 및 표기	27

나. 트리형 네트워크 설계 문제 모형	28
2. 트리형 네트워크 설계의 탐색체 표현방법	29
3. 트리형 네트워크 설계	31
가. 트리형 네트워크 설계를 위한 주요 절차	31
나. 초기 개체집단 생성	35
다. 교배 및 돌연변이	35
라. 서브트리 탐색	37
마. 타부리스트	39
4. 트리형 네트워크 설계를 위한 알고리즘	40
가. 노드 차수 계산 알고리즘	40
나. 서브트리 탐색 및 트래픽 계산 알고리즘	40
다. 타부리스트 관리 알고리즘	41
IV. 성능 평가	43
1. 문제 1 : 13개의 노드, 서브트리 트래픽 용량 3	44
2. 문제 2 : 13개의 노드, 서브트리 트래픽 용량 10	48
3. 문제 3 : 17개의 노드, 서브트리 트래픽 용량 5	51
4. 문제 4 : 17개의 노드, 서브트리 트래픽 용량 12	54
5. 문제 5 : 21개의 노드, 서브트리 트래픽 용량 6	57
6. 문제 6 : 21개의 노드, 서브트리 트래픽 용량 15	60
7. 노드 수의 증가에 따른 평균 설계비용의 차이	63
V. 결 론	66
참고문헌	68

< 표 목 차 >

<표 II-1> 물렛휠 선택 예제	17
<표 III-1> 링크의 인접행렬	30
<표 IV-1> 노드의 좌표	44
<표 IV-2> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 1) ...	45
<표 IV-3> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 1)	46
<표 IV-4> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 2) ...	49
<표 IV-5> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 2)	50
<표 IV-6> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 3) ...	52
<표 IV-7> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 3)	53
<표 IV-8> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 4) ...	55
<표 IV-9> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 4)	56
<표 IV-10> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 5) ...	58
<표 IV-11> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 5) ...	59
<표 IV-12> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 6) ...	61
<표 IV-13> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 6) ...	62

< 그림 목 차 >

<그림 II-1> 네트워크의 분류	6
<그림 II-2> 노드와 링크	7
<그림 II-3> 유전자 알고리즘의 기본 구조	10
<그림 II-4> 염색체와 형질의 표현방법	12
<그림 II-5> 해밍절벽을 가지는 적합도 함수	14
<그림 II-6> 기호코딩의 염색체 표현방법	15
<그림 II-7> 룰렛휠 선택 예제	17
<그림 II-8> 일점교배의 예시	20
<그림 II-9> 균등교배의 예시	21
<그림 II-10> 사이클 교배 예제-1	21
<그림 II-11> 사이클 교배 예제-2	22
<그림 II-12> 사이클 교배 예제	22
<그림 II-13> 순서 교배 예제	23
<그림 II-14> 교배의 예제	24
<그림 III-1> 트리형 네트워크 설계 구조의 예	30
<그림 III-2> 이진 염색체의 표현 예제	31
<그림 III-3> 타부리스트가 결합된 유전자 알고리즘을 이용한 트리형 네트워크 설계 절차	32
<그림 III-4 > 초기 염색체	35
<그림 III-5> 균등교배 예제	36
<그림 III-6> 돌연변이 후 염색체	36
<그림 III-7> 서브트리 탐색 전 노드의 차수	37
<그림 III-8> 서브트리 탐색 전 노드의 트래픽 용량	37
<그림 III-9> 서브트리 탐색 후 차수와 트래픽 용량	38
<그림 III-10> 서브트리 탐색 전 노드의 트래픽 용량	39

<그림 III-11> 서브트리 탐색 후 노드의 트래픽 용량	39
<그림 IV-1> 초기 가장 우수한 해(문제 1 : 390.46\$)	45
<그림 IV-2> 가장 우수한 최종해(문제 1 : 281.17\$)	46
<그림 IV-3> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 1)	47
<그림 IV-4> 초기 가장 우수한 해(문제 2 : 396.64\$)	49
<그림 IV-5> 가장 우수한 최종해(문제 2 : 277.66\$)	49
<그림 IV-6> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 2)	50
<그림 IV-7> 초기 가장 우수한 해(문제 3 : 603.88\$)	52
<그림 IV-8> 가장 우수한 최종해(문제 3 : 334.66\$)	52
<그림 IV-9> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 3)	53
<그림 IV-10> 초기 가장 우수한 해(문제 4 : 539.19\$)	55
<그림 IV-11> 가장 우수한 최종해(문제 4 : 337.55\$)	55
<그림 IV-12> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 4)	56
<그림 IV-13> 초기 가장 우수한 해(문제 5 : 681.97\$)	58
<그림 IV-14> 가장 우수한 최종해(문제 5 : 416.12\$)	58
<그림 IV-15> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 5)	59
<그림 IV-16> 초기 가장 우수한 해(문제 6 : 734.625\$)	61
<그림 IV-17> 가장 우수한 최종해(문제 6 : 454.71\$)	61
<그림 IV-18> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 6)	62
<그림 IV-19> 노드의 증가에 따른 평균 설계비용의 차이(트래픽 1)	64
<그림 IV-20> 노드의 증가에 따른 평균 설계비용의 차이(트래픽 랜덤)	65

Seong-hwan Lee

*Major in e-Business, Department of Business Administration
Graduate School of Business Administration
Pukyong National University*

Abstract

This study aims to consider an economic design problem of a tree network which is a kind of computer network. This problem can be modeling to be an objective function to minimize installation costs, on the constraints of spanning tree and maximum traffic capacity of sub-tree network.

The existing researches have used Heuristic Algorithm or Meta-Heuristic Algorithm such as Simulated Annealing, Tabu Search, and Genetic Algorithm to solve out the problem. Of them, the genetic algorithm is being used most widely. However, it produces numerous the same candidate solutions in the population, during an evolutionary computation process. The result disturbs efficient exploration of optimal solutions.

To overcome this defect of the genetic algorithm, in this study, Tabu List Coupled Genetic Algorithm is proposed. A proposed algorithm approach for tree network design is described in details.

Experimental results showed evidence that the approach performs more efficiently for finding a good solution or near optimal solution in comparison with the genetic algorithm approach.

Keywords : Tree Network Design, Spanning Tree, Maximum Traffic Capacity, Tabu List Coupled Genetic Algorithm, Genetic Algorithm

I. 서론

1. 연구의 배경 및 목적

현대 사회는 사회 활동이 점점 복잡, 다양해지며 하루만 지나도 엄청난 양의 정보가 쏟아져 들어온다. 초기의 컴퓨터는 주로 군사용 또는 과학 산술용으로 사용되었으나, 오늘날은 대량 정보의 구축과 컴퓨터의 응용으로 대량 정보 처리 위주의 정보화 사회로 발전함에 따라 데이터 통신의 역할은 그 만큼 중요하게 되었다(손동욱, 1995).

정보화 시대에 이르러 컴퓨터 네트워크는 기업 내부뿐만 아니라 기업 외부의 여러 곳에 분산·저장 되어 있는 정보를 효율적으로 사용할 수 있게 해준다. 뿐만 아니라 기업에 있어서 중요한 전략적 수단으로써 신속한 정보처리와 의사결정을 가능하게 해준다.

컴퓨터 네트워크는 정보통신기술의 발전과 함께 기업의 전략적 수단으로 그 중요성이 강조되고 있으며 컴퓨터 네트워크 구축에 대한 수요는 증가하고 있다. 이런 이유로 기업에 있어 효율적인 컴퓨터 네트워크의 설계는 매우 중대한 문제이다.

따라서 기업은 기업 내 정보처리기기들의 연결을 통해 정보자원을 공유하고 다양한 부문에서 생성되는 정보자원을 빠르게 교환하여 과업수행에 활용하고자, 기업 내부 네트워크로 근거리 통신망(LAN : Local Area Network)을 설치·운영하고 있다(염창선 등, 2005).

컴퓨터 네트워크는 대량의 통신 트래픽을 전송하는 기간망(backbone networks)과 종단 사용자와 기간망 사이의 트래픽을 전송하는 근거리 통

신망으로 구성된다.

컴퓨터 네트워크의 설계라 함은 넓게는 세 가지의 의미를 포함할 수 있다. 첫 번째는 교환기의 설계이며, 두 번째는 교환 네트워크의 구조 설계이면, 세 번째는 통신 프로토콜의 설계이다(Ahuja, 1985).

이 중 교환 네트워크의 구조 설계 문제는 노드의 수가 증가할수록 계산의 시간과 탐색공간이 지수적으로 증가하는 NP(Non-Polynomial)-hard 문제에 속한다. 이러한 문제들은 분기 한계(branch and bound)기법 등의 전통적인 방법들을 이용하면, 네트워크의 규모가 커지는 경우 네트워크 설계 문제를 풀이하기가 힘들어진다. 그래서 시뮬레이티드 어닐링(SA : Simulated Annealing), 타부 서치(TS : Tabu Search), 유전자 알고리즘(GA : Genetic Algorithm) 등의 메타휴리스틱 알고리즘이 널리 사용되고 있다(Papadimitriou, 1978). 메타휴리스틱 알고리즘 중에서는 유전자 알고리즘이 가장 많이 사용되고 있다.

본 연구에서는 유전자 알고리즘을 개선한 새로운 알고리즘을 이용하여 근거리 통신망의 한 유형인 트리형 네트워크 구조를 경제적으로 설계하는 방법에 대해 연구하고자 한다.

트리형 네트워크 설계는 CMST(Capacitated Minimum Spanning Tree) 문제로 불리며 집중기 노드(기간망 노드)의 위치와 집중기에 할당된 터미널 노드의 위치 등이 주어진 상태에서 신장 트리(spanning tree)와 서브트리의 최대 트래픽 용량을 제약조건으로 하면서 설계비용을 최소화 하는 터미널의 연결 형태를 결정하거나, 신장 트리와 노드 간 링크의 최대 트래픽 용량을 제약조건을 하면서 설계비용을 최소화하는 터미널의 연결 형태를 결정하는 것이다.

트리형 네트워크 설계를 위해 메타휴리스틱 기법을 이용한 기존 연구를 살펴보면 다음과 같다. Ersoy et al.(1993)은 SA를 이용하여 신장 트리 제

약조건 하에서 평균 네트워크 지연(average network delay)을 최소화하는 트리형 네트워크 설계 방법을 제시하였다.

Torres-Jiménez et al.(1999)은 SA를 이용하여 노드 간 링크의 최대 트래픽 용량 제약조건 하에 설계비용을 최소화하는 트리형 네트워크 설계 방법을 제시하였고, Sharaiha et al.(1997)은 TS를 이용하여 신장 트리와 노드 간 링크의 최대 트래픽 용량 제약조건 하에서 설계비용을 최소화하는 트리형 네트워크 설계 방법을 제시하였다. Zhou et al.(2006)은 GA를 이용하여 신장 트리와 서브트리의 최대 트래픽 용량 제약조건 하에서 설계비용을 최소화하는 트리형 네트워크 설계 방법을 제시하였다. Gamvros et al.(2006)은 GA를 이용하여 신장 트리와 노드 간 링크의 최대 트래픽 용량 제약조건 하에서 여러 개의 링크 속성을 갖는 트리형 네트워크의 설계비용을 최소화하는 설계 방법을 제시하였다.

GA의 해 탐색과정은 확률에 의한 연산과정을 포함하고 있어 지속적인 진화 연산과정 중 개체집단 내에 여러 개체가 동일한 해를 가질 수 있다. 개체집단 내의 동일한 해들의 존재는 GA의 효율적인 최적해 탐색을 방해하는 요인이 될 수 있다. 이러한 GA의 비효율적인 탐색과정을 개선하기 위해 TS의 타부리스트 기능이 이용될 수 있다. 타부리스트 기능은 개체집단 내의 개체들이 동일한 해를 갖는 것을 배제하는 역할을 수행한다.

그러므로 본 연구에서는 트리형 네트워크를 효율적으로 설계하기 위해, 유전자 알고리즘에 타부리스트 기능을 결합시킨 형태인 타부리스트가 결합된 유전자 알고리즘(TLCGA : Tabu List Coupled Genetic Algorithm)을 제안한다. 그리고 새로 제안한 알고리즘을 이용하여 신장 트리와 서브트리의 최대 트래픽 용량을 제약조건으로 하고 목적함수로 네트워크 총 설계비용을 최소화하는 트리형 네트워크의 설계 방법을 제시한다. 그리고 실험적 문제들을 가지고 본 연구에서 제시한 설계방법과 GA를 이용하여 설계하는

방법을 비교 평가함으로써 제안하는 본 연구에서 제시한 설계방법의 우수성을 입증하고자 한다.

2. 연구의 방법 및 구성

본 연구에서는 신장 트리와 서브트리의 최대 트래픽 용량 제약조건을 충족시키면서 설계비용을 최소화시키는 트리형 네트워크의 경제적 설계를 위해, 타부리스트가 결합된 유전자 알고리즘을 이용하는 설계 방법을 제시한다.

본 연구의 구성은 총 5개의 장으로 구성되어 있다. 제 1 장에서는 연구의 배경 및 목적, 연구의 방법 및 구성에 관하여 기술하고, 제 2 장에서는 트리형 네트워크 설계 문제에 대한 모델, 유전자 알고리즘의 개요와 타부서치의 개요 및 주요 연산자에 관한 이론적 고찰을, 제 3 장에서는 트리형 네트워크 설계 문제에 대한 기술과 문제 모형, 유전자 알고리즘의 염색체 표현방법, 유전자 알고리즘 및 타부 리스트의 연산자 및 해 알고리즘에 대해 기술한다. 제 4 장에서는 실험적 문제를 통해 타부리스트가 결합된 유전자 알고리즘의 성능을 평가한다. 마지막으로, 제 5 장에서는 연구의 결론과 한계점에 대해 기술한다.

Ⅱ. 네트워크, 유전자 알고리즘 및 타부서치

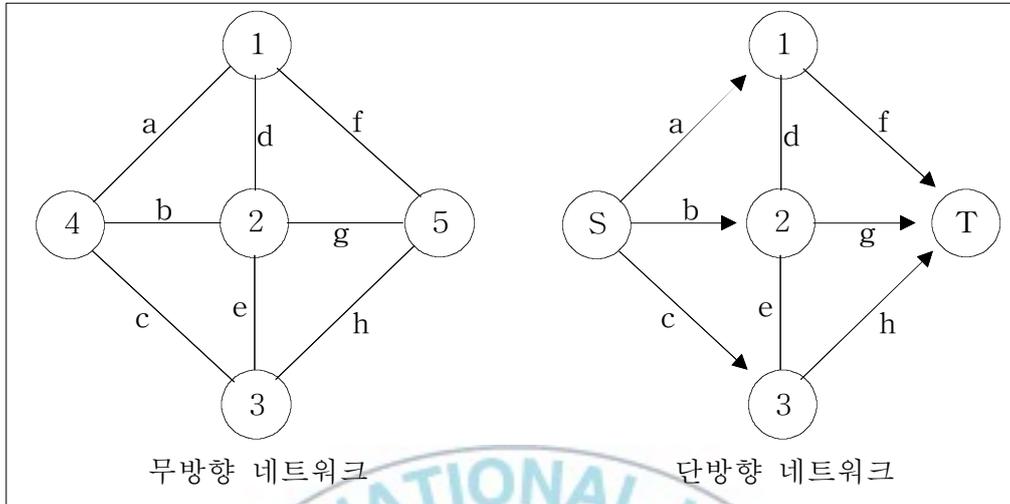
1. 네트워크

가. 네트워크의 종류

네트워크란 데이터 통신 그 자체로서 단말장치(terminal) 간의 통신경로를 구성하는 자원의 집합을 의미한다(신재호 등, 1998). 이러한 네트워크는 링크(link)의 방향성과 노드(node)의 시점과 종점에 따라 분류될 수 있다.

먼저 링크의 방향성에 따라 단방향 네트워크(directed network), 무방향 네트워크(undirected network)로 나눌 수 있다. 단방향 네트워크는 노드 간에 링크의 방향이 원천노드(시점)인 S노드에서 목적노드(종점)인 T노드까지의 화살표 방향으로 연결된 형태를 말한다. S노드는 나가는 가지만을, T노드는 들어오는 가지만을 갖는다. 여기서 가지는 노드에서 분리된 링크를 의미한다.

무방향 네트워크는 링크의 방향이 없는 상태로 노드들 간에 상호 연결된 형태를 말하며 노드 사이에서 링크는 나가는 가지, 들어오는 가지가 될 수 있다. 아래 <그림 Ⅱ-1>은 단방향 네트워크와 무방향 네트워크의 예시이다(고대식 등, 2000).



<그림 II-1> 네트워크의 분류

다음으로 노드와 노드의 연결 구성 형태에 따라 스타형(star) 네트워크, 링형(ring) 네트워크, 트리형(tree) 네트워크, 그물형(mesh) 네트워크로 분류할 수 있다.

스타형 네트워크는 모든 노드들이 중앙의 노드에 접속되어 있고, 중앙의 노드의 제어를 받는 네트워크를 의미한다. 링형 네트워크는 이웃 노드 간에 접속이 되어있는 형태로 주로 근거리 망에 사용되어진다. 트리형 네트워크는 각 노드가 계층적으로 구성 되어 있는 네트워크를 의미하며 계층형이라고도 한다. 그물형 네트워크는 각 노드가 완전하게 서로 서로 연결되어진 형태로 시스템 신뢰성이 가장 높다. 하지만 상호 연결에 대한 회선 비용이 증가하여 경제적 부담이 크다(손동욱, 1995).

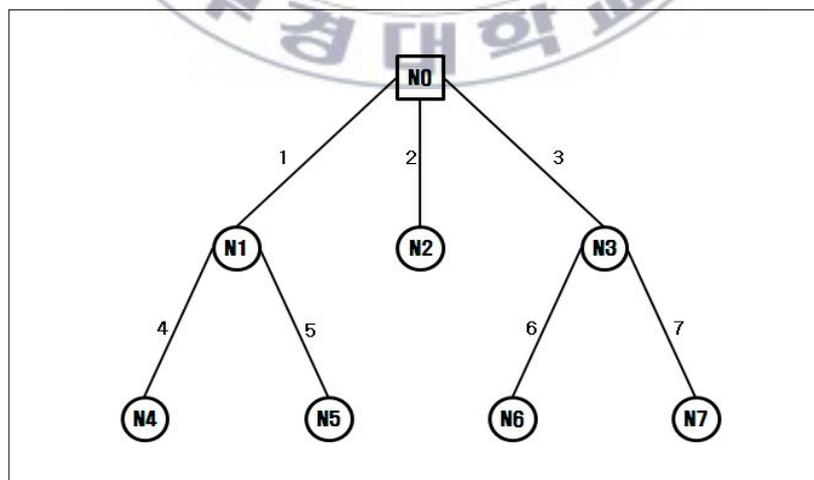
트리형 네트워크는 데이터가 양방향으로 모든 노드에 전송이 가능하고, 통신 회선 수가 절약되고, 통신 선로가 가장 짧다는 장점이 있다. 또한 네트워크 구축은 쉬우나 확장이 많은 경우 트래픽이 과중 되는 단점이 있다. 그리고 상위 노드의 고장은 하위 노드의 사용에 영향을 미친다.

특히 트리형 네트워크에서 신장 트리와 노드의 최대 트래픽 용량 또는 노드와 노드 간 링크의 최대 트래픽 용량을 제약조건으로 최소 비용으로 만족하는 네트워크의 연결 형태를 결정하는 문제를 그래프 용어로 CMST (Capacitated Minimum Spanning Tree) 문제라고 한다(Kershenbaum, 1993). 이 문제는 중단 노드의 수가 많아지게 되면 계산 시간이 지수적으로 증가하는 NP-hard 문제로 알려져 있다(Papadimitriou, 1978).

본 연구에서는 트리형 네트워크의 설계 문제를 다루며, 제약조건은 신장 트리와 서브트리의 최대 트래픽 용량으로 서브트리의 최대 트래픽 용량을 제한함으로써 임의의 서브트리가 고장 나는 경우 그 라인에만 피해를 국한시켜 네트워크 전체 피해를 최소화하기 위함이다.

나. 네트워크의 설계

본 연구에서 사용하는 기본 개념을 정리하기 위해 <그림 II-2>의 트리형 네트워크를 예로 설명하고자 한다.



<그림 II-2> 노드와 링크

<그림 II-2>를 보면 $N_0, N_1, N_2, \dots, N_7$ 을 노드라고 하며, 루트노드인 N_0 는 사각형으로 표시되어 있다. 노드는 각 터미널 전송을 요구하는 데이터들을 목적지별로 분류하고 전송경로를 결정하여 결정된 경로를 통해 데이터를 전송한다. 또한 노드와 노드사이에 번호가 붙은 실선은 링크라고 하며 각 노드사이의 경로를 링크번호로 나타내고 있다. 트리형 네트워크에서 노드와 노드사이의 링크의 수는 $n-1$ 개이다(Papadimitriou C. H., 1978).

트리형 네트워크에서 연결 가능한 경로 수는 노드 수의 증가에 따라 지수적으로 증가하여 네트워크 설계 문제를 더욱 복잡하게 한다.

트래픽이란 노드와 노드사이에 전송하고자 하는 목적지별로 모아진 메시지를 의미한다.

네트워크를 설계하는 것은 주어진 비용 및 자원 하에서 적절한 노드들의 위치 선정 및 링크용량의 적절한 선택, 정확한 수요 예측, 미래에 발생하게 될 네트워크의 확장의 용이성, 네트워크 기기들의 올바른 선택, 네트워크의 신뢰성 등 종합적인 사항들을 고려하여 이루어져야 한다. 즉 네트워크 설계가 네트워크를 구성하는 참여자들의 직관에 의하여 이루어져서는 안 된다는 것을 의미하며 보다 정확하고 체계적인 분석이 요구됨을 의미한다.

네트워크 설계라 함은 넓게는 세 가지의 의미를 포함할 수 있다. 첫째로는 교환기의 설계이며, 둘째로는 교환 네트워크 구조의 설계이며, 셋째로는 통신 프로토콜의 설계이다(Ahuja, 1985). 첫째는 주로 하드웨어 관련사항이고, 셋째는 소프트웨어의 설계와 관련된 것으로써 본 연구의 관심사항이 아니므로 두 번째의 교환 네트워크 구조를 설계하는 문제를 다루는 좁은 의미로 간주한다.

네트워크 설계의 목적은 주어진 제약조건을 만족하는 네트워크를 최소의 비용으로 구성하는 방법을 찾아내는 것으로 막대한 투자비용이 요구되는 네트워크 구축에서 서비스 수요, 네트워크의 성능, 신뢰도, 환경요건 등을 고려하여 경제적이며 최적의 네트워크 구성을 제공하는 것이다.

트리형 네트워크의 경우 원천노드와 목적노드사이에 하나 이상의 경로가 존재하기 때문에 복잡한 경로선택 방법과 링크용량 배분 문제들에 관한 해법들이 요구되어진다. 네트워크를 물리적으로 구성하는 데는 막대한 비용과 시간이 소요되고 미래의 네트워크의 성능에 결정적인 요인으로 작용하기 때문에 설계 단계는 매우 중요하다.

네트워크의 설계는 네트워크의 노드들을 연결하기 위해 링크들을 할당하고 경로선택방법과 각 링크들의 용량을 선택해야 하는 것이며 여러 가지의 성능 및 경제성 등을 고려하여 이루어져야 하는데, 특히 노드의 위치, 링크의 연결 관계, 경로선택 및 링크용량 등이 직접적으로 네트워크의 전송시간을 결정하게 된다(고대식 등, 2000).

2. 유전자 알고리즘

가. 유전자 알고리즘의 개요

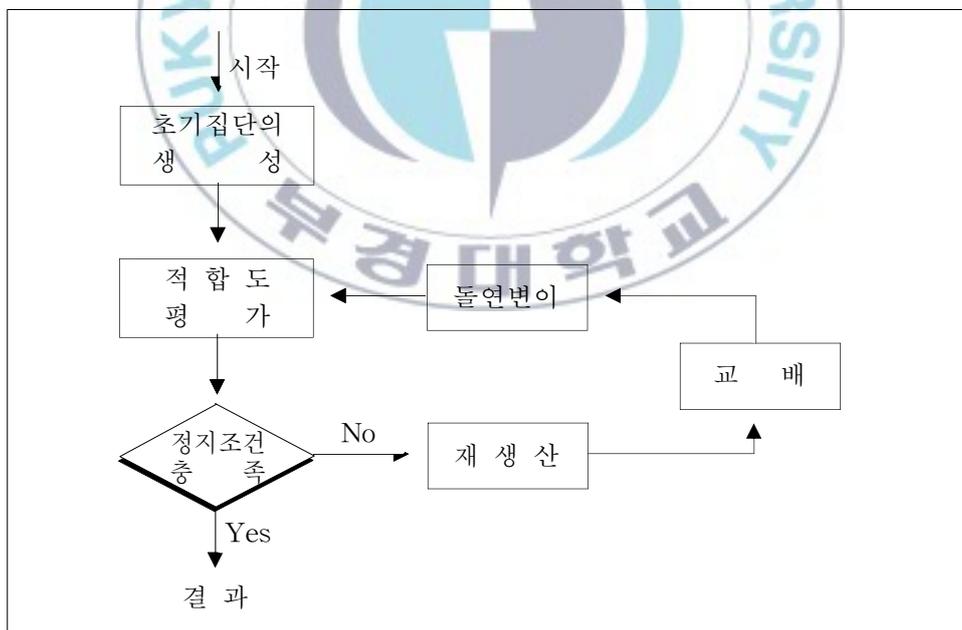
유전자 알고리즘은 Holland(1975)에 의해 처음 소개되었으며, 적자생존의 유전법칙을 컴퓨터 알고리즘에 적용하여 확률적 최적해를 탐색하는 기법이다. 즉, 탐색해들 중에서 좀 더 나은 해는 다음 세대에 살아남고 그렇지 않은 해는 도태된다.

난수발생에 의해 생성된 s 개의 염색체(또는 스트링)로 구성된 개체집단(population) 내에서 부모해(1세대)로부터 선택(selection), 재생산(reproduction), 교배(crossover), 돌연변이(mutation) 등의 연산을 통해 자손해(2세대)를 생성시키게 된다. 이러한 과정을 반복하여 준 최적 및 최적의 해를 찾게 된다. 유전자 알고리즘의 기본 구조는 크게 초기화, 적합도

(fitness) 평가, 재생산, 교배, 돌연변이 5단계로 <그림 II-3>과 같이 구분된다(진강규, 2000).

초기화 단계에서는 주어진 문제의 해가 될 가능성이 있는 개체집단이 생성된다. 초기점들은 해 공간 내에 무작위로 분포되도록 선택되거나 아니면 경험적인 방법으로 선택된다.

다음 단계에서는 개체들의 강점과 약점이 적합도 항목으로 평가된다. 여기서 각 개체는 복호화 되어 목적함수를 제공하고 이로부터 적합도가 계산된다. 적합도는 자연계에서 환경의 역할을 수행한다. 그들의 상대적인 적합도 값에 따라 현재 집단 내의 개체들이 재생산을 위해 선택된다. 우수한 개체들은 더 많이 선택되어 복제되고 반면에 나쁜 개체들은 집단으로부터 소멸되게 된다.



<그림 II-3> 유전자 알고리즘의 기본 구조

선택된 개체들은 교배를 통해 재결합되는데 서로 간에 유전정보를 교환함으로써 집단에 새로운 개체를 도입하게 된다. 교배는 일반적으로 현 집단 내에 존재하는 정보만을 이용하여 변화를 시도하게 된다. 만약 이전 세대의 선택과정에서 특정 정보를 잃게 되면 교배는 이러한 정보를 보유하는 새로운 개체들을 생산할 수 없다.

돌연변이는 선택된 개체의 하나 이상의 유전정보를 임의로 변경하여 집단에 새로운 정보를 도입하는 수단을 제공한다. 그러나 이러한 변화는 보통 낮은 확률을 가지고 이행된다. 돌연변이를 이용한다는 것은 탐색공간상의 어떤 점에 도달할 수 있는 확률이 결코 0이 아니라는 것을 의미한다.

이렇게 해서 새롭게 형성된 집단은 다시 평가되고 최적의 해가 발견될 때까지 앞서 수행한 일련의 연산과정을 종료조건이 충족될 때까지 반복한다(Melanie Mitchell, 1997).

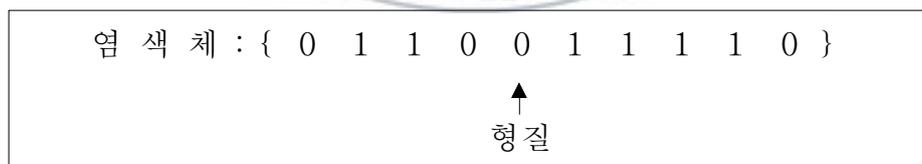
유전자 알고리즘이 기존의 탐색 알고리즘과 구분되는 몇 가지 특징들은 다음과 같다(Goldberg, 1989)

- 문제의 파라미터를 직접 이용하기보다는 이를 적절히 기호로 표현하여 이용한다.
- 하나의 해를 다루는 것이 아니라 집단을 취급한다.
- 목적함수 값만을 필요로 하고, 연속성(continuity), 미분가능성(differentiability), 단봉성(unimodality) 등과 같은 다른 보조정보를 요구하지 않는다.
- 유전 연산자들을 사용한다.
- 얻어진 결과는 결정적이기 보다는 확률적이다.

유전자 알고리즘은 해의 탐색능력이 우수하여 공학, 자연과학 분야뿐만 아니라 경영학 및 사회과학 분야의 최적화 또는 의사결정 문제에 다양하게 적용이 가능하다. 특히 유전자 알고리즘은 복잡한 해 공간의 탐색능력이 우수하여 변수와 제약이 많은 대형 수리 문제를 푸는 데 적합한 기법이고 모형에 대한 유연성이 높아 제약조건 첨가나 목적함수의 변경이 용이하다는 장점이 있다.

나. 염색체

염색체(chromosome)는 유전자 알고리즘에서 가장 기본이 되는 것으로 이것을 기본 단위로 유전 연산을 하게 된다. 어떤 문제이든 유전자 알고리즘을 적용하기 위해서는 반드시 염색체로 표현해야 하며, 염색체의 표현은 문제에 의존적이므로 염색체는 각 해의 특성을 표시할 수 있어야 한다. 염색체 내의 요소를 형질(allele)이라고 하며, 재생산 연산은 염색체 단위로 이루어지고 교배, 돌연변이 연산은 염색체 내의 형질에서 이루어진다. 염색체의 표현방법과 형질의 표현은 아래 <그림 II-4>와 같다.



<그림 II-4> 염색체와 형질의 표현방법

초기의 유전자 알고리즘은 염색체를 이진 스트링(염색체)으로 표현하였다. 그러나 이러한 방법으로 공학용 수치 계산 문제나 다차원 함수의 최소값을 찾는 문제들을 해결하기는 매우 어렵다(Goldberg, 1991). 이진 스트링

의 표현으로는 한계를 가지는 문제들에 대해서 염색체 표현에 대한 연구가 이루어지고 있다(장수현 등, 1998).

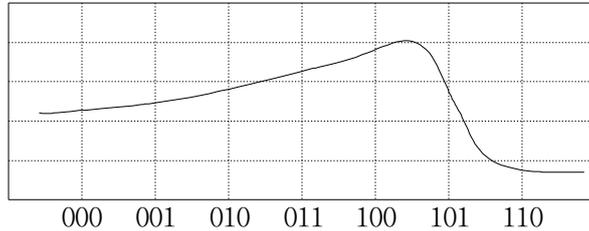
염색체의 표현방법으로는 이진코딩(binary encoding), k-ary 코딩(k-ary encoding), 기호코딩(symbolic encoding), 실수코딩(real-value encoding) 등이 있다.

본 연구에서 다루는 트리형 네트워크 설계 문제는 네트워크를 구성하는 링크의 유무에 대한 정보만을 필요로 하므로 염색체 표현 방법으로 이진코딩을 사용한다.

(1) 이진코딩

이진코딩은 초기의 유전자 알고리즘이 이러한 형태를 취해 왔기 때문에 아직도 폭 넓게 사용되고 있다. 이 표현은 이론적 분석을 용이하게 해서 특정 유전자들의 성장과 쇠퇴를 예측할 수 있는 메커니즘을 제공한다(Goldberg, 1989).

이진코딩은 아직도 폭 넓게 사용되고 있지만 해 공간에서 가까운 두 점이 부호공간에서도 항상 가깝도록 사상되지 않는 단점을 가진다. 한 예로 해 공간에서 인접한 수 3과 4는 부호공간에서 각각 011과 100이 되어 3비트의 차이가 나고 이러한 현상은 스트링의 길이가 길면 길수록 확대되는데 이와 같이 인접한 두 수 사이에 발생하는 급격한 비트 차이를 해밍절벽(hamming cliff)이라고 한다. 어떤 상황에서는 해밍절벽은 유전자 알고리즘을 비전역 최적해 쪽으로 수렴케 하는 문제를 일으킨다(Goldberg, 1991).



<그림 II-5> 해밍절벽을 가지는 적합도 함수

<그림 II-5>에서와 같이 스트링 100이 전역해이고 왼쪽의 4개 스트링들은 적합도가 높은 반면에 오른쪽의 2개 스트링은 적합도가 낮게 나타나고 있다. 이는 적합도가 높은 쪽에서 수렴할 가능성이 높은 것을 의미하며, 비록 011과 100이 인접해 있지만 둘 사이의 해밍절벽이 존재하므로 011이 100에 도달하기 위해서는 3비트의 차이를 극복할 수 있어야 한다. 하지만 단순 돌연변이 연산을 통해 3개의 비트를 동시에 변경되는 것을 기대하기는 힘들다. 그러므로 비록 전역해는 아니지만 적합도가 높은 011로 수렴해가기 쉽다.

(2) k-ary 코딩

k-ary 코딩은 염색체의 형질 값이 링크타입 k 가 가질 수 있는 값들 중 하나의 값으로 염색체를 표현하는 방법이다. 네트워크 설계 문제를 염색체로 표현하는데 있어서 k 값은 링크타입의 개수로 정의할 수 있으며 링크타입이 5개인 경우, k 가 가질 수 있는 값은 $k \in \{0, 1, 2, 3, 4\}$ 이다. 두개의 노드를 연결한 링크의 존재 유무에 따라서 0과 1이 아닌 값으로 나뉘지며 링크 용량, 비용의 링크 속성에 따라 1과 4사이의 값 하나를 가진다.

이 코딩방법은 네트워크 설계 구조를 쉽게 표현할 수 있지만 돌연변이 연산에서 똑같은 확률로 0과 4사이의 값 중의 하나로 반전되기 때문에 돌연변이 연산의 효과를 제대로 얻지 못할 가능성이 높아진다(진광규, 2000).

(3) 기호코딩

기호코딩 염색체는 문제의 노드들을 의미하는 알파벳의 열로 표시된다.

염색체 : $(v_1, v_2, \dots, v_i, \dots, v_n)$

<그림 II-6> 기호코딩의 염색체 표현방법

여기서 $v_i (1 \leq i \leq n)$ 는 이진표현에서 유전자로서의 의미를 가지며, 흔히 어떤 범위에서 취한 정수 집합의 원소가 된다. n 은 노드 수로서 염색체의 길이와 일치한다. 이 코딩의 특징은 유전자간의 순서는 의미를 가지나, 이들 간의 거리는 의미를 갖지 않는다.

이 염색체의 구조는 이진코딩의 구조와는 다르므로 기존의 교배와 돌연변이 연산을 적용하는 것이 불가능하며 특수 연산자가 요구된다. 비록 특수 연산자를 사용한다 하더라도 염색체의 적법성 유지 문제는 계속 남게 된다(진강규, 2000).

(4) 실수코딩

실수코딩은 염색체를 실수 값의 열로 표현하는 것이다.

실제 값의 이진코딩은 하나의 수로 표현되어 사용된 bit들의 수에 의존하므로 정밀도의 손실을 겪게 된다. 역시 실수코딩에서도 염색체의 스트링들이 매우 짧게 되어 이후 이것은 유전자 알고리즘이 큰 문제들을 해결하고자 할 때 실패하는 원인인데, 왜냐하면 매우 큰 스트링 크기를 수반해야 되기 때문이다. 실수코딩을 사용할 경우에 정밀도는 계산 환경에 따라 다

르겠지만 더 많은 비트를 사용함으로써 이진표현의 정확도가 더욱 개선될 수 있으나 계산속도가 느리다는 단점이 있다. 그리고 이 코딩 또한 새로운 교배와 돌연변이 연산자를 필요로 한다(장수현 등, 1998).

다. 재생산

자연에서 환경은 더 적합한 생물들이 더 높은 생존 기회를 갖도록 이들의 생존을 조절하는 기능을 가진다. 이런 환경을 통하여 생물들은 생존경쟁에 이기면 살아남고 지면 도태하게 되는데, 재생산은 적자생존(survival of the fittest) 또는 자연도태(natural selection) 현상을 모방하려는 인위적인 메커니즘이다.

이를 알고리즘으로 구현한 재생산은 만들어진 해들 중에서 가장 우수한 해들을 골라 그것을 그대로 자손세대에 물려주는 것이다. 이것의 의미는 좋은 형질을 그대로 보존하는 것에 있다. 이 단계에서는 얼마만큼을 자손세대에 물려 줄 것인가를 결정하여야 하는데 이 비율에 따라 세대가 진화하면서 생기는 해의 수렴도가 변화하게 된다. 비율이 높을수록 빨리 안정되게 수렴을 하나 해의 질이 좋지 않을 수 있고, 부분 최적해(local optimal)에 빠질 수 있다. 반면 비율이 낮을수록 더 다양한 해를 탐색하기 때문에 더 나은 질의 해를 찾을 수 있으나 수렴도는 떨어진다. 일반적으로 재생산 비율은 40~50%를 주로 사용하며, 가변적으로 쓰일 수 있다.

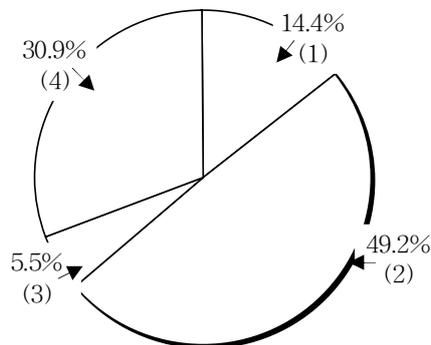
재생산 연산 이후의 교배 연산에서 어떻게 부모를 선택할 것인가가 중요한 문제가 되는데, 재생산을 알고리즘으로 구현한 방법으로는 룰렛휠 선택(roulette wheel selection), 순위기반 선택(rank-based selection), 토너먼트 선택(tournament selection) 등이 있다(Melanie Mitchell, 1997).

(1) 룰렛휠 선택

가장 기본적인 선택방법으로 각 염색체들은 각각 적합도를 평가한 다음 각 염색체의 적합도를 모두 합한 값만큼의 크기를 가진 룰렛휠을 가정한다. 그런 다음 이 룰렛휠 상에 자신의 적합도의 비율만큼 공간을 배정 받는 방법이다. 가령 적합도 함수를 $f(x) = x^2$ 로 가정하고 개체집단의 염색체의 수가 4개일 때 각 염색체에 대한 적합도와 공간배정 비율에 대한 예를 보면 다음과 같다.

<표 II-1> 룰렛휠 선택 예제

번호	염색체	적합도	총합에 대한 비율(%)
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
총합	.	1170	100



<그림 II-7> 룰렛휠 선택 예제

(2) 토너먼트 선택

이 방법은 가장 간단한 선택방법으로 두 개의 검색체를 임의로 선택한 다음 $[0, 1]$ 범위의 난수를 발생시킨다. 이 값이 t 보다 작으면 두 검색체 중 적합도가 높은 것을 선택하고 그렇지 않다면 적합도가 낮은 것을 선택하는 방법이다. 여기서 t 는 파라미터화 할 수 있는 것으로 t 값이 높을수록 선택 압이 높아진다. 두 개의 검색체를 선택하는 대신 $2k$ 개의 검색체를 선택한 다음 이들을 난수 발생에 의해 토너먼트 형식으로 비교하여 마지막으로 남은 것을 선택하는 방법도 있다. 토너먼트 선택은 대체로 수행에 필요한 시간이 매우 짧은 것이 장점이다. 하지만 이 방법은 우수한 개체가 도태(소멸)될 가능성을 배제할 수 없는 단점이 있다.

(3) 순위기반 선택

룰렛휠 선택은 적합도의 값을 가지고 교배할 개체를 정하는 방법이다. 하지만 이 방법의 단점은 적합도가 특히 높은 개체가 발생하면 그 개체의 영향으로 전체의 개체가 유사해지게 되어 조기수렴 문제를 일으킬 수 있다. 반대로 적합도의 차이가 거의 없으면 개체의 선택 확률은 거의 균등해지기 때문에 임의의 선택과 유사한 효과를 가진다. 그러므로 룰렛휠 선택법과 토너먼트 선택법의 단점을 보완하기 위한 방법으로 순위기반 선택법이 있다.

순위기반 선택은 검색체들을 적합도 순으로 순위를 매긴 다음, 가장 좋은 검색체부터 일차 함수적으로 적합도를 배정하는 방법이다.

이 선택법은 룰렛휠 선택에 비해 적합도가 높은 검색체와 적합도가 낮은 검색체 사이의 적합도의 차이가 크게 나는 것을 막을 수 있다.

라. 교배(crossover)

자연계의 생물들은 그들의 유전자를 서로 섞는 유성생식(sexual mating)을 통하여 자손을 생산하게 되고, 이런 성적 재생산을 알고리즘으로 구현한 것이 교배 연산이다. 교배는 탐색공간 상의 가능한 새로운 점을 찾기 위하여 개체집단으로부터 부모(parent) 염색체 쌍을 임의로 선택하고 교배 방법에 의한 교배 연산을 통하여 자손(offspring) 염색체를 생성시킨다.

따라서 교배연산은 우수한 두 형질의 염색체를 교배했을 때 더 우수한 형질의 결과가 나올 것이라는 가정에서 행해지는 연산이다. 즉 우수한 형질을 공유함으로써 더 우수한 형질을 찾아내고자 하는 시도이다. 개체집단에서 재생산에 의해 생성된 자손을 제외한 나머지를 교배를 통해 만들어 내게 된다.

교배의 방법으로는 일점교배(one-point crossover), 다점교배(multi-point crossover), 순환교배(cycle crossover), 균등교배(uniform crossover), 순서교배(order crossover) 등이 있다(Melanie Mitchell, 1997).

(1) 일점교배

일점교배는 가장 기본적인 교배연산이다. 초기의 유전자 알고리즘에서 많이 사용되었다. 길이가 n 인 일차원 문자열로 된 염색체 상에서 일점 교배로 자르는 방법의 총 수는 $n-1$ 가지이다. 6번째와 7번째 형질 사이에서의 일점교배의 예는 다음 <그림 II-8>과 같다.

부모 염색체 1 : (1010101101)
부모 염색체 2 : (1101010000)
자손 염색체 1 : (1010100000)
자손 염색체 2 : (1101011101)

<그림 II-8> 일점교배의 예시

(2) 다점교배

다점교배는 일점교배보다 자르는 방법의 수가 훨씬 많다. 염색체의 길이가 n 일 때 k 점 교배로 자르는 방법의 총 가지 수는 ${}_{n-1}C_k$ 이다. 다점교배는 일점교배보다 교란(perturbation)의 정도가 크다. 따라서 보다 넓은 탐색공간을 탐색할 수 있다. 반면에 교란이 강하면 수렴성이 떨어진다.

교란이 강하다는 것은 스키마가 파손될 확률이 높다는 것이다. 대신 새로운 스키마의 생성은 더 다양하게 할 수 있다. 따라서 항상 최적인 교란의 정도란 있을 수 없다. 일반적으로 다점 교배는 순수 유전 알고리즘보다 혼합형 유전 알고리즘에 더 어울린다. 왜냐하면 혼합형 유전 알고리즘에는 다소의 지역 최적화 기능이 있으므로 순수 유전 알고리즘보다 교란에 대한 회복력이 강하다. 혼합형 유전 알고리즘의 경우, 교란의 정도가 너무 미약하면 후반부에 부모 해와 같은 자손 해가 만들어질 확률이 높아져 탐색 시간의 낭비가 커진다(Eshelman et al., 1989).

(3) 균등교배

일점교배와 다점교배가 자름선을 이용하여 이루어지는 데 반해 균등교배는 자름선을 이용하지 않는다. 균등 교배는 먼저 임계 확률 P 를 설정한다. 가장 일반적인 임계 확률은 0.5이다. 두 부모해를 부모1, 부모2라 하고 각

각의 유전자 위치에 대하여 난수를 발생한 다음 이 값이 P 이상이면 해당 부모1의 같은 위치로부터 유전자를 복사해오고, 그렇지 않으면 부모2의 같은 위치로부터 복사를 한다. 아래 <그림 II-9>는 P = 0.5인 균등 교배의 한 예를 보인다.

부모1 :	1	0	1	0	1	0	1	1	0	1
부모2 :	1	1	0	1	0	1	0	0	0	0
난 수 :	.8	.2	.3	.8	.9	.6	.4	.7	.7	.3
자 식 :	1	1	0	0	1	0	0	1	0	0

<그림 II-9> 균등교배의 예시

균등 교배는 자름선을 이용하지 않으므로 일점 교배나 다점 교배에 비해 스키마의 결합 형태가 다양하다. 대신 일반적으로 교란의 정도가 크므로 수렴 시간이 오래 걸린다. Syswerda(1989)에 의해 제안된 이후 많은 주목을 받았고 폭넓게 이용되고 있는 교배 연산이다.

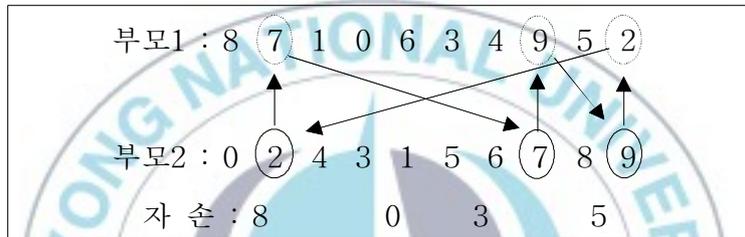
(4) 사이클 교배

사이클 교배는 TSP(Traveling Salesman Problem)의 문제에서와 같이 염색체가 순열로 표현되는 경우에 적용 가능한 교배 연산이다. 아래 <그림 II-10>와 같은 두 부모해 부모1, 부모2로부터 사이클 교배를 한다고 가정하면

부모 1 :	8	7	1	0	6	3	4	9	5	2
부모 2 :	0	2	4	3	1	5	6	7	8	9

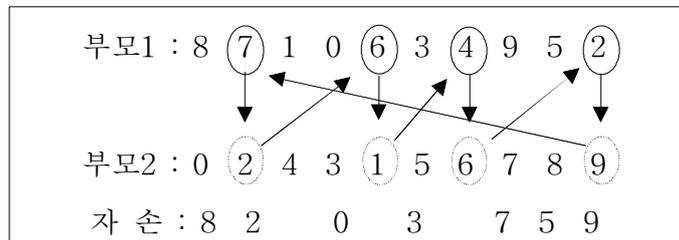
<그림 II-10> 사이클 교배 예제-1

부모1의 첫 번째 유전자 {8}로부터 복사를 시작한다. 방금 복사한 위치의 부모2 유전자는 0이므로 부모1 상에서 0이 있는 위치를 찾아 자손해의 같은 위치에 0을 복사한다. 같은 원리로 부모1상에서 3이 있는 위치를 찾아 자손해의 같은 위치에 복사한다. 마찬가지로 5가 복사된다. 이제 5와 같은 위치에는 8이 있는데 8은 이미 복사가 되었으므로 더 이상 진행 할 수가 없다. 이 과정을 아래 <그림 II-11>처럼 화살표를 따라 표시해 보면 하나의 사이클이 만들어진다는 데서 사이클 교배란 이름이 붙게 되었다.



<그림 II-11> 사이클 교배 예제-2

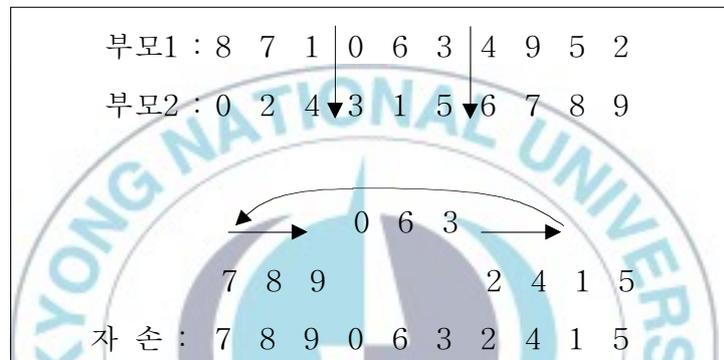
다음은 자손해의 유전자 값이 아직 결정되지 않은 위치들 중 가장 왼쪽의 위치로부터 시작하는데 이번에는 부모2로부터 시작한다. 2, 7, 9 순으로 복사하면서 아래 <그림 II-12>처럼 다시 사이클이 만들어진다. 이렇게 부모1과 부모2를 번갈아 시작해 가면서 더 이상 결정되지 않은 유전자가 없을 때까지 계속하면 최종적인 자손해는 {8 2 1 0 6 3 4 7 5 9}가 된다.



<그림 II-12> 사이클 교배 예제

(5) 순서교배

순서 교배 역시 염색체가 순열로 표시되는 경우를 위하여 고안된 교배 연산자이다. 먼저 임의로 두 개의 자름선을 정한 다음 두 자름선 사이에 있는 부분을 부모1로부터 복사한다(0 6 3). 나머지 위치는 부모2로부터 복사하되, 두 번째 자름선 바로 다음 위치부터 시작하여 사용되지 않은 기호들 중 부모2에서 나타난 순서대로 {2 4 1 5 7 8 9}를 복사한다.



<그림 II-13> 순서 교배 예제

마. 돌연변이(mutation)

진화가 계속되는 동안 재생산과 교배 연산을 통해 개체집단을 더욱 강하게 해주고 이로 인하여 염색체들은 서로 닮아가게 된다. 이런 현상은 말기 세대에서는 바람직하지만 초기 세대에 발생하게 되면 지역 최적해에 빠질 가능성이 많다. 따라서 재생산과 교배 연산을 통한 개체집단 내에서의 염색체 진화에는 한계가 있다. 다시 말해서 개체집단 내에서의 염색체 스키마(schema)를 극복할 수 없는 경우가 생길 수 있다. 그 예로, 개체집단 내 염색체들의 특정위치의 비트가 모두 같게 되는 경우 그러한 현상이 생길 수 있다.

부모 염색체 1 : 01110
부모 염색체 2 : 10000
부모 염색체 3 : 11010
부모 염색체 4 : 10110

<그림 II-14> 교배의 예제

위의 <그림 II-14>에서 두 염색체가 아무리 교배를 한다고 하더라도 5 번째 형질의 “0”은 변화하지 않으므로 ****1이라는 염색체는 생성될 수가 없다. 이를 보완하기 위한 연산이 돌연변이 연산이다.

따라서 돌연변이 연산자는 염색체의 기존 형질과는 다른 새로운 형질을 만듦으로서 모든 염색체의 특정 비트가 고정되는 것을 방지하고 탐색공간을 넓히려는 시도이다. 이 돌연변이 연산은 해가 어느 정도 수렴되었을 때 중요하게 작용하는 연산이다. 돌연변이 발생 확률은 가변적으로 쓰일 수 있으며 보통 10~30%의 확률을 적용한다(진강규, 2000).

3. 타부서치

타부서치는 1960년대 후반과 1970년대 초반에 그 기원을 두고 있으며, Glover(1977)에 의해 현재의 형태로 제시된 후 많은 조합 최적화문제에 적용되고 있다.

조합 최적화문제에서 기존 알고리즘의 기본원리는 개선하는 방향으로만 진행하는 것이다. 이것의 한계점은 더 개선할 수 없어 멈출 때 지역 최적해를 얻으므로, 이것이 전체 최적해라는 보장이 없다는 것이다. 타부서치는 개선하는 움직임을 한쪽 방향으로만 고정하지 않고 이미 지나왔던 지역 최적해에 다시 빠지는 일 없이 탐색을 계속하게 휴리스틱을 인도한다. 타부서치는 조합적 문제를 풀기위한 메타전략으로 볼 수 있다(김여근 등, 1997).

타부서치는 트리형 네트워크 설계 문제에도 적용되어 우수한 성능을 발휘하는 것으로 알려져 있다(Sharaiha et al., 1997). 타부서치는 유전자 알고리즘과 마찬가지로 상위수준의 발견적 해법(meta heuristic)으로 다른 발견적 해법에서 사용할 수 있고 다른 발견적 해법을 제어할 수 있다.

타부서치 과정에서 중요한 점은 타부리스트를 사용함으로써 탐색접근은 '제한적인 탐색'을 하게 되고 생성된 해들은 타부리스트의 내용에 결정적으로 의존한다는 것이다.

타부서치의 가장 큰 특징은 타부리스트라고 할 수 있는데 타부란 일종의 기억장치로 해의 순환이 일어나는 것을 방지하는 것이다. 타부리스트는 타부 속성이 기록된 목록이다. 타부 속성은 해의 탐색 동안 금지되어야 할 내용을 의미한다(김여근 등, 1997).

본 연구에서는 유전자 알고리즘에 타부리스트를 결합한 새로운 알고리즘을 이용하여 트리형 네트워크를 설계한다.

Ⅲ. 타부리스트가 결합된 유전자 알고리즘을 이용한 트리형 네트워크 설계

1. 트리형 네트워크 설계 문제의 기술

Ⅱ장에서 살펴본 바와 같이 일반적인 네트워크 설계 문제 모델에 모든 고려사항을 포함시켜 하나의 모델로 설계하는 것은 쉽지 않다.

본 연구에 적용한 문제 모델을 다음과 같이 정의한다.

1) 주어진 조건	노드의 개수 각 노드에 대한 데카르트 좌표 각 노드의 트래픽 용량
2) 목적함수	총 링크비용의 최소화
3) 제약조건	서브트리의 최대 트래픽 용량
4) 결정변수	토폴로지 서브트리 서브트리 최대 트래픽 용량

본 연구에서 제안하는 트리형 네트워크 설계 문제와 이를 풀기 위한 TLCGA를 설명하기 위해 필요한 가정 및 표기는 다음과 같다.

가. 트리형 네트워크 설계에 대한 가정 및 표기

(1) 트리형 네트워크 설계를 위한 가정

- 각 노드(허브 또는 터미널)의 위치는 주어지 있다.
- 노드는 완전히 신뢰할 수 있다.
- 각 링크는 무방향이다.
- 노드의 메모리는 무한하다.
- 네트워크에 있어 중복된 링크는 존재하지 않는다.

(2) 트리형 네트워크 설계를 위한 표기 형식

- n : 전체 노드의 수
- s_i : i 번째 노드, $i = 1, 2, 3, \dots, n$
- s_{ij} : s_i 와 s_j 간 링크
- d_{ij} : s_i 와 s_j 간 거리 또는 s_i 와 s_j 간 비용
- T_j : j 번째 서브트리($s_i \in T_j$)
- $C(x)$: 총 설계비용
- q_i : i 번째 노드의 트래픽 용량
- K : 하나의 서브트리에서 전송 가능한 최대 트래픽 용량
- u_i : i 번째 노드의 차수(degree)

(3) TLCGA에 대한 표기 형식

- g : 세대 수
- g_{\max} : 최대 세대수

- s : 개체집단 크기
- $m\%$: 각 개체집단에 발생하는 돌연변이 비율
- r_m : 염색체내의 돌연변이 비율
- T_s : 타부리스트의 크기

나. 트리형 네트워크 설계 문제 모형

트리형 네트워크 설계 문제는 아래와 같이 모형화 될 수 있다.

목적함수 :

$$\text{Min } C(x) = \sum_{i=0}^{n-1} \sum_{j=1}^n d_{ij} s_{ij} \quad \text{-----} \quad (1)$$

제약조건 :

$$s_{ij} = \begin{cases} 1, & \text{if } (i, j) \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \quad \text{-----} \quad (2)$$

$$\sum_{i=0}^{n-1} \sum_{j=1}^n s_{ij} = n - 1 \quad \text{-----} \quad (3)$$

$$\sum_{i=0}^n s_{ij} \geq 1, j = 1, 2, \dots, n \quad \text{-----} \quad (4)$$

$$\sum_{s_i \in T_j} q_i \leq K \quad \text{-----} \quad (5)$$

목적함수인 식(1)은 노드 간 거리의 합을 최소화를 의미한다. 본 연구에서 노드 간의 거리는 곧 노드 간 링크 설치비용의 최소화를 나타낸다. 즉 트리 네트워크를 구성하는 총 설계비용의 최소화를 의미한다.

제약조건 식(2)에서 1은 노드 s_i 와 노드 s_j 간 링크가 연결되었음을 의미하고 0은 연결되지 않았음을 의미한다.

제약조건 식(2)는 모든 스패닝 트리의 제약조건으로 n 개의 노드를 가지는 신장 트리는 $n-1$ 개의 링크를 가져야 한다는 것을 의미한다.

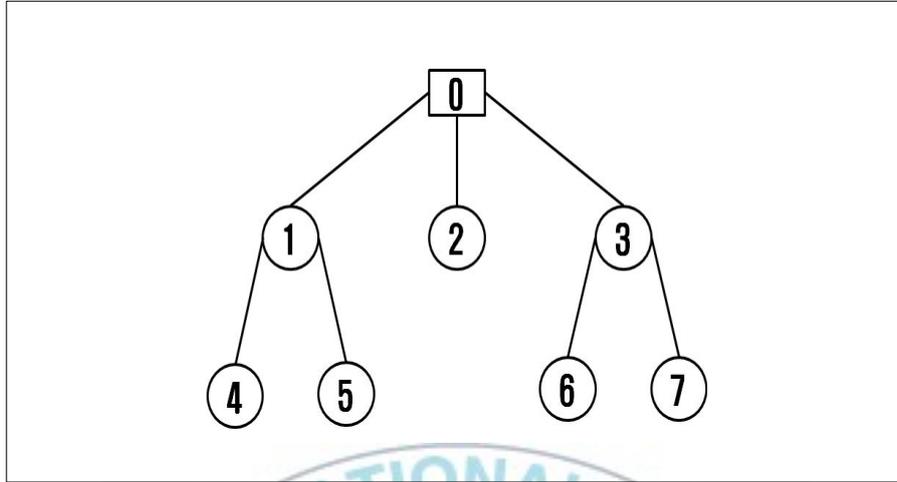
제약조건 식(3)은 각각의 노드는 반드시 하나 이상의 다른 노드와 연결되어야 함을 의미한다.

제약조건 식(4)는 루트노드(s_0)의 어떤 서브트리(T_j)에 속하는 노드들의 트래픽 용량의 합이 최대 트래픽(K)이하가 되어야 한다는 것을 의미한다. 특히, q_i 의 값이 전부 1인 경우, 어떠한 서브트리에도 K 개 이하의 노드가 연결되어야 한다는 것을 의미한다.

2. 트리형 네트워크 설계의 염색체 표현방법

유전자 알고리즘에 문제를 적용하기 위해서는 문제의 부호화가 필요하며 문제를 부호화한 것이 염색체이다. 이 염색체는 유전자 알고리즘에서 가장 기본이 되는 것으로 이것을 기본 단위로 하여 유전 연산을 하게 된다. 염색체의 표현은 문제에 의존적이므로 염색체는 각 해의 특성을 표시할 수 있어야 한다. 트리형 네트워크 설계 문제는 이진 코딩으로 쉽게 변환할 수 있다.

<그림 III-1>은 1개의 루트노드(사각형)에 7개의 노드를 연결하는 트리형 네트워크 설계 구조의 예시를 보여주고 있다.



<그림 III-1> 트리형 네트워크 설계 구조의 예

<그림 III-1>과 같은 네트워크 구조를 가정했을 경우, 링크를 인접행렬로 나타내면 아래 <표 III-1>와 같다.

<표 III-1> 링크의 인접행렬

노드 번호	0	1	2	3	4	5	6	7
0	-	1	1	1	0	0	0	0
1		-	0	0	1	1	0	0
2			-	0	0	0	0	0
3				-	0	0	1	1
4					-	0	0	0
5						-	0	0
6							-	0
7								-

염색체는 인접행렬표의 위 삼각형의 각 행을 가로방향으로 연결한 것이 된다. 염색체의 표현 예는 다음 <그림 III-2>와 같다.

이진 염색체 : {11100000011000000000110005000}

<그림 III-2> 이진 염색체의 표현 예제

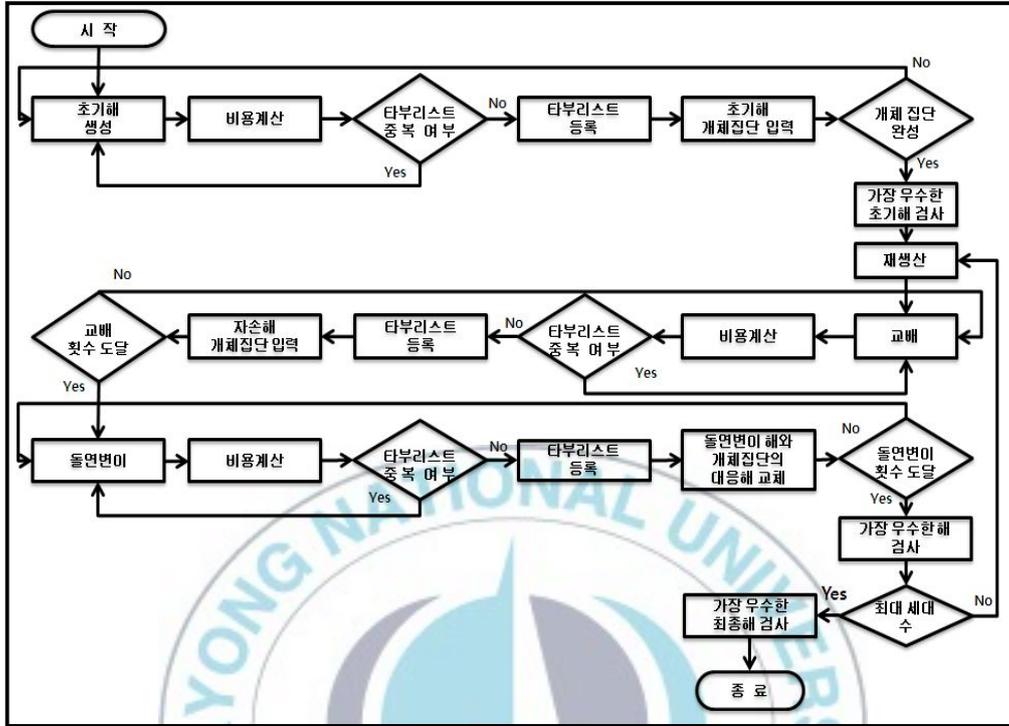
예제의 트리형 네트워크 설계 문제에 대한 해의 경우의 수는 $x^{n(n-1)/2}$ 로 나타낼 수 있다. x 는 링크타입의 개수이고 n 은 노드의 수이다. 따라서 예시의 해의 경우의 수는 $2^{8(8-1)/2} = 268,435,456$ 개이다.

3. 트리형 네트워크 설계

본 연구에서 제안하는 TLCGA를 실행하기 위해 기존의 GA 설계 절차 (염창선 등, 2005)에 TS의 Tabu List 기능을 결합하여 다음과 같이 트리형 네트워크 설계 절차를 제시한다.

가. 트리형 네트워크 설계를 위한 주요 절차

TLCGA를 이용한 트리형 네트워크 설계의 주요 절차는 <그림 III-3>과 같다.



<그림 III-3>타부리스트가 결합된 유전자 알고리즘을 이용한 트리형 네트워크 설계 절차

1) 파라미터 초기화(initialize parameters)

- g : 세대 수
- g_{\max} : 최대 세대수
- s : 개체집단 크기
- $m\%$: 각 개체집단에 발생하는 돌연변이 비율
- r_m : 염색체내의 돌연변이 비율
- Ts : 타부리스트의 크기

2) 초기 개체집단 생성($g=1$)

- (1) 제약조건을 만족하는 초기해를 무작위로 생성한다.
- (2) 초기해를 비용계산함수로 보낸다.
- (3) 초기해를 타부 리스트로 보낸다.
- (4) 타부 리스트가 비어 있다면, 초기해를 타부 리스트에 등록한다.
타부 리스트가 비어 있지 않다면, 타부 리스트와 중복되는 초기해는 버리고 중복되지 않는 초기해는 타부 리스트의 해보다 우수할 경우 타부 리스트에 등록한다.
- (5) 타부리스트와 중복되지 않는 초기해는 초기 개체집단에 넣는다.
- (6) 초기 개체집단(s)이 만들어질 때까지 (1)-(5)를 반복한다.
- (7) 초기 개체집단 중 가장 우수한 해를 검사한다.

3) 교배 및 돌연변이 연산

a) 교배

- (1) 순위기반 선택법에 따라 현재 개체집단으로부터 서로 다른 두 부모해가 선택된다.
- (2) 균등교배를 통해 자손해가 생성된다.
- (3) 자손해를 비용계산 함수에 보낸다.
- (4) 자손해를 타부 리스트로 보낸다.
- (5) 타부 리스트와 중복되는 자손해는 버리고 중복되지 않는 자손해는 타부 리스트의 해보다 우수할 경우 타부 리스트에 등록한다.
- (6) 타부리스트와 중복되지 않는 자손해는 개체집단에 저장된다.

(7) 개체집단(s)이 만들어질 때까지 (1)-(6)를 반복한다.

b) 돌연변이

- (1) 개체집단으로부터 돌연변이 대상 해를 무작위로 선택한다.
- (2) 돌연변이 대상 해에 대해 돌연변이 비율(r_m)에 따라 돌연변이 연산을 한다.
- (3) 돌연변이 해를 비용계산 함수에 보낸다.
- (4) 돌연변이 해를 타부 리스트로 보낸다.
- (5) 타부 리스트와 중복되는 돌연변이 해는 버리고 중복되지 않는 돌연변이 해는 타부 리스트의 해보다 우수할 경우 타부 리스트에 등록한다.
- (6) 타부리스트와 중복되지 않는 돌연변이 해는 개체집단의 대응하는 해와 교체한다.
- (7) 돌연변이 횟수($= m\% \times s$) 만큼 (1)-(6)을 반복한다.
- (8) 개체집단(s) 중 가장 우수한 해를 검사한다.

C) 교배 및 돌연변이 연산 반복

- (1) 최대 세대 수($g = g_{max}$)까지 교배 및 돌연변이 연산을 반복한다.
- (2) 가장 우수한 최종해를 확인한다.

나. 초기 개체집단 생성

초기 세대의 개체집단을 구성을 위해 제약조건을 만족하는 노드 간 링크를 무작위로 결정한다. <그림 III-4>은 루트 노드가 1개이고 노드가 7개이고 모든 노드의 트래픽 용량이 1인 네트워크 설계의 경우 신장 트리를 형성하고 서브트리 트래픽 용량 제약조건 $k=3$ 을 만족하는 네트워크 구조의 염색체를 나타내고 있다.

초기 염색체 : { 1110000001100000000011000000 }

<그림 III-4 > 초기 염색체

다. 교배 및 돌연변이

교배 연산을 위해 서로 다른 두 개의 부모 염색체를 순위기반 선택(rank based quadratic procedure)을 적용하여 선택한다(Tate et al., 1998). 두 부모 염색체 내의 각각의 형질에 대해 0과 1사이의 난수를 발생시켜 임계확률 0.5 이상인 경우 부모 1의 형질이 유전되고 그 외의 형질에 대해서는 부모 2의 형질이 자손에게 유전된다(균등교배). 예를 들어, 아래 <그림 III-5>와 같이 두 부모가 선택되었고, 각 형질에 난수가 발생되었다고 가정하면, 자손의 발생과정은 아래와 같다.

부모 1 :	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	
난 수 :	3	9	4	5	6	2	5	1	5	3	8	6	7	2	4	1	3	6	7	8	8	3	8	2	1	6	2	5
부모 2 :	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	
자 손 :	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	

<그림 III-5> 균등교배 예제

위의 예제에서 난수가 임계확률 0.5 이상인 2, 4, 5, 7, 9, 11, 12, 13, 18, 19, 20, 21, 23, 26, 28번째 형질에 대해서는 부모 1의 형질을 자손에게 물려주고 그 외의 형질에 대해서는 부모 2의 형질을 자손에게 물려주어 자손 염색체를 만든다.

위 과정을 통해 얻어진 자손 염색체는 타부 리스트에 저장되며 이후에 동일한 염색체가 나왔을 경우 그 해는 선택하지 않으며 타부 리스트와 비교해서 해가 다를 경우에만 선택되어 진다.

그리고 돌연변이 확률($r_m = 0.15$)에 따라 28개의 형질 중에서 4개의 형질이 선택되고 선택된 형질의 값이 0이면 1로, 1이면 0으로 바뀐다. 위 예에서 자손 염색체의 11, 16, 17, 21번째 형질이 돌연변이 확률에 의해 선택 되어 졌다고 가정하면 자손의 염색체의 돌연변이 된 이후의 염색체는 아래 <그림 III-6>와 같다.

돌연변이 후 염색체 : { 1110000001100000100001000000 }

<그림 III-6> 돌연변이 후 염색체

위 과정을 통해 얻어진 해는 타부 리스트에 저장되며 이후에 동일한 해가 나왔을 경우 그 해는 선택하지 않으며 타부 리스트와 비교해서 해가 다를 경우에만 선택되어 진다.

라. 서브트리 탐색

루트노드의 서브트리(T_j)를 찾고 서브트리의 트래픽 총량을 계산하기 위해서 먼저 각 노드들의 차수(u_i)를 구한다. 차수는 각각의 노드가 다른 노드와 몇 개의 링크를 가지고 있는가를 의미한다. <그림 III-1> 네트워크 구조의 차수를 표현하면 아래 <그림 III-7>과 같다.

차수 : { 3 3 1 3 1 1 1 1 }

<그림 III-7> 서브트리 탐색 전 노드의 차수

루트노드를 제외한 노드들의 트래픽 용량이 1 이라고 가정했을 경우 각 노드의 트래픽 용량을 표현하면 아래 <그림 III-8>과 같다.

트래픽 용량 : { 0 1 1 1 1 1 1 1 }

<그림 III-8> 서브트리 탐색 전 노드의 트래픽 용량

먼저 루트노드를 제외하고 차수가 1인 노드를 검색한다. 트리형 네트워크에서 차수가 1인 노드는 항상 leaf-노드이다. 예제의 경우 2, 4, 5, 6, 7번

노드가 leaf-노드이다.

leaf-노드와 연결된 상위 노드(루트노드는 제외)를 검색하고 상위 노드에 leaf-노드의 트래픽을 전송하고 노드의 연결을 제거한다.

예제에서는 {1, 4, 5}, {2}, {3, 6, 7} 3개의 서브트리가 존재한다.

leaf-노드의 트래픽을 상위 노드로 전송하고 노드의 연결을 제거한 후의 차수와 트래픽 용량은 아래 <그림 III-9>과 같다.

차 수 : { 3 1 1 1 0 0 0 0 }
트래픽 용량 : { 0 3 1 3 0 0 0 0 }

<그림 III-9> 서브트리 탐색 후 차수와 트래픽 용량

위 예제의 경우 한 번의 계산으로 서브트리 탐색이 가능하지만 노드의 수가 증가하여 계산이 복잡해질 경우 이 과정을 전체 노드의 수 / 루트노드의 차수(n / u_0 , 예제의 경우 $8 / 3 = 2.67$ 이므로 2번 반복)만큼 반복하여 서브트리를 탐색할 수 있다. 서브트리의 트래픽 용량의 합이 최대 트래픽 용량(K)이하일 경우 후보해가 된다.

루트노드를 제외한 노드들의 트래픽 용량이 서로 다른 경우도 동일한 방법으로 서브트리 탐색이 가능하다. 노드의 트래픽 용량이 서로 다른 경우도 서브트리의 탐색 과정은 트래픽 용량이 1인 경우와 동일하지만 서브트리 탐색 전 각 노드의 트래픽 용량이 <그림 III-10>와 같이 { 0 2, 3, 1, 1, 2, 2, 2 }일 때, 서브트리 탐색 후 각 서브트리의 트래픽 용량은 <그림 III-11>와 같이 { 0 5, 3, 5, 0, 0, 0, 0 }로 계산되며, 서브트리의 트래픽 용량의 합이 최대 트래픽 용량(K)이하일 경우 후보해가 된다.

트래픽 용량 : { 0 2, 3, 1, 1, 2, 2, 2 }

<그림 III-10> 서브트리 탐색 전 노드의 트래픽 용량

트래픽 용량 : { 0 5, 3, 5, 0, 0, 0, 0 }

<그림 III-11> 서브트리 탐색 후 노드의 트래픽 용량

마. 타부리스트

타부리스트는 재생산, 교배 및 돌연변이 연산으로 선택된 해의 중복을 제거하기 위한 기능이다. 유전자 알고리즘은 연산과정 중 선택된 개체집단 내에 동일한 해가 존재할 가능성이 있다.

재생산, 교배 및 돌연변이 연산을 통해 선택된 해는 타부리스트에 저장된다. 재생산, 교배 및 돌연변이 연산을 거쳐 새로운 해가 추가되면 타부리스트와 비교하여 중복 해인지 검사하고 중복 해인 경우 해를 버리고, 중복 해가 아닌 경우 후보해가 되고 타부리스트에 타부리스트의 크기(T_s)만큼 저장된다.

타부리스트의 크기를 초과할 경우 비용이 가장 큰 해를 삭제하고 새로운 해를 추가하여 타부리스트의 크기를 유지한다.

4. 트리형 네트워크 설계를 위한 알고리즘

트리형 네트워크 설계를 위한 주요 알고리즘은 다음과 같다.

가. 노드 차수 계산 알고리즘

input : 초기해 생성 또는 교배 및 돌연변이 연산에서 얻은 노드 간

링크 s_{ij}

output : 각 노드의 차수 u_i

variable : u_i - 각 노드의 차수

Process :

step 1 : /* 초기화 */

$u_i \leftarrow 0$

step 2 : /* 차수 계산 */

if $s_{ij} > 0$

노드 i 의 차수 증가

노드 j 의 차수 증가

나. 서브트리 탐색 및 트래픽 계산 알고리즘

input : 초기해 생성 또는 교배 및 돌연변이 연산에서 얻은 노드 간

링크 s_{ij} , 차수 계산에서 얻은 각 노드의 차수 u_i

output : 서브트리의 총 트래픽 $\sum_{s_i \in T_j} q_i$

variable : q_i - 각 노드의 트래픽

n - 전체 노드의 수

process :

step 1 : /* 서브트리 탐색 및 서브트리 총 트래픽 계산 */

while(n / u_0){

① $u_i = 1$ 인 노드 검색

② $u_i = 1$ 인 노드에 연결된 링크($s_{ij} = 1$) 검색

③ 노드 i 의 트래픽을 노드 j 로 전송

④ 링크 s_{ij} 의 연결 제거

⑤ 노드 i 의 트래픽 초기화

};

다. 타부리스트 관리 알고리즘

input : 초기해 생성 또는 교배 및 돌연변이 연산에서 얻은 노드 간 링크 s_{ij} , 네트워크 총 설계비용 $C(x)$

output : 노드 간 링크 s_{ij} , 네트워크 총 설계비용 $C(x)$

variable : T_s - 타부리스트의 크기

num_tabu - 현재 타부리스트 갯수

process :

step 1 : /* 타부리스트 정렬 */

① 타부리스트에 저장된 목록을 $C(x)$ 가 낮은 순부터 올림차순 정렬

step 2: /* 타부리스트 생성 및 유지 */

if (num_tabu < T_s)

① input의 s_{ij} , $C(x)$ 를 타부리스트에 저장

else

① input의 $C(x)$ 가 타부리스트의 $C(x)$ 보다 크거나 같으면
해를 버림

② 타부리스트의 어떤 $C(x)$ 보다 input의 $C(x)$ 가 작으면
input의 s_{ij} , $C(x)$ 와 타부리스트의 가장 큰 $C(x)$ 와 교체



IV. 성능 평가

트리형 네트워크 설계의 실험을 위한 컴퓨터의 시스템 사양은 Pentium Core2Duo(2.2GHz), 1GB RAM이고 컴파일러로 Visual C++ 6.0가 사용되었다.

앞서 기술한 바와 같이 네트워크 설계 문제는 NP-hard 문제로 지수형태의 복잡도를 가진다. 따라서 상대적으로 탐색공간의 규모가 작은 문제 $n=13$ 이고 각 노드의 트래픽 용량이 모두 1이며 서브트리 총 트래픽 용량이 3 ($K=3$) 인 문제 1에 대하여 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 성능 비교를 통해 효율성을 증명한다.

그리고 좀 더 현실적인 네트워크 설계 상황을 고려하여 동일한 문제에서 각 노드의 트래픽 용량이 서로 다른 경우, 서브트리 총 트래픽 용량이 10 ($K=10$) 인 문제 2에 대하여 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 성능 비교를 통해 효율성을 증명한다.

문제 3($n=17$)을 통해 노드의 추가에 따른 급격히 증가하는 탐색공간에 대해서 각 노드의 트래픽 용량이 1이고 서브트리 총 트래픽 용량이 5 ($K=5$)인 경우 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 비교평가를 통해 효율성을 증명한다.

문제 4($n=17$)에 대해서도 각 노드의 트래픽 용량이 서로 다른 경우, 서브트리 총 트래픽 용량이 12($K=12$)인 경우 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 성능 비교를 통해 효율성을 증명한다.

그리고 보다 큰 문제로 노드의 수가 21개($n=21$)이고 각 노드의 트래픽 용량이 1이고 서브트리 총 트래픽 용량이 6($K=6$)인 문제 5, 노드의 수가

21개($n=21$)이고 각 노드의 트래픽 용량이 서로 다른 경우 서브트리 총 트래픽 용량이 15($K=15$)에 문제 6을 통해 타부리스트가 결합된 유전자 알고리즘의 효율성을 증명한다.

타부리스트가 결합된 유전자 알고리즘의 평가를 위해 <표 IV-1>과 같이 노드 좌표를 설정하였으며 유클리디안 식을 이용하여 노드간의 거리를 계산한다. 그리고 노드 간의 거리 자체를 링크 비용(d_{ij})으로 간주하며 그 값은 대칭으로 가정하며 단위는 '\$' 이다.

<표 IV-1> 노드의 좌표

노드	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
가로 좌표	50	90	59	92	94	70	80	35	37	19	23	3	9	75	96	76	14	38	27	19
세로 좌표	50	3	63	8	31	20	45	9	59	17	90	20	85	76	36	13	33	87	10	73

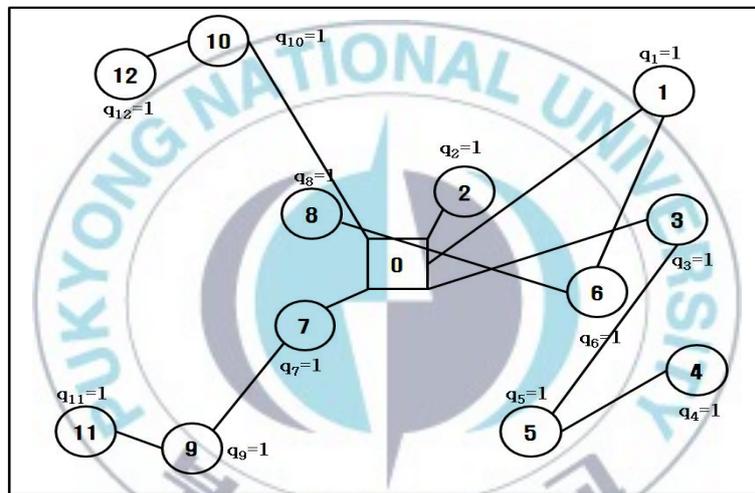
(주) 문제 1은 노드 번호 0 ~ 12로 한정한다.

1. 문제 1 : 13개의 노드, 서브트리 트래픽 용량 3

노드가 13개이고 각 노드의 트래픽 용량이 모두 1인 경우를 네트워크로 연결하는 문제 1에 대해 타부리스트가 결합된 유전자 알고리즘을 이용한 트리형 네트워크 설계를 수행하였다. 초기 탐색 실험을 통해 초기화 값 ($g_{max} = 100$, $s = 100$, $m\% = 0.3$, $r_m = 0.2$, $Ts = 100$)을 얻은 후 앞서 제시한 설계 절차에 따라 실험을 수행하였다.

그 결과 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었으며, <그림 IV-2>에 나타나 있는 것과 같은

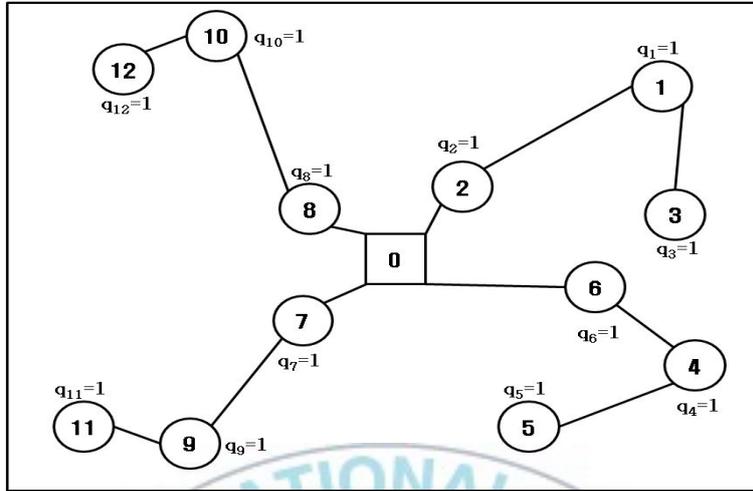
설계 구조를 갖는 비용이 281.17\$ 인 가장 우수한 해를 찾았다. <그림 IV-1>은 초기 가장 우수한 해의 네트워크 설계 구조를 나타내고 있다. 반면 유전자 알고리즘을 이용한 설계 결과 최적해의 비용은 335.69\$였다. 타부리스트가 결합된 유전자 알고리즘을 이용한 초기 가장 우수한 해와 가장 우수한 최종해의 서브트리 및 트래픽 용량은 <표 IV-2>, <표 IV-3>과 같다.



<그림 IV-1> 초기 가장 우수한 해(문제 1 : 390.46\$)

<표 IV-2> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 1)

서브트리	{2}	{1, 6, 8}	{3, 5, 4}	{7, 9, 11}	{10, 12}
트래픽 용량	1	3	3	3	2



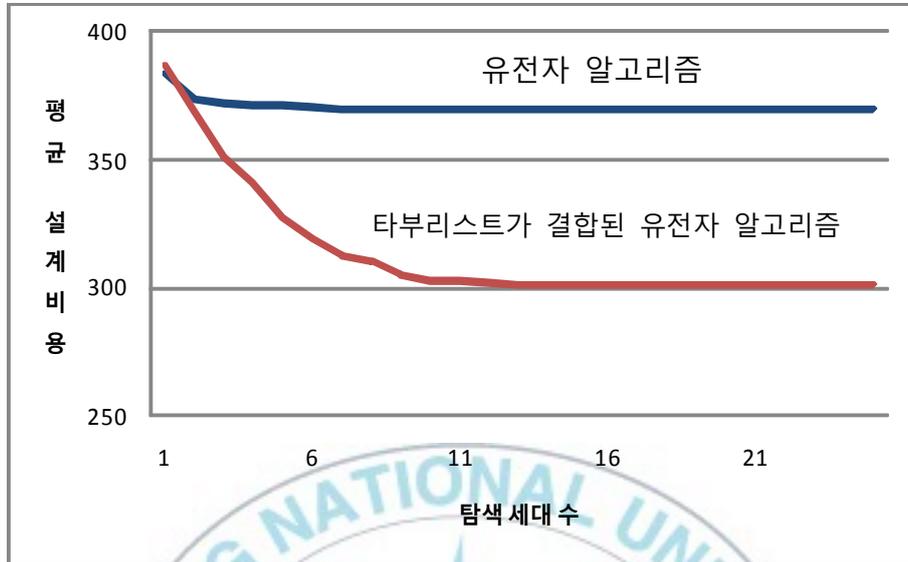
<그림 IV-2> 가장 우수한 최종해(문제 1 : 281.17\$)

<표 IV-3> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 1)

서브트리	{2, 1, 3}	{6, 4, 5}	{7, 9, 11}	{8, 10, 12}
트래픽 용량	3	3	3	3

그리고 신뢰성 있는 분석을 위하여 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘을 이용한 설계 방법 각각에 대해 서로 다른 10가지의 random number seed에 적용하여 문제를 풀었다.

그 결과 10가지의 경우 모두 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었다. 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색을 통해 도출된 10가지 seed의 해에 대해 탐색 세대 수의 증가에 따른 평균 설계비용의 추이를 살펴보면 <그림 IV-3>와 같다.



<그림 IV-3> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 1)

타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법은 탐색 세대가 증가함에 따라 평균 설계비용이 급격히 떨어져 13세대에서 약 300\$로 수렴하는 반면에 유전자 알고리즘을 이용한 탐색 방법은 9세대에 이르기까지 7번의 설계비용 감소를 가졌으며, 9세대에서 평균 설계비용 369.501\$를 가져 약 18% 개선 효과를 보여주고 있다.

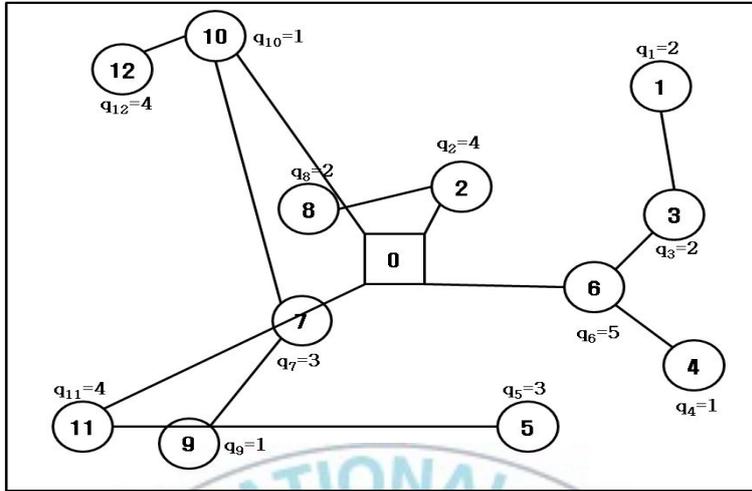
이 결과는 결합된 유전자 알고리즘을 이용한 설계 방법이 유전자 알고리즘을 이용한 탐색 방법보다 효율적인 설계 방법이라는 것을 보여 주고 있다.

2. 문제 2 : 13개의 노드, 서브트리 트래픽 용량 10

실제 네트워크 설계 과정에서 모든 노드의 트래픽 용량이 동일한 경우는 존재하지 않는다. 좀 더 현실적인 상황에서 제안하는 타부리스트가 결합된 유전자 알고리즘의 성능을 평가하기 위하여 13개의 노드를 네트워크로 연결하는 문제 1에서 다른 모든 초기화 값이 동일하나 각 노드의 트래픽 용량이 서로 다른 문제 2에 대해 동일한 설계 절차에 따라 실험을 수행하였다.

각 노드의 트래픽 용량은 1~5 사이의 random number를 적용하여 문제를 풀었다.

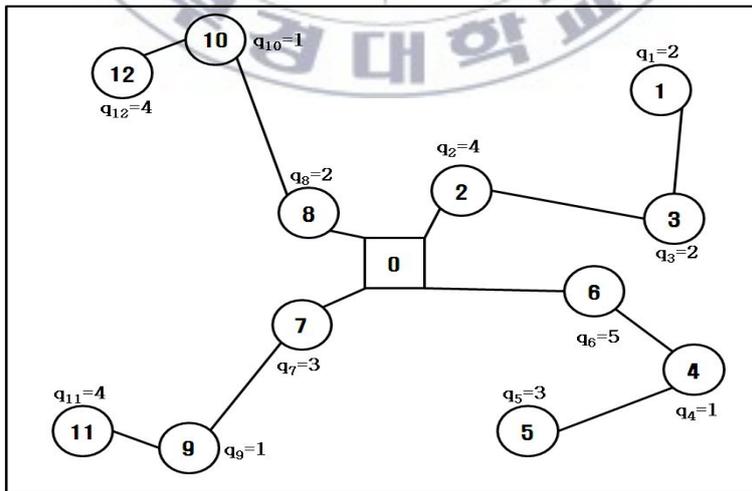
그 결과 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었으며, <그림 IV-5>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 277.66\$인 가장 우수한 최종해를 찾았다. <그림 IV-4>는 초기 가장 우수한 해의 네트워크 설계 구조를 나타내고 있다. 반면 유전자 알고리즘을 이용한 설계 결과 최적해의 비용은 316.01\$였다. 타부리스트가 결합된 유전자 알고리즘을 이용한 초기 가장 우수한 해와 가장 우수한 최종해의 서브트리 및 트래픽 용량은 <표 IV-4>, <표 IV-5>와 같다.



<그림 IV-4> 초기 가장 우수한 해(문제 2 : 396.64\$)

<표 IV-4> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 2)

서브트리	{2, 8}	{6, 3, 1, 4}	{11, 5}	{10, 12, 7, 9}
트래픽 용량	6	10	7	9



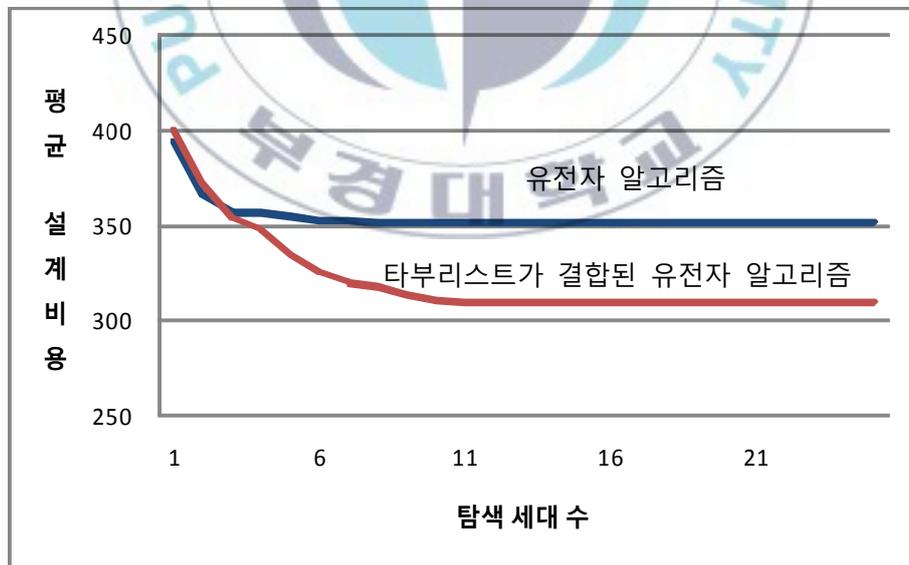
<그림 IV-5> 가장 우수한 최종해(문제 2 : 277.66\$)

<표 IV-5> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 2)

서브트리	{ 2, 1, 3 }	{ 6, 4, 5 }	{ 7, 9, 11 }	{ 8, 10, 12 }
트래픽 용량	8	9	8	7

문제 2에 대해서도 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘을 이용한 설계 방법 각각에 서로 다른 10가지의 random number seed에 적용하여 문제를 풀었다.

그 결과 10가지 경우 모두 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었다. 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색을 통해 도출된 10가지 seed의 해에 대해 탐색 세대 수의 증가에 따른 평균 설계비용의 추이를 살펴보면 <그림 IV-6>와 같다.



<그림 IV-6> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 2)

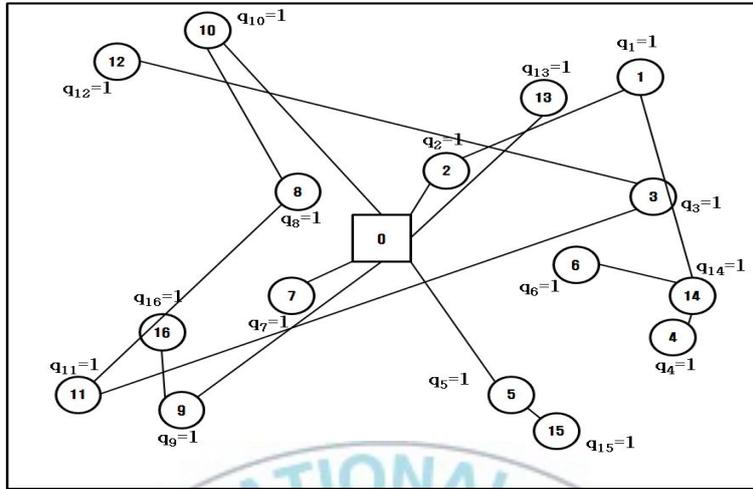
타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법은 탐색 세대 수가 증가함에 따라 평균 설계비용이 급격히 떨어져 13세대에서 약 310\$로 수렴하는 반면에 유전자 알고리즘을 이용한 탐색 방법은 8세대에 이르기까지 7번의 설계비용 감소를 가졌으며, 9세대에서 평균 설계비용 351.48\$를 가져 약 11% 개선 효과를 보여주고 있다.

이 결과는 각 노드의 트래픽 용량이 서로 다른 경우에도 타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법이 유전자 알고리즘을 이용한 탐색 방법보다 효율적인 설계 방법이라는 것을 보여 주고 있다.

3. 문제 3 : 17개의 노드, 서브트리 트래픽 용량 5

노드가 17개이고 각 노드의 트래픽 용량이 모두 1인 경우를 네트워크로 연결하는 문제 3에 대해 먼저 타부리스트가 결합된 유전자 알고리즘을 이용한 트리형 네트워크 설계를 수행하였다. 초기 탐색 실험을 통해 초기화 값($g_{max} = 100$, $s = 120$, $m\% = 0.3$, $r_m = 0.2$, $Ts = 120$)을 얻은 후 앞서 제시한 설계 절차에 따라 실험을 수행하였다.

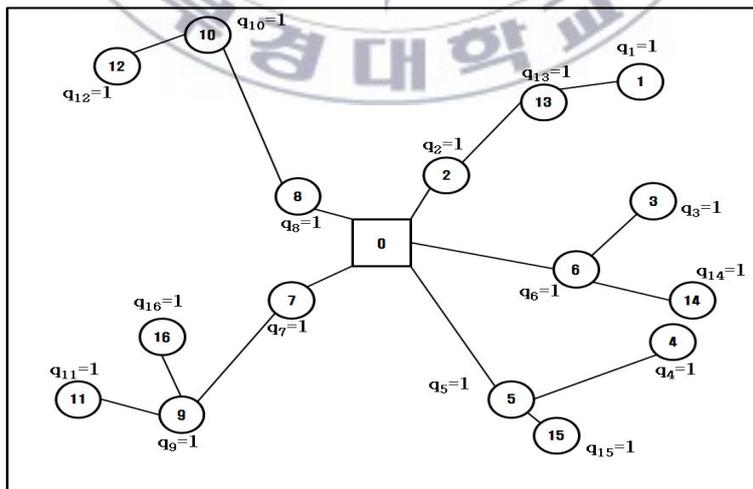
그 결과 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었으며, <그림 IV-8>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 334.66\$ 인 가장 우수한 해를 찾았다. <그림 IV-7>은 초기 가장 우수한 해의 네트워크 설계 구조를 나타내고 있다. 반면 유전자 알고리즘을 이용한 설계 결과 최적해의 비용은 539.15\$였다. 타부리스트가 결합된 유전자 알고리즘을 이용한 초기 가장 우수한 해와 가장 우수한 최종해의 서브트리 및 트래픽 용량은 <표 IV-6>, <표 IV-7>과 같다.



<그림 IV-7> 초기 가장 우수한 해(문제 3 : 603.88\$)

<표 IV-6> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 3)

서브트리	{2, 1, 14, 4, 5}	{5, 15}	{7}	{9, 16}	{10, 8, 11, 3, 12}	{13}
트래픽 용량	5	2	1	2	5	1



<그림 IV-8> 가장 우수한 최종해(문제 3 : 334.66\$)

<표 IV-7> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 3)

서브트리	{2, 13, 1}	{5, 4, 15}	{6, 3, 14}	{7, 9, 11, 16}	{8, 10, 12}
트래픽 용량	3	3	3	4	3

문제 3에서 대해서도 신뢰성 있는 분석을 위하여 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘을 이용한 설계 방법 각각에 대해 서로 다른 10가지의 random number seed에 적용하여 문제를 풀었다.

그 결과 10가지 경우 모두 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었다. 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색을 통해 도출된 10가지 seed의 해에 대해 탐색 세대 수의 증가에 따른 평균 설계비용의 추이를 살펴보면 <그림 IV-9>와 같다.



<그림 IV-9> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 3)

타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법은 탐색 세대 수가 증가함에 따라 평균 설계비용이 급격히 떨어져 18세대에서 약 410 \$로 수렴하는 반면에 유전자 알고리즘을 이용한 탐색 방법은 5세대에 이르기까지 5번의 설계비용 감소를 가졌으며, 5세대에서 평균 설계비용 560 \$를 가져 약 25% 개선 효과를 보여주고 있다.

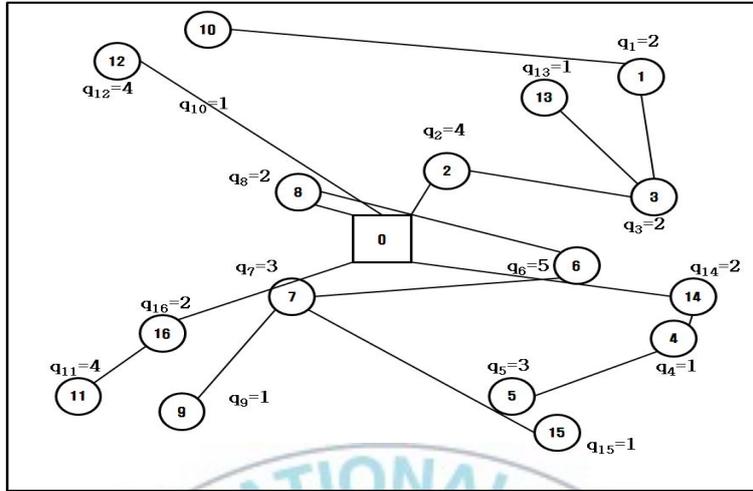
이 결과는 타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법이 유전자 알고리즘을 이용한 탐색 방법보다 효율적인 설계 방법이라는 것을 보여 주고 있다.

4. 문제 4 : 17개의 노드, 서브트리 트래픽 용량 12

문제 2와 마찬가지로 좀 더 현실적인 상황에서 타부리스트가 결합된 유전자 알고리즘의 성능평가를 위해 17개의 노드를 네트워크로 연결하는 문제 3에서 다른 모든 초기화 값이 동일하나 각 노드의 트래픽 용량이 서로 다른 문제 4에 대해 동일한 설계 절차에 따라 실험을 수행하였다.

각 노드의 트래픽 용량은 1~5 사이의 random number를 적용하였다.

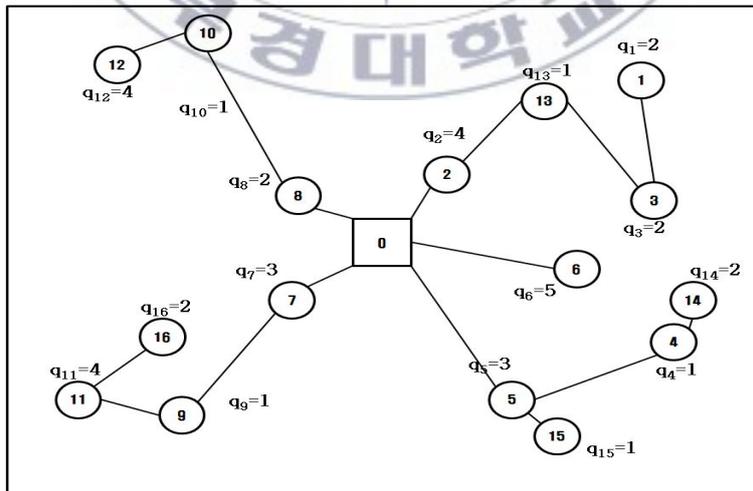
그 결과 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었으며, <그림 IV-11>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 337.55\$인 가장 우수한 해를 찾았다. <그림 IV-10>은 초기 가장 우수한 해의 네트워크 설계 구조를 나타내고 있다. 반면 유전자 알고리즘을 이용한 설계 결과 최적해의 비용은 521.38\$였다. 타부리스트가 결합된 유전자 알고리즘을 이용한 초기 가장 우수한 해와 가장 우수한 최종해의 서브트리 및 트래픽 용량은 <표 IV-8>, <표 IV-9>와 같다.



<그림 IV-10> 초기 가장 우수한 해(문제 4 : 539.19\$)

<표 IV-8> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 4)

서브트리	{2, 3, 1, 10, 13}	{8, 6, 7, 9}	{12}	{14, 4, 5}	{16, 11}
트래픽 용량	10	12	4	6	6



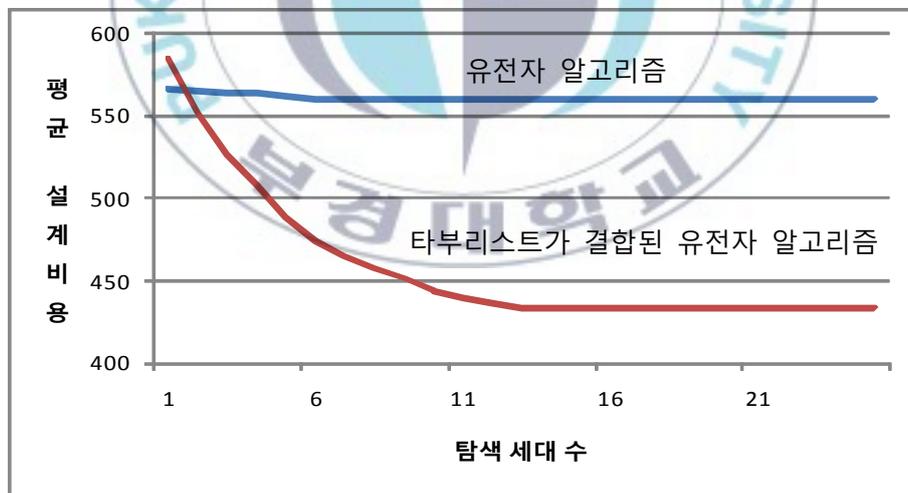
<그림 IV-11> 가장 우수한 최종해(문제 4 : 337.55\$)

<표 IV-9> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 4)

서브트리	{2, 13, 3, 1}	{5, 4, 14, 15}	{6}	{7, 9, 11, 16}	{8, 10, 12}
트래픽 용량	9	7	5	10	7

문제 4에 대해서도 신뢰성 있는 분석을 위하여 서로 다른 10가지의 random number seed에 적용하여 문제를 풀었다.

그 결과 10가지 경우 모두 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었다. 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색을 통해 도출된 10가지 seed의 해에 대해 탐색 세대 수의 증가에 따른 평균 설계비용의 추이를 살펴보면 <그림 IV-12>와 같다.



<그림 IV-12> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 4)

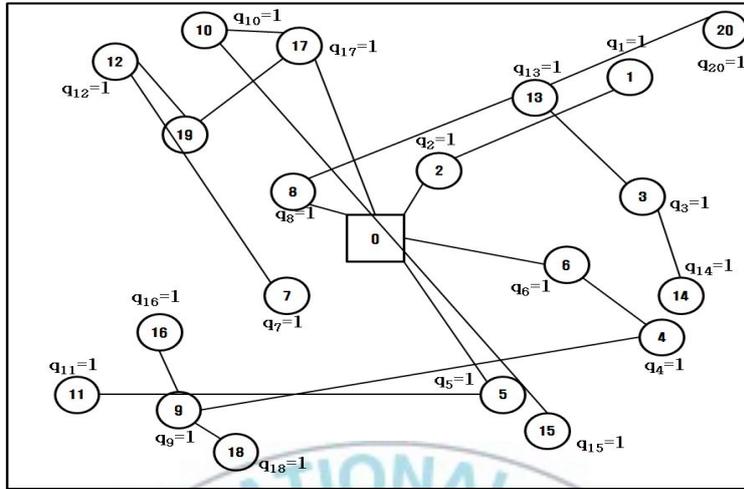
타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법은 탐색 세대가 증가함에 따라 평균 설계비용이 급격히 떨어져 14세대에서 약 430\$로 수렴하는 반면에 유전자 알고리즘을 이용한 탐색 방법은 6세대에 이르기까지 5번의 설계비용 감소를 가졌으며, 6세대에서 평균 설계비용 560\$를 가져 약 23% 개선 효과를 보여주고 있다.

이 결과는 각 노드의 트래픽 용량이 서로 다른 경우에도 타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법이 유전자 알고리즘을 이용한 탐색 방법보다 효율적인 설계 방법이라는 것을 보여 주고 있다.

5. 문제 5 : 21개의 노드, 서브트리 트래픽 용량 6

노드가 21개이고 각 노드의 트래픽 용량이 모두 1인 경우를 네트워크로 연결하는 문제 5에 대해 먼저 타부리스트가 결합된 유전자 알고리즘을 이용한 트리형 네트워크 설계를 수행하였다. 초기 탐색 실험을 통해 초기화 값($g_{max} = 100$, $s = 150$, $m\% = 0.3$, $r_m = 0.2$, $Ts = 150$)을 얻은 후 앞서 제시한 설계 절차에 따라 실험을 수행하였다.

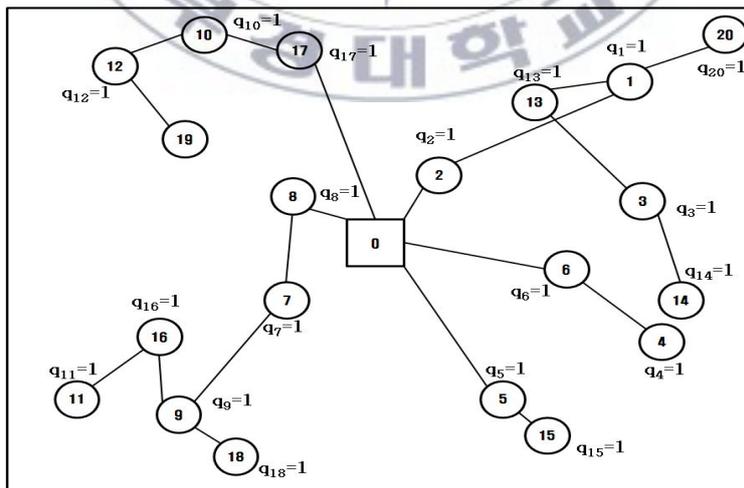
그 결과 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었으며, <그림 IV-14>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 416.12\$ 인 가장 우수한 해를 찾았다. <그림 IV-13>은 초기 가장 우수한 해의 네트워크 설계 구조를 나타내고 있다. 반면 유전자 알고리즘을 이용한 설계 결과 최적해의 비용은 686.70\$였다. 타부리스트가 결합된 유전자 알고리즘을 이용한 초기 가장 우수한 해와 가장 우수한 최종해의 서브트리 및 트래픽 용량은 <표 IV-10>, <표 IV-11>과 같다.



<그림 IV-13> 초기 가장 우수한 해(문제 5 : 681.97\$)

<표 IV-10> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 5)

서브트리	{2, 1}	{5, 11}	{6, 4, 9, 16, 18}	{8, 13, 3, 14, 20}	{17, 10, 15, 19, 12, 7}
트래픽 용량	2	2	5	5	6



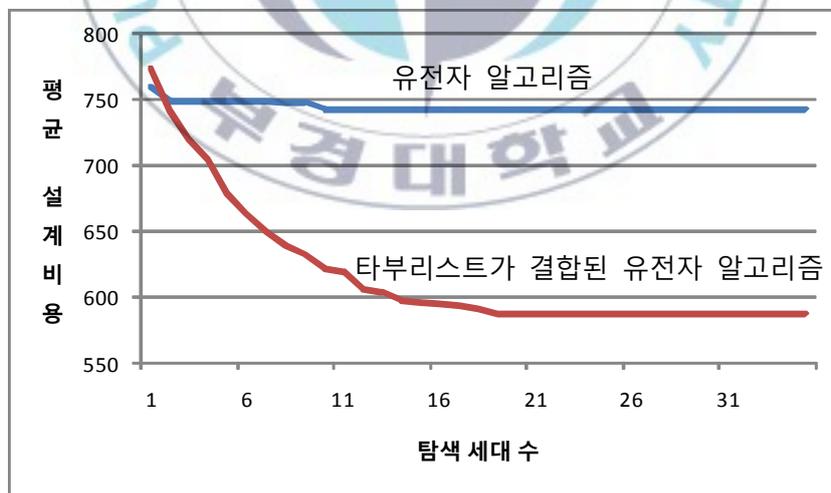
<그림 IV-14> 가장 우수한 최종해(문제 5 : 416.12\$)

<표 IV-11> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 5)

서브트리	{2, 1, 13, 3, 14, 20}	{5, 15}	{6, 4}	{8, 7, 9, 15, 11, 18}	{17, 10, 12, 19}
트래픽 용량	6	2	2	6	4

문제 5에서 대해서도 신뢰성 있는 분석을 위하여 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘을 이용한 설계 방법 각각에 대해 서로 다른 10가지의 random number seed에 적용하여 문제를 풀었다.

그 결과 10가지 경우 모두 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었다. 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색을 통해 도출된 10가지 seed의 해에 대해 탐색 세대 수의 증가에 따른 평균 설계비용의 추이를 살펴보면 <그림 IV-15>와 같다.



<그림 IV-15> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 5)

타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법은 탐색 세대 수가 증가함에 따라 평균 설계비용이 급격히 떨어져 28세대에서 약 590 \$로 수렴하는 반면에 유전자 알고리즘을 이용한 탐색 방법은 10세대에 이르기까지 6번의 설계비용 감소를 가졌으며, 8세대에서 평균 설계비용 740 \$를 가져 약 21% 개선 효과를 보여주고 있다.

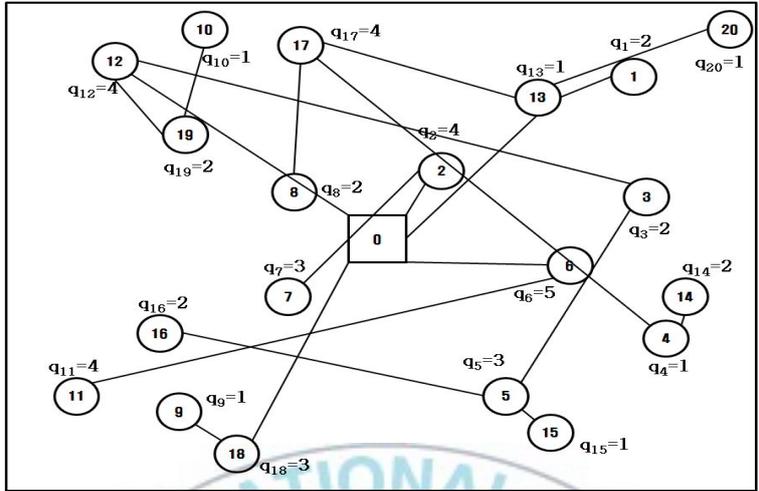
이 결과는 타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법이 유전자 알고리즘을 이용한 탐색 방법보다 효율적인 설계 방법이라는 것을 보여 주고 있다.

6. 문제 6 : 21개의 노드, 서브트리 트래픽 용량 15

문제 2, 4와 마찬가지로 좀 더 현실적인 상황에서 타부리스트가 결합된 유전자 알고리즘의 성능평가를 위해 21개의 노드를 네트워크로 연결하는 문제 5에서 다른 모든 초기화 값이 동일하나 각 노드의 트래픽 용량이 서로 다른 문제 6에 대해 동일한 설계 절차에 따라 실험을 수행하였다.

각 노드의 트래픽 용량은 1~5 사이의 random number를 적용하였다.

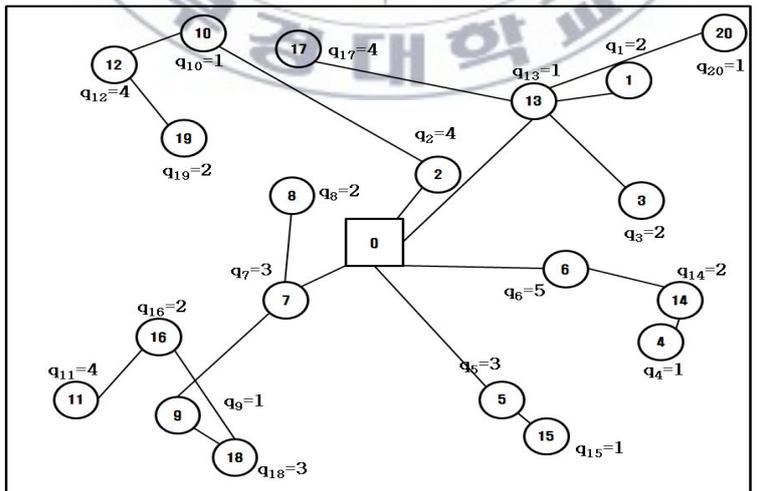
그 결과 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었으며, <그림 IV-17>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 454.71\$인 가장 우수한 해를 찾았다. <그림 IV-16>은 초기 가장 우수한 해의 네트워크 설계 구조를 나타내고 있다. 반면 유전자 알고리즘을 이용한 설계 결과 최적해의 비용은 631.856\$였다. 타부리스트가 결합된 유전자 알고리즘을 이용한 초기 가장 우수한 해와 가장 우수한 최종해의 서브트리 및 트래픽 용량은 <표 IV-12>, <표 IV-13>와 같다.



<그림 IV-16> 초기 가장 우수한 해(문제 6 : 734.625\$)

<표 IV-12> 초기 가장 우수한 해의 서브트리 및 트래픽 용량(문제 6)

서브트리	{2, 7}	{6, 11}	{12, 19, 10, 3, 5, 15, 16}	{13, 1, 20, 17, 8, 4, 14}	{18, 9}
트래픽 용량	7	9	15	13	4



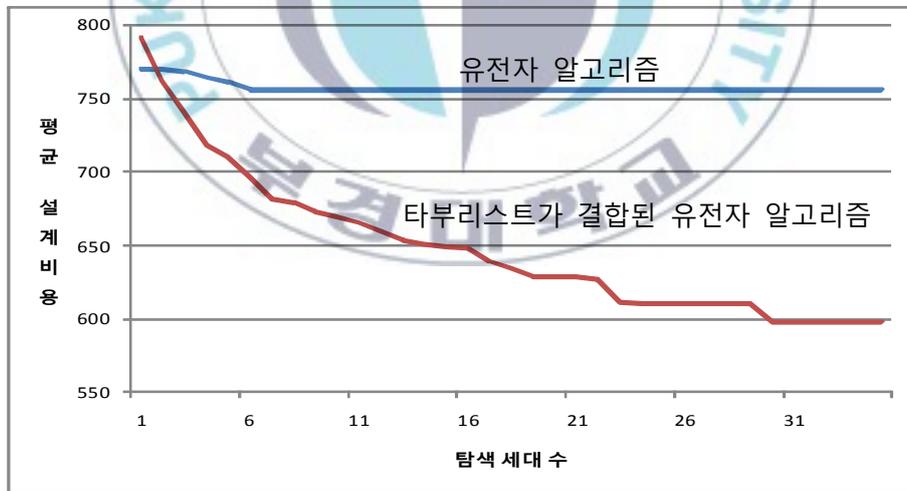
<그림 IV-17> 가장 우수한 최종해(문제 6 : 454.71\$)

<표 IV-13> 가장 우수한 최종해의 서브트리 및 트래픽 용량(문제 6)

서브트리	{2, 10, 12, 19}	{5, 15}	{6, 14, 4}	{7, 8, 9, 18, 16, 11}	{13, 1, 3, 17, 20}
트래픽 용량	11	4	8	15	0

문제 6에 대해서도 신뢰성 있는 분석을 위하여 서로 다른 10가지의 random number seed에 적용하여 문제를 풀었다.

그 결과 10가지 경우 모두 타부리스트가 결합된 유전자 알고리즘이 유전자 알고리즘에 비해 더 좋은 해를 도출해 내었다. 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색을 통해 도출된 10가지 seed의 해에 대해 탐색 세대 수의 증가에 따른 평균 설계비용의 추이를 살펴보면 <그림 IV-18>와 같다.



<그림 IV-18> 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘 탐색 방법의 성능 비교(문제 6)

타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법은 탐색 세대가 증가함에 따라 평균 설계비용이 급격히 떨어져 31세대에서 약 580\$로 수렴하는 반면에 유전자 알고리즘을 이용한 탐색 방법은 6세대에 이르기까지 5번의 설계비용 감소를 가졌으며, 6세대에서 평균 설계비용 756.02\$를 가져 약 21% 개선 효과를 보여주고 있다.

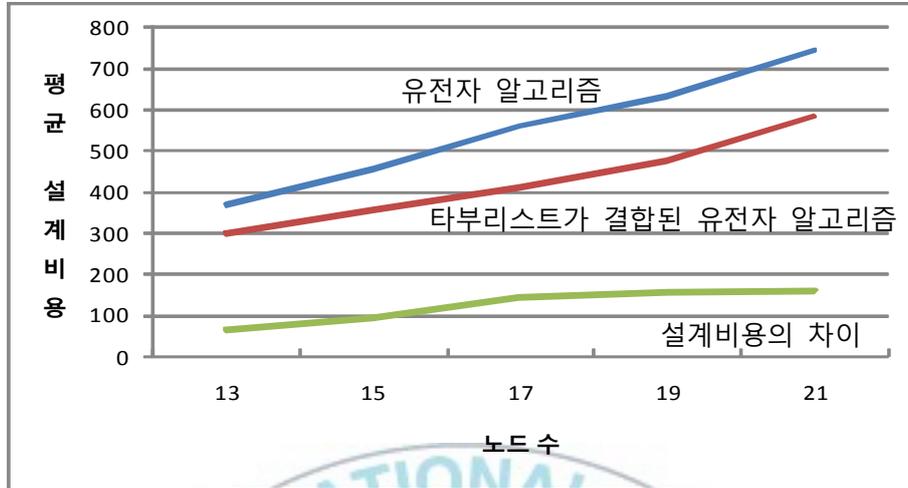
이 결과는 각 노드의 트래픽 용량이 서로 다른 경우에도 타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법이 유전자 알고리즘을 이용한 탐색 방법보다 효율적인 설계 방법이라는 것을 보여 주고 있다.

7. 노드 수의 증가에 따른 평균 설계비용의 차이

노드 수의 증가에 따른 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 평균 설계비용의 차이가 어떻게 변화하는지 평가하기 위하여 13, 15, 17, 19, 21개의 노드에 각 노드의 트래픽이 1인 경우, 각 노드의 트래픽이 서로 다른 경우에 대해 실험을 실시하였다.

신뢰성 있는 평가를 위해 각각의 상황에 10가지의 random number seed를 적용하여 평균 설계비용을 계산하였다.

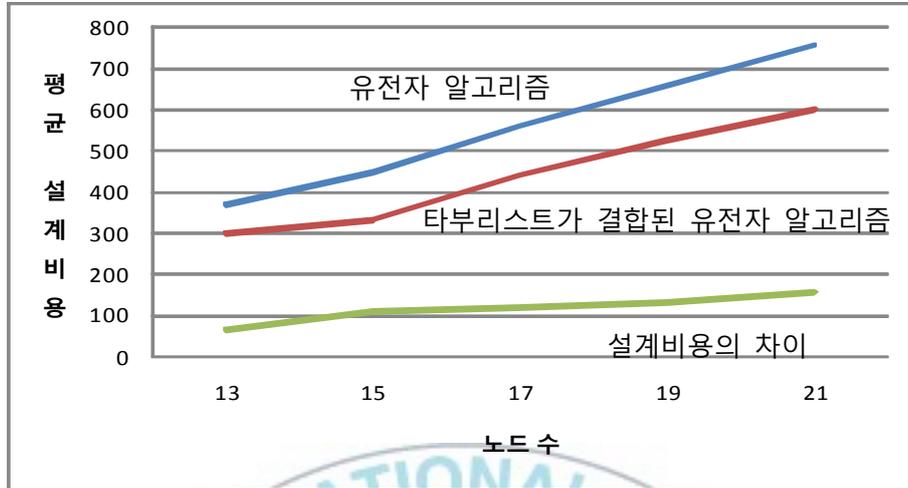
아래의 그림<IV-19>은 각 노드의 트래픽이 1일 때 노드 수의 증가에 따른 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 설계비용의 차이를 나타내고 있다.



<그림 IV-19> 노드의 증가에 따른 평균 설계비용의 차이(트래픽 1)

각 노드의 트래픽이 1인 경우, 노드의 수가 증가함에 따라 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 평균 설계비용의 차이가 점차 커지는 것을 보여주고 있다.

아래 <그림 IV-20>는 각 노드의 트래픽이 서로 다른 경우 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 평균 설계비용의 차이를 나타내고 있다.



<그림 IV-20> 노드의 증가에 따른 평균 설계비용의 차이(트래픽 랜덤)

각 노드의 트래픽이 서로 다른 경우도 마찬가지로 노드의 수가 증가함에 따라 타부리스트가 결합된 유전자 알고리즘과 유전자 알고리즘의 평균 설계비용의 차이가 점차 커지는 것을 보여주고 있다.

V. 결 론

네트워크가 기업 내의 자원공유와 정보교환을 위한 중요한 전략적 수단으로 활용되면서, 효율적인 네트워크의 설계는 기업 활동에 직접적인 영향을 미치게 되었다. 네트워크를 구축하기 위해서는 막대한 투자비용이 소요되기 때문에 중대한 의사결정 문제이다.

기업은 기업 내 자원을 공유하고 다양한 정보자원을 빠르게 교환하여 과업수행에 활용하고자 최소한의 비용으로 효율적인 네트워크를 설치하여 운영하길 원한다.

본 연구에서는 이러한 기업의 요구에 맞추어, 서브트리의 최대 트래픽 용량 제약조건으로 한 비용을 최소화하는 트리형 네트워크 최적화 문제를 효율적으로 풀기 위해 유전자 알고리즘에 타부서치 중 타부리스트의 기능을 결합한 타부리스트가 결합된 유전자 알고리즘을 이용한 설계 방법을 제시하였다.

본 연구에서 13개 노드에 각 노드의 트래픽 용량이 1인 문제 1, 13개 노드에 각 노드의 트래픽 용량이 서로 다른 경우인 문제 2, 17개 노드에 각 노드의 트래픽 용량이 1인 문제 3, 17개 노드에 각 노드의 트래픽 용량이 서로 다른 경우인 문제 4, 21개 노드에 각 노드의 트래픽 용량이 1인 문제 5, 21개 노드에 각 노드의 트래픽 용량이 서로 다른 경우인 문제 6를 통해 기존의 유전자 알고리즘보다 타부리스트가 결합된 유전자 알고리즘의 성능이 네트워크 설계비용과 최적해로의 수렴도가 우월하다는 것을 밝혔다. 또한 앞선 실험적 문제에 노드의 수가 15, 19개이고 각 노드의 트래픽 용량이 1인 경우와 각 노드의 트래픽 용량이 서로 다른 경우에 대한 실험을 추가하여 노드 수의 증가에 따라 타부리스트가 결합된 유전자 알고리즘과

유전자 알고리즘의 평균 설계비용의 차이가 점점 증가하는 것을 밝혔다.

유전자 알고리즘에 결합된 타부 리스트는 세대를 거치면서 반복되는 진화 연산 과정 중에 동일한 해가 생성되는 것을 방지하여 효율적인 최적해의 탐색을 가능하게 하였다.

본 연구에서는 단일 링크 유형, 트래픽 용량 제약조건 하에서의 트리형 네트워크의 경제적 설계 방법을 제시하였으나, 향후에는 다양한 링크 유형, 평균 패킷 지연 시간, 루트노드의 총 트래픽 용량 제약조건 하에서의 설계 방법에 대한 연구가 수행되어야 할 것이다.



참 고 문 헌

1. 국내 문헌

Melanie Mitchell(1997), “유전자 알고리즘 입문,” 진영사.

고대식 · 서태석 · 류명주 · 최세하(2000), “데이터 통신 및 LAN 설계,” 도서출판 아진, 2000.

김여근 · 윤복식 · 이상복(1997), “메타 휴리스틱, 유전알고리즘 · 시뮬레이티드 어닐링 · 타부서치,” 영지문화사.

손동욱(1995), “정보통신의 이해,” 생능출판사.

신재호 · 권병희 · 임강진(1998), “Network Bible,” 영진출판사.

진강규(2000), “유전자알고리즘과 그 응용,” 교우사.

염창선 · 이한진(2005), “유전자 알고리즘을 이용한 경제적 LAN 설계,” *Journal of the Society of Korea Industrial and Systems Engineering*, 제28권, 제2호, pp. 101-108.

오동헌(2003), “지역탐색 유전자 알고리즘을 이용한 네트워크 설계에 관한 연구,” 부경대학교 대학원 정보시스템학과, 학위논문.

윤현식(2002), “유전자 알고리즘과 타부서치를 이용한 차량경로문제에 관한 연구,” 국방대학교 대학원 운영분석전공, 학위논문.

이용진 · 김태윤(1994), “근거리 통신망에서의 CMST 알고리즘,” *한국정보과학회, 정보과학회논문지*, 제20권 제3호, pp. 1097-1106.

이용진 · 김태진(1995), “동적 계획법에 기초한 로컬 액세스 트리 네트워크의 토폴로지 설계 알고리즘,” *한국통신학회논문지*, 제20권, 제8호, pp. 2067-2078.

이용진(2003), “평균 지연 시간과 트래픽 용량이 제한되는 스페닝 트리 문제의 2단계 휴리스틱 알고리즘,” 한국정보처리학회, 정보처리학회 논문지C, 제10권, 제3호, pp. 367-376.

장수현 · 윤병주(1998), “유전자 알고리즘에서의 실수처리 방법 비교,” 한국정보처리학회 논문지, 제5권, 제2호, pp. 361-371.

2. 외국 문헌

A. Kershenbaum(1993), “Telecommunication Network Design Algorithm”, McGraw-Hill, pp. 183-200.

Ahuja V.(1985), “Design and Analysis of Computer Communication Networks”, pp. 117-145, McGrawHill,.

Goldberg, D. E.(1989), “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison Wesley.

Elbaum, R. and Sidi, M.(1995), “Topological Design of Local Area Networks using Genetic Algorithms”, *INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE* , Vol. 1, pp. 64-71.

Ersoy, C. and Panwar, S. S.(1992), “Topological Design of Interconnected LAN-MAN Networks”, *INFOCOM '92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE*, Vol. 3, pp. 2260-2269.

Esau L. R. and Williams(1966), “On teleprocessing system design, part II”, *IBM syst.J.*, Vol. 5, No. 3, pp. 142-147.

- Eshelman, L. J. and Caruana, R. A.(1989), "Bases in the Crossover Landscape," Proc. 3rd Int. Conf. on Genetic Algorithms, J. Schaffer(Ed.), Morgan Kaufmann Publishers, LA.
- Gamvros I., Golden B. and Raghavan S.(2006), "The Multilevel Capacitated Minimum Spanning Tree Problem", *Journal on Computing*, Vol. 18, No. 3, pp. 348-365.
- Glover F.(1977), "Heuristic for Integer Programming Using Surrogate Constraints", *Decision Sciences*, Vol. 8, pp. 156-166.
- Goldberg, D. E.(1991), "Real-Coded Genetic Algorithms, Virtual Alphabets, and Blocking", *Complex Systems*, Vol. 5, pp. 139-167.
- Gouveia L. and paixao J.(1991), "Dynamic Programming Based Heuristics for the topological design of local access networks", *Annals of Operations Research 33*, pp. 305-327.
- Holland, J. H.(1975), "Adaptation in Natural and Artificial Systems", The University of Michigan Press, Michigan.
- Kershenbaum A. and Boorstyn R. R.(1983), "Centralized Tele Processing Network Design", *Networks*, Vol. 13, pp. 279-293.
- Papadimitriou C. H.(1978), "The Complexity of the Capacitated Tree Problem", *Networks*, Vol. 8, pp. 217-230.
- Sharaiha Y. M., Gendreau M., Laporte G., and Osman I. H.(1997), "A Tabu Search Algorithm for the Capacitated Shortest Spanning Tree Problem", *Networks*, Vol. 29, Issue 3, pp. 161-171.
- Signorile R. and Kershenbaum A.(1986), "Multipoint Line Optimization Using fuzzy set partitioning", *Proceedings of GLOBECOM-86*, pp. 1571-1576.

- Tate, D. M. and Smith, A. E.(1995), "A Genetic Approach to the Quadratic Assignment Problem", *Computers and Operations Research*, Vol. 22, No. 1, pp. 73-83.
- Torres-Jiménez J., Pinto-Elías R. and García-Romero A.(1999), "A Simulated Annealing Approach for the Capacitated minimum Spanning Tree Problem", *iccima*, p. 432, Third International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'99).
- Zhou G. and Meng Z.(2003), "A genetic algorithm approach on tree-like telecommunication network design problem", *Journal-Operational Research Society*, Vol. 54, pp. 248-254/.
- Zhou G., Cao Z., Cao J, and Meng Z.(2006), "A Genetic Algorithm Approach on Capacitated Minimum Spanning Tree Problem", *Computational Intelligence and Security*, 2006 International Conference on, pp. 215-218.