

經營學博士 學位論文

유전자 알고리즘을 이용한 경제적
네트워크 설계

2006 年 2 月

釜慶大學校 大學院

經 營 學 科

李 漢 珍

經營學博士 學位論文

유전자 알고리즘을 이용한 경제적
네트워크 설계

指導教授 廉 昌 善

이 論文을 經營學博士 學位論文으로 提出함.

2006 年 2 月

釜慶大學校 大學院

經 營 學 科

李 漢 珍

李漢珍의 經營學博士 學位論文을 認准함

2006 年 2 月 24 日

主 審 經營學博士 魚 允 洋 ㉠

委 員 工學博士 韓 載 昊 ㉠

委 員 經營學博士 金 河 均 ㉠

委 員 工學博士 趙 宰 均 ㉠

委 員 工學博士 廉 昌 善 ㉠

< 목 차 >

제 1 장 서 론	1
제 1 절 연구 배경 및 목적	1
제 2 절 연구 방법 및 구성	5
제 2 장 네트워크 설계 및 유전자 알고리즘에 관한 고찰	7
제 1 절 네트워크 설계	7
1. 네트워크의 개념 및 정의	7
2. 네트워크의 분류	9
3. 네트워크의 그래프적 표현	10
4. 네트워크 설계 문제	11
5. 네트워크 신뢰도 계산	12
6. 신뢰성 있는 네트워크 설계	14
제 2 절 유전자 알고리즘	15
1. 유전자 알고리즘의 개념	15
2. 유전자 알고리즘의 특징	16
3. 유전자 알고리즘의 기본 요소	17
4. 유전자 알고리즘의 구조	19
5. 유전자 알고리즘의 표현(Representations)과 관련된 문제	20
6. 유전자 알고리즘의 염색체적 표현	24
1) 이진 코딩	24
2) k-ary 코딩	25
3) 기호 코딩	25
4) 실수 코딩	26
7. 유전자 알고리즘의 연산자	27
1) 선택(selection) 연산	27
2) 교배(crossover) 연산	31

3) 돌연변이(mutation) 연산	38
4) 종료조건	40
8. 네트워크 설계를 위한 유전자 알고리즘	41
1) 염색체 표현	42
2) 적합도 계산	43
3) 초기 개체집단 생성	44
4) 선택 연산	45
5) 교배 연산	46
6) 돌연변이 연산	46
7) 신뢰도 계산	47
8) 전체 알고리즘	48
제 3 장 경제적 백본(Backbone) 네트워크 확장 설계	50
제 1 절 경제적 백본 네트워크 확장 설계 문제의 표현	50
1. 기본 가정 및 표기 형식	50
2. 백본 네트워크 확장 문제의 모형	51
3. 백본 네트워크 설계 구조의 염색체 표현	52
제 2 절 경제적 백본 네트워크 확장 설계	56
1. 백본 네트워크 확장 설계를 위한 절차	56
2. 초기 세대의 개체집단 생성	57
3. 교배 및 돌연변이 연산	58
4. 백본 네트워크 신뢰도 계산	59
제 3 절 신뢰도를 제약조건으로 하는 경제적 백본 네트워크 확장 설계의 성능 실험	60
1. 문제 1 : 기존 3개 노드에 2개 노드 추가	61
2. 문제 2 : 기존 5개 노드에 3개 노드 추가	64
제 4 절 비용을 제약조건으로 하는 경제적 백본 네트워크 확장 설계의 성능 실험	66
1. 문제 3 : 기존 3개 노드에 2개 노드 추가	67
2. 문제 4 : 기존 5개 노드에 3개 노드 추가	69

제 5 절 성능 비교 및 민감도 분석	71
1. 유전자 알고리즘의 성능 비교	71
2. 민감도 분석	76
제 4 장 경제적 LAN 설계	79
제 1 절 경제적 LAN 설계에 대한 문제 표현	79
1. 기본 가정 및 표기 형식	79
2. LAN 설계 문제의 모형	81
3. LAN 설계 구조의 탐색체 표현	83
제 2 절 경제적 LAN 설계	86
1. LAN 설계를 위한 절차	86
2. 초기 세대의 개체집단 생성	87
3. 교배 및 돌연변이 연산	88
4. LAN 신뢰도 계산 방법	89
제 3 절 신뢰도를 제약조건으로 하는 경제적 LAN 설계의 성능 실험	91
1. 문제 1 : 3개의 허브와 13명의 사용자	94
2. 문제 2 : 5개의 허브와 30명의 사용자	94
제 4 절 비용을 제약조건으로 하는 경제적 LAN 설계의 성능 실험	96
1. 문제 3 : 4개의 허브와 30명의 사용자	96
2. 문제 4 : 5개의 허브와 30명의 사용자	97
제 5 절 성능 비교 및 민감도 분석	99
1. 유전자 알고리즘의 성능 비교	99
2. 민감도 분석	105
제 5 장 결 론	110
제 1 절 연구 요약	110
제 2 절 연구의 한계점 및 향후 연구 방향	114

참 고 문 헌	116
Abstract	126

<표 목 차>

<표 2-1> 확률바퀴에 의한 선택 확률	29
<표 2-2> 노드 간 링크 유형	43
<표 3-1> 노드 간 링크 유형	53
<표 3-2> 노드 간 링크 유형	55
<표 3-3> 링크 유형의 속성	61
<표 3-4> 노드 간 거리 (단위 : m)	62
<표 3-5> 문제 1에 대한 최적해	63
<표 3-6> 문제 1의 탐색 결과	64
<표 3-7> 노드 간 링크 유형	65
<표 3-8> 노드 간 거리 (단위: m)	65
<표 3-9> 문제 3에 대한 최적해	67
<표 3-10> 문제 3의 탐색 결과	68
<표 3-11> 노드 간 링크 유형	70

<표 4-1> 허브 간 링크 유형	83
<표 4-2> 허브와 사용자 간 연결 유무	84
<표 4-3> 허브 간 링크 유형의 속성	91
<표 4-4> 허브의 속성	92
<표 4-5> 허브 간 거리 (단위 : m)	93
<표 4-6> 허브와 사용자 간 거리 (단위 : m)	93

<그 립 목 차>

<그림 2-1> 로컬 네트워크 위상(이문구, 2000)	9
<그림 2-2> 단방향 네트워크	11
<그림 2-3> 무방향 네트워크	11
<그림 2-4> LAN 구조의 그래프적 표현	11
<그림 2-5> 유전자 알고리즘의 구조	19
<그림 2-6> 유전자 알고리즘에서의 해공간과 부호공간	20
<그림 2-7> 부호공간에서 해공간으로의 맵핑(mapping)	22
<그림 2-8> 실행 불가능성(infeasibility)과 위법성(illegality)	23
<그림 2-9> 확률바퀴 예	29
<그림 2-10> 네트워크 설계 예	42
<그림 3-1> 네트워크 확장 설계 문제	53
<그림 3-2> 후보 해	55
<그림 3-3> 네트워크 확장 설계를 위한 문제 (=0.900)	64

<그림 3-4> 가장 우수한 해(=4,208\$)	66
<그림 3-5> 네트워크 확장 설계를 위한 문제 (=4,700\$)	70
<그림 3-6> 가장 우수한 해(=0.944)	71
<그림 3-7> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 2)	74
<그림 3-8> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 4)	75
<그림 3-9> 백본 네트워크 신뢰도 변화에 따른 민감도 분석결과(문제 2)	76
<그림 3-10> 백본 네트워크 설치비용 변화에 따른 민감도 분석결과(문제 4)	77
<그림 4-1> LAN 설계의 예	83
<그림 4-2> 허브와 사용자 위치	92
<그림 4-3> 가장 우수한 해(=1161.5\$)	94
<그림 4-4> 가장 우수한 해(=3366\$)	95
<그림 4-5> 가장 우수한 해(=0.902)	97
<그림 4-6> 가장 우수한 해(=0.895)	98
<그림 4-7> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 1)	99
<그림 4-8> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 2) ..	101

<그림 4-9> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 3) ..	102
<그림 4-10> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 4)	104
<그림 4-11> LAN 신뢰도 변화에 따른 민감도 분석결과(문제 1)	105
<그림 4-12> LAN 신뢰도 변화에 따른 민감도 분석결과(문제 2)	106
<그림 4-13> LAN 설치비용 변화에 따른 민감도 분석결과(문제 3)	107
<그림 4-14> LAN 설치비용 변화에 따른 민감도 분석결과(문제 4)	108

제 1 장 서 론

제 1 절 연구 배경 및 목적

오늘날 기업은 급변하는 환경 속에서 살아남고 치열한 국내외 경쟁에서 우위를 확보하기 위한 수단으로 정보기술을 활용하고 있다. 정보기술 중 컴퓨터 네트워킹은 기업 구성원들 간에 고가의 하드웨어와 소프트웨어 자원을 공유할 수 있게 할 뿐만 아니라 물리적으로 떨어져 있는 자원들 간을 연결함으로써 보다 적은 비용으로 다양한 정보에 접근할 수 있도록 해준다.

컴퓨터 네트워킹이 기업의 정보 공유를 위한 중요한 역할을 하게 됨에 따라 네트워크의 단절은 기업 활동에 직접적인 영향을 미치게 되었다.

기업의 입장에서 네트워크 설계에 있어 중점적인 요소는 경제성과 신뢰성을 들 수 있다. 이에 따라 충분한 자금을 확보하고 있는 기업의 경우, 신뢰성 있는 컴퓨터 네트워크 설계의 필요성이 네트워크 설계 시 중요한 요소로써 고려될 수 있다. 그러나 한편으로는 네트워크 설계와 관련한 충분한 자금획득이 어려운 중소기업의 경우 어느 정도의 네트워크 신뢰성(예 : 0.900)만 확보되면 네트워크 설치에 필요한 비용의 최소화에 관심을 가지게 된다.

신뢰성 및 경제성 있는 컴퓨터 네트워크 설계는 조합 최적화(combinatorial optimization)와 관련된 문제로써, 대부분이 정확한 해를 도출하는 알고리즘을 구하기 힘들며, 주어진 문제에 대한 해의 탐색공간이 문제의 복잡성에 따라 지수적으로 증가하는 NP-hard 문제에 해당된다(Wood, 1986; Cancela & Khadiri, 1995; Dengiz 등, 1997).

1960년대 이래, NP-hard 문제와 같이 난해한 최적화 문제들에 대해 다양한 알고리즘에 적용시키고자 하는 노력들이 진행되어 왔다. 일반적으로 이러한 기술들을 진화 연산(evolutionary computation)이라 한다.

이러한 기술들 중 가장 잘 알려진 알고리즘으로는, Holland(1975)에 의해 개발된 유전자 알고리즘(GA: Genetic Algorithm), Recheberg와 Schwefel(1995)에 의해 개발된 진화 전략(evolution strategies), Fogel 등(1966)에 의해 개발된 진화 프로그래밍(evolutionary Programming) 등이 있다. 또한 이러한 알고리즘들의 여러 가지 특징들을 결합한 하이브리드 형태들이 있다. 진화 계산론의 최근 형태들은 Bäck and Schwefel(1993), Michalewicz(1992), 그리고 Fogel(1995)등에 의해 제시되고 있다(Gen and Cheng, 2000).

신뢰성 있는 컴퓨터 네트워크 설계와 관련된 연구는 크게 소규모 네트워크 설계에 적합한 계산 중심적 접근방법(enumerative-based approach)과, 보다 더 큰 규모의 네트워크에 적용할 수 있는 타부서치(tabu serch), 시뮬레이티드 어닐링(simulated annealing), 유전자 알고리즘(genetic algorithm) 등의 휴리스틱 접근(heuristic approach)으로 크게 나눌 수 있다. 이 중 NP-hard 문제에 다양한 형태로 사용되어지고 있는 휴리스틱 접근방법은 빠른 시간에 근사 최적해(우수해)를 찾을 수 있는 기법으로써 주어진 가정 하에 최적해 또는 근사 최적해(우수해)를 도출하는데 사용되어진다. 이러한 방법 중 확률론적 탐색(stochastic search)과 최적화 기법(optimization technique)을 통해 강력하고 광범위하게 적용되는 유전자 알고리즘은 오늘날 진화 연산 방법에 있어 가장 광범위하게 알려진 형태이다.

네트워크 설계 문제에 대해 유전자 알고리즘을 적용한 기존의 연구를 크게 백본 네트워크(Backbone Network) 설계 문제와 LAN 설계 문제로 나누

어 살펴보면 다음과 같다.

백본 네트워크 설계와 관련된 연구로서, Jan 등(1993)은 네트워크 신뢰도를 제약조건으로 하고 비용을 최소화하는 경제적 백본 네트워크 설계 방법을 제시하였다. 그리고 Deeter와 Smith(1998)는 유전자 알고리즘을 이용하여 신뢰도를 제약조건으로 하고 비용을 최소화하는 경제적 백본 네트워크 설계 방법을 제시하였다. 이 설계 방법은 Jan 등(1993)이 네트워크의 링크를 단일 유형만을 고려한데 반해 여러 유형을 고려하고 있어 좀 더 발전된 설계 방법이라 할 수 있다.

LAN 설계와 관련된 연구로서, Elbaum 등(1996)은 유전자 알고리즘을 이용하여 신장트리(spanning tree)를 제약조건으로 하고 평균 네트워크 지연(average network delay)을 최소화하는 LAN 설계 방법을 제시하였다. Gen 등(1998)은 유전자 알고리즘을 이용하여 신장트리를 제약조건으로 하고 평균 메시지 지연과 비용을 최소화하는 다목적 LAN 설계 방법을 제시하였다.

이들과는 달리, Kumar 등(1995)은 유전자 알고리즘을 네트워크 성능(network performance)에 작용하는 중요한 비용요소인 유효경로(diameter), 거리(distance), 각 노드의 차수(degree) 및 평균 거리(average distance)와 네트워크 신뢰도(reliability)를 통해 다양한 형태의 백본 네트워크 확장 설계에 관해 연구하였다. 그리고 Shao와 Zhao(1998)는 신뢰도를 제약조건으로 하여 기존의 네트워크에 여러 후보 노드 중 한 개의 노드를 선택하여 추가하는 경제적 백본 네트워크 확장 설계 방법을 제시하였다. 최근 염창선과 배상욱(2005)은 수행가능성(Performability) 제약 하에 비용을 최소화하는 네트워크 확장 설계에 관한 연구를 유전자 알고리즘을 이용하여 수행하였다.

백본 네트워크 설계 중 확장 설계와 관련된 기존의 연구에서는 기존의 네트워크 구조에 새로운 노드만을 추가하는 문제만으로 한정하거나, 노드를

부가하더라도 네트워크의 중요한 요소인 신뢰도를 제약조건으로 하여 다양한 링크 수준까지 고려하는 경제적 네트워크 확장 설계에 관한 연구가 미흡하다. 또한, Kumar 등(1995)의 연구에서는 비용요소를 제약조건으로 하고 네트워크 신뢰도를 최대화하는 문제를 다루고 있으나, 연구에서 비용요소로 사용된 유효경로나 정점의 차수 등은 실제 네트워크 설계문제에 적용함에 있어 현실성이 낮다는 단점이 있다. 그러므로 본 연구에서는 백본 네트워크 확장 설계 시 복수개의 새로운 노드 부가 이외에 이들 노드를 연결함에 있어 다양한 링크 수준 및 기존 링크 교체 시 현실적으로 고려해야 하는 교체 비용까지 포함한 네트워크 설계방법을 제시하고자 한다.

LAN 설계와 관련된 기존의 연구들에 있어서는 허브 간의 연결 및 허브와 사용자 간의 연결을 위해 한 가지의 허브와 링크 유형만을 고려한 LAN 설계 방법에 한정되었다. 그러므로 본 연구에서는 LAN 설계 시 스타링형을 대상으로 다양한 허브 유형과 링크 유형을 고려한 설계방법을 제시하고자 한다.

본 연구에서는 첫째, 백본 네트워크 확장 설계 문제에 있어 네트워크 신뢰도를 제약조건으로 하고 네트워크 설계비용을 최소화하는 문제와 네트워크 설계비용을 제약조건으로 하고 네트워크 신뢰도를 최대화 하는 문제를 통한 경제적 네트워크 설계방법을 제안한다.

둘째, LAN 설계 방법에 있어 백본 네트워크 확장 설계문제와 동일하게 LAN 신뢰도를 제약조건으로 하고 LAN 설치비용을 최소화하는 문제와 LAN 설치비용을 제약조건으로 하고 LAN 신뢰도를 최대화하는 문제를 통한 경제적 LAN 설계 방법을 제안한다.

경제적 백본 네트워크 확장 설계 및 LAN 설계 방법에 있어 효율적인 탐색을 위해 유전자 알고리즘을 이용한다.

제 2 절 연구 방법 및 구성

본 연구는 백본 네트워크 확장 설계와 LAN 설계 문제에 있어 주어진 최소 네트워크 신뢰도(network reliability)를 충족시키면서, 목적함수인 네트워크 설치비용을 최소화하는 문제와 최대 네트워크 설치비용을 만족하면서 네트워크 신뢰도를 최대화하는 문제를 해결하기 위해 휴리스틱 접근기법 중 하나인 유전자 알고리즘을 이용한다.

본 연구는 첫째, 유전자 알고리즘을 이용한 백본 네트워크 확장설계에 있어, 네트워크 신뢰도와 네트워크 설치비용을 제약조건 및 목적함수로 하여 다양한 노드를 부가할 뿐만 아니라 이러한 노드를 다양한 링크수준으로 연결하는 설계방법을 제안한다. 이를 위해, 백본 네트워크 확장 설계 구조에 적합한 염색체 구조를 k-nergy코딩(k-nergy encoding)을 이용하여 표현하며 유전자 알고리즘의 중요요소인 초기 개체집단 형성 방법, 교배 및 돌연변이 연산 방법을 제시한다. 또한 백본 네트워크 설계의 중요요소인 네트워크 신뢰도 계산을 위해 본 논문에서 제안하는 백본 네트워크 신뢰도 계산방법을 이용한다.

둘째, 유전자 알고리즘을 이용한 LAN 설계에 있어 LAN 신뢰도와 설치비용을 제약조건 및 목적함수로 하여 각 사용자들을 연결하기 위한 설계 방법을 제시한다. 이러한 LAN 설계에 있어 사용자와 허브 및 허브와 허브 간 연결을 위해 사용된 허브의 유형과 링크유형을 다양화한다. 유전자 알고리즘을 적용함에 있어 다양한 허브 유형과 링크 유형 및 사용자를 고려한 LAN 연결 형태를 염색체로 표현하기 위해 k-nergy 코딩방법을 이용한 염색체 표현방법을 제시한다. 그리고 유전자 알고리즘을 LAN 설계에 적용하기 위해 중요한 세대 진행과정인 초기 개체집단 형성 방법, 교배 및 돌연변이

연산 방법 및 본 논문에서 제안하는 LAN 신뢰도 계산방법을 이용한다.

마지막으로, 본 논문에서 제안하는 백본 네트워크 확장설계 및 LAN 설계 방법의 성능 평가를 위해 SA(Simulated Annealing) 및 Greedy 알고리즘과 비교하여 성능의 우수성을 입증하고, 네트워크 신뢰도 및 설치비용 제약 조건의 변화에 따라 각 목적함수의 변화를 살펴보기 위해 민감도 분석을 실시하고자 한다.

본 연구는 총 5장으로 구성되었다. 제 1 장은 서론으로써 연구의 배경 및 목적, 연구 방법 및 구성을 제시하고, 제 2 장에서는 네트워크 설계와 관련된 기본적인 개념과 유전자 알고리즘을 구성하고 있는 중요 개념 및 기존 연구에 적용된 유전자 알고리즘에 대해 고찰한다. 제 3 장에서는 백본 네트워크 확장 설계 문제에 대한 문제 표현 및 문제 모형, 유전자 알고리즘을 통한 백본 네트워크 설계문제에 대한 염색체 표현 및 절차를 제시한다. 그리고 유전자 알고리즘을 이용한 백본 네트워크 확장 설계 방법을 SA와 Greedy 기법과의 성능 비교 및 제약조건의 변화에 따른 민감도 분석을 통해 제안된 알고리즘의 우수성을 평가한다. 제 4 장에서는 경제적 LAN 설계 문제에 대한 문제 표현 및 문제 모형, 유전자 알고리즘을 통한 LAN 설계문제에 대한 염색체 표현 및 절차를 제시한다. 그리고 LAN 설계 문제에 대해 주어진 실험 예제를 SA와 Greedy 기법과의 성능 비교 및 민감도 분석을 통해 제안된 알고리즘의 우수성을 평가한다. 마지막으로 제 5 장은 결론으로써 본 연구에 대한 연구 요약 및 본 연구의 한계점과 향후 연구 방향에 대해 기술한다.

제 2 장 네트워크 설계 및 유전자 알고리즘에 관한 고찰

제 1 절 네트워크 설계

1. 네트워크의 개념 및 정의

네트워크란 데이터 통신 그 자체로서 단말장치(terminal) 간의 통신경로를 구성하는 자원의 집합을 의미한다(신재호 등, 1998).

네트워크는 송신 장치, 전송 매체, 수신 장치로 이루어지고, 이 세 가지를 묶어 통신 링크(communication link)라고 하며 통신링크는 네트워크에서 공유되는 요소들이다. 통신 링크 공유의 목적은 하나의 공유 백본을 구축하여 각 지역의 연결을 구현하는 것이다.

통신 링크 공유의 중요성은 바로 비용과 연결되는 민감한 요소이기 때문이다. 이 통신 링크의 공유는 보통 네트워크에서 백본(backbone)이라는 기술로 적용된다.

백본이란 용어는 사용되는 환경에 따라 각각 다른 의미를 갖지만 일반적으로 구축된 케이블 환경에서의 백본은 주 또는 중간 절체 접속점과 통신을 연결하는데 사용된 중심 케이블을 말한다. 이러한 관점에서 네트워크 최하위 레벨로서, 네트워크의 중심을 이루는 주요간선이라고 정의되며 많은 랜이나 원거리 통신망들을 서로 연결시켜 주는 초고속 통신망을 백본이라고 부른다(고대식 등, 2000).

LAN이란 용어에 대해 전기전자협회(IEEE)에서는 “다수의 개별적인 장치

들이 어느 정도의 전송속도를 갖춘 물리적인 통신채널을 이용하여 적당한 면적 안에서 서로 직접적으로 통신이 가능하도록 하는 데이터 통신 시스템”으로 정의하고 있다.

그러므로 LAN은 다음과 같은 특징을 가진 네트워크라 할 수 있다(이문구, 2000).

첫째, 사무실용 빌딩, 공장, 연구소 또는 학교 등의 구내에 분산적으로 설치된 여러 장비들을 연결할 수 있다.

둘째, 사무실 또는 공장 내와 같은 한 장소에 여러 대의 컴퓨터 네트워크가 설치된 규모에서부터 대학이나 공장 등의 일정 지역 내에 설치된 대규모적인 네트워크까지 존재한다.

셋째, LAN은 종래의 컴퓨터 시스템과 같이 호스트 컴퓨터와 단말을 트리(tree)와 같은 구조로 접속할 뿐만 아니라 버스 형태나 링 형태로도 접속하므로 케이블의 배선이 대폭 생략, 간소화 되고 있으며, 또한 LAN에 다른 LAN을 덧붙여 사용이 가능하다.

넷째, LAN의 전송방식으로는 변조하지 않는 베이스밴드(baseband) 방식과 변조를 하는 브로드밴드(broadband) 방식이 있으며, 도입하는 규모, 목적, 가격 등에 따라 다양하게 선택할 수 있다.

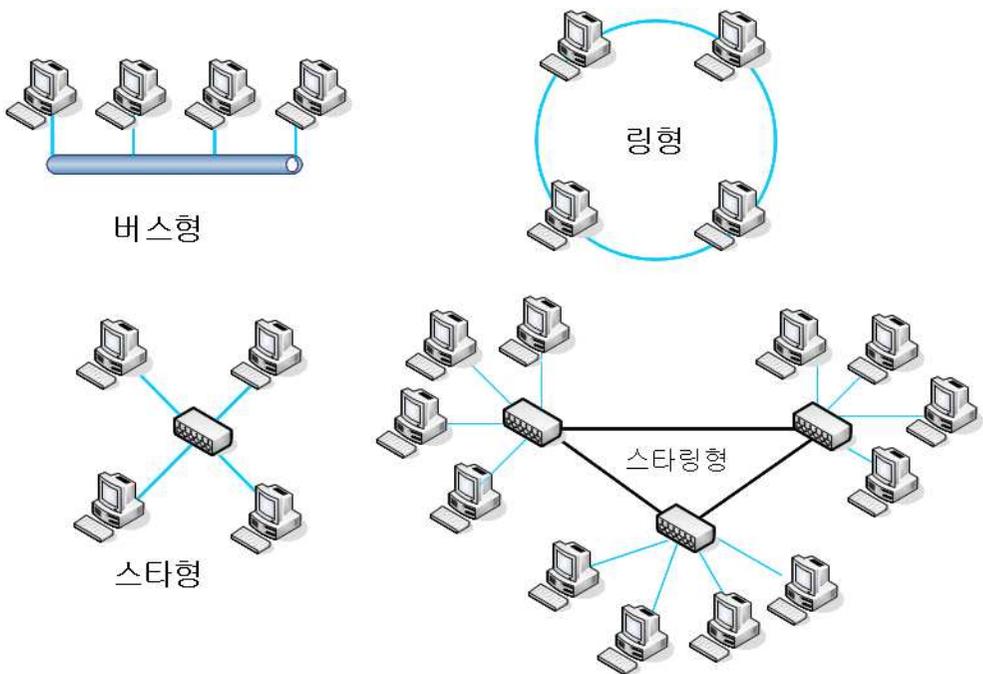
LAN의 관점에서 본 백본의 또 다른 의미는 둘 이상의 LAN을 연결하는 것으로써, 하나의 LAN으로부터 다른 곳까지 통신을 할 수 있는 전송경로를 제공한다. 하나의 백본에 연결된 후에 LAN은 별개로 남게 되거나 다른 곳에 병합된다.

개별적인 LAN은 백본의 형태에 의해 결정된 네트워크 장비 또는 구성을 의미한다. 그러므로 허브와 같은 장비는 가장 하단의 사용자 바로 위단에 위치하고 이러한 허브들이 이런 백본 장비에 연결이 되어 데이터가 모이지

며, 이런 데이터들이 라우터를 통해 필요한 경로로 나가게 되는 것이다.

2. 네트워크의 분류

네트워크의 성질은 주로 위상(topology)에 의해 결정되는데, 4가지의 일반적인 구조가 있으며, <그림 2-1>은 로컬 네트워크의 4가지 일반적인 위상이다.



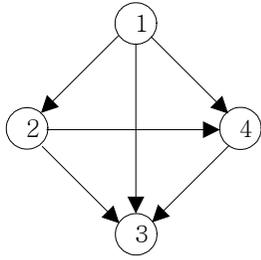
<그림 2-1> 로컬 네트워크 위상(이문구, 2000)

버스형은 장치들이 하나의 통신매체를 공유함으로 한 쌍의 디바이스만이 동시에 통신할 수 있다. 링형은 닫혀진 원의 형태로 구성되며, 데이터는 리피터(repeater)들 사이에서 두 지점 간 전송방식으로 돌아간다. 그러므로 거의 모든 전송매체를 사용할 수 있다. 스타형은 중앙 스위치를 사용하여 네트워크의 모든 요소를 연결한다. 중앙 스위치에서 회선교환(circuit switching) 방법을 사용하여 통신하기를 원하는 2개의 장소에 전용회선을 만들어 준다. 링형과 스타형이 결합된 스타링형은 네트워크 케이블이 스타형과 비슷하게 배치되고 중앙에 있는 허브에 링이 구현된다. 본 연구에서는 스타링형을 대상으로 한다.

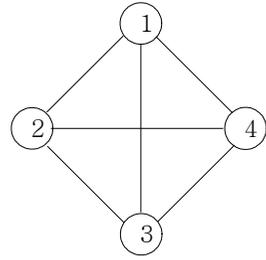
3. 네트워크의 그래프적 표현

네트워크를 그래프로 표현하는데 있어 우선 링크의 방향성에 따라 단방향 네트워크(directed network), 무방향 네트워크(undirected network)로 나눌 수 있다. 단방향 네트워크는 노드 간에 링크의 방향이 원천노드(시점)인 S(source)노드에서 목적노드(종점)인 T(terminal)까지의 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.

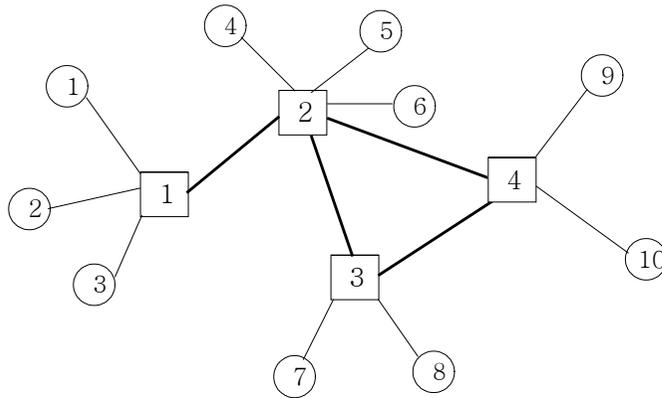


<그림 2-2> 단방향 네트워크



<그림 2-3> 무방향 네트워크

LAN의 경우는 다수의 허브와 다수의 사용자를 포함한 형태로써 아래와 같이 표현될 수 있다.



<그림 2-4> LAN 구조의 그래프적 표현

4. 네트워크 설계 문제

네트워크를 설계하는 것은 주어진 비용 및 자원 하에서 적절한 노드들의 위치 선정 및 링크용량의 적절한 선택, 정확한 수요 예측, 네트워크 확장의

용이성, 올바른 네트워크 장치 선택, 네트워크 신뢰성 등 종합적인 사항들을 고려하여야 한다.

네트워크 설계라 함은 넓게 교환기의 설계, 교환 네트워크 구조의 설계 및 통신 프로토콜의 설계의 의미를 포함한다. 본 연구에서는 교환 네트워크 구조를 설계하는 문제를 다루게 된다.

네트워크 설계에 포함되는 문제들을 살펴보면, 각 노드와 노드간의 비용 예측, 노드 위치의 결정, 최적 네트워크 위상의 결정, 링크 용량 및 링크 신뢰도의 최적화, 적절한 서비스 수준의 결정 등이 포함된다. 그러므로 네트워크 설계시 이상과 같은 문제들을 고려하여 결정하게 되며, 이러한 문제들은 학문적으로나 현실적으로 네트워크 위상 설계에 있어 핵심이 된다.

네트워크 설계의 목적은 주어진 신뢰도 및 성능을 만족하는 네트워크를 최소의 비용으로 구성하는 방법을 찾아내는 것으로써 막대한 투자비용이 드는 네트워크 구축에서 서비스 수요, 네트워크의 성능, 신뢰도 등을 고려하여 경제적인 최적 네트워크 구성을 제공하고자 하는 것이다.

5. 네트워크 신뢰도 계산

일반적으로 신뢰도(reliability)는 시스템 수행의 평가 척도이다. 시스템들이 보다 더 복잡해짐에 따라, 신뢰도가 확보되지 않은 시스템은 시간적 물리적 비용의 증가를 가져오기 때문에 시스템 신뢰도의 평가, 시스템 신뢰도의 향상의 필요조건은 보다 중요한 요소로 작용된다.

네트워크의 신뢰도는 “주어진 환경 내에서 명시된 시간동안 성공적으로 작동할 수 있는 가능성”으로 정의될 수 있다.

네트워크 신뢰도 계산 문제는 다음과 같은 6가지 경우로 분류할 수 있다 (Satyanarayana, 1982).

- 1) S - T 신뢰도 문제 : 주어진 네트워크의 S에서 T까지의 모든 경로를 통해서 목적을 수행하게 될 확률을 계산하는 문제
- 2) S - K 터미널 신뢰도 문제 : 주어진 네트워크가 하나의 S를 갖고 K개의 종점이 목적을 수행하게 될 확률을 계산
- 3) S - All 터미널 신뢰도 문제 : 하나의 S와 나머지 모든 노드들이 T가 되는 네트워크의 문제
- 4) 2 터미널 신뢰도 문제 : 두 개의 노드가 동시에 S와 T가 되는 무방향 네트워크 문제
- 5) K 터미널 신뢰도 문제 : K개의 노드가 S와 T인 무방향 네트워크 문제
- 6) 전 터미널(all-terminal) 신뢰도 문제 : 네트워크상의 모든 노드가 시점과 종점이 되는 경우

위의 6가지 문제 유형 중 전자의 3가지는 단방향 네트워크의 신뢰도 계산과 관련된 문제이고, 후자의 3가지는 무방향 네트워크 신뢰도와 관련된 문제이다.

본 연구에서의 네트워크 신뢰도는 무방향 네트워크 중 전 터미널 네트워크 신뢰도 문제만으로 한정한다.

6. 신뢰성 있는 네트워크 설계

신뢰성 있는 네트워크 설계 문제는 고가의 하드웨어 및 소프트웨어 자원의 공유와 원거리에 위치한 메인시스템에 대한 접근을 위해 많은 연구자들, 특히 네트워크 디자이너, 네트워크 분석가 그리고 네트워크 관리자들의 주된 관심사이다.

네트워크 설계시 중요한 단계 중의 하나는 비용, 신뢰도와 같은 주어진 성과 기준을 최적화 할 수 있는 네트워크 구성요소들의 설계 방법을 찾아내는데 있다.

시스템 성과 기준은 특히 중요한 부분이며, 네트워크 위상(topology)에 의해 주로 결정되어진다. 네트워크 설계 문제에 있어 신뢰도를 고려한 네트워크 설계는 전형적인 네트워크 설계 문제 중의 하나이며, 최적 네트워크 설계 문제는 조합 최적화 문제로서 일반적으로 NP-hard 문제로 알려져 있다.

네트워크 설계 문제와 관련된 신뢰도는 네트워크를 구성하는 모든 노드들이 서로 연결될 수 있는 확률 즉, 네트워크상의 모든 노드들이 명시된 수행 시간 내에 모든 다른 노드들과 통신을 할 수 있는 것을 말한다(Gen & Cheng, 2000).

신뢰성을 고려한 최적 네트워크 설계 문제에 대한 접근 방법은 아래와 같이 3가지로 요약될 수 있다.

- 1) 계산 중심적 접근방법(enumeration-based approach)
- 2) 휴리스틱 중심적 접근방법(heuristic-based approach)
- 3) 유전자 알고리즘 중심적 접근방법(genetic algorithm-based approach)

계산 중심적 접근방법은 주로 소규모 네트워크에 적용될 수 있는 방법으로서 형태(states), 최소경로(minpaths), 최소조각(mincuts)의 계산에 기반하고 있다. 계산 중심적 접근방법은 신뢰성 측정에 있어 상호 배타적인 확률사건들(probabilistic events)의 집합이다.

타부서치(tabu search), 시뮬레이티드 어닐링(simulated annealing) 등으로 대표되는 휴리스틱 중심적 접근방법은 대규모 네트워크에 적용가능하나 최적해를 보장하지는 못한다.

최근 유전자 알고리즘 중심적 접근방법이 신뢰성을 고려한 최적 네트워크 설계문제에 있어 새로운 문제해결 방법으로 주목받기 시작하고 있다(Gen & Cheng, 2000).

제 2 절 유전자 알고리즘

1. 유전자 알고리즘의 개념

유전자 알고리즘(genetic algorithm)은 Darwin이 주장한 자연 진화 법칙인 적자생존(survival of fittest)과 자연도태(natural selection)의 원리를 토대로 하여 정립된 최적화 알고리즘이다. 자연 진화의 법칙에 의하면, 일정한 환경에 무리를 지어 살고 있는 생명체들은 그 환경에 적합한 형질을 가진 개체가 생존할 확률을 크게 가진다. 그리고 교배(crossover)와 돌연변이(mutation)등의 과정을 통해 보다 좋은 방향으로 진화해 나가면서 부적합한 형질의 개체들은 진화의 과정을 통해 점차 도태되어 간다. 그러므로 이와

같은 진화의 과정을 되풀이하면 결국 주어진 환경에 가장 적합한 형질의 개체들이 형성될 것이라는 것이 주요내용이다.

유전자 알고리즘은 이러한 생물의 진화과정, 즉 자연선택(natural selection)과 유전법칙을 최적화 문제에 적용한 확률적 탐색기법으로써, 설계 영역 내에 다수의 개체를 분포시켜 목적함수 값과 제약조건의 위반정도에 따라 각 개체의 적합도(fitness)를 부여한다. 그러므로 적합도가 클수록 다음 단계인 교배와 변이의 과정에 선택(selection)될 확률을 크게 하여 적합도가 높은 개체들이 다음 세대에 보다 많이 형성되고, 교배와 돌연변이의 과정을 통해 전체 개체들은 좋은 방향으로 탐색이 진행되게 한다.

유전자 알고리즘에 대한 연구는 1960년대 초반 Michigan 대학의 John Holland와 그의 연구진이 컴퓨터화된 유전학을 광범위한 함수의 최적화에도입함으로써 처음으로 소개되었다. 유전자 알고리즘은 1970년대에 Holland가 스키마타(schemata) 이론을 개발함으로써 유전자 스스로 적합한 생성물을 창조해 내는 이유에 대한 이론적인 기반이 제공되었고 이후 다양한 분야에서 적용되어 왔다(Berry and Linoff, 1997).

2. 유전자 알고리즘의 특징

유전자 알고리즘은 기존의 최적화 방법과는 다른 다음과 같은 특징이 있다.

첫째, 타부서치(tabu search), 시물레이티드 어닐링(simulated annealing)과 같은 이웃해(neighborhood solution) 탐색기법이 하나의 해를 운용하는 데 반하여, 유전자 알고리즘은 복수 개의 잠재된 해들로 이루어진 모집단

(population)을 운용한다는 것이다. 그러므로 이러한 해집단에 자연선택과 유전법칙의 메카니즘을 적용하여 세대(generation)를 진행시키면서 해공간을 탐색해 간다(김여근 등, 2004).

둘째, 문제의 표현을 숫자 배열로 코드화 하여 사용함으로써 교배 및 돌연변이와 같은 연산 수행이 단순하고 편리하다는 점을 들 수 있다. 이러한 문자열은 조합 최적화 문제에 맞게 이산적인 형태로 이루어져 있으므로 정수 및 이산형 설계변수가 혼합된 혼합형 최적화 문제에 효과적으로 사용할 수 있으며 탐색설계공간의 종류에 관계없이 유전자 알고리즘을 사용할 수 있다.

셋째, 함수값에 관한 정보만을 이용하기 때문에 미분 불가능한 문제에도 쉽게 적용할 수 있다. 이러한 특성은 유전자 알고리즘을 적용할 수 있는 문제의 범위를 확장시키며, 형상 최적화나 다목적 함수와 같은 문제에도 적용 가능하고, 다른 방법과의 결합도 쉽게 할 수 있도록 한다.

넷째, 일반적으로 최적화 방법에서는 탐색을 위한 변환과정을 확정론적인 방법을 사용한다. 이러한 탐색방법은 설계 공간과 초기점이 주어지면 최종 결과가 결정되어지므로, 초기점의 선정이 매우 중요한 요인이 된다. 그러나 유전자 알고리즘은 탐색과정을 확률론적으로 진행하므로 설계집단을 통한 탐색과 더불어 설계 공간 전체에 대한 정보 활용이 매우 용이하다(김기화, 1996).

3. 유전자 알고리즘의 기본 요소

일반적으로 유전자 알고리즘은 5가지 요소들을 가지고 있으며 이는 다음

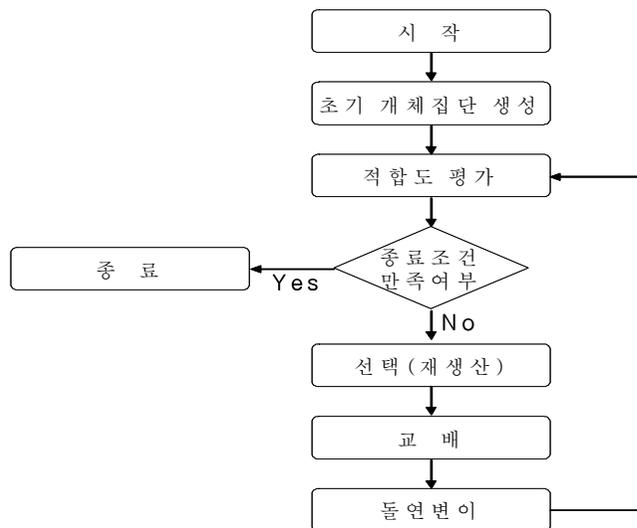
과 같다(Michalewicz, 1992).

- 1) 특정 문제에 대한 해(solution)의 유전자적 표현
- 2) 초기 시작해 집단의 생성 방법
- 3) 적합도에 따른 진화 함수 평가
- 4) 재생산(reproduction) 동안 자손의 유전자 조합을 변경할 유전자적 연산자
- 5) 유전자 알고리즘의 파라미터(parameters)에 대한 값

유전자 알고리즘은 t 세대 동안 개체 집단(population)인 $P(t)$ 를 유지한다. 각 개체들은 문제에 대한 잠재적인 해로 표현되며 주어진 척도들에 대해 적합도(fitness)가 평가되어진다. 주어진 개체들은 유전 연산자들에 의한 복잡한 변형과정(transformation)을 겪은 후 새로운 개체들을 형성한다. 이러한 변형과정에는 단일 개체 내에서의 변화에 의해 새로운 개체를 생성하는 돌연변이(mutation)와 2개의 개체들로부터 부분을 조합함으로써 새로운 개체를 생성하는 교배(crossover) 등 2가지 형태가 있다. 다음으로 새로운 개체 즉, 자손(offspring)인 $C(t)$ 가 평가되어진다. 새로운 개체 집단(new population)은 부모 개체 집단(parent population)과 자손 개체 집단(offspring population)으로부터 가장 적합한 개체들을 선택함으로써 형성되어진다. 몇몇 세대를 지남에 따라, 유전자 알고리즘은 이상적으로 특정 문제에 대한 최적(optimal) 또는 부분 최적(suboptimal) 해인 우수 개체(the best individual)로 수렴하게 된다.

4. 유전자 알고리즘의 구조

유전자 알고리즘은 문제를 표현하는 개체들로 이루어진 모집단으로부터 시작된다. 모집단은 매 세대마다 주어진 조건에 맞게 개체를 유지하고 매 세대에서 각 개체의 적합도(fitness)를 평가하여 이에 따라 자손 세대에서 생존할 개체들을 확률적으로 선택하게 된다. 선택된 개체들 중 일부의 개체들이 교배(crossover)단계를 거쳐 부모세대의 유전자를 자손세대에 상속한 후 돌연변이(mutation)연산을 통해 부모세대에 없는 속성을 도입하여 자손의 다양성을 높이는 과정을 수행한다. 그러므로 자손은 부모로부터 우수한 유전형질을 상속받는다고 가정함으로써 다음 세대가 평균적으로 이전 세대보다 더 좋아진다고 본다. 이러한 진화과정은 종료조건을 만족할 때까지 반복하게 된다.



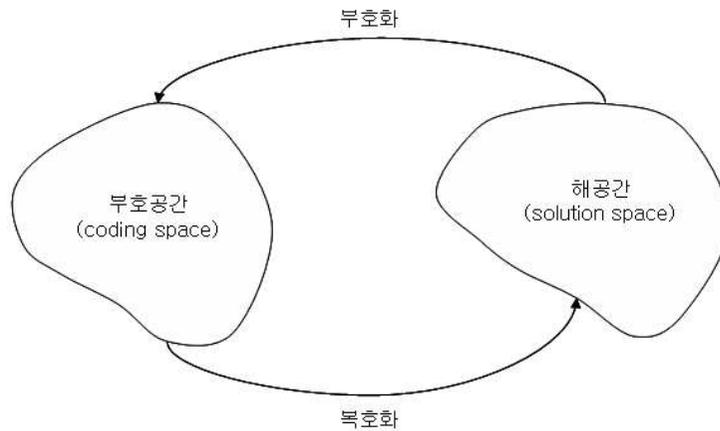
<그림 2-5> 유전자 알고리즘의 구조

<그림 2-5>는 이러한 유전자 알고리즘의 구조를 나타낸 것이다.

먼저 초기 개체집단을 무작위로 생성하여 일정한 형질을 갖는 염색체들의 집합으로 구성하여야 한다. 다음 세대로의 진화를 위해 초기 개체집단에 대해 적합도 함수로써 각 개체에 대해 적합도를 평가한다. 이후 주어진 종료 조건을 만족할 때까지 선택, 교배 및 돌연변이 과정을 거쳐 생성된 개체들에 대해 적합도 평가를 실시한다.

5. 유전자 알고리즘의 표현(Representations)과 관련된 문제

유전자 알고리즘은 자연 진화과정을 충실히 반영하기 위해 두 개의 공간, 즉 부호공간(coding space)과 해공간(solution space) 상에서 동작되도록 고안되어 왔다.



<그림 2-6> 유전자 알고리즘에서의 해공간과 부호공간

해공간 상의 파라미터는 연산에 적합한 부호 공간상의 자연 파라미터 (natural parameter)로 변환되고 유지된다. 변환된 결과를 스트링 또는 검색 체라 부른다. 이렇게 함으로써 검색체 내의 비트(형질)들은 자연계의 유전자로서의 역할을 반영하게 된다. 검색체로부터 적합도를 계산하는 작업은 해공간에서 이루어지기 때문에 검색체는 파라미터로 변환된다. <그림 2-6>은 해공간과 부호공간 사이의 관계를 보여주고 있다.

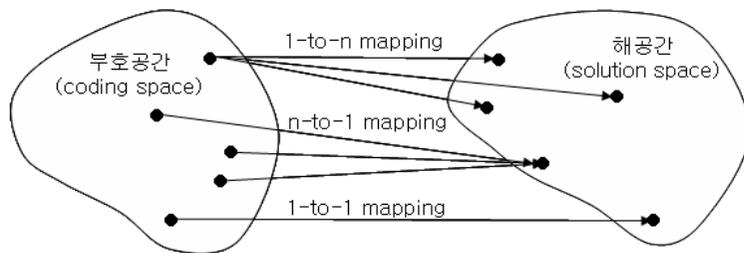
유전자 알고리즘은 선택적으로 두 가지 형태인 부호공간과 해공간, 달리 말하면 유전자형 공간 또는 표현형 공간에 영향을 미친다. 유전 연산자는 유전자 공간에 영향을 미치고, 평가와 선택은 표현형 공간에 영향을 미친다. 자연 선택은 검색체들과 복호화된 해의 수행 사이 연결고리이다. 유전자형 공간으로부터 표현형 공간으로의 매핑은 유전자 알고리즘의 수행에 상당한 영향력을 가지고 있다. 매핑과 관련된 주요한 문제는 몇몇 개체가 주어진 문제에 대한 실행불가능 해에 대응한다는 점이다. 이 문제는 제약 최적화 문제와 조합 최적화 문제에 있어 매우 엄격하게 적용된다.

다음으로 유전자 알고리즘의 표현방법에 있어 부호화와 해 사이의 매핑은 반드시 1대1 대응이어야 한다는 점이다. 해에 대한 코드화로부터의 매핑은 다음과 같이 3가지 경우 중 하나를 포함한다.

- 1) 1-to-1 mapping
- 2) n-to-1 mapping
- 3) 1-to-n mapping

가장 이상적인 매핑인 1-to-1 매핑은 자손을 생성할 때 어떠한 중(trival)의 연산이 발생되지 않는다. 만약 n-to-1 매핑이 발생한다면, 유전자 알고리

즘은 탐색동안 시간적 낭비가 발생한다. 각 개체가 유전자형 공간에서가 아닌 표현형 공간에서 복제(duplicated)되기 때문에 유전자형 공간에서의 거리 척도(distance measures)들은 동일한 개체들로 취급할 수 없다. 이것은 유전자 알고리즘이 성급하게 수렴하는 하나의 이유가 된다. 가장 이상적이지 못한 경우는 1-to-n 매핑인데, 이는 많은 가능 해들 사이에서 하나의 해를 결정하기 위해 표현형공간에서 또 다른 절차를 수행시킬 필요가 있기 때문이다(Gen & Cheng, 2000).



<그림 2-7> 부호공간에서 해공간으로의 맵핑(mapping)

유전자 알고리즘을 통한 해를 평가함에 있어 실행불가능성(infeasibility)과 위법성(illegality)들을 구분할 필요가 있다.

실행불가능성은 탐색체로 복호화된 어떠한 해가 주어진 문제의 실행가능 영역 외부에 위치하는 현상(phenomenon)을 일컫는 것인 반면, 위법성(illegality)은 탐색체가 주어진 문제에 대한 해로 표현하지 못하는 현상을 말한다. 탐색체의 실행불가능성은 제약 최적화 문제의 성질에 기인한다. 전통적인 방법이든 유전자 알고리즘이든 어떠한 기법을 사용하던 간에 이러한 제약들을 다룰 수 있어야 한다. 많은 최적화 문제에 대해 실행가능영역은 동질(equalities) 또는 비동질(inequalities)시스템으로써 표현되어질 수 있다.

많은 경우 페널티(penalty) 방법들이 실행 불가능 염색체를 다루기 위해 사용되어질 수 있다. 제약 최적화 문제들에 있어서, 최적점(optimum)은 전형적으로 실행가능영역과 실행불가능 영역사이의 경계에서 발생한다. 페널티 접근법은 실행가능영역과 실행불가능 영역으로부터의 최적점에 접근하기 위한 유전자적 접근을 강화시킨다.



<그림 2-8> 실행 불가능성(infeasibility)과 위법성(illegality)

염색체들의 위법성은 코드화 기법의 성질에서 기인한다. 많은 조합 최적화 문제에서, 문제정의(problem-specific)적 코드화 방법들이 사용되고, 그러한 코드화 방법들은 보통 단순한 이점(one-cutpoint)교배 과정에 의해 위법적인 자손을 생성한다. 위법적인 염색체는 특정 해로 복호화 될 수 없기 때문에, 페널티 기법은 이러한 상황에 적합하지 않다. 교정기법(repair technique)들이 위법적인 염색체를 적법한 염색체로 변환시키는데 사용되어진다. 예를 들면, 잘 알려진 PMX연산은 필수적으로 순열표현을 위해 이점(two-cutpoint)교배를 이용하고, 동시에 단순 이점교배에 의해 야기되는 위법성을 해결하기 위한 교정기법을 사용한다. Orvosh and Davis(1994)는 많은 조합 최적화 문제에서 실행불가능 또는 위법적인 염색체를 교정하는 것

은 상대적으로 용이하고, 교정전략은 거부(rejecting) 또는 페널티 전략과 같은 다른 전략을 능가한다는 것을 보여주었다.

6. 유전자 알고리즘의 염색체적 표현

앞서 언급한 바와 같이 유전자 알고리즘에서는 문제의 잠재 해를 표현하기 위한 기본적인 개체를 염색체(chromosome)라 하고, 각 개체를 이루는 요소를 형질(allele)이라 부른다.

일반적으로 개체의 선택은 염색체 단위로 이루어지며, 교배 및 돌연변이 연산은 형질 단위로 이루어지는 것이 보통이다.

유전자 알고리즘에서 모든 해는 염색체로 표현되어야 한다. 가장 전형적인 염색체 표현방법은 이진 코딩(binary encoding) 형태로 해를 표현하는 방법이다. 그러나 유전자 알고리즘이 많은 분야에 적용됨에 따라 k-ary 코딩(k-ary encoding), 실수 코딩(real-value encoding), 기호 코딩(symbolic encoding) 등 다양한 표현방법이 사용되고 있다.

1) 이진 코딩

Holland는 그의 연구에서 표현방법으로 이진 코드를 사용하였다(Holland, 1992). 이진 코딩은 초기 유전자 알고리즘이 이러한 형태를 취해왔기 때문에 아직도 폭 넓게 사용되고 있다. 이 표현 방법은 이론적 분석을 용이하게 해서 특정 유전자들의 성장과 쇠퇴를 예측할 수 있는 메커니즘을 제공한다(Goldberg, 1989).

이진 코딩은 해공간에서 가까운 두 점이 부호공간에서도 항상 가깝도록 표현되지 않는 단점을 가진다. 한 예로 해공간에서 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 해밍절벽이라고 한다. 어떤 상황에서는 해밍절벽이 유전자 알고리즘을 비전역 최적해 쪽으로 수렴케 하는 문제를 일으킨다 (Goldberg, 1991).

2) k-ary 코딩

k-ary 코딩은 염색체의 형질 값이 링크 유형의 가지수(k)가 가질 수 있는 값들 중 하나의 값으로 염색체를 표현하는 방법이다. 네트워크 설계 문제를 염색체로 표현하는데 있어서 k값은 링크 유형의 가지수로 정의할 수 있으며 링크 유형의 가지수가 4개인 경우, k가 가질 수 있는 값은 4개 ($k \in \{0, 1, 2, 3\}$)가 된다. 즉 두개의 노드를 연결한 링크의 존재 유무에 따라 0과 1이 아닌 값으로 나뉘지며 링크의 형태에 따라 1과 3사이의 값 중 하나를 가진다.

이러한 코딩 방법은 네트워크 설계 구조를 쉽게 표현할 수 있지만, 돌연변이 연산에서 똑같은 확률로 0과 3사이의 값 중 하나로 반전되기 때문에 돌연변이 연산의 효과를 제대로 얻지 못할 가능성이 존재한다.

3) 기호 코딩

이 방법에서 염색체는 다음과 같이 문제의 노드들을 의미하는 영문자의

열로 표시된다. 여기서 $v_i (1 \leq i \leq n)$ 는 이진표현에서 유전자로서의 의미를 가지며, 흔히 어떤 범위에서 취한 정수 집합의 원소가 된다. n 은 노드수로서 염색체의 길이와 일치하게 된다.

$$\text{염색체} : \{ v_1, v_2, \dots, v_n \}$$

이 코딩 방법의 특징은 유전자 간의 순서는 의미를 가지지만, 이들 간의 거리는 아무런 의미를 갖지 않는다.

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

4) 실수 코딩

실수 코딩은 염색체를 실수 값의 열로 표현하는 것이다. 실수 코딩은 함수 최적화 문제에 가장 많이 사용되어진다. 실수 코딩이 함수 최적화와 제약 최적화 문제에 대해 이진 코딩보다 더 효율적으로 수행되어진다는 것은 광범위하게 알려져 있다(Gen and Cheng, 2000).

실수 코딩에 있어 유전자형 공간의 위상 구조는 표현형 구조와 동등하기 때문에 효과적인 유전 연산을 수행하기가 용이하다. 그러나 실제 값의 이진 코딩은 하나의 수로 표현되어 사용된 비트들의 수에 의존하므로 정밀도의 손실을 겪게 된다. 이러한 결점은 실수 코딩에서도 염색체의 스트링들을 매

우 짧게 하여 유전자 알고리즘을 통해 매우 큰 스트링 크기를 수반하는 문제 해결에 있어 실패의 원인으로써 작용된다.

실수 코딩을 사용할 경우에 정밀도는 계산 환경에 따라 다르겠지만 더 많은 비트를 사용함으로써 이진 표현의 정확도가 더욱 개선될 수 있으나 계산 속도가 느리다는 단점이 있다.

7. 유전자 알고리즘의 연산자

1) 선택(selection) 연산

선택 연산은 자연법칙에 기초한다. 즉, 주어진 환경에 대한 적응 정도에 의해 현재 세대의 모집단으로부터 다음 세대에 생존할 개체를 선택하는 과정이다. 다시 말해, 주어진 환경에서 더 적합한 생물들이 더 높은 생존 기회를 갖도록 이들의 생존을 조절하는 기능을 말한다. 이러한 과정을 통하여 생물들은 생존경쟁에서 우위를 가지면 살아남고 그렇지 못하면 도태하게 되는데, 선택은 적자생존(survival of the fittest)과 자연 도태(natural selection) 현상을 모방하려는 인위적인 메커니즘이다.

유전자 알고리즘에서 선택연산은 모집단의 다양성과 우성개체의 선별이 조화를 이룰 수 있어야 한다. 높은 우성개체의 선별은 모집단의 개체들을 조기 수렴시키는 경향을 갖는다. 즉, 모집단의 다양성을 약화시켜 탐색공간의 다양한 탐색을 막음으로써 지역 해에 빠지는 결과를 가져오게 된다. 이와 달리, 낮은 우성개체의 선별은 모집단의 다양성은 유지되지만, 결과적으로 좋은 해를 효율적으로 탐색하지 못하여 무작위 탐색(random search)과

비슷한 결과를 초래할 수 있다. 따라서 모집단의 다양성을 유지시키면서 지역 해에 빠지지 않는 개체를 선택하는 것이 무엇보다 중요한 요소이다. 개체 선택방법으로는 여러 가지 방법이 존재하나, 크게 확률바퀴 선택(roulette wheel selection), 토너먼트 선택(tournament selection) 및 순위기반 선택(rank-based selection) 등이 대표적으로 사용되고 있다.

① 확률바퀴 선택

유전자 알고리즘에서 흔히 사용하는 선택방법은 적합도 비례 선택(fitness-proportionate selection)인데, 어떤 개체가 재생산될 것으로 기대되는 횟수는 그 적합도를 개체집단의 평균 적합도로 나눈 것과 같다 (Goldberg, 1989).

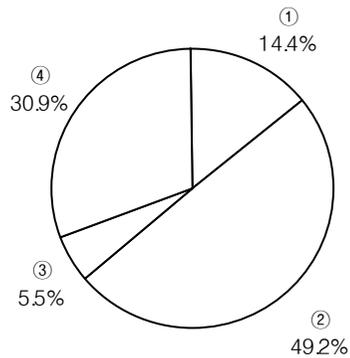
이러한 적합도 비례 선택을 구현하는 가장 대표적인 방법으로써 개체의 적합도에 비례하여 개체가 선택될 확률을 부과하는 확률바퀴 선택방법이다 (Holland, 1975).

예를 들어 적합도 함수를 $f(x) = x^2$ 로 가정하고 5개의 염색체를 가진 4개의 개체집단에서 각 개체에 대한 적합도와 선택확률이 <표 2-1>에 나타나 있다.

<표 2-1> 확률바퀴에 의한 선택확률

번호	염색체	적합도 : $f(x)$	선택확률(%)
1	(0 1 1 0 1)	169	14.4
2	(1 1 0 0 0)	576	49.2
3	(0 1 0 0 0)	64	5.5
4	(1 0 0 1 1)	361	30.9
합계		1170	100

선택확률은 각 개체의 적합도를 모든 개체의 적합도의 합으로 나눈 값이다. 예에서 개체 1, 2, 3, 4의 선택확률은 $169/1170$, $576/1170$, $64/1170$, $361/1170$ 이 된다. 이 경우 확률바퀴는 <그림 2-9>와 같이 된다.



<그림 2-9> 확률바퀴 예

실제 유전자 알고리즘에서는 0과 1사이의 난수를 발생시켜 하나의 개체를 선택하게 되며, 만약 4개의 개체를 선택한다면, 이와 같은 과정을 4번 반복

하여 다음 세대에 생존할 개체를 선택하게 된다.

② 토너먼트 선택

토너먼트 선택방법은 두 개 또는 그 이상의 개체들을 비교하여 그 중에서 생존할 개체를 선택하는 방법이다. 그 절차는 다음과 같다(김여근 외 2인, 2004).

단계 1. 토너먼트 크기 $k(\geq 2)$ 를 결정한다.

단계 2. 현재의 모집단에 있는 모든 개체를 임의의 순서로 나열한다.

단계 3. 나열된 개체들 중에서 처음 k 개의 개체 적합도를 비교하여, 그 중에서 가장 좋은 개체를 다음 세대에 생존시킨다. 그리고 비교된 k 개의 개체를 삭제한다.

단계 4. 나열된 개체가 모두 비교되었으면 현재 모집단의 개체들을 새로이 임의의 순서로 나열한다.

단계 5. 다음 세대에서 요구되는 개체의 수만큼 개체가 선별될 때까지 단계 3과 단계4를 반복한다.

이 방법에서 k 는 파라미터로 k 값이 높을수록 우수개체가 선택될 확률이 높아지지만 모집단의 다양성은 감소한다. 즉 k 가 크면 우수개체가 중복해서 여러 번 선택될 수 있어 다음 세대에 똑같은 개체가 복수로 생존하게 된다.

③ 순위기반 선택

확률바퀴 선택은 적합도의 값을 가지고 교배할 개체를 정하는 형태로서

적합도가 특히 높은 개체가 발생되면 그 개체의 영향으로 전체의 개체가 유사해지게 되어 조기 수렴의 문제가 발생할 수 있다. 반대로 적합도의 차이가 거의 없으면 개체의 선택 확률은 거의 균등해지기 때문에 임의의 선택과 유사한 효과가 있을 수 있다. 이와 비슷하게 토너먼트 선택은 모집단의 다양성을 유지하기 힘들다는 단점이 있다. 그러므로 확률바퀴 선택과 토너먼트 선택의 단점을 보완하기 위한 방법으로 순위기반 선택이 있다.

이 방법은 각 개체들의 적합도의 크기 순서에 따라 순위를 매긴 다음, 순위에 따라 다음 세대에 자손을 남길 확률을 결정하는 방법으로써, 가장 좋은 염색체로부터 일차 함수적으로 적합도를 배정하는 방법이다.

이 선택법은 확률바퀴 선택에 비해 적합도가 높은 염색체와 적합도가 낮은 염색체 사이의 적합도 차이가 크게 나는 것을 막을 수 있다.

2) 교배(crossover) 연산

자연계의 생물들은 일반적으로 암수의 유성생식(sexual mating)을 통해 자손을 생성하고, 이러한 자연계의 유성생식과정을 알고리즘 형태로 흉내낸 것이 교배이다. 교배연산은 좋은 해를 이용하는 역할을 담당하는 연산으로써, 부모의 좋은 형질이 가능한 파괴되지 않고 자손에 상속될 수 있어야 한다. 교배 연산자는 무작위로 특정 위치를 선택하고, 부모의 두 염색체 사이에 그 이전과 이후 형질의 일부를 교환하여 두 개의 자손을 형성하며, 이러한 연산은 교배된 자손의 수가 부모 집단의 크기와 같을 때까지 반복된다.

자연계의 유성생식을 알고리즘 형태로 구현한 교배의 방법으로는 일점 교배(one-point crossover), 다점 교배(multi-point crossover), 균등 교배(uniform crossover), 순환 교배(cycle crossover), 순서 교배(order

crossover), 부분 사상 교배(partially mapped crossover) 및 산술적 교배(arithmetical crossover) 등이 있다.

① 일점 교배

일점 교배는 가장 고전적인 교차방법으로 흔히 표준교배로도 불린다. 이는 염색체 사이의 임의의 한 점(절단점 또는 교차점)을 기준으로 절단하여 두 부모로부터 각각 절단된 한 부분의 스트링씩을 상속받아 자손을 생산하는 방법이다. 그러므로 길이가 n 인 일차원 문자열로 된 염색체 상에서 일점 교차로 자르는 방법의 총 수는 $n-1$ 가지이다. 만약 절단점이 3번째와 4번째 형질 사이인 일점 교배의 예는 다음과 같다.

부 모 1	(0 1 0	1 1)
부 모 2	(1 0 0	1 0)
자 손 1	(0 1 0	1 0)
자 손 2	(1 0 0	1 1)

② 다점 교배

다점교배는 일점교배와 비슷하나 그 절단점이 2개 이상 다수이므로 일점 교차보다 자르는 방법의 수가 더 다양하다.

염색체의 길이가 n 이라고 하면, k 개의 절단점을 기준으로 자르는 방법의 총수는 ${}_{n-1}C_k$ 가지가 된다. 다점교차는 일점교차보다 교란(perturbation)의 정도가 크다. 그러므로 보다 넓은 탐색공간을 탐색할 수 있는 반면에 교란이 강하면 수렴성이 떨어져 주어진 시간 내에 제대로 수렴하지 않을 가능성

이 존재한다. 만약 3개의 절단점을 가지고 다점교차를 실시하고자 한다면 다음과 같다.

부 모 1	(1 0	0 1 1 0	0 1 1	0)
부 모 2	(0 1	1 0 0 1	1 0 0	1)
자 손 1	(1 0	1 0 0 1	0 1 1	1)
자 손 2	(0 1	0 1 1 0	1 0 0	0)

교란이 강하다는 것은 염색체 구조가 파손될 확률이 높다는 것이지만 대신 새로운 염색체의 생성은 더 다양해질 수 있다는 의미이다. 그러므로 항상 최적인 교란의 정도란 있을 수 없다.

일반적으로 다점교차는 순수 유전자 알고리즘보다 혼합형 유전자 알고리즘에 더 어울린다. 왜냐하면 혼합형 유전 알고리즘에는 다소의 지역 최적화 기능이 있으므로 순수 유전자 알고리즘보다 교란에 대한 회복력이 강하다. 혼합형 유전자 알고리즘의 경우 교란의 정도가 너무 미약하면 후반부에 부모 해와 같은 자식해가 만들어질 확률이 높아져 탐색시간의 낭비가 더 커진다.

③ 균등 교배

일점 교배와 다점 교배가 절단점을 이용하여 이루어지는 데 반해 균등 교배는 절단점을 이용하지 않는다. 균등 교배는 먼저 임계 확률 P_0 를 설정한다. 가장 일반적인 임계 확률은 0.5이다. 두 부모 해의 유전자 위치에 대해 난수를 발생한 다음 이 값이 P_0 이상이면 부모 1의 같은 위치로부터 형질을

복사하고, 그렇지 않으면 부모 2의 같은 위치로부터 형질을 복사한다 (Syswerda, 1989). 다음은 $P_0 = 0.6$ 인 균등 교배의 한 예이다.

부 모 1	(1 0	0 1 1 0	0 1 1	0)
부 모 2	(0 1	1 0 0 1	1 0 0	1)
난 수	.8 .7	.3 .8 .9 .6	.4 .7 .2	.1
자 손	(1 0	1 1 1 0	1 1 0	1)

균등 교배는 절단점을 이용하지 않으므로 일점 교차나 다점 교차에 비해 스키마의 결합 형태가 다양하다. 즉, 스키마 내의 특정 기호의 위치가 영향을 미치지 않는 대신 교란의 정도가 크므로 수렴 시간이 오래 걸린다(문병로, 2001).

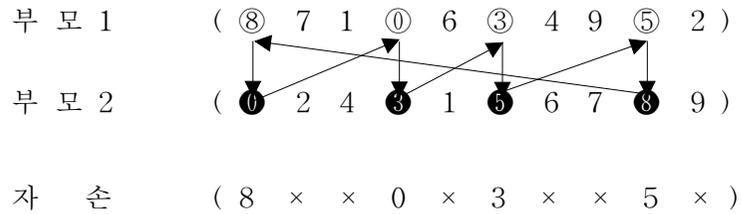
④ 순환 교배

Oliver(1987)에 의해 제안된 순환 교배는 TSP(Traveling Salesman Problem)의 예에서와 같이 염색체가 순열로 표현되는 경우에 적용 가능한 교배 연산이다. 아래와 같이 두 부모 해로부터 순환 교배를 한다고 하자.

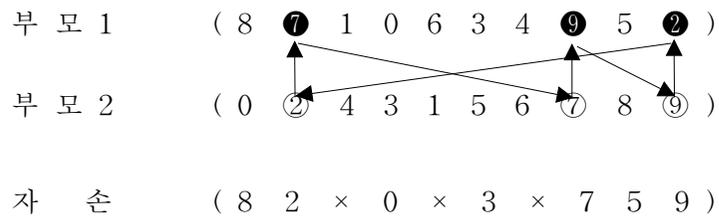
부 모 1	(8 7 1 0 6 3 4 9 5 2)
부 모 2	(0 2 4 3 1 5 6 7 8 9)

부모 1의 첫 번째 형질(8)로부터 복사를 시작한다. 방금 복사한 위치에 대응되는 부모 2의 유전자는 0이므로 부모 1에서 0이 있는 위치를 찾아 자손

해의 같은 위치에 0을 복사한다. 같은 원리로 부모 1의 염색체에서 3이 있는 위치를 찾아 자손 해의 같은 위치에 5를 복사한다. 이제 5와 같은 위치에 있는 8은 이미 복사가 한 번 이루어졌으므로 더 이상 진행 할 수 없다. 이 과정을 아래와 같이 화살표를 따라 표시해 보면 하나의 순환경로가 만들어진다는 데에서 순환 교배라는 이름이 붙게 되었다(문병로, 2001).

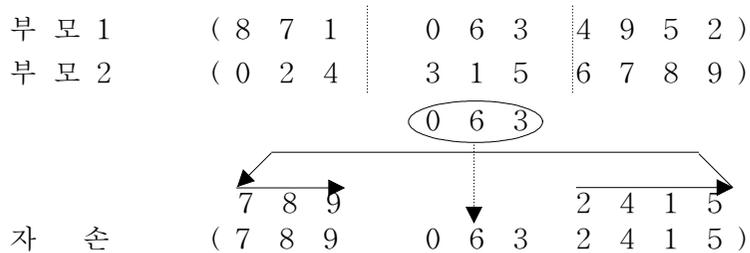


다음은 자손의 형질 값이 아직 결정되지 않은 위치들 중 가장 왼쪽의 위치로부터 시작하는데 이번에는 부모 2로부터 시작한다. 2, 7, 9 순으로 복사하면서 아래 그림처럼 다시 순환 경로가 만들어진다. 이렇게 부모 1과 부모 2를 번갈아 시작해 가면서 더 이상 결정되지 않은 유전자가 없을 때까지 계속하면 최종적인 자손은 { 8 2 1 0 6 3 4 7 5 9}가 된다.



⑤ 순서 교배

순서 교배 역시 염색체가 순열로 표시되는 경우를 위하여 고안된 교차 연산자이다. 두 부모 해가 주어졌다면, 먼저 임의로 두 개의 절단점을 정한 다음 두 절단점 사이에 있는 부분을 부모 1로부터 복사한다(063). 나머지 위치는 부모 2로부터 복사하되, 두 번째 절단점 바로 다음 위치부터 시작하여 사용되지 않는 기호들 중 부모 2에서 나타난 순서대로(2 4 1 5 7 8 9) 복사한다(David, 1985).



⑥ 부분 사상 교배

부분 사상 교배 역시 염색체가 순열로 표시되는 경우를 위해 고안된 교배 연산자이다(Goldberg & Lingle, 1985). 두 부모에 대해 두 개의 절단점을 정한 다음 두 절단점 사이에 있는 부분을 순서 교배와 같이 부모 1로부터 복사한다. 나머지 위치는 부모 2로부터 복사하되 만일 해당 값이 이미 사용되었으면 같은 값을 가진 부모 1의 위치를 찾아 같은 위치의 부모 2로부터 복사한다(0과 6). 이 값은 아직 사용되지 않을 수도 있고(1), 이미 사용되었을 수도 있다(중간 과정의 ③). 마지막으로 이미 사용되어 중복이 일어난 유전자는 교체된다.

부 모 1	(8 7 1		0 6 3		4 9 5 2)
부 모 2	(0 2 4		3 1 5		6 7 8 9)
			0 6 3		
			① 2 4		③ 7 8 9
			③ 2 4		1 7 8 9
자 손	(7 8 9		0 6 3		2 4 1 5)

⑦ 산술적 교배

산술적 교배는 실수 염색체를 사용하는 경우에 사용할 수 있는 교배 연산이다(Michelewicz, 1992). 염색체의 각 위치에 대해 두 부모 염색체의 두 형질 값의 평균을 내어 자식해의 해당 위치의 값으로 배정한다. 다음은 간단한 산술적 교차의 한 예이다(문병로, 2001).

부 모 1	1.98	3.31	20.43	12.01	-2.34	8.34	98.86
부 모 2	11.28	2.21	12.39	1.44	2.45	3.55	87.44
자 손	6.63	2.76	16.41	6.73	0.06	5.95	93.15

산술적 교차는 수의 '크기'라는 개념을 교차 행위에 사용하는 점에 큰 매력이다. 그렇지만 수들의 산술 평균을 지향하므로 매우 빠른 수렴을 보인다. 변이 등을 적절히 조절하여 조기 수렴을 막을 수 있어야 한다.

3) 돌연변이(mutation) 연산

진화가 진행되는 동안 선택 연산과 교배 연산은 집단을 더욱 강하게 해주지만 이로 인해 각 염색체들은 서로 유사한 형태로 닮아가게 된다. 이러한 현상이 세대 말기에 진행되면 바람직한 형태로 나타나지만 세대 초기에 발생할 경우 유전자의 다양성 결핍으로 지역 최적해(local optimal solution)나 사점(deal corner)에 빠지게 되는 원인으로 작용하게 된다.

선택 연산과 교배 연산은 지역 최적해나 사점으로부터 벗어날 수 있는 메커니즘을 근본적으로 가지고 있지 않다. 즉 세대 내 염색체들의 특정위치 비트가 모두 동일하게 되면 선택 연산과 교배 연산은 자체적으로 이를 변경할 수 없게 된다는 것이다. 예를 들어 다음의 5개의 개체가 아무리 많은 수의 교배과정을 거치더라도 4번째 형질과 10번째 형질은 각각 0과 1외의 형질은 가질 수가 없게 된다.

부 모 1	(1 0 0	0	1 0 0 1 1	1)
부 모 2	(0 1 1	0	0 1 1 0 0	1)
부 모 3	(1 1 1	0	1 1 0 0 0	1)
부 모 4	(0 1 1	0	0 0 1 1 0	1)
부 모 5	(0 0 0	0	1 1 1 0 1	1)

이러한 원치 않는 해로부터 벗어나기 위해 유전자 알고리즘에서 제공되는 메커니즘이 돌연변이 연산이다. 현실에서의 돌연변이라는 의미는 좋지 않은 이미지로써 받아들여지고 있지만 유전자 알고리즘에 있어서는 개체의 다양화에 중요한 역할을 하게 된다.

돌연변이 연산을 통해 생성되는 새로운 개체는 우수한 염색체를 가질 수도 있지만 그렇지 못한 경우도 발생할 수 있다. 그러나 새로 만들어진 열성 개체를 이전의 개체와의 교배를 통해 양호한 개체 또는 매우 우수한 개체로 나타날 수 있다. 만일 돌연변이 연산을 너무 많이 하게 되면 특정 값으로 수렴을 하는 것이 아닌 무작위(random) 방식의 탐색만 행해지는 결과를 가져옴으로써 유전자 알고리즘의 의미를 상실하게 된다. 반면에 너무 적은 돌연변이는 주어진 초기 형질의 조합 이외의 공간은 탐색할 수 없으며, 결국 원하는 최적해를 구할 수도 없게 된다.

그러므로 돌연변이 연산은 부모 세대에 없는 형질을 도입하여 모든 염색체의 특정 비트가 고정되는 것을 방지하고 탐색공간을 넓히려는 시도로써 지역 최적해에 빠질 확률을 줄여주는 효과를 가진다.

돌연변이 연산에도 다양한 방법이 존재하지만 본 논문에서 사용된 균등 돌연변이(uniform mutation)가 대표적이다. 즉, 각각의 유전자에 대해서 0과 1사이의 난수를 발생시켜 미리 정한 임의의 임계값 미만의 수가 나오면 해당 유전자를 임의로 돌연변이를 일으키게 하고 그 이상의 수가 나오면 돌연변이를 일으키지 않게 하는 형태이다. 전형적인 임계값으로는 15%를 들 수 있으나 문제의 형태와 유전자 알고리즘의 형태에 따라 상당한 폭(10%~30%)으로 조정될 수 있다. 이외에 유전자 알고리즘의 초기 해는 품질이 낮고 시간이 지남에 따라 해가 상당한 수준에 이른 후반에는 돌연변이가 강하면 거의 품질 향상이 일어나기 힘들다는데서 착안하여 유전자 알고리즘이 진행됨에 따라 점점 돌연변이의 강도를 줄여가는 비균등 돌연변이(non-uniform mutation)가 있다(Michalewicz, 1992). 이외에 기타 k개의 연속된 유전자를 취하여 값을 뒤집는 형태(Colorni 등, 1991), 일련의 연속된 유전자를 혼합하는 방법(Davis, 1991), 임의의 두 유전자 값을 서로 치환하

는 방법(Laszewsk, 1991) 등이 있다.

4) 종료조건

개체 집단이 진화를 거듭하게 되면서 집단 내의 우수한 해쪽으로 수렴해 갈 때 언제 정지할 것인지를 결정하는 것도 중요한 유전자 알고리즘의 요소 중에 하나이다. 일반적으로 다음의 두 가지 방법 중 한 가지를 이용하여 진행되고 있는 유전자 알고리즘을 종료시키게 된다.

첫째, 현재의 세대 또는 반복횟수가 사전에 지정한 지정 값(예를 들면, 최대 세대 수)에 도달하면 종료시키는 것이다. 개체 집단이 해 가까이 수렴해 가면 개체들의 대부분은 한 세대에서 다음 세대로 아무런 변경 없이 생존해 간다. 이 경우에는 모든 개체들의 적합도를 평가할 필요가 없어지게 되며, 이 경우 지정한 세대에 도달하게 되면 종료한다. 다시 말하면 유전자 알고리즘을 반복 수행하면서 더 이상 상당한 개선이 일어날 가능성이 매우 적다고 판단될 때의 세대를 종료 조건으로 선택하는 방법이다. 그러나 이 방법은 사용자가 탐색공간의 특성을 사전에 잘 알고 있다는 가정 하에서 성립하게 된다. 탐색공간의 특성은 탐색의 길이에 영향을 주기 때문에 최대 세대 수를 임의적으로 정하기보다는 주어진 문제가 사전에 연구되었을 경우나, 주어진 문제에 대해 시행착오적으로 결정하는 것이 바람직하다.

두 번째 방법은 개체의 유전자형에 기초한 것과 표현형에 기초한 것으로 다시 나눌 수 있다. 전자는 매 세대마다 염색체들의 유전자 수를 조사하여 수렴성을 판단하는 것으로써, 수렴된 유전자의 수가 총 유전자 수의 일정비율을 넘게 되면 탐색을 종료한다. 후자는 매 세대마다 얻어지는 잠정적인 해로부터 수렴성을 조사하는 것인데 해의 정밀도를 나타내는 양의 상수 값

만큼 개선되지 않을 경우 종료시키는 방법이다(박우열, 2003).

8. 네트워크 설계를 위한 유전자 알고리즘

앞서 설명한 바와 같이 최근 유전자 알고리즘 중심적 접근방법이 신뢰성을 고려한 최적 네트워크 설계문제에 있어 새로운 문제해결 방법으로 주목받기 시작하고 있다.

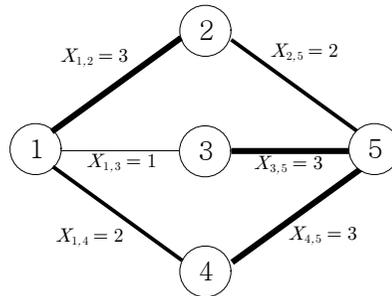
유전자 알고리즘을 이용한 신뢰성 있는 네트워크 설계와 관련된 연구로써, Kumar 등(1995-b)은 분산 시스템 위상설계에 있어 유전자 알고리즘을 제안하였다. Kumar 등(1995-a)은 또한 컴퓨터 네트워크 설계와 네트워크 확장에 있어 직경(diameter), 거리(distance) 그리고 신뢰도를 고려하는 유전자 알고리즘을 제안하였다. Walters와 Smith(1995)는 트리 구조를 가진 최적 네트워크 설계를 위해 진화 알고리즘(evolutionary algorithm)을 제안하였다. Deeter와 Smith(1998)는 유전자 알고리즘을 이용하여 신뢰도를 제약 조건으로 하고 비용을 최소화하는 경제적 네트워크 설계 방법을 제안하였다. 이들은 수준에 따라 다양한 비용과 신뢰도가 부여된 링크를 대상으로 유전자 알고리즘적인 접근방법을 통해 전체 네트워크 신뢰도를 고려한 백본 네트워크 설계방법을 제시하였다. 이와 유사하게 Dengiz 등(1995)도 유전자 알고리즘을 이용하여 네트워크 신뢰도 제약 하에서의 대규모 백본 통신망 설계에 관한 연구를 실시하였다.

본 연구 중 경제적 네트워크 확장설계 문제에 있어 중요한 사전연구로써 Deeter와 Smith(1998)의 연구는 다양한 비용과 신뢰도 수준을 고려함으로써 단일 비용 및 신뢰도 수준을 고려한 이전의 연구를 보다 확장시켰다. 그러

므로 경제적 네트워크 확장설계 문제를 다루기 전에 Deeter와 Smith의 유전자 알고리즘을 고려한 백본 네트워크 설계연구를 살펴보고자 한다.

1) 염색체 표현

각 네트워크 설계 구조 x 는 정수 벡터(integer vector)로 쉽게 변환되기 때문에 유전자 알고리즘의 염색체로 사용되어질 수 있다. 염색체의 각 요소는 네트워크 설계에서 연결 가능한 링크를 표현하기 때문에 각 후보 네트워크 구조 x 에는 $n \times (n-1)/2$ 개의 벡터 요소들이 존재할 수 있다. 각 형질 값은 특정 링크의 연결 형태를 표현함으로써 한 개의 링크는 연결된 두 개의 노드로 구성됨을 의미한다. 예를 들어, 5개의 노드와 4개의 링크 유형 가지수(k)를 가진 네트워크는 <그림 2-10>과 <표 2-2>로 나타낼 수 있다.



<그림 2-10> 네트워크 설계 예

5개의 노드와 4개의 링크유형 가지수를 고려한다면, 10개($(5 \times 4)/2$)의 연결 가능 링크가 존재하지만, <그림 2-10>의 예에서는 6개의 연결링크만으로 구성되어짐을 볼 수 있다.

<표 2-2> 노드 간 링크 유형

	1	2	3	4	5
1	-	3	1	2	0
2	3	-	0	0	2
3	1	0	-	0	3
4	2	0	0	-	3
5	0	2	3	3	-

<표 2-2>에서 보는 바와 같이, 행렬은 대칭적인 형태를 가지고 있으며, 위상향 삼각형에 이러한 구조정보가 기록되어진다. 이러한 정보는 다음과 같이 염색체로 표현되어질 수 있다.

염색체 : {3 1 2 0 0 0 2 0 3 3}

단, 염색체의 각 형질이 가질 수 있는 값은 $0, 1, \dots, k-1$ 개 이므로, 네트워크 구조의 탐색공간은 $k^{\{n \times (n-1)\}/2}$ 이다.

2) 적합도 계산

Deeter와 Smith의 연구에서 유전자 알고리즘은 주어진 네트워크 신뢰도 (R_0)이상에서 최소 비용을 갖는 네트워크 구조를 찾고자 하는 목적으로 이용된다. 여기서, 여러 가지 후보 해 중 실행 불가능(infeasible) 네트워크 구조를 가진 후보 해를 고려하여야 하는데, 이를 위해 적합도 함수가 이용된다. 실행 불가능해 모두가 불필요한 정보를 가지는 것이 아니고 경우에 따라 유용한 정보를 가질 수도 있다. 즉, 두 개의 실행 불가능 해로 이루어지

거나, 각각 한 개의 실행 가능해와 실행 불가능해로도 우수한 실행 가능 해를 생산할 수도 있다는 의미이다. 또한 최소비용 네트워크 설계 문제에 있어 하나의 제약 조건만이 주어지기 때문에 최적 네트워크 설계 구조는 실행 가능 설계구조와 실행 불가능 설계구조 사이의 경계에 놓이는 경우가 많다. 그러므로 적합도 함수는 다음과 같이 주어진다.

$$C_p(X) = C(X) + C(X^*) \times (1 + R_0 - R(x))^{r_p + (s \times g/50)}$$

여기서,

$C_p(X)$: 페널티 비용

$C(X)$: 비 페널티 비용

$C(X^*)$: 개체 집단 내의 가장 우수한 실행 가능해의 비용

r_p : 페널티 비율

s : 개체 집단의 크기

g : 개체 수

3) 초기 개체집단 생성

초기 개체집단은 무작위로 다음의 과정에 따라 생성되었다.

단계 1. 링크의 신뢰도와 비용 옵션에 따라 무작위로 초기 개체집단을 생성한다.

단계 2. 초기 개체집단을 신뢰도 계산 과정으로 보낸다.

단계 3. 초기 개체집단을 비용 계산 과정으로 보낸다. 만약 실행 불가능한

염색체가 존재한다면, 그 염색체는 페널티 비용(penalty cost)이 부과된다.

단계 4. 초기 염색체를 검사한 후, 가장 우수한 염색체를 저장한다. 만약 어떠한 염색체도 실행 가능 염색체가 아니라면, 가장 우수한 실행 불가능 염색체가 저장된다.

4) 선택 연산

순위 기반 2차 선택(rank-based quadratic selection) 과정을 이용하여 다음과 같은 과정에 따라 두 개의 부모 개체가 선택되어진다(Tate and Smith, 1995).

단계 1. 적합도 수준에 따라 개체를 내림차순 정렬한 후 순위를 부여한다.

여기서, 가장 높은 적합도를 가진 개체를 1로 둔다.

단계 2. 0과 $\sqrt{\text{개체집단크기}}$ 사이의 난수를 생성한다.

단계 3. 생성된 난수를 제공한다.

단계 4. 제공된 수에서 소수점 이하를 버린다.

단계 5. 1을 제공된 수에 더한다.

단계 6. 결과에 따라 첫 번째 부모개체를 선택한다. 다음으로 두 번째 부모개체를 선택하기 위해 단계 2로 간다. 만약 동일한 개체가 선택된다면, 상이한 개체를 선택할 때까지 반복한다.

예를 들어, 개체 집단의 크기가 20이라면, 순위기반 2차 선택과정은 아래와 같다.

단계 1. 20개의 염색체를 적합도에 따라 내림차순 정렬한 후 순위를 부여한다.

단계 2. 0과 $\sqrt{20}$ 사이의 난수를 생성한다.

단계 3. 생성된 난수가 2.5라면, 제공된 수는 6.25가 된다.

단계 4. 만약 6.25를 소수점 이하 버림을 수행하면, 6이 된다.

단계 5. 6에 1을 더하면, 그 수는 7이 된다.

단계 6. 7번째 개체를 선택한다.

5) 교배 연산

다음과 같이 균등교배를 실행한다.

단계 1. 선택과정에 따라 2개의 개체를 선택한다.

단계 2. 교배비율에 따라 각 개체의 한 형질을 자손 염색체로 복사한다.

단계 3. 자손 염색체가 완전해질 때까지 단계 2를 반복한다.

6) 돌연변이 연산

돌연변이는 새로운 유전자 자원을 개체집단에 포함시키는 과정이며, 유전자 알고리즘을 통해 생성된 해가 지역 최적해(local optimal solution)로 빠질 확률을 줄이기 위한 과정이다.

단계 1. 돌연변이 대상 염색체를 선택하기 위해 0과 1사이의 실수형 난수를 생성한다.

단계 2. 만약 생성된 난수가 돌연변이 염색체 비율($m\%$)보다 작다면, 그 염색체가 돌연변이 대상 염색체로 선택되고 단계 3으로 간다. 그렇지 않으면, 단계 6으로 간다.

단계 3. 돌연변이 대상 형질을 선택하기 위해 0과 1사이의 실수형 난수를 생성한다.

단계 4. 만약 생성된 난수가 돌연변이 형질의 비율(r_m)보다 작다면, 단계 5로 간다. 그렇지 않으면, 단계 6으로 간다.

단계 5. $\{0, 1, 2, \dots, k-1\}$ 에서 무작위로 형질을 선택한다.

단계 6. 돌연변이 과정을 종료한다.

7) 신뢰도 계산

다음의 backtracking 알고리즘(Ball and Van Slyke, 1977)을 사용하여 정확한 시스템 비신뢰도($1 - R(x)$)를 구한다.

단계 1. (초기화) 모든 링크를 free로 표시하고, 스택(stack)을 초기화한다.

단계 2. 수정된 cutset을 생성한다.

(a) 네트워크 cutset을 구성할 모든 비작동(inoperative) 링크로 구성된 free 링크의 집합을 찾는다.

(b) 단계 (a)에서 찾은 모든 링크들을 비작동으로 표시하고 이를 스택에 넣는다.

(c) 현재 스택에 들어있는 것은 수정된 cutset이고 이들의 확률을 누적합계 한다.

단계 3. Backtrack

- (a) 만약 스택이 비었다면, 단계 4로 간다. 그렇지 않다면 단계 (b)로 간다.
- (b) 스택의 가장 상단에 있는 링크를 꺼낸다.
- (c) 만약 링크가 비작동이고, 이것을 작동시켜 신장트리(spanning tree)를 이루게 되면, 이 링크를 free로 표시하고 단계 (a)로 간다.
- (d) 만약 링크가 비작동이고, 이것을 작동시키더라도 신장트리를 이룰 수 없다면, 이 링크를 operative로 표시하고 스택의 마지막에 넣은 후 단계 2로 간다.
- (e) 만약 링크가 작동중이면, 이 링크를 free로 표시하고 단계 (a)로 간다.

단계 4. 네트워크 비신뢰도를 반환하고 신뢰도 계산과정을 종료한다.

8) 전체 알고리즘

단계 1. 매개변수를 초기화한다.

단계 2. 초기화

- (a) 초기 세대의 개체집단을 무작위로 생성한다.
- (b) 초기 세대의 개체집단을 신뢰도 계산함수로 보낸다.
- (c) 초기 세대의 개체집단을 비용 계산함수로 보낸다. 단, 실행 불가능 해가 존재하면 패널티 비용을 부과한다.
- (d) 초기 세대의 가장 우수한 해를 검사한다. 만약 실행 가능해가 하나도 존재하지 않는다면, 가장 우수한 실행 불가능 해가 저

장된다.

단계 3. 선택

- (a) 가장 우수한 개체를 새로운 개체집단으로 포함시킨다.
- (b) 순위 기반 2차 선택과정에 의해 현재 개체집단으로부터 두 개의 상이한 후보 개체를 선택한다.

단계 4. 교배 및 돌연변이 과정을 수행한다. 두 개의 자손 개체를 획득하기 위해 균등 교배 연산자를 수행한다. 그 다음으로 생성된 개체에 대해 돌연변이과정을 적용한다.

단계 5. 생성된 자손 개체의 수를 검사한다. 만약 개체 수가 개체집단의 크기보다 작다면, 단계 3으로 간다. 그렇지 않으면, 단계 6으로 간다.

단계 6. 새로운 개체집단을 생성한다. 즉, 생성된 자손개체에 의해 부모개체가 교체된다.

단계 7. 평가

- (a) 새로운 개체집단을 신뢰도 계산함수로 보낸다.
- (b) 새로운 개체집단을 비용 계산함수로 보낸다. 만약 실행 불가능 해가 존재한다면, 페널티 비용을 부과한다.

단계 8. 새로운 개체 중 가장 우수한 해를 검사한다. 새로운 우수 해가 이전 우수 해보다 더 우수하다면 새로운 우수 해를 저장한다. 만약 실행 가능 해가 하나도 존재하지 않는다면, 가장 우수한 실행 불가능 해가 저장된다.

단계 9. 종료조건을 검사한다. 만약 현재 세대 수가 최대 세대 수보다 작다면 단계 3으로 간다. 그렇지 않으면 종료한다.

제 3 장 경제적 백본(Backbone) 네트워크 확장 설계

제 1 절 경제적 백본 네트워크 확장 설계 문제의 표현

백본 네트워크 확장설계는 네트워크 사용자의 요구에 따라 새로운 노드와 링크를 추가함에 있어 기존 네트워크 효율성을 유지하거나 보다 향상된 형태로 이루어져야 한다. 즉, 새로운 노드나 링크를 기존 네트워크에 부가하는 것으로 한정하지 않고, 최소한의 설치비용과 최대한의 네트워크 신뢰도를 보장할 수 있는 네트워크 설계가 중요한 문제이다.

본 장에서는 주어진 비용 및 신뢰도 제약조건을 만족하면서 최소 네트워크 설치비용 및 최대 네트워크 신뢰도를 달성할 수 있는 경제적 백본 네트워크 확장 설계방법을 제시한다. 특히, 백본 네트워크와 관련된 기존의 대표적인 연구로써 Deeter와 Smith(1998)의 연구를 백본 네트워크의 확장 설계 문제에 적용하고자 한다.

본 연구에서 제안하는 경제적 백본 네트워크 확장 설계 문제를 유전자 알고리즘에 적용함에 있어 필요한 기본 가정 및 표기 형식은 다음과 같다.

1. 기본 가정 및 표기 형식

백본 네트워크 확장 설계를 위한 가정은 다음과 같다.

- 각 노드의 위치는 주어져 있다.

- 노드들은 완전히 신뢰할 수 있다.
- 각 링크는 무방향적(bi-directional)이다.
- 네트워크에 있어 중복된 링크는 존재하지 않는다.
- 링크의 상태는 작동(operational)이거나 고장(failed) 상태로 한정한다.
- 링크의 고장확률은 독립적이다.

백본 네트워크 확장 설계와 관련된 표기는 다음과 같다.

- E : 네트워크 설계에 있어 연결된 링크들의 집합
- $\{i, j\}$: 노드 i 와 j 를 연결하는 링크
- $x_{i,j} (\in \{0, 1, 2, \dots, k-1\})$: $\{i, j\}$ 의 링크 유형. 여기서, k 는 링크 유형의 가지 수
- $\mathbf{x} (= \{x_{1,2}, x_{1,3}, \dots, x_{n-1,n}\})$: 네트워크 설계 구조
- $C(\mathbf{x})$: 네트워크 설계의 총 비용
- $R(\mathbf{x})$: 네트워크 설계의 신뢰도
- R_0 : 요구된 최소의 네트워크 신뢰도
- C_0 : 요구된 최대의 네트워크 설치비용

2. 백본 네트워크 확장 문제의 모형

본 연구에서 네트워크 확장 설계 문제에 있어 네트워크 신뢰도를 제약조건으로 하고 비용을 최소화하는 문제는 다음과 같이 모형화될 수 있다.

목적함수 :

$$\text{Minimize } C(\mathbf{x}) = \sum_{\{i,j\} \in E} c_{ij}$$

제약조건 : $R(\mathbf{x}) \geq R_0$

여기서 c_{ij} 는 링크 $\{i, j\}$ 의 비용을 의미하며, $c_{ij} = a_{ij} \cdot d_{ij}$ 로 표현될 수 있다. a_{ij} 는 단위거리 당 링크 $\{i, j\}$ 의 비용을, d_{ij} 는 링크 $\{i, j\}$ 의 거리를 의미한다.

기존 네트워크에 설치된 링크를 더 나은 수준으로 교체하는 경우에는 새 링크 설치비용 이외에 추가적인 교체비용이 소요되므로 $c_{ij} = a_{ij} \cdot d_{ij} \times (1 + r_e)$ 로 표현될 수 있다. 여기서 r_e 는 교체비용율을 의미한다.

다음으로 네트워크 확장설계 문제를 네트워크 설계비용을 제약조건으로 하고 신뢰도를 최대화하는 문제에 적용할 경우에는 다음과 같이 모형화될 수 있다.

목적함수 :

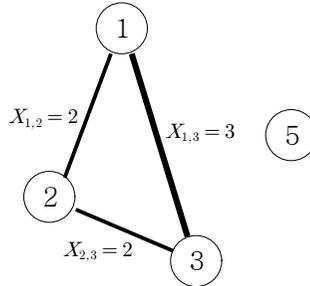
$$\text{Maximize } R(\mathbf{x})$$

제약조건 : $C(\mathbf{x}) = \sum_{\{i,j\} \in E} c_{ij} \leq C_0$

3. 백본 네트워크 설계 구조의 염색체 표현

<그림 3-1>은 하나의 네트워크 확장 설계 문제를 보여주고 있다. 3개의

노드가 링크 $x_{1,2}$, $x_{1,3}$, $x_{2,3}$ 으로 연결되어 있는 기존 네트워크에 새로운 노드 4와 5를 확장하는 문제이다.



④

<그림 3-1> 네트워크 확장 설계 문제

이 문제에서 각 링크를 위해 고려하는 선택 가능한 링크 유형의 가지수 (k)는 4이다. <표 3-1>은 <그림 3-1>의 네트워크 확장 설계 구조의 노드 간 링크의 유형을 보여 주고 있다.

<표 3-1> 노드 간 링크 유형

	1	2	3	4	5
1	-	2	3	0	0
2		-	2	0	0
3			-	0	0
4				-	0
5					-

주) 링크 유형은 4가지로 0, 1, 2, 3이다. 그리고 음영은 기존 네트워크의 링크 유형을 나타낸다.

<그림 3-1>의 확장 네트워크 설계 구조는 다음과 같은 염색체(chromosome)로 표현될 수 있다.

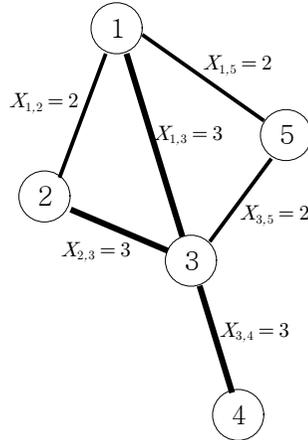
$$\text{염색체} : \{\boxed{2} \boxed{3} 0 0 \boxed{2} 0 0 0 0\}$$

이는 <표 3-1>의 우 상향 삼각형 모양의 각 행을 한 줄로 나열한 것으로, 네트워크 설계의 링크 유형이 염색체의 구성요소인 형질(allele)로 표현되었다. 여기서, 사각형으로 표시된 형질은 기존 네트워크의 링크를 의미하며 제약된 형질(constrained allele)로 명명된다.

백본 네트워크 확장 설계에 있어 염색체 구조 \mathbf{x} 의 탐색 대상 해의 수는 기존 네트워크의 링크 유형이 더 나은 수준으로만 교체되어야 하므로 다음과 같다.

$$k^{\{n(n-1)/2 - m_0\}} \cdot \prod_{\{i,j\} \in E_0} (k - x_{i,j})$$

여기서 n 은 확장 설계의 총 노드 수, 그리고 E_0 , m_0 는 각각 기존 네트워크에 연결된 링크들의 집합과 노드 수를 의미한다.



<그림 3-2> 후보 해

그러므로 <그림 3-1>의 백분 네트워크 확장 설계 문제는 $4^{\{5(5-1)/2-3\}} \times (2 \times 1 \times 2) = 65,536$ 개의 탐색 대상 해를 갖는다. <그림 3-2>와 <표 3-2>는 이 탐색 대상 해 중 하나인 후보 해와 이의 노드 간 링크 유형을 보여주고 있다.

<표 3-2> 노드 간 링크 유형

	1	2	3	4	5
1	-	2	3	0	2
2		-	3	0	0
3			-	3	2
4				-	0
5					-

주) 링크 유형은 4가지로 0, 1, 2, 3이다. 그리고 음영은 기존 네트워크의 링크 유형을 나타낸다.

이 후보 해의 염색체 표현은 다음과 같다.

염색체 : { $\boxed{2}$ $\boxed{3}$ 0 2 $\boxed{3}$ 0 0 3 2 0}

제 2 절 경제적 백본 네트워크 확장 설계

1. 백본 네트워크 확장 설계를 위한 절차

유전자 알고리즘을 이용한 경제적 백본 네트워크 확장 설계의 단계별 주요 절차는 아래와 같다.

(1) 매개변수(parameter) 초기화

- g : 세대(generation) 수
- g_{\max} : 최대 세대 수
- s : 개체집단(population)의 크기
- $m\%$: 돌연변이 염색체의 비율
- r_m : 돌연변이 형질의 비율

(2) 초기 세대의 개체집단 생성($g = 1$)

- (a) 초기 세대의 개체집단을 무작위로 생성한다.
- (b) 초기 세대의 개체집단을 신뢰도(비용) 계산함수로 보낸다.

- (c) 초기 세대의 개체집단을 비용(신뢰도) 계산함수로 보낸다.
- (d) 초기 세대의 가장 우수한(best) 해를 검사한다.

(3) 반복적인 교배 및 돌연변이 연산

(a) 교배 및 돌연변이

- 순위기반 2차 선택에 따라 현재 세대의 개체집단으로부터 서로 다른 두 부모가 선택된다.
- 균등교배를 통해 자손이 생성된다.
- 생성된 자손을 돌연변이 비율에 따라 돌연변이 연산을 한다.
- 자손이 충분히 생성되었을 때, 부모들은 자손에 의해 대체된다.

(b) 새로운 개체집단을 신뢰도(비용) 계산함수로 보낸다.

(c) 새로운 개체집단을 비용(신뢰도) 계산함수로 보낸다.

(d) 새로운 가장 우수한 해를 검사한다.

(e) 최대 세대 수($g = g_{\max}$)까지 반복한다.

2. 초기 세대의 개체집단 생성

초기 세대의 개체집단 구성을 위해 각 형질에 대한 링크 유형을 무작위로 결정한다. 단, 기존 네트워크 링크는 더 나은 수준으로의 교체만이 가능하므로, 제약된 형질인 경우는 기존 링크 수준 이상의 유형들 중에서 무작위로 결정한다. 예를 들어, <그림 3-1>의 네트워크 확장 설계 구조를 표현한 염색체 { $\boxed{2} \ \boxed{3} \ 0 \ 0 \ \boxed{2} \ 0 \ 0 \ 0 \ 0 \ 0$ }을 기반으로 구성되는 초기 개체집단 내의 하나의 염색체로 { $\boxed{3} \ \boxed{3} \ 1 \ 0 \ \boxed{2} \ 0 \ 0 \ 0 \ 2 \ 0$ }이 생성될 수 있다.

3. 교배 및 돌연변이 연산

교배 연산을 위해 두 개의 염색체를 순위기반 선택(rank based quadratic procedure)을 적용하여 선택한다(Tate와 Smith, 1995). 선택된 두 부모 염색체를 대상으로 균등교배를 실시한다. 예를 들어, 아래와 같이 부모 1과 부모 2가 선택되었고 각 형질에 난수가 발생되었다고 가정하자.

부 모 1	{	3	3	1	0	2	0	0	0	2	0	}
부 모 2	{	2	3	0	0	3	0	1	3	0	0	}
난 수		.4	.5	.4	.2	.3	.9	.7	.4	.3	.6	
자 손	{	2	3	0	0	3	0	0	3	0	0	}

자손 염색체는 난수가 임계확률(0.5)보다 같거나 큰 2, 6, 7, 10번째 형질에 대해서는 부모 1의 형질을 물려받고 난수가 임계확률보다 작은 1, 3, 4, 5, 6, 8, 9번째 형질에 대해서는 부모 2의 형질을 물려받아 생성된다.

다음으로 행해지는 돌연변이 연산은 모든 염색체의 특정 비트가 고정되는 것을 방지하고 탐색공간을 넓히려는 시도으로써 지역 최적해(local optimal)에 빠질 확률을 줄여준다.

돌연변이 연산은 먼저 자손 염색체들 중에서 돌연변이 염색체의 비율($m\%$)에 따라 돌연변이 대상 염색체를 무작위로 선택한다. 그리고 돌연변이 형질의 비율(r_m)에 따라 형질을 무작위로 선택하고, 역시 선택한 형질에 대해 무작위로 링크 유형을 결정한다. 단, 제약된 형질인 경우는 기존 링크 수준 이상의 링크 유형들 중에서 무작위로 결정한다. 이는 기존 링크의 교

체는 더 나은 수준으로 이루어져야 한다는 것을 의미한다. 예를 들어, 돌연변이 대상 염색체가 {2 3 0 0 3 0 0 3 0 0}이고, 돌연변이 형질의 비율 ($r_m = 0.2$)에 따라 1번째 형질과 9번째 형질이 무작위로 선택되고 각각의 링크 유형이 3과 2로 결정되었다면, 돌연변이된 염색체는 {3 3 0 0 3 0 0 3 2 0}이다.

4. 백본 네트워크 신뢰도 계산

본 연구에서 백본 네트워크 신뢰도 계산을 위해 다음과 같은 알고리즘을 사용한다(Cancela와 El Khadiri, 1995; Van Slyke와 Frank, 1972).

(1) 매개변수(parameter) 초기화

- $R(x) = 0$

- r : 네트워크 상태 인스턴스의 신뢰도

(2) 네트워크 상태(network state)의 인스턴스(instance) 생성

(a) n 개의 노드와 h 개의 링크로 구성된 네트워크 설계 x 에 대해 신장 트리(spanning tree)를 구성할 수 있는 최소한의 링크 수($n-1$ 개) 이상의 각 링크의 상태(고장 또는 비고장)로 인해 발생 가능한 모든 네트워크 상태 인스턴스를 생성한다. 그러므로 n 개의 노드와

h 개의 링크를 갖는 네트워크 설계 x 는 $\sum_{i=n-1}^h {}_h C_i$ (단, $h \geq n-1$)

개의 네트워크 상태 인스턴스를 갖는다.

(b) 생성한 모든 네트워크 상태 인스턴스를 스택(stack)에 담는다.

(3) 신장트리(spanning tree) 검사

(a) 스택이 비어있으면 종료한다.

(b) 스택에서 맨 위의 네트워크 상태 인스턴스를 꺼낸다.

(c) 네트워크 상태 인스턴스가 신장트리를 형성하는지를 검사한다.

- 신장트리를 형성하지 않으면, (3)(a)로 간다.

- 신장트리를 형성하면, (4)로 간다.

(4) 네트워크 신뢰도 계산

(a) 네트워크 상태 인스턴스에 대한 신뢰도(r)를 계산한다.

- 각 링크의 유형에 따른 신뢰도를 확인한다.

- 고장 상태인 링크는 $(1 - \text{신뢰도})$ 를 신뢰도로 사용한다.

- 모든 링크의 신뢰도를 곱한다.

(b) 네트워크 신뢰도를 계산한다. : $R(x) = R(x) + r$

(c) (3)(a)로 간다.

제 3 절 신뢰도를 제약조건으로 하는 경제적 백본 네트워크 확장 설계의 성능 실험

본 절에서는 네트워크 신뢰도를 제약조건으로 하고 네트워크 설치비용을 최소화하는 두 개의 백본 네트워크 확장 문제를 통해, 본 연구에서 제시하

는 유전자 알고리즘을 이용한 백본 네트워크 확장 설계 방법의 성능을 실험한다. 실험은 PentiumIV(1.8GHz)와 512MB RAM으로 구성된 PC 환경에서 이루어졌다.

첫 번째 문제(문제 1)는 3개의 기존 노드에 새로운 2개의 노드를 추가하는 경우이고, 두 번째 문제(문제 2)는 5개의 노드에 새로운 3개의 노드를 추가하는 경우이다. 두 문제를 위해 사용된 각 링크 유형의 속성은 <표 3-3>과 같고, 기존 링크 교체비용율(r_c)은 0.2로 한다.

<표 3-3> 링크 유형의 속성

링크 유형	신뢰도	단위거리 당 비용(\$)
0(비연결)	0	0
1	0.7	8
2	0.8	10
3	0.9	14

1. 문제 1 : 기존 3개 노드에 2개 노드 추가

기존 3개 노드에 새로운 2개 노드를 추가하는 문제 1을 위한 네트워크 확장 설계 구조가 <그림 3-1>에 표현되어 있다. 그리고 <표 3-4>는 노드간 거리를 나타내고 있다.

<표 3-4> 노드 간 거리 (단위 : m)

	1	2	3	4	5
1	-	32	54	62	25
2		-	34	58	45
3			-	36	52
4				-	29
5					-

문제 1은 네트워크 확장 설계 문제에 있어 상대적으로 규모가 작기 때문에, 계산 중심적(enumeration-based) 탐색 방법으로 최적해(optimal solution)의 탐색이 가능하다. 1.3절에서 살펴본 바와 같이 이 문제의 탐색 대상 해의 수는 65,536개이다. 다양한 신뢰도 제약조건($R_0 = 0.850 \sim 0.999$)에 대해 계산 중심적 탐색 방법으로 구한 최적해가 <표 3-5>에 나타나 있다. 신뢰도 제약조건이 엄격해 질수록 링크의 수준과 네트워크 설계비용이 증가함을 볼 수 있다.

<표 3-5> 문제 1에 대한 최적해

제약조건 (R_0)	비용(\$)	신뢰도	최적해의 구조
0.999	5,654	0.999	3323333333
0.995	4,418	0.995	3303222323
0.990	3,886	0.991	2303223303
0.950	2,882	0.951	2302202203
0.930	2,742	0.932	2301201203
0.900	2,532	0.906	2303200203
0.850	2,344	0.853	2303200102

문제 1에 대해 유전자 알고리즘을 이용한 확장 설계 방법을 적용하여 성능을 알아보고자 한다.

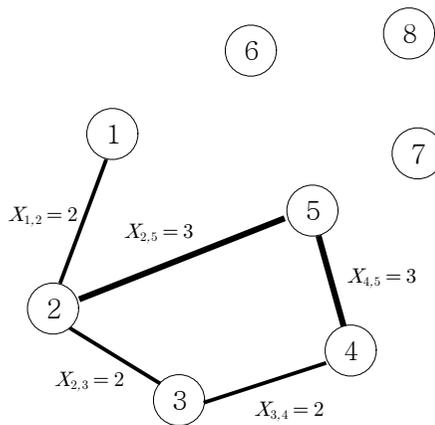
먼저 초기 탐색 실험을 통해 초기화 값($g_{\max} = 500$, $s = 30$, $m\% = 25$, $r_m = 0.25$)을 얻은 후, 각 신뢰도 제약조건($R_0 = 0.850 \sim 0.999$)에 대해 10개의 seed를 가지고 3.1절에서 제시한 설계 절차에 따라 실험을 수행하였다. 그 결과가 <표 3-6>에 나타나 있다. 각 신뢰도 제약조건에서 최소한 7개 이상의 최적해가 도출되었다. 신뢰도 제약조건에 대한 평균 탐색 해의 수(= 10개의 seed에 대한 총 탐색 해의 수 / 10)는 2,397개에서 4,032개 사이로 나타났으며, 평균 탐색 비율(= 평균 탐색 해의 수 / 탐색 대상 해의 수)은 3.658%에서 6.152% 사이로 나타났다. 이 결과는 본 연구에서 제시한 설계 방법이 높은 비율로 빠르게 최적해를 찾는다고 볼 수 있다. 즉 신뢰도를 제약조건으로 하고 비용을 최소화하는 문제에 있어 유전자 알고리즘을 이용한 설계 방법이 상당히 효과적이고 효율적인 설계 방법이라는 것을 보여 주고 있다.

<표 3-6> 문제 1의 탐색 결과

제약조건 (R_0)	평균 탐색 해의 수	평균 탐색 비율(%)	최적해의 수
0.999	3,006	4.587	10 of 10
0.995	4,032	6.152	7 of 10
0.990	2,442	3.726	10 of 10
0.950	2,982	4.550	8 of 10
0.930	2,736	4.175	9 of 10
0.900	3,654	5.576	10 of 10
0.850	2,397	3.658	10 of 10

2. 문제 2 : 기존 5개 노드에 3개 노드 추가

기존 5개 노드에 새로운 3개 노드를 추가하고 신뢰도 제약조건(R_0)을 0.900으로 하는 문제 2에 대한 네트워크 확장 설계 구조가 <그림 3-3>에 표현되어 있다.



<그림 3-3> 네트워크 확장 설계를 위한 문제 ($R_0=0.900$)

그리고 노드 간 링크 유형 및 거리가 각각 <표 3-7>과 <표 3-8>에 나타나 있다.

<표 3-7> 노드 간 링크 유형

	1	2	3	4	5	6	7	8
1	-	2	0	0	0	0	0	0
2		-	2	0	3	0	0	0
3			-	2	0	0	0	0
4				-	3	0	0	0
5					-	0	0	0
6						-	0	0
7							-	0
8								-

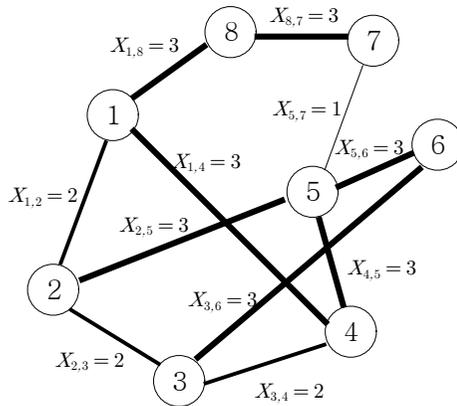
주) 링크 유형은 4가지로 0, 1, 2, 3이다. 음영은 기존 네트워크의 링크 유형을 나타낸다.

<표 3-8> 노드 간 거리 (단위: m)

	1	2	3	4	5	6	7	8
1	-	47	41	24	48	54	33	34
2		-	20	28	39	31	65	68
3			-	34	19	15	49	53
4				-	50	51	54	56
5					-	13	41	45
6						-	54	58
7							-	30
8								-

먼저 유전자 알고리즘을 이용한 네트워크 확장 설계를 수행하였다. 초기 탐색 실험을 통해 초기화 값($g_{\max} = 1,500$, $s = 60$, $m\% = 25$, $r_m = 0.25$)

을 얻은 후, 2.1절에 제시한 설계 절차에 따라 실험을 수행하였다. 그 결과 <그림 3-4>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 4,208\$이고 네트워크 신뢰도가 0.904인 가장 우수한 해 {2030003203000203003000310003}를 찾았다.



<그림 3-4> 가장 우수한 해 ($C(x)=4,208\$$)

제 4 절 비용을 제약조건으로 하는 경제적 백본 네트워크 확장 설계의 성능 실험

본 절에서는 3절과는 달리 네트워크 설치비용을 제약조건으로 하고 네트워크 신뢰도를 최대화하는 두 개의 백본 네트워크 확장 문제를 통해, 본 연구에서 제시하는 유전자 알고리즘을 이용한 백본 네트워크 확장 설계 방법의 성능을 실험한다. 실험은 3절과 동일한 PC 환경에서 이루어졌다.

첫 번째 문제(문제 1)는 3개의 기존 노드에 새로운 2개의 노드를 추가하는 경우이고, 두 번째 문제(문제 2)는 5개의 노드에 새로운 3개의 노드를 추가하는 경우이다. 두 문제를 위해 사용된 각 링크 유형의 속성은 <표 3-3>과 같고, 기존 링크 교체비용율(r_e)은 3절과 마찬가지로인 0.2로 한다.

1. 문제 3 : 기존 3개 노드에 2개 노드 추가

문제 3은 3절의 신뢰도를 제약 조건으로 하고 네트워크 설계비용을 최소화 하는 문제에 적용된 방법이 서로 상반되는 문제에 대해서도 유사한 성능을 나타내는지를 살펴보고자 한다. 기존 3개 노드에 새로운 2개 노드를 추가하는 문제 3을 위한 네트워크 확장 설계 구조는 <그림 3-1>에 표현되어 있다. 노드간의 거리는 3절과 동일하게 <표 3-4>에 제시되어 있다.

<표 3-9> 문제 3에 대한 최적해

제약조건 (C_0)	신뢰도	비용(\$)	최적해의 구조
5,654	0.999	5,654	3323333333
4,418	0.995	4,418	3303222323
3,886	0.991	3,886	2303223303
2,882	0.951	2,882	2302202203
2,742	0.932	2,742	2301201203
2,532	0.906	2,532	2303200203
2,344	0.853	2,344	2303200102

네트워크 설계비용을 제약조건으로 하고 신뢰도를 최대화 하는 문제 3은 3절과 동일한 탐색 대상 해의 수(65,536개)를 가진다. 3절에서 도출된 최적해의 비용을 제약조건($C_0 = 2,344 \sim 5,654$)으로 하여 앞서 계산 중심적 탐색 방법으로 구한 최적해가 <표 3-9>에 나타나 있다. 이는 3절의 결과와는 반대로 비용제약조건이 완화될수록 링크의 수준과 네트워크 신뢰도가 증가함을 볼 수 있다.

계산 중심적 탐색방법으로 구한 최적해에 대해 유전자 알고리즘을 이용한 확장 설계 방법을 적용하여 성능을 알아보려고 한다.

먼저 초기화 값을 설정함에 있어 3절의 문제 1과 동일한 초기화 값($g_{max} = 500, s = 30, m\% = 25, r_m = 0.25$)으로 구성되었으며, 각 비용 제약조건 ($C_0 = 2,344 \sim 5,654$)에 대해 10개의 seed를 가지고 앞서 제시한 설계 절차에 따라 실험을 수행하였다.

<표 3-10> 문제 3의 탐색 결과

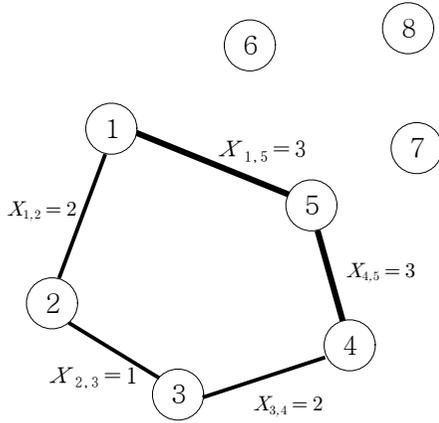
제약조건 (C_0)	평균 탐색 해의 수	평균 탐색 비율(%)	최적해의 수
5,654	2,079	3.172	10 of 10
4,418	7,038	10.739	10 of 10
3,886	2,088	3.186	10 of 10
2,882	2,184	3.333	6 of 10
2,742	2,226	3.397	9 of 10
2,532	492	0.751	10 of 10
2,344	1,497	2.284	9 of 10

그 결과가 <표 3-10>에 나타나 있다. 각 네트워크 설계비용 제약조건에

서 최소한 6개 이상의 최적해가 도출되었다. 이는 네트워크 신뢰도를 제약 조건으로 하고 네트워크 설계비용을 최소화하는 문제에서 도출된 최적해의 최소 개수인 7개 보다는 낮으나, 각 7개의 제약조건에 대해 평균 9.14개를 찾고 있다. 이는 네트워크 신뢰도를 제약조건으로 하고 네트워크 설계비용을 최소화하는 문제와 그 비율이 동일하다. 다음으로, 네트워크 설계비용 제약조건에 대한 평균 탐색 해의 수(= 10개의 seed에 대한 총 탐색 해의 수 / 10)는 492개에서 7,038개 사이로 나타났으며, 평균 탐색 비율(= 평균 탐색 해의 수 / 탐색 대상 해의 수)은 0.751%에서 10.739% 사이로 나타났다. 이러한 결과는 본 연구에서 제시한 설계 방법이 앞서 3절의 목적함수와 제약조건을 상반되게 설정하는 경우에도 높은 비율로 빠르게 최적해를 찾는다고 판단할 수 있다. 즉 네트워크 설계비용을 제약조건으로 하고 네트워크 신뢰도를 최대화 하는 문제에 있어서도 유전자 알고리즘을 이용한 설계 방법이 상당히 효과적이고 효율적인 설계 방법이라는 것을 보여 주고 있다.

2. 문제 4 : 기존 5개 노드에 3개 노드 추가

기존 5개 노드에 새로운 3개 노드를 추가하고 비용 제약조건(C_0)을 4,700(\$)
으로 하는 문제 4에 대한 네트워크 확장 설계 구조가 <그림 3-5>에 표현되어 있다.



<그림 3-5> 네트워크 확장 설계를 위한 문제 ($C_0=4,700\$$)

그리고 노드 간 링크 유형은 <표 3-11>에 제시되어 있으며, 노드간 거리는 앞서 3절에 제시된 <표 3-8>과 동일하다.

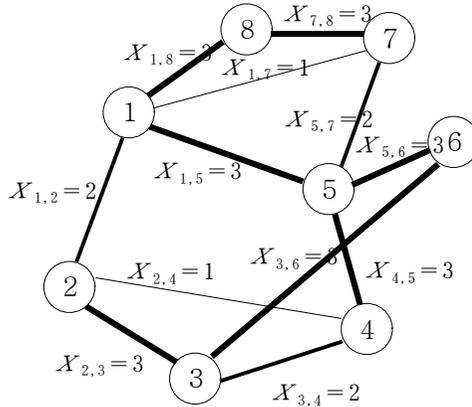
<표 3-11> 노드 간 링크 유형

	1	2	3	4	5	6	7	8
1	-	2	0	0	3	0	0	0
2		-	1	0	0	0	0	0
3			-	2	0	0	0	0
4				-	3	0	0	0
5					-	0	0	0
6						-	0	0
7							-	0
8								-

주) 링크 유형은 4가지로 0, 1, 2, 3이다. 음영은 기존 네트워크의 링크 유형을 나타낸다.

먼저 유전자 알고리즘을 이용한 네트워크 확장 설계를 수행하였다. 초기 탐색 실험을 통해 초기화 값($g_{max} = 1,500$, $s = 60$, $m\% = 25$, $r_m = 0.25$)

은 3절의 문제 2와 동일한 값으로 설정하였으며, 이를 2.1절에 제시한 설계 절차에 따라 실험을 수행하였다. 그 결과 <그림 3-6>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 4,680\$이고 네트워크 신뢰도가 약 0.944인 가장 우수한 해 {2003013310000203003000320003}를 찾았다.



<그림 3-6> 가장 우수한 해($R(x)=0.944$)

제 5 절 성능 비교 및 민감도 분석

1. 유전자 알고리즘의 성능 비교

앞서 신뢰도를 제약조건으로 하고 비용을 최소화 하는 3절의 문제 2는 계산 중심적 탐색 방법을 할 수 없을 정도로 탐색 대상 해의 수($4^{\lceil (8-1)/2 - 5 \rceil} (2 \times 2 \times 2 \times 1 \times 1) = 562,949,953,421,312$)가 매우 크며, 비용을 제약

조건으로 하고 신뢰도를 최대화하는 4절의 문제 4는 3절의 문제 2보다 그 탐색 대상 해의 수($4^{(8-1)/2-5}(2 \times 1 \times 3 \times 2 \times 1) = 844,424,930,131,968$)가 더 크다. 그러므로 계산 중심적 탐색 방법 대신에 SA(Costamagna 등, 1995) 및 Greedy 탐색 방법(Deeter와 Smith, 1998)을 사용하여 유전자 알고리즘을 이용한 설계 방법과의 성능을 비교하고자 한다.

본 연구에서 적용한 SA 탐색 방법은 Kirkpatrick 등(1983)에 의해서 제안된 최적화 기법으로써, 기존의 반복적인 개선(iterative improvement)에 근거한 휴리스틱 기법들이 지역 최소점(local minimum point)에 빠져버리는 단점을 개선한 범용의 최적화 기법이다(김여근 등, 2004). SA는 본질적으로 언덕 오르기(hill climbing) 휴리스틱을 변형한 것으로, 주어진 현재 해의 인접 해에서 품질이 나은 지점으로 이동할 경우, 이 이동은 긍정적 전이(positive move)로써 받아들여지게 된다. 반면, 품질이 더 나쁜 지점으로도 낮은 확률로 이동할 수 있는데 이를 부정적 전이(negative move)라 한다. SA에서 부정적 전이는 시간이 지남에 따라 감소한다. 그러므로 SA는 현재 해로부터 인접 해로 전이하는 반복적인 과정을 통해 전체 최적해를 찾아 나가는 방법으로써, 부정적 전이를 빼면 SA가 항상 현재해보다 나은 쪽으로만 이동하는 언덕 오르기 방식과 유사해진다. 결국 이러한 부정적 전이를 통해 SA는 열등한 해로부터의 전이도 확률적으로 허용함으로써 지역 최소점에서 빠져 나올 수 있는 가능성을 열어둔다는 특징을 가지고 있다(Arts & Korst, 1997).

SA는 폭넓은 응용 가능성과 최상에 가까운 해답을 얻을 수 있다는 장점이 있으나, 상당히 좋은 해답을 얻는데 걸리는 계산 시간이 엄청나게 길다는 단점이 존재한다. 그러나 SA는 유전자 알고리즘과 더불어 조합 최적화 문제에 있어 전역 최적해를 구할 수 있는 메타 휴리스틱 알고리즘으로써 광

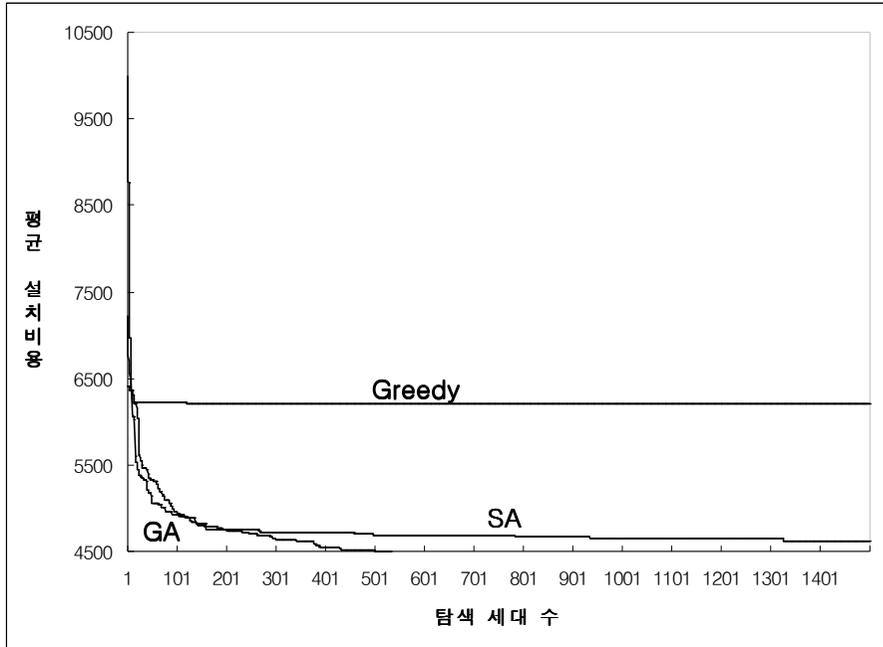
범위하게 사용되기 때문에 이를 본 연구에서 제안된 유전자 알고리즘과 비교 분석하였다.

조합 최적화 문제에 SA를 적용하기 위해 필요한 파라미터로써는 초기해, 초기온도, 쿨링 스케줄이 정의되어야 한다. 본 연구에서는 제안된 유전자 알고리즘과의 비교를 위해 Costamagna(1995)의 연구에 사용된 SA 파라미터를 이용하였다.

이에 비해, Greedy 탐색 방법은 정확한 알고리즘보다 빠른 단계적 최적화를 구할 때 사용되나, 유전자 알고리즘이나 SA보다는 가장 간단한 설계방법이고 좀 더 다양한 문제들에 적용할 수 있다는 점이 그 특징이다. 본 연구에서 사용된 Greedy 탐색 방법은 유전자 알고리즘을 이용한 설계 절차와 동일하나, 단지 교배 및 돌연변이를 통해 새로운 개체집단을 형성하는 대신에 염색체의 형질 값으로 무작위로 선택한 링크 유형을 사용함으로써 새로운 개체 집단을 생성하는 기법이다.

유전자 알고리즘을 이용한 설계 방법과 SA 및 Greedy 탐색 방법 각각에 대해 10개의 seed를 사용하여, 탐색 세대 수의 증가에 따른 평균 설계비용의 추이를 살펴보았다. <그림 3-7>은 신뢰도를 제약조건으로 하고 비용을 최소화하는 3절의 문제 2에 대한 비교 결과이다.

유전자 알고리즘을 이용한 설계 방법은 탐색 세대 수가 증가함에 따라 평균 설계비용이 급격히 떨어져 약 500세대 수에서 4,500\$이하로 수렴하는 반면에, SA는 초기 세대에서 GA보다 빠른 속도로 비용의 감소가 이루어지고 있으나, 1,500세대 수에 이르기까지 약 4,600\$이상의 비용에서 더 이상 감소가 이루어지지 않고 있음을 볼 수 있다. 이 두 가지 탐색방법에 비해, Greedy 탐색 방법은 초기 해에서 1,500세대 수가 넘어갈 때까지 단 2번의 설계비용 감소가 이루어 졌다.



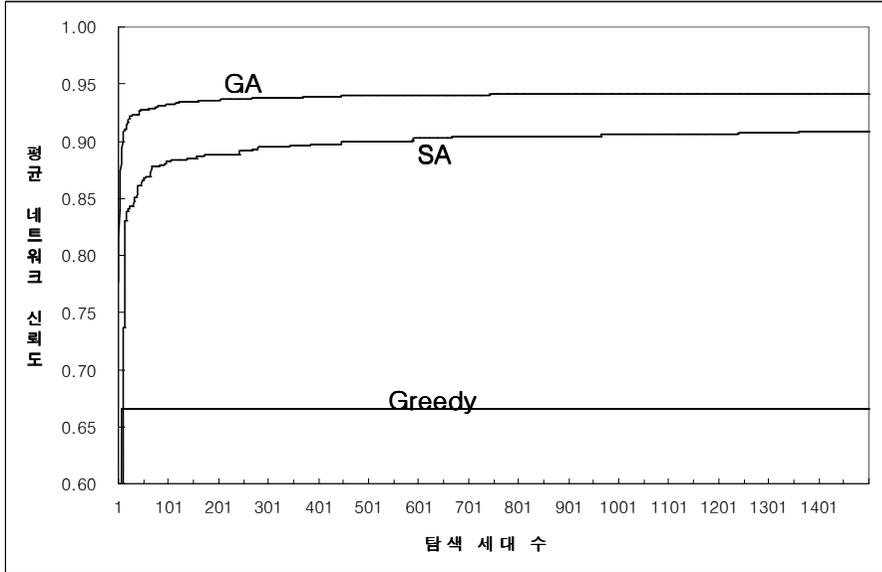
<그림 3-7> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 2)

이 결과는 백본 네트워크 확장 설계 문제에 있어 유전자 알고리즘을 이용한 설계 방법이 SA나 Greedy 탐색 방법 보다 훨씬 효율적인 설계 방법이라는 것을 보여 주고 있다.

<그림 3-8>은 네트워크 설계비용을 제약조건으로 하고 네트워크 신뢰도를 최대화 하는 4절의 문제 4에 대한 SA 및 Greedy 탐색방법과의 비교 결과이다.

유전자 알고리즘을 이용한 설계방법은 탐색 세대 수가 증가함에 따라 평균 네트워크 신뢰도가 지속적으로 증가하여 약 750세대 수에서 약 0.9400으로 수렴하고 있으나, SA는 약 590세대 수에서 약 0.9020으로 수렴하고 있으며 1,500세대 수에 이르기까지 약 0.908의 신뢰도에서 더 이상 증가가 이루

어지지 않고 있다. 이에 비해, Greedy 탐색 방법은 앞선 네트워크 신뢰도를 제약조건으로 하는 경우와 마찬가지로 초기 해에서 1,500세대 수가 넘어갈 때까지 단 3번의 네트워크 신뢰도의 증가로 약 0.6700에서만 수렴됨을 볼 수 있다.



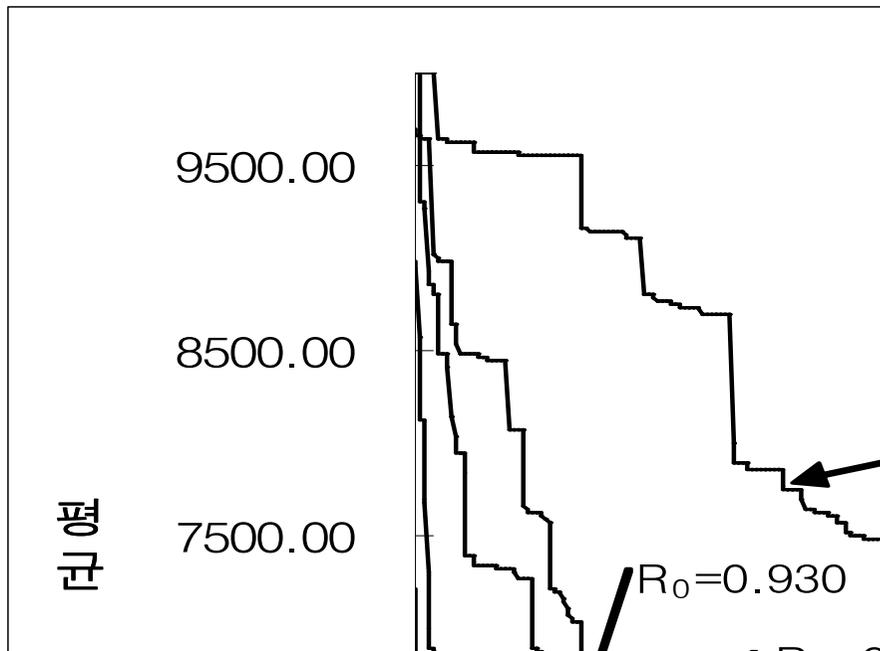
<그림 3-8> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 4)

이 결과는 네트워크 설계비용을 제약조건으로 하고 네트워크 신뢰도를 최대화 하는 백분 네트워크 확장 설계 문제에 있어서도 유전자 알고리즘을 이용한 설계 방법이 SA나 Greedy 탐색 방법보다 훨씬 더 효율적인 설계 방법이라는 것을 보여주고 있다.

2. 민감도 분석

본 절에서는 주어진 네트워크 신뢰도 및 설치비용 제약조건의 변화에 따른 목적함수의 변화를 분석하기 위해 <그림 3-9> 및 <그림 3-10>과 같이 제약조건을 다양하게 변화시켜 그 결과를 분석하고자 한다.

<그림 3-9>는 문제 2의 네트워크 신뢰도 제약조건의 변화에 따른 네트워크 평균 설계비용의 변화를 나타내고 있다.

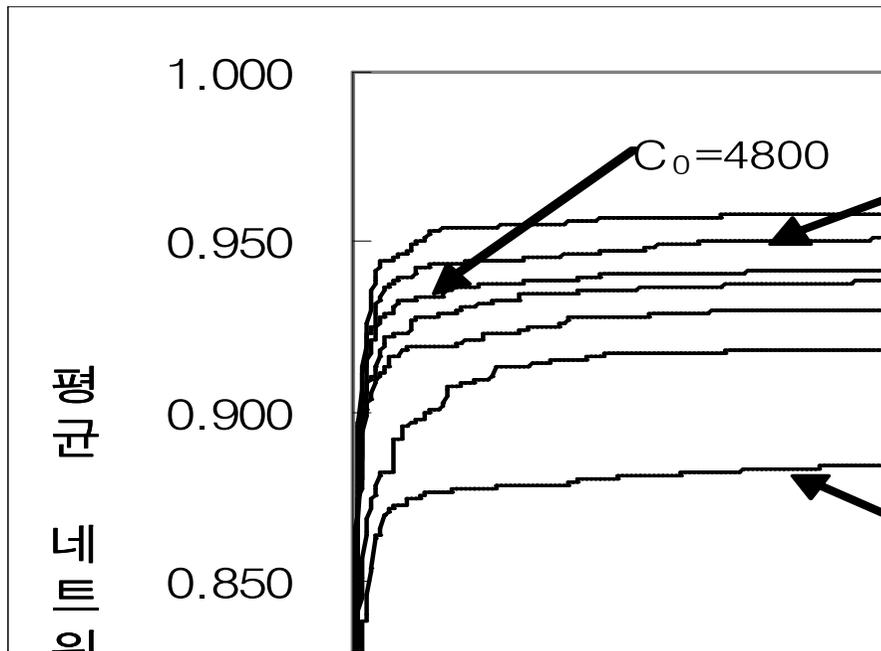


<그림 3-9> 백본 네트워크 신뢰도 변화에 따른 민감도 분석결과(문제 2)

그림에서 보는 바와 같이 네트워크 신뢰도 제약조건을 높일 경우, 탐색 세대 수가 증가함에 따라 평균 설계비용이 낮게 나타남을 볼 수 있다. 특히,

네트워크 신뢰도가 0.900에서 0.930일 경우, 수립되는 평균 설계비용이 크게 차이가 나지 않으나, 0.940일 경우는 이들 보다는 높은 비용에서 수립됨을 볼 수 있다. 이는 0.940일 경우 10개의 seed 중에서 신뢰도 제약조건을 만족하는 seed의 수가 많지 않음으로 인해 나타난 결과로 볼 수 있다.

<그림 3-10>은 문제 4의 네트워크 설계비용 제약조건의 변화에 따른 네트워크 평균 신뢰도의 변화를 나타내고 있다.



<그림 3-10> 백분 네트워크 설치비용 변화에 따른 민감도 분석결과(문제 4)

네트워크 신뢰도 제약조건이 달라질 경우와 달리, 네트워크 설계비용 제약조건을 높일 경우, 탐색 세대 수가 증가함에 따라 평균 네트워크 신뢰도가 증가함을 볼 수 있다. 특히, 네트워크 설계비용이 4,000\$에서 4,500\$일 경우, 각

네트워크 비용 제약조건이 수렴되는 네트워크 평균 신뢰도가 확연히 차이가 나고 있으며 이와는 달리, 네트워크 설계비용이 4,500\$에서 5,000\$인 경우에는 그 차이가 크질 않음을 볼 수 있다. 이는 네트워크 비용의 구간 차이에서 기인하는 원인일 수도 있지만, 네트워크 설계비용의 차이에 따라 수렴하는 신뢰도가 다양함으로 인해 일반 기업에서 특정 수준의 네트워크 신뢰도 (예를 들면, 0.900)만을 요구한다면 선택할 수 있는 네트워크 설계비용이 다양해질 수 있음을 나타낸다.

제 4 장 경제적 LAN 설계

제 1 절 경제적 LAN 설계에 대한 문제 표현

LAN은 제한된 공간에서 다수의 컴퓨터나 OA 기기 등을 속도가 빠른 통신선로로 연결하여 사용자들 간에 통신이 가능하도록 하는 기업 내 대표적인 통신 기반시설로써 컴퓨터의 급속한 보급 및 네트워크 기술의 발달에 따라 기업 내에서 다량의 정보가 교환되고 있는 중요 네트워크 통로이다. 기업의 업무수행에 직접적인 영향을 미치는 LAN의 설계문제에 있어 무엇보다도 최대 신뢰도 확보와 최소 설치비용 할당이 기업이 우선적으로 고려해야 할 중요한 요소이다.

그러므로 본 장에서는 경제적 LAN 설계 문제를 주어진 신뢰도 및 설치비용 제약조건 하에서 최소 LAN 설치비용 및 최대 LAN 신뢰도를 달성할 수 있는 유전자 알고리즘을 이용한 LAN 위상설계 방법을 제안한다.

본 연구에서 제안하는 유전자 알고리즘을 이용한 경제적 LAN 설계 문제를 해결하기 위해 필요한 기본 가정 및 표기 형식은 다음과 같다.

1. 기본 가정 및 표기 형식

LAN 설계 문제를 위한 기본 가정은 다음과 같다.

- 각 노드(허브 또는 사용자(PC))의 위치는 주어져 있다.

- 각 링크는 무방향적(bi-directional)이다.
- 네트워크에 있어 중복된 링크는 존재하지 않는다.
- 허브의 상태, 허브 간 링크의 상태는 고장(failed)이거나 비고장 상태로 한정한다.
- 각 허브와 링크의 실패는 독립적이다.

LAN 설계 문제를 위한 표기 형식은 다음과 같다.

- n : 허브의 수
- m : 사용자의 수
- s_i : i 번째 허브, $i = 1, 2, \dots, n$
- u_j : j 번째 사용자, $j = 1, 2, \dots, m$
- $ks_i (\in \{1, 2, 3, \dots, v\})$: s_i 의 허브 유형, 여기서 v 는 허브 유형의 가지 수
- ms_i : s_i 에 연결 가능한 최대 노드(다른 허브와 사용자) 수
- l_{ij} : s_i 와 s_j 간 링크
- $kl_{ij} (\in \{0, 1, 2, 3, \dots, t-1\})$: s_i 와 s_j 간의 링크 유형. 여기서, t 는 링크 유형의 가지 수
- d_{ij} : s_i 와 s_j 간 거리
- h_{ij} : s_i 와 u_j 간 거리
- $y_{ij} (\in \{0, 1\})$: s_i 와 u_j 간 연결 유무
- cs_i : s_i 의 설치비용(\$)
- cl_{ij} : l_{ij} 의 설치비용(\$)
- cu : 허브와 사용자간 단위거리 당 링크 설치비용(\$/m)
- $\mathbf{x} (= \{ ks_1, ks_2, \dots, ks_n ; kl_{1,2}, kl_{1,3}, \dots, kl_{n-1,n} ; (y_{1,1}, y_{2,1}, \dots, y_{n,1}), (y_{1,2}, y_{2,2}, \dots, y_{n,2}), \dots,$

- $(y_{1,m}, y_{2,m}, \dots, y_{n,m})$: LAN 설계 구조
- $su_j (\in \{1, 2, \dots, n\})$: u_j 와 연결된 허브의 번호
 - $\tilde{\mathbf{x}} (= \{ks_1, ks_2, \dots, ks_n ; kl_{1,2}, kl_{1,3}, \dots, kl_{n-1,n} ; su_1, su_2, \dots, su_m\})$: 염색체 표현을 위한 LAN 설계구조
 - $C(\mathbf{x})$: LAN 설계의 총 비용
 - $R(\mathbf{x})$: LAN 설계의 신뢰도
 - R_0 : 요구되는 최소의 LAN 신뢰도
 - C_0 : 요구되는 최대의 LAN 설치비용

2. LAN 설계 문제의 모형

$$\text{목적함수 : Minimize } C(\mathbf{x}) = \sum_{i=1}^n cs_i + \sum_{i=1}^n \sum_{j>i}^n cl_{ij} + cu \sum_{i=1}^n \sum_{j=1}^m h_{ij} \cdot y_{ij} \quad (1)$$

$$\text{제약조건 : } R(\mathbf{x}) \geq R_0 \quad (2)$$

$$\sum_{\substack{j=1 \\ (j \neq i)}}^n b_{ij} + \sum_{j=1}^m y_{ij} \leq ms_i, \quad i = 1, 2, \dots, n \quad (3)$$

$$b_{ij} = \begin{cases} 0, & kl_{ij} = 0 \text{ 이면} \\ 1, & kl_{ij} > 0 \text{ 이면} \end{cases}$$

$$\sum_{i=1}^n y_{ij} = 1, \quad j = 1, 2, \dots, m \quad (4)$$

$$\sum_{\substack{j=1 \\ (j \neq i)}}^n kl_{ij} \geq 1, \quad i = 1, 2, \dots, n \quad (5)$$

목적함수인 식 (1)은 허브 설치비용, 허브 간 링크 설치비용, 그리고 허브

와 사용자 간 링크 설치비용의 합으로 구성되는 총 설계비용의 최소화를 의미한다. 허브 설치비용은 각 허브 설치비용(cs_i)의 합으로 계산된다. 각 허브의 설치비용은 그 허브의 유형(ks_i)에 의존한다. 허브 간 링크 설치비용은 각 허브 간 링크 설치비용(cl_{ij})의 합으로 계산된다. 각 허브 간 링크 설치비용은 그 링크의 유형(kl_{ij})에 의존하는 단위거리 당 설치비용과 거리(d_{ij})의 곱으로 계산된다. 그리고 허브와 사용자 간 링크 설치비용은 모든 허브와 사용자간에 동일한 링크가 설치된다는 가정 하에, 각 허브와 사용자 간에 연결이 있는 경우(y_{ij})의 단위거리 당 링크 설치비용(cu)과 거리(h_{ij})의 곱에 대한 총합으로 계산된다.

제약조건인 식 (2)는 LAN 신뢰도 제약을 나타낸 것이며, 식 (3)은 각 허브는 연결할 수 있는 최대의 노드 수(ms_i)에 제약 받는다는 것을 의미한다. 허브의 연결 가능한 최대 노드 수는 허브의 유형에 의존한다. 그리고 식 (4)는 각각의 이용자는 반드시 한 개의 허브와 연결되어야 함을, 식 (5)는 한 개의 허브는 반드시 한 개 이상의 다른 허브와 연결되어야 함을 의미한다.

LAN 설계 문제를 LAN 설치비용을 제약조건으로 하고 LAN 신뢰도를 최대화는 문제에 적용할 경우에는 다음과 같이 모형화될 수 있다.

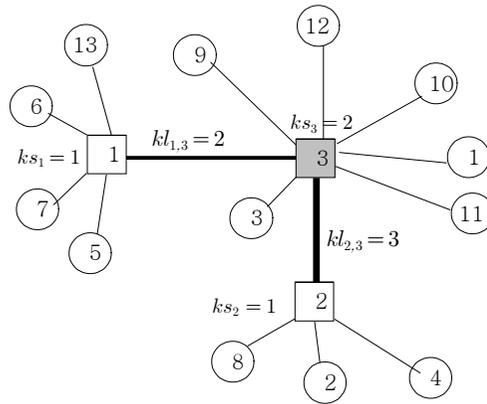
목적함수 : Maximize $R(\mathbf{x})$

제약조건 : $C(\mathbf{x}) \leq C_0$

3. LAN 설계 구조의 탐색체 표현

<그림 4-1>은 3개의 허브(사각형)에 13명의 사용자(원형)를 연결하는 LAN 설계의 예를 보여주고 있다. 이 예에서 허브 간 링크를 위해 선택 가능한 링크 유형은 0, 1, 2, 3으로 가지 수(t)는 4이다.

허브의 유형은 두 가지로 최대 연결 노드 수가 5인 유형 1과 8인 유형 2이다. <표 4-1>은 <그림 4-1>의 허브 간 링크 유형을 보여 주고 있으며, <표 4-2>는 <그림 4-1>의 허브와 사용자간 연결 유무를 보여주고 있다.



<그림 4-1> LAN 설계의 예

<표 4-1> 허브 간 링크 유형

kl_{ij}	1	2	3
1	-	0	2
2		-	3
3			-

(주) 허브간 링크 유형은 4가지로 0, 1, 2, 3이다.

<표 4-2> 허브와 사용자 간 연결 유무

y_{ij}		사 용 자												
		1	2	3	4	5	6	7	8	9	10	11	12	13
허 브	1	0	0	0	0	1	1	1	0	0	0	0	0	1
	2	0	1	0	1	0	0	0	1	0	0	0	0	0
	3	1	0	1	0	0	0	0	0	1	1	1	1	0

(주) 1은 허브와 사용자간에 연결이 있음을 의미하고, 0은 연결이 없음을 의미한다.

<그림 4-1>의 예는 LAN 설계 구조 \mathbf{x} (= {1, 1, 2 ; 0, 2, 3 ; (0, 0, 1), (0, 1, 0), (0, 0, 1), (0, 1, 0), (1, 0, 0), (1, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1), (0, 0, 1), (0, 0, 1), (0, 0, 1), (1, 0, 0)})로 표현된다. 이 설계 구조의 전반부(허브 영역)는 허브의 번호 순에 의한 허브 유형을 나타내고, 중반부(허브 간 링크 영역)는 <표 4-1>의 우상향 삼각형 모양의 각 행을 한 줄로 나열한 것으로 허브 간 링크 유형을 나타낸다. 그리고 후반부(사용자 영역)는 <표 4-2>의 각 열을 괄호로 묶어 나열한 것으로 사용자와 허브간의 연결 유무를 나타낸다.

위의 LAN 설계 구조 \mathbf{x} 는 염색체 표현을 위한 설계 구조 $\tilde{\mathbf{x}} = \{1, 1, 2 ; 0, 2, 3 ; 3, 2, 3, 2, 1, 1, 1, 2, 3, 3, 3, 3, 1\}$ 형태로 나타낼 수 있다.

설계 구조 $\tilde{\mathbf{x}}$ 는 설계 구조 \mathbf{x} 와 동일하나, 단지 사용자 영역에 표현된 이진 형태의 특정 사용자와 허브 간의 연결 유무를 n진 형태로 압축하여 표현한 것으로 특정 사용자가 어느 허브와 연결되었는지를 나타낸다. 즉, $\tilde{\mathbf{x}}$ 의 사용자 영역은 <표 4-2>의 사용자 번호 순서대로 사용자와 연결된 허브 번

호를 나타낸다. 후반부 첫 번째 숫자 3은 사용자 1번이 허브 3번과 연결되어 있음을 의미한다.

설계 구조 $\tilde{x} = \{1, 1, 2 ; 0, 2, 3 ; 3, 2, 3, 2, 1, 1, 1, 2, 3, 3, 3, 3, 1\}$ 는 유전자 연산을 위해 동일한 형태인 염색체(chromosome) $\{1\ 1\ 2 ; 0\ 2\ 3 ; 3\ 2\ 3\ 2\ 1\ 1\ 1\ 2\ 3\ 3\ 3\ 3\ 1\}$ 로 표현될 수 있다.

n 개의 허브와 m 개의 사용자를 갖는 LAN 설계의 탐색 대상 해의 수는 다음과 같다.

$$v^n \times t^{n(n-1)/2} \times n^m$$

v^n 는 n 개의 허브가 v 개의 허브 유형 중 하나를 선택할 수 있는 경우의 수이고, $t^{n(n-1)/2}$ 은 t 개의 허브 간 링크 유형을 가지고 n 개의 허브 간을 연결할 수 있는 경우의 수이다. 그리고 n^m 은 m 명의 사용자가 각각 n 개의 허브 중 하나에 연결될 수 있는 경우의 수이다. 그러므로 3개의 허브와 13명의 사용자, 그리고 4가지의 허브 간 링크 유형과 2가지의 허브 유형이 고려된 <그림 4-1>과 같은 형태의 LAN 설계 문제는 $2^3 \times 4^{3(3-1)/2} \times 3^{13} = 816,293,376$ 개의 탐색 대상 해를 갖는다.

제 2 절 경제적 LAN 설계

본 연구에서 제안하는 유전자 알고리즘을 이용한 경제적 LAN 설계 문제는 다음과 같이 모형화될 수 있다.

1. LAN 설계를 위한 절차

유전자 알고리즘을 이용한 LAN 설계의 주요 절차는 다음과 같다.

(1) 매개변수(parameter) 초기화

- g : 세대(generation) 수
- g_{\max} : 최대 세대 수
- s : 개체집단(population)의 크기
- $m\%$: 돌연변이 염색체의 비율
- r_m : 돌연변이 형질의 비율

(2) 초기 세대의 개체집단 생성($g = 1$)

- (a) 초기 세대의 개체집단을 무작위로 생성한다.
- (b) 초기 세대의 개체집단을 신뢰도 계산함수와 비용 계산함수로 보낸다.
- (c) 초기 세대의 가장 우수함(best) 해를 검사한다.

(3) 반복적인 교배 및 돌연변이 연산

(a) 교배 및 돌연변이

- 순위기반 선택에 따라 현재 세대의 개체집단으로부터 서로 다른 두 부모가 선택된다.
- 균등교배를 통해 자손이 생성된다.
- 생성된 자손을 돌연변이 비율에 따라 돌연변이 연산을 한다.
- 자손이 충분히 생성되었을 때, 부모들은 자손에 의해 대체된다.

(b) 새로운 개체집단을 신뢰도 계산함수와 비용 계산함수로 보낸다.

(c) 새로운 가장 우수한 해를 검사한다.

(d) 최대 세대 수까지($g = g_{\max}$) 반복한다.

2. 초기 세대의 개체집단 생성

초기 세대의 개체집단 구성을 위해, 염색체 허브 영역의 각 형질에 대해 허브 유형을 무작위로 결정하고, 염색체 허브 간 링크 영역의 각 형질에 대해 허브 간 링크 유형을 무작위로 결정한다. 그리고 염색체 사용자 영역의 각 형질에 대해 허브 번호를 무작위로 결정한다. 단, n 개의 허브 간 연결이 최소한 연결 가능한 형태(신장트리)가 되도록 하기 위해 허브 간 링크 영역의 형질은 0이 아닌 값을 $n-1$ 개 이상 갖도록 한다(Kershenbaum, 1993; Rao, 1998). 예를 들어, <그림 4-1>의 LAN 설계 예에 있어 초기 개체집단 내의 후보 염색체는 다음과 같이 생성될 수 있다.

{ 2 1 2 ; 2 0 3 ; 1 2 3 2 2 1 1 2 3 2 3 3 3 }

3. 교배 및 돌연변이 연산

LAN 설계 문제에서도 백본 네트워크 설계 문제에서와 동일하게 두 개의 염색체를 순위기반 선택(rank based quadratic procedure)을 적용하여 선택한다. 선택된 두 부모 염색체 내의 각각의 형질에 대해 0과 1사이의 난수를 발생시켜 임계확률(0.5) 이상인지 여부에 따라 형질(allele)에 대해 교배를 하는 균등교배를 실시한다. 예를 들어, 아래와 같이 부모 1과 부모 2가 선택되었고, 각 형질에 난수가 발생되었다고 가정하자.

부모1 : {2 1 1 ; 2 0 3 ; 1 2 3 2 2 1 2 2 3 2 3 3 3}

부모2 : {1 2 1 ; 1 1 0 ; 3 2 2 2 2 1 1 2 3 3 1 3 3}

난 수 : .2 .6 .1 ; .5 .3 .7 ; 4 .5 .3 .7 .8 .5 .4 .6 .2 .1 .6 .3 .2

자 손 : {1 1 1 ; 2 1 3 ; 3 2 2 2 2 1 1 2 3 3 3 3 3}

위의 예에서 난수가 임계확률 0.5이상인 2, 4, 6, 8, 10, 11, 12, 14, 17번째 형질에 대해서는 부모 1의 형질을 자손에게 물려주고 그 외의 형질에 대해서는 부모 2의 형질을 자손에게 물려주어 자손 염색체를 만든다.

다음으로 행해지는 돌연변이 연산은 모든 염색체의 특정 비트가 고정되는 것을 방지하고 탐색공간을 넓히려는 시도으로써 지역 최적해(local optimal)에 빠질 확률을 줄여준다.

돌연변이 연산은 먼저 자손 염색체들 중에서 돌연변이 염색체의 비율($m\%$)에 따라 돌연변이 대상 염색체를 무작위로 선택한다. 그리고 돌연변이 형질의 비율(r_m)에 따라 형질을 무작위로 선택하고, 역시 선택한 형질에 대해 무작위로 형질의 값을 결정한다. 예를 들어, 돌연변이 대상 염색체가

{1 1 1 ; 2 1 3 ; 3 2 2 2 2 1 1 2 3 3 3 3}이고 돌연변이 형질의 비율($r_m = 0.2$)에 따라 1, 4, 13, 18번째 형질이 무작위로 선택되고 허브 영역의 형질인 1번째 형질에 대해 허브 유형이 3으로, 허브 간 링크 영역의 형질인 4번째 형질에 대해 허브 간 링크 유형이 3으로, 그리고 사용자 영역의 형질인 13, 18번째 형질에 대해 허브 번호가 2, 1로 결정되었다면, 돌연변이된 염색체는 {3 1 1 ; 3 1 3 ; 3 2 2 2 2 1 2 2 3 3 3 1 3}이다.

4. LAN 신뢰도 계산 방법

LAN 설계 문제에 있어 각 사용자 그리고 사용자와 허브 간의 링크 고장은 LAN 시스템의 정상 운영에 영향을 주지 않고 독립적으로 수리가 가능하다. 그러므로 본 연구에서는 LAN 설계 문제에 있어 LAN 시스템 정상 운영에 중요한 구성요소인 각 허브 및 허브 간 링크만을 대상으로 다음과 같은 알고리즘을 사용하여 계산한다.

(1) 매개변수(parameter) 초기화

- $R(x)=0$

- r : 네트워크 상태 인스턴스의 신뢰도

(2) LAN 상태(network state)의 인스턴스(instances) 생성

(a) n 개의 허브와 z 개의 허브 간 링크로 구성된 LAN 설계는 허브의 상태(고장 또는 비고장)와 허브 간에 연결된 링크의 상태(고장 또

는 비고장)로 인해 발생 가능한 $2^{(n+z)}$ 개의 LAN 상태 인스턴스를 가질 수 있다. LAN이 정상적으로 작동하기 위해서는 모든 허브가 비고장이고 허브간 비고장의 링크로 최소한 신장트리를 형성해야 한다. 그러므로 n 개의 비고장 허브와 z 개의 허브 간 링크 중 최소한 $n-1$ 개 이상의 허브 간 비고장 링크를 포함하는 LAN 상태 인스턴스들을 생성한다. 즉 n 개의 허브와 z 개의 허브 간 링크로 구성된 LAN 설계에 대해 n 개의 허브가 모두 비고장인 $\sum_{i=n-1}^z C_i$ 개의 네트워크 상태 인스턴스를 생성한다(단, $z \geq n-1$).

(b) 생성한 모든 LAN 상태 인스턴스를 스택(stack)에 담는다.

(3) 비고장 링크 기반의 신장트리 검사

(a) 스택이 비어있으면 종료한다.

(b) 스택에서 맨 위의 LAN 상태 인스턴스를 꺼낸다.

(c) LAN 상태 인스턴스의 허브 간 비고장 링크들만으로 신장트리를 형성하는지를 검사한다.

- 신장트리를 형성하지 않으면, 3(a)로 간다.

- 신장트리를 형성하면, (4)로 간다.

(4) LAN 신뢰도 계산

(a) LAN 상태 인스턴스에 대한 신뢰도(r)를 계산한다.

- 각 허브 및 허브 간 링크의 유형에 따른 신뢰도를 확인한다.

- 고장 상태인 링크는 $(1 - \text{신뢰도})$ 를 신뢰도로 사용한다.

- 허브 및 허브 간 링크의 모든 신뢰도를 곱한다.

(b) LAN 신뢰도를 계산한다. : $R(x) = R(x) + r$

(c) (3)(a)로 간다.

제 3 절 신뢰도를 제약조건으로 하는 경제적 LAN 설계의 성능 실험

본 절에서는 네트워크 신뢰도를 제약조건으로 하는 두 개의 LAN 설계 문제를 통해, 본 연구에서 제시하는 유전자 알고리즘을 이용한 설계 방법에 대한 성능을 실험한다. 본 실험은 PentiumIV(1.8GHz)과 512MB RAM으로 구성된 PC 환경에서 이루어졌다.

첫 번째 문제(문제 1)는 3개의 허브와 13명의 사용자를 LAN으로 연결하는 경우이고, 두 번째 문제(문제 2)는 5개의 허브와 30명의 사용자를 네트워크로 연결하는 경우이다. 두 문제 모두 신뢰도 제약조건(R_0)은 0.850으로 한다.

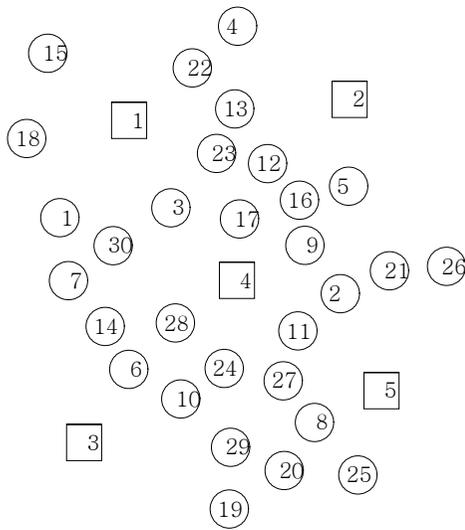
<표 4-3> 허브 간 링크 유형의 속성

링크 유형	신뢰도	단위거리 당 비용(\$/m)
0(비연결)	0	0
1	0.910	1.0
2	0.930	1.5
3	0.950	4.0

<표 4-4> 허브의 속성

허브 유형	신뢰도	허브 당 설치 비용(\$/허브)	최대 연결 노드 수
1	0.940	110	5
2	0.960	200	8
3	0.980	580	16

(주) 3절의 문제 1은 허브 유형 1과 2로 한정한다.



<그림 4-2> 허브와 사용자 위치

<표 4-5> 허브 간 거리 (단위 : m)

d_{ij}	1	2	3	4	5
1	-	84	65	72	95
2		-	78	56	69
3			-	82	87
4				-	100
5					-

(주) 3절의 문제 1은 허브 번호 1~3으로, 4절의 문제 1은 허브번호 1~4로 한정한다.

<표 4-6> 허브와 사용자 간 거리 (단위 : m)

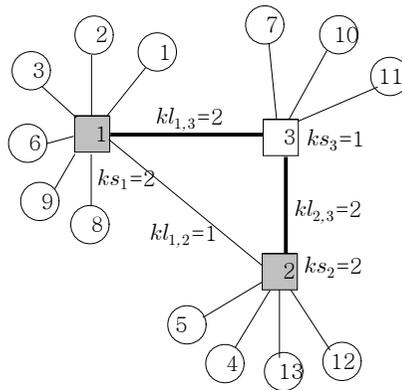
h_{ij}	사 용 자																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	22	27	15	39	32	35	50	35	28	49	32	34	33	43	41	26	35	17	20	32	49	14	33	46	42	46	34	52	50	37
2	46	44	33	19	25	49	44	43	37	46	45	26	29	49	46	49	50	43	36	42	18	23	19	43	25	18	47	79	46	43
3	32	43	24	44	45	50	20	41	41	37	35	47	48	16	43	50	48	32	13	33	50	44	32	36	34	30	32	23	21	33
4	34	42	17	46	26	19	43	29	26	15	16	38	47	35	49	36	45	49	34	30	48	47	18	21	47	26	19	28	48	37
5	45	14	50	45	34	45	49	31	42	40	29	46	37	49	49	24	44	35	15	21	13	37	25	46	38	15	23	44	49	43

(주) 3절의 문제 1은 허브 번호 1~3과 사용자 번호 1~13으로, 4절의 문제 1은 허브 번호 1~4와 사용자 번호 1~30으로 한정한다.

이와 같은 두 문제를 위해 사용된 허브 간 링크 유형의 속성은 <표 4-3>과 같다. 허브와 사용자 간의 단위거리 당 링크 설치비용(cu)은 1.0\$/m이다. 그리고 허브와 사용자 위치가 <그림 4-2>에 표현되어 있다. 허브의 속성, 허브간 거리 및 허브와 사용자간 거리가 각각 <표 4-4>, <표 4-5>, <표 4-6>에 나타나 있다. 문제 1을 위해, <그림 4-2>, <표 4-5>, 그리고 <표 4-6>에서는 허브 번호가 1~3, 사용자 번호가 1~13으로 한정되어 사용되고, <표 4-4>에서는 허브 유형이 1~2로 한정되어 사용된다.

1. 문제 1 : 3개의 허브와 13명의 사용자

3개의 허브와 13명의 사용자를 네트워크로 연결하는 문제 1에 대해 먼저 유전자 알고리즘을 이용하여 LAN 설계를 수행하였다. 초기 탐색 실험을 통해 초기화 값($g_{\max}=700$, $s=20$, $m\%=25$, $r_m=0.20$)을 얻은 후, 2.1절에서 제시한 설계 절차에 따라 실험을 수행하였다.



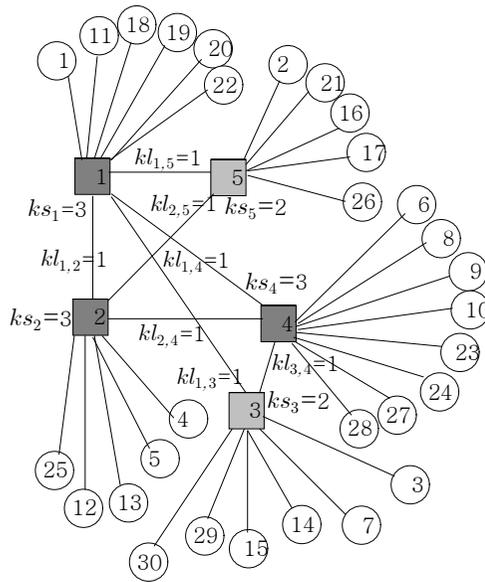
<그림 4-3> 가장 우수한 해($C(x)=1161.5\$$)

그 결과 <그림 4-3>에 나타나 있는 것과 같은 설계 구조를 갖는 LAN 설치비용이 1161.5\$이고 LAN 신뢰도가 0.852인 가장 우수한 해 {221 ; 122 ; 1112213113322}를 찾았다.

2. 문제 2 : 5개의 허브와 30명의 사용자

보다 넓은 탐색공간에 대한 유전자 알고리즘의 성능을 수행하기 위해 5개

의 허브에 30명의 사용자를 LAN으로 연결하는 문제 2에 대해 유전자 알고리즘을 이용하여 그 성능을 실험하였다. 초기 탐색 실험을 통해 초기화 값 ($g_{\max}=1,000$, $s=30$, $m\%=25$, $r_m=0.25$)을 얻은 후, 문제 1과 동일한 설계 절차에 따라 실험을 수행하였다. 그 결과 <그림 4-4>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 3366\$이고 네트워크 신뢰도가 0.851인 가장 우수한 해 {33232 ; 1111011100 ; 153224344412233551115144254433}를 찾았다.



<그림 4-4> 가장 우수한 해 ($\alpha_x=3366\$$)

제 4 절 비용을 제약조건으로 하는 경제적 LAN 설계의 성능 실험

본 절에서는 3절과는 달리 LAN 설치비용을 제약조건으로 하는 두 개의 LAN 설계 문제를 통해, 본 연구에서 제시하는 유전자 알고리즘을 이용한 설계 방법의 성능 실험을 앞서 3절과 동일한 실험환경에서 실시하였다.

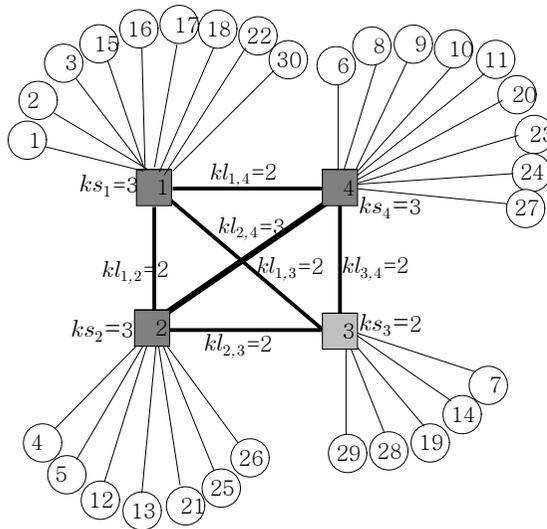
첫 번째 문제(문제 3)는 LAN 설치비용 제약조건(C_0)이 3,500\$일 경우에서 4개의 허브와 30명의 사용자를 네트워크로 연결하는 경우이고, 두 번째 문제(문제 4)는 LAN 설치비용 제약조건(C_0)을 4,300\$로 하여 5개의 허브와 30명의 사용자를 네트워크로 연결하는 경우이다.

이와 같은 두 문제를 위해 사용된 허브 간 링크 유형의 속성, 허브와 사용자 간의 단위거리 당 링크 설치비용(c_u), 허브와 사용자 간의 위치 및 허브의 속성은 3절과 동일하다. 그리고 허브 간 거리 및 허브와 사용자 간 거리가 각각 <표 4-5>, <표 4-6>에 나타나 있으나, 4개의 허브에 30명의 사용자를 연결하는 문제 1을 위해 <그림 4-2>, <표 4-5>, 그리고 <표 4-6>에서 허브 번호를 1~4로 한정하였다. 본 절의 두 문제 모두 허브 유형은 1~3, 사용자 번호는 1~30으로 한정한다.

1. 문제 3 : 4개의 허브와 30명의 사용자

LAN 설치비용 제약조건하에서 4개의 허브와 30명의 사용자를 LAN으로 연결하는 문제 3에 대해 먼저 유전자 알고리즘을 이용하여 LAN 설계를 수행하였다. 참고로 문제 3의 해 탐색공간은 $3.82512E+23(=3^4 \times 4^{4(4-1)/2} \times 4^{30})$

개 이다. 초기 탐색 실험을 통해 초기화 값($g_{\max}=800$, $s=30$, $m\%=25$, $r_m=0.20$)을 얻은 후, 2.1절에서 제시한 설계 절차에 따라 실험을 수행하였다. 그 결과 <그림 4-5>에 나타나 있는 것과 같은 설계 구조를 갖는 LAN 설치비용이 3,415.5\$이고 네트워크 신뢰도가 0.902인 가장 우수한 해 {3323; 222232 ; 111224344442231111342144224331}를 찾았다.

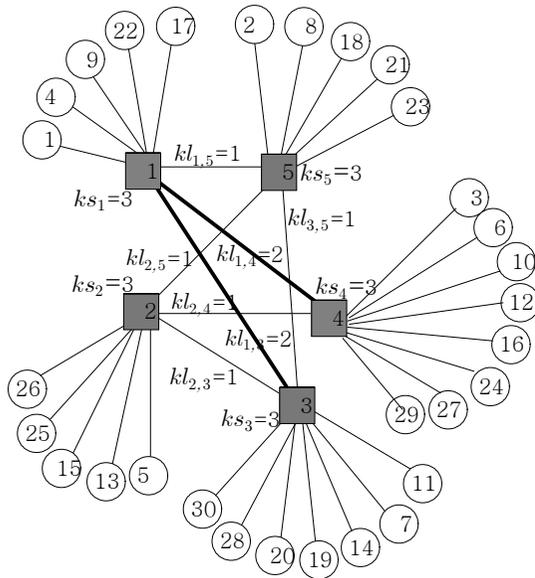


<그림 4-5> 가장 우수한 해 ($R(x)=0.902$)

2. 문제 4 : 5개의 허브와 30명의 사용자

LAN 설치비용 제약조건하에서 앞서 문제 3과 동일한 사용자 수에서 1개의 허브를 추가한 5개의 허브에 30명의 사용자를 LAN으로 연결하는 문제 4에 대해 유전자 알고리즘을 이용한 LAN 설계를 수행하였다. 초기 탐색 실험

험을 통해 초기화 값($g_{\max}=1,000$, $s=30$, $m\%=25$, $r_m=0.25$)을 얻은 후, 문제 3과 동일한 설계 절차에 따라 실험을 수행하였다. 그 결과 <그림 4-6>에 나타나 있는 것과 같은 설계 구조를 갖는 비용이 4,279.5\$이고 네트워크 신뢰도가 0.895인 가장 우수한 해 {33333 ; 0221111010 ; 154124351434232415335154224344}를 찾았다.



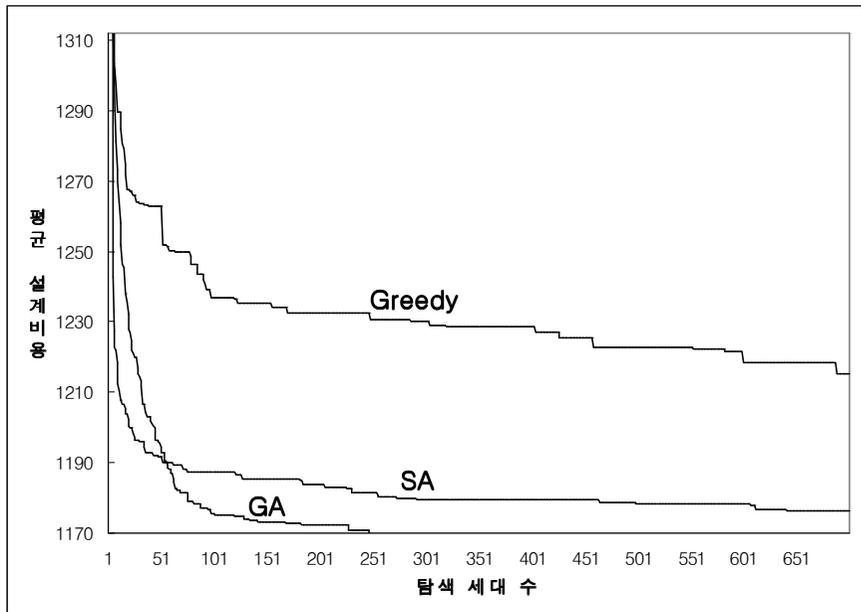
<그림 4-6> 가장 우수한 해 ($R(x)=0.895$)

제 5 절 성능 비교 및 민감도 분석

1. 유전자 알고리즘의 성능 비교

LAN 신뢰도를 제약조건으로 하는 문제 1은 계산 중심적 탐색을 할 수 없을 정도로 탐색 대상 해의 수(816,293,376)가 매우 크다. 그러므로 계산 중심적 탐색 방법 대신에 SA와 Greedy 탐색 방법을 사용하여 유전자 알고리즘을 이용한 LAN 설계 방법과의 성능을 비교하고자 한다.

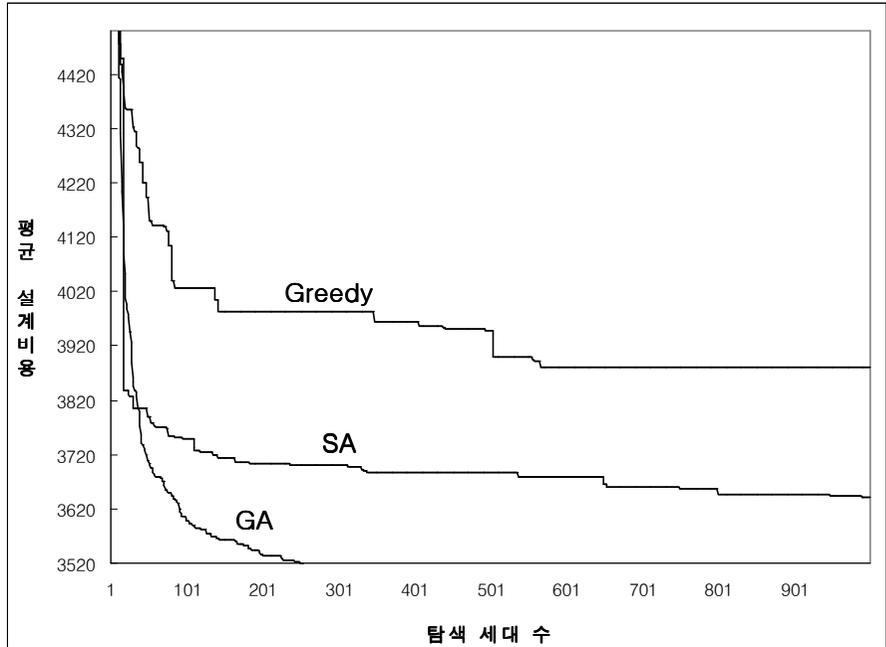
유전자 알고리즘을 이용한 LAN 설계 방법과 SA 및 Greedy 탐색 방법 각각에 대해 10개의 seed를 사용하여, 탐색 세대 수의 증가에 따른 평균 설치비용의 변화 추이를 살펴보았다. 그 결과가 <그림 4-7>에 나타나 있다.



<그림 4-7> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 1)

유전자 알고리즘을 이용한 설계 방법은 탐색 세대 수가 증가함에 따라 평균 설계비용이 급격히 떨어져 약 70세대 수에서 1,180\$ 이하로 수렴하는 반면에 SA는 초기 세대에서 GA보다 급격한 비용감소가 이루어지고 있으나 700세대 수에 이르기까지 약 1,176\$ 이상에서 더 이상 비용의 감소가 이루어지지 않고 있음을 볼 수 있다. 이에 비해, Greedy 탐색 방법은 700세대 수에 이르기까지 약 50번의 설계비용 감소를 가졌으며, 700세대 수에서 평균 설계비용 1,215\$를 가졌다. 이 결과는 LAN 신뢰도를 제약조건으로 하고, LAN 설치비용을 최소화 하는 문제에 있어 유전자 알고리즘을 이용한 LAN 설계 방법이 SA나 Greedy 탐색 방법보다 효율적인 설계 방법이라는 것을 보여 주고 있다.

5개의 허브에 30명의 사용자를 LAN으로 연결하는 문제 2는 문제 1보다 훨씬 더 탐색 대상 해의 수($3^5 \times 5^{5(5-1)/2} \times 5^{30} = 2.21007E+30$)가 크다. 그러므로 계산 중심적 탐색 방법을 적용하기 어려워 문제 1에서와 마찬가지로 SA와 Greedy 탐색 방법을 사용하여 유전자 알고리즘을 이용한 설계 방법과 성능을 비교하였다. 세 가지 설계 방법을 적용한 탐색 세대 수의 증가에 따른 평균 설치비용의 변화 추이가 <그림 4-8>에 표현되었다.

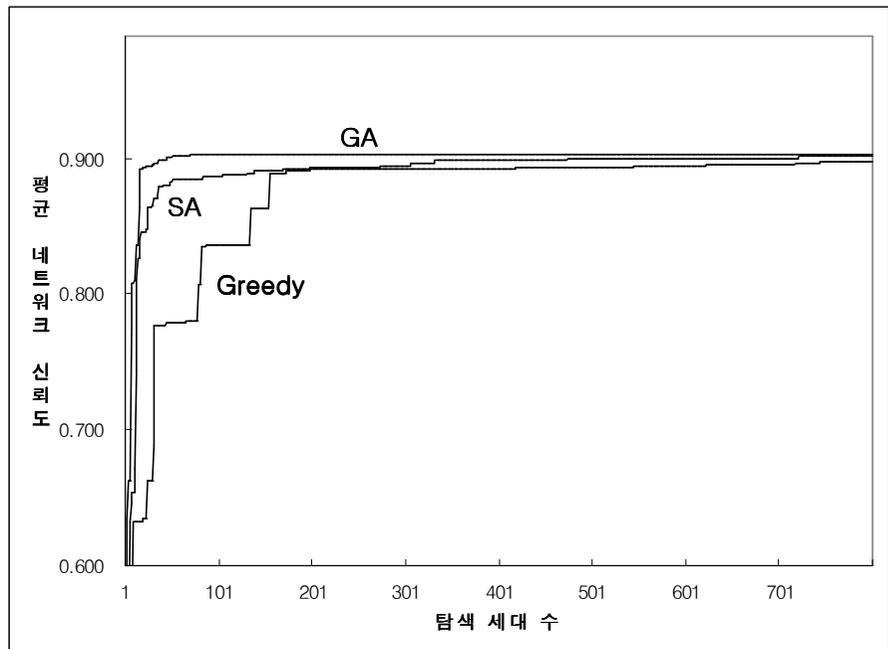


<그림 4-8> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 2)

유전자 알고리즘을 이용한 설계 방법은 탐색 세대 수가 증가함에 따라 평균 설계비용이 급격히 떨어져 약 250세대 전후에서 3,520\$ 이하로 수렴하고 있으나, SA를 이용한 설계방법은 이전과 마찬가지로 초기에는 GA보다 급속한 비용감소가 이루어지고 있으나, 1,000세대 수에 이르기까지 약 3,642이상의 비용에서 더 이상 감소가 이루어지지 않고 있음을 볼 수 있다. 이 두 가지 방법에 비해, Greedy 탐색 방법은 초기 해에서 1,000세대 수에 이르기까지 약 40번의 설계비용 감소를 가졌으며, 1,000세대 수에서 평균 설계비용 3,880\$를 가졌다. 이 결과는 문제 1보다 더 큰 규모인 문제 2에 대해서도 문제 1에서 얻은 결과와 유사하게 유전자 알고리즘을 이용한 LAN 설계 방법이 SA나 Greedy 탐색 방법보다 효율적인 설계 방법이라는 것을 보여 주

고 있다.

<그림 4-9>는 앞서 LAN 신뢰도를 제약조건으로 한 두 문제와는 달리 LAN 설치비용을 제약조건으로 하고 LAN 신뢰도를 최대화 하는 4절의 문제 3에 대해 제약 조건(C_0)의 다양한 변화에 따른 평균 LAN 신뢰도 변화 추이를 SA 및 Greedy 탐색방법과 비교한 결과이다.

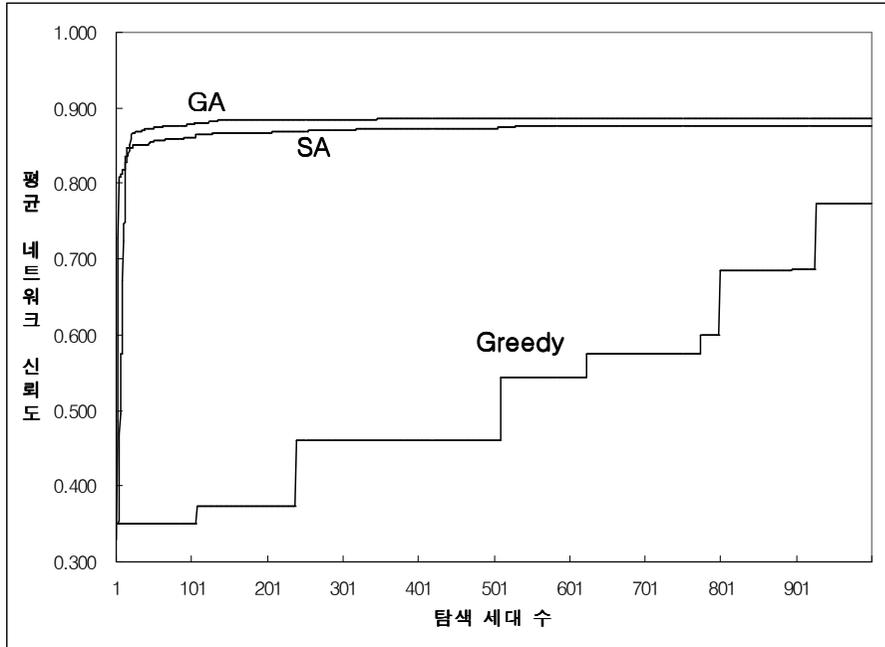


<그림 4-9> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 3)

유전자 알고리즘을 이용한 설계 방법은 탐색 세대 수가 증가함에 따라 평균 네트워크 신뢰도가 급격히 증가하여 약 50세대 수에서 0.902 이상으로 증가한 후 800세대 수에 이르러 평균 0.9024로 수렴하고 있으나, SA방법은 800세대 수에 이르기까지 평균 0.897이하로 수렴하고 있음을 볼 수 있다. 이

에 비해, Greedy 탐색 방법은 유전자 알고리즘과는 달리 약 720세대 수에서 비로소 0.900 이상으로 증가하고, 800세대 수에 이르러서도 LAN 신뢰도가 평균 0.901이하로 수렴하고 있음을 알 수 있다. 이 결과에서 특이한 점은 이전의 결과와는 달리 SA방법보다 Greedy 탐색방법이 더 효율적이라는 것이다. 이는 사용자 수에 비해 충분한 허브용량의 확보로 전체 LAN 신뢰도 확보가 용이해졌기 때문인 것으로 판단되며, 이로 인해 이웃 해를 탐색하는 SA보다는 다소 다양한 영역에서 해를 탐색하는 Greedy 알고리즘이 보다 효율적일 수 있음을 보여주는 결과라 할 수 있다. 그러나 이러한 결과에도 불구하고, 유전자 알고리즘을 이용한 방법이 우수한 해에 접근하는 속도에서나 평균 네트워크 신뢰도에서 여타 두 가지 탐색방법보다 효율적으로 우수 해를 도출하고 있음을 알 수 있다. 그러므로 LAN 설치비용을 제약조건으로 하고 LAN 신뢰도를 최대화 하는 문제 있어서도 유전자 알고리즘을 이용한 LAN 설계 방법이 SA나 Greedy 탐색 방법보다 다소 효율적인 설계 방법이라는 것을 보여 주고 있다. 이 결과는 설치비용을 제약조건으로 4개의 허브에 30명의 사용자를 연결하는 경우, 본 연구에서 제시한 유전자 알고리즘이 주어진 실험환경에서 다소 효율적이지만, SA나 Greedy 탐색 방법도 차선책으로 고려해 볼 수 있는 방법이라 할 수 있다.

문제4는 3절의 문제 2와 마찬가지로 탐색 대상 해의 수($=2.21007E+30$)가 크다. 그러므로 SA와 Greedy 탐색 방법을 사용하여 유전자 알고리즘을 이용한 설계 방법과 성능을 비교하였다. 두 설계 방법을 적용하여 실험한 결과인 탐색 세대 수의 증가에 따른 평균 LAN 신뢰도 변화 추이가 <그림 4-10>에 표현되어 있다.



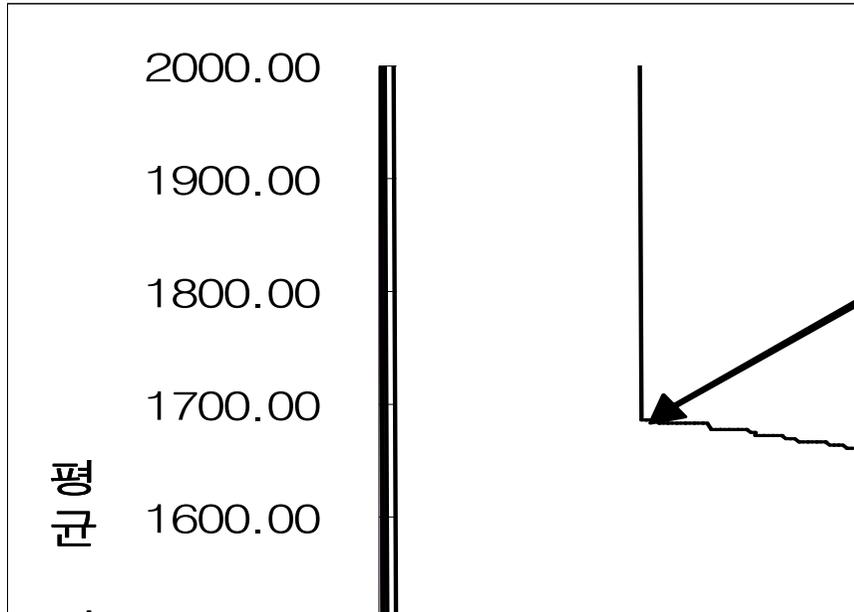
<그림 4-10> GA 설계 방법과 SA 및 Greedy 탐색 방법의 성능 비교(문제 4)

유전자 알고리즘을 이용한 설계 방법은 탐색 세대 수가 증가함에 따라 평균 네트워크 신뢰도가 급격히 증가하여 약 100세대 전후에서 0.880 이상으로 수렴하고 있다. 그러나 SA를 이용한 설계방법은 초기 GA와 비슷한 네트워크 신뢰도 증가가 이루어지고 있지만 1,000세대 수에 이르기까지 약 0.877이하의 신뢰도에서 더 이상 증가가 이루어지지 않고 있음을 볼 수 있다. 이에 비해, Greedy 탐색 방법은 초기 해에서 1,000세대 수에 이르기까지 약 0.773이상으로 네트워크 신뢰도가 증가하지 않았다. 그러므로 문제 3보다 더 큰 규모에서도 유전자 알고리즘이 Greedy 탐색 방법보다는 훨씬 효율적이지만, SA와 비교해서는 그 결과에 있어 다소 효율적이나 월등하게 우월하다고는 할 수 없다. 이 결과는 LAN 설치비용을 제약조건으로 하는 LAN

설계 문제에 있어, LAN 설계 규모가 보다 확대될 경우에는 유전자 알고리즘을 이용한 LAN 설계 방법이 Greedy 탐색 방법에 비해 보다 더 효율적인 설계 방법이지만, SA 탐색 방법도 차선택으로 사용될 수 있는 설계 방법이라는 것을 보여 주고 있다.

2. 민감도 분석

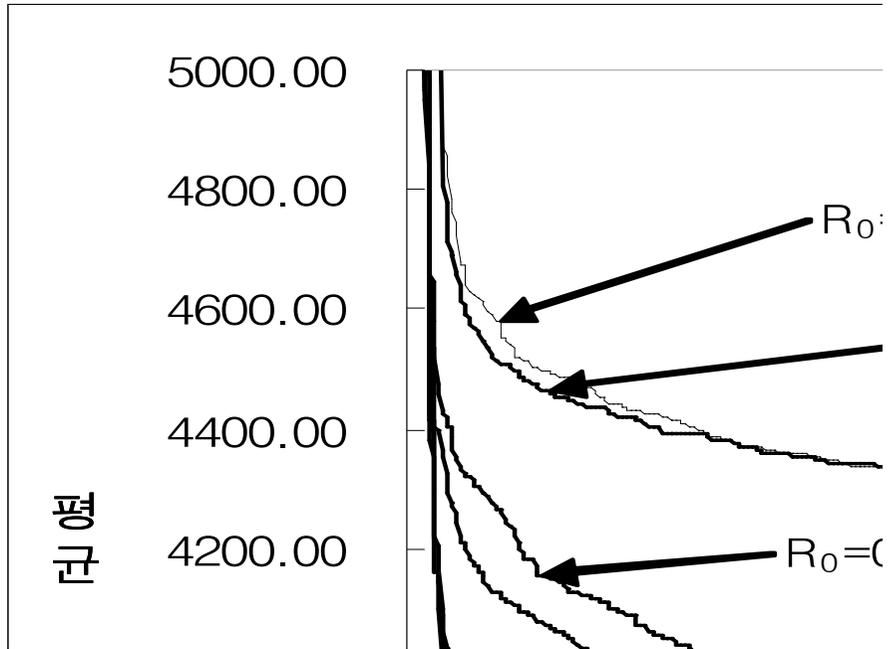
<그림 4-11>은 LAN 신뢰도를 제약조건으로 하는 3절의 문제 1에 있어, LAN 신뢰도 제약조건(R_0)의 변화에 따른 평균 설계비용의 변화를 관찰하기 위해 신뢰도 제약조건을 다양하게 변화시킨 결과이다.



<그림 4-11> LAN 신뢰도 변화에 따른 민감도 분석결과(문제 1)

이 결과를 보면, 신뢰도가 0.840 ~ 0.870인 경우 비슷한 세대 수에서 특정 비용으로 수렴하고 있으나, 0.875의 경우 보다 높은 설계비용으로 수렴하고 있음을 볼 수 있다. 이는 신뢰도를 0.875로 제한하는 경우 주어진 유전자 알고리즘의 조건하에서 우수 해를 찾는 seed의 수가 다른 조건들보다 적은 결과로 판단된다.

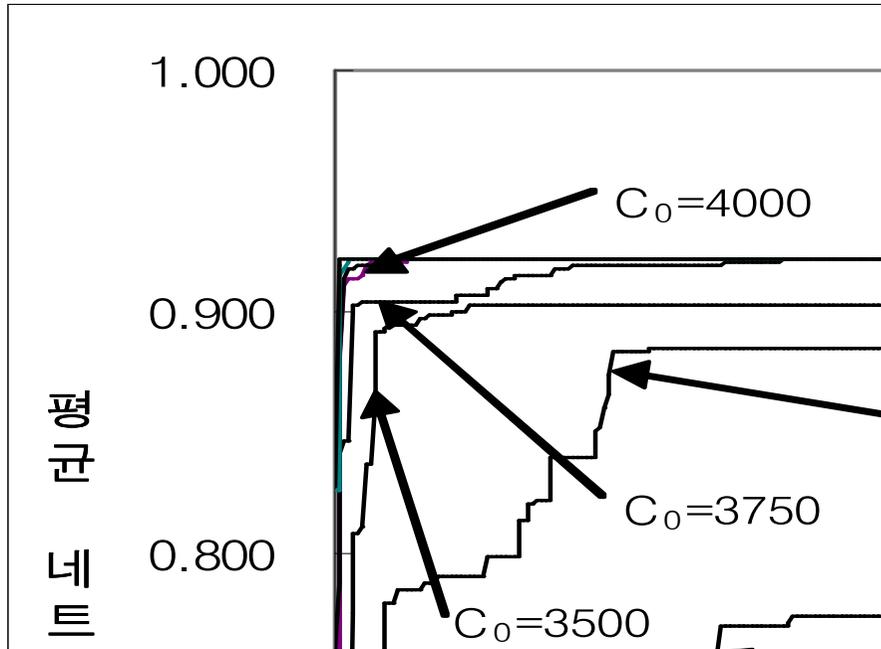
<그림 4-12>는 문제 1보다 탐색공간이 큰 문제 2의 경우에 대해 LAN 신뢰도 제약조건(R_0)의 변화에 따른 평균 설계비용의 변화를 관찰하기 위한 민감도 분석결과를 그래프로 표현한 것이다.



<그림 4-12> LAN 신뢰도 변화에 따른 민감도 분석결과(문제 2)

이 결과에서도 신뢰도 제약조건을 높일 경우, 탐색 세대 수가 증가함에 따라, 평균 설계비용이 급격히 낮아짐을 볼 수 있으며, 0.900과 0.890, 0.880과 0.870 및 0.860과 0.850의 설계비용이 비슷함을 볼 수 있다. 이는 각 10개의 seed 중 비슷한 수의 우수 해를 찾은 결과 때문인 것으로 판단된다.

<그림 4-13>은 앞선 두 문제와는 달리 LAN 설치비용을 제약조건으로 하는 4절의 문제 3에 대해 LAN 설치비용의 다양한 변화에 따른 평균 네트워크 신뢰도의 변화를 관찰하기 위한 민감도 분석결과이다.

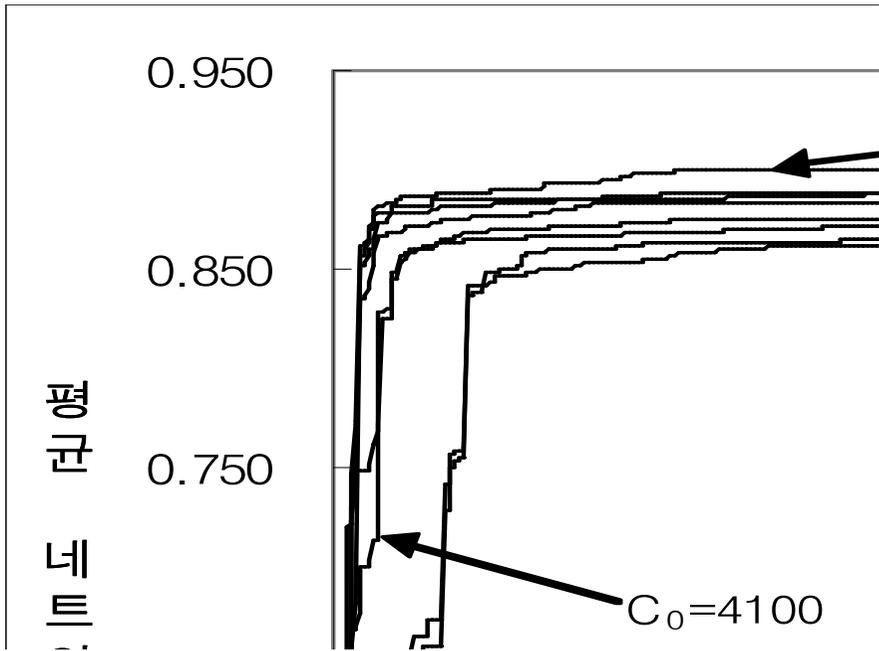


<그림 4-13> LAN 설치비용 변화에 따른 민감도 분석결과(문제 3)

이 결과는 제약조건인 LAN 설치비용을 완화할 경우, 보다 높은 수준의 LAN 신뢰도를 확보할 수 있음을 나타내고 있다. 특히, 하위 2개의 조건에서는 그 설계비용의 차이가 500\$정도에도 두 조건의 신뢰도 차이가 높게 나타나는 것에 비해, 상위 500\$의 차이를 갖는 $C_0=4,000\$$ 와 $C_0=4,500\$$ 일 경우에도 그 신뢰도에서 크게 차이가 나타나지 않음을 볼 수 있다. 이는 LAN 설치비용 제약이 4,000\$를 초과하는 경우 주어진 조건하에서 도출할 수 있는 LAN 신뢰도가 상한에 근접한 결과 때문인 것으로 판단된다.

<그림 4-14>는 LAN 설치비용을 제약조건으로 하는 4절의 문제 4에 대해 LAN 설치비용의 다양한 변화에 따른 평균 네트워크 신뢰도의 변화를 관찰하기 위해 LAN 설치비용 제약조건을 다양하게 변화시킨 민감도 분석

결과이다.



<그림 4-14> LAN 설치비용 변화에 따른 민감도 분석결과(문제 4)

이 결과를 보면, <그림 4-13>의 문제 3과 동일하게 세대 수가 증가함에 따라 평균 네트워크 신뢰도가 급격히 높아지고 있다. 특히, 문제 4의 경우는 문제 3과 동일한 사용자수에서 허브의 수를 하나 증가시킨 결과로써, 주어진 제약조건의 변화에 대해 모두 0.9이상으로 수렴함을 볼 수 있다. 그러므로 동일한 사용자 환경에서 새로운 허브의 추가는 보다 원활한 네트워크 흐름을 보장할 수 있는 것으로 판단된다.

제 5 장 결 론

제 1 절 연구 요약

컴퓨터 네트워크가 기업의 정보 공유, 신속한 의사결정 그리고 고객 서비스를 위한 중요한 수단으로 활용되면서, 네트워크의 단절은 기업 활동에 직접적인 영향을 미치게 되었다. 이에 따라 기업은 최소의 설치비용으로 안정적인 컴퓨터 네트워크를 설치하여 운영하길 원한다. 기존에 기업에 설치되어 운영되고 있는 컴퓨터 네트워크는 경영환경의 변화에 따라 기업의 조직 및 시장 확장, 그리고 정보기술의 발전에 맞추어 네트워크의 확장, 개선 및 새로운 네트워크 설계가 요구된다.

기업의 입장에서 보면, 기존 백본 네트워크를 확장 설계함에 있어 네트워크의 경제성 문제는 무엇보다도 우선적으로 고려되어야 할 사항이다. 그러므로 네트워크 신뢰성을 제약조건으로 하고 최소 네트워크 설계비용을 목적함수로 하여 네트워크를 설계하는 방법과 네트워크 설계비용을 제약조건으로 하고 최대 네트워크 신뢰성을 목적함수로 하여 네트워크를 설계하는 방법 등이 고려되어질 수 있다.

백본 네트워크 확장 시 기존 설치된 네트워크 매체(통신 케이블, 허브 등)의 교체 또한 고려되어야 할 사항이다. 즉 네트워크를 확장 할 경우, 기존에 설치된 네트워크 매체의 교체를 통해 보다 높은 신뢰성을 확보 할 수 있다면 기존 케이블이나 허브 등의 교체도 기업 입장에서는 고려할 수 있는 사항이며 이러한 교체비용도 운영비용으로 포함되어져야 한다. 그러나 모든

통신 매체들을 동시에 동일한 사양으로 교체하는 것은 여러 가지 측면에서 제약이 있을 수 있다.

네트워크 신뢰도를 중시하는 기업에서는 백본 네트워크 확장 설계 시 비용측면보다는 신뢰성 있는 네트워크의 운영이 보다 더 중요한 부분이기 때문에 주어진 네트워크의 신뢰도를 달성하는데 드는 비용을 최소화 하는 것에 많은 관심을 둔다.

이에 비해, 네트워크 설치비용을 중시하는 기업에서는 백본 네트워크 신뢰도의 확보에 대해 객관적인 기준이 없을 경우, 일반적으로 허용될 수 있는 신뢰도 기준(예 : 0.9) 이상의 네트워크 신뢰도만 확보된다면, 네트워크 설치비용이 기업차원에서 고려되어야 할 중요 요소로써 작용된다.

백본 네트워크 설계와 더불어 기업 내 제한된 공간에서 사용자들 간의 정보교환이 증가함에 따라 기업의 업무 수행에 직접적인 영향을 미치는 LAN 은 기업의 네트워크를 구성하는 중요 요소로써 이용되고 있다. LAN 설계 문제는 두 개의 중요한 이슈로써 사용자들을 각 허브에 할당하는 방법 (clustering)과 각 허브 간 연결방법(routing)이 포함되어져야 한다(Elbaum & Sidi, 1996). 백본 네트워크 확장 설계 문제와 마찬가지로 LAN 설계 문제 또한 네트워크 설계와 관련된 예산 범위 내에서 최대한의 네트워크 신뢰도를 확보하는 경제성 문제로 대표될 수 있다.

그러므로 본 연구에서는 이러한 기업의 요구사항 및 현실성에 맞추어 네트워크 신뢰도 제약 및 비용 제약 하에서 새 노드 부가 및 여러 유형의 링크 수준과 함께 기존 매체의 교체비용까지 고려하는 경제적 백본 네트워크 확장 설계 방법을 제시하였다. 더불어 다양한 허브의 수준 및 링크 수준을 고려하여 사용자들을 각 허브에 할당하고 할당된 허브 간 연결을 포함하는 경제적 LAN 설계 방법을 제시하였다.

본 논문의 주요 결과를 요약하면 다음과 같다.

첫째, 유전자 알고리즘을 이용한 백본 네트워크 확장 설계 문제에 있어, 신뢰도를 제약조건으로 하여 다양한 노드를 부가할 뿐만 아니라, 다양한 링크 수준까지 고려한 네트워크 설계 방법을 제안하였다. 유전자 알고리즘을 네트워크 설계 문제에 적용함에 있어 중요한 염색체 구조를 k -nery 코딩 방법으로 표현하였으며, 백본 네트워크 신뢰도 계산을 위한 신뢰도 계산방법을 제시하였다.

본 연구에서 제안된 유전자 알고리즘을 이용한 백본 네트워크 설계 방법을 계산 중심적 탐색, SA 및 Greedy 탐색 방법과 비교함으로써 성능의 우수성을 입증하였으며, 민감도 분석을 통해 신뢰도 제약 조건의 변화에 따라 평균 설계비용이 다양한 수준에서 수렴하고 있음을 제시하였다.

이 결과를 통해, 기업에서 기존에 구성된 백본 네트워크에서 기존 네트워크의 신뢰도를 떨어뜨리지 않거나 또는 기존의 네트워크 신뢰도를 높이면서 새로운 노드를 추가하는 경우에 이용될 수 있다. 또한 기존의 케이블을 교체해야 할 경우에 보다 효율적으로 네트워크를 설계할 수 있을 것으로 판단된다.

둘째, 유전자 알고리즘을 이용한 백본 네트워크 확장 설계 문제에 있어, 네트워크 설계비용을 제약조건으로 하여 새로운 노드 부가 및 다양한 링크 수준을 고려하여 네트워크 신뢰도를 최대화 하는 네트워크 설계 방법을 제안하였다. 유전자 알고리즘을 이용하여 이러한 설계 방법을 적용함에 있어 신뢰도를 제약조건으로 하고 비용을 최소화하는 방법과 동일하게 염색체 구조와 신뢰도 설계 방법을 이용하였다.

제안된 유전자 알고리즘을 이용한 백본 네트워크 설계 방법을 계산 중심

적 탐색, SA 및 Greedy 탐색 방법과 비교함으로써 성능의 우수성을 입증하였으며, 민감도 분석을 통해 네트워크 설치비용 제약 조건의 변화에 따라 평균 네트워크 신뢰도가 다양한 수준에서 수렴하고 있음을 제시하였다. 이는 네트워크 신뢰도를 제약조건으로 하고 비용을 최소화 하는 방법뿐만 아니라, 네트워크 설계비용을 제약조건으로 하고 네트워크 신뢰도를 최대화 하는 방법에서도 본 연구에서 제안된 네트워크 설계방법이 우수함을 입증할 수 있었다.

이 결과를 볼 때, 기업에서 기존에 구성된 백본 네트워크에서 주어진 네트워크 설계비용의 예산에 맞게 최소한의 네트워크 신뢰도를 확보하면서 새로운 노드를 추가하는 경우에 이용될 수 있다. 나아가 기존의 케이블을 교체해야 할 경우 보다 효율적으로 네트워크를 설계할 수 있을 것으로 판단된다.

셋째, 유전자 알고리즘을 이용한 LAN 설계 문제에 있어, 신뢰도를 제약조건으로 하여 각 사용자들을 연결하기 위해 다양한 허브 유형과 링크 유형을 고려한 설계 방법을 제안하였다. 유전자 알고리즘을 이용한 LAN 설계에 있어 중요한 구성요소인 허브, 링크 및 사용자들을 k-nergy 코딩 방법을 통해 염색체로 표현하였다. 또한, LAN 신뢰도를 계산하기 위한 신뢰도 계산 방법을 제시하였다.

본 연구에서 제안된 유전자 알고리즘을 이용한 LAN 설계 방법을 SA 및 Greedy 탐색 방법과 비교한 결과, 거의 유사한 수준에서 수렴하고 있으나, 주어진 실험환경 내에서는 유전자 알고리즘이 다소 우수함을 알 수 있었다. 또한, 민감도 분석을 통해 유전자 알고리즘을 이용한 LAN 설계방법이 신뢰도 제약 조건의 변화에 따라 평균 설계비용이 다양한 수준에서 수렴하고 있음을 제시하였다.

이러한 결과를 통해, 기업들은 본 연구에서 제시한 유전자 알고리즘을 통해 기업 내 새로운 LAN 설계시 필요로 하는 LAN 신뢰도를 만족시키면서 경제적인 비용으로 LAN 네트워크를 설계할 수 있을 것으로 판단된다.

넷째, 유전자 알고리즘을 이용한 LAN 설계 문제에 있어, 설치비용을 제약조건으로 하고 최대의 네트워크 신뢰도로 각 사용자들을 연결하기 위한 LAN 설계방법을 또한 제시하였다.

이는 신뢰도를 제약조건으로 하는 LAN 설계방법과 동일한 절차로 이루어졌다. 타 기법과의 비교 실험 결과, 본 연구에서 제시한 유전자 알고리즘을 이용한 LAN 설계 방법이 Greedy 탐색 방법보다는 월등하게 우수하였으나, SA 탐색 방법에 비해서는 약간 우수함을 알 수 있었다. 이는 차선책으로써 SA 탐색방법도 설치비용 제약 조건하에서 효율적인 LAN 설계가 가능한 방법이라는 것을 의미한다. 마지막으로 민감도 분석을 통해 본 연구에서 제시한 유전자 알고리즘을 이용한 LAN 설계방법이 설치비용 제약 조건의 변화에 따라 평균 네트워크 신뢰도가 다양한 수준에서 수렴하고 있음을 제시하였다.

이러한 결과를 통해, 기업은 새로운 LAN 설계시 주어진 설치비용을 초과하지 않는 범위 내에서 최대 네트워크 신뢰도를 확보할 수 있는 경제적 LAN 네트워크를 설계할 수 있을 것으로 판단된다.

제 2 절 연구의 한계점 및 향후 연구 방향

본 연구에서 수행된 연구결과를 바탕으로 연구의 한계점 및 향후 연구 방향을 제시하면 다음과 같다.

본 연구에서 제시된 유전자 알고리즘의 성능 우수성을 비교 분석함에 있어 SA와 Greedy 알고리즘을 통한 결과를 제시하였다. 특히, SA를 이용한 탐색 방법은 이전 연구에서 제시된 각 파라미터 값을 그대로 사용하여 성능의 우수성을 제시하였다. 그러나 이 결과는 유전자 알고리즘과의 탐색 구조상의 차이에 기인한 측면도 있으므로, 차후 연구를 통해 본 논문에 제시된 유전자 알고리즘의 파라미터 값과 비교가 가능하도록 SA의 파라미터 값을 조절한 후 그 결과를 분석할 필요가 있을 것으로 판단된다. 더불어 최근 많이 사용되고 있는 Tabu Search나 Ant Colony방법과의 비교분석도 필요할 것으로 판단된다.

본 연구에서는 백본 네트워크 확장문제를 다룸에 있어서, 네트워크 설계와 관련되어 가장 많이 사용되고 있는 요소인 네트워크 신뢰도와 설치비용만을 다루고 있으나, 또 다른 중요 요소인 평균 패킷 지연시간(mean packet delay)을 제약조건으로 한 최대 신뢰도 및 최소 설치비용 확보방안도 고려되어져할 문제로 판단된다.

본 연구에서는 네트워크 확장문제를 다룸에 있어 백본 네트워크만을 그 대상으로 하였다. 그러나 기업 내에서 기존에 구축된 LAN의 확장 설계 문제도 빈번히 고려될 수 있는 사항이므로 이 부분에 대해서도 백본 네트워크의 확장 문제와 유사한 방법의 연구가 향후 이루어져야 할 것으로 본다.

또한 본 연구에서는 네트워크 및 LAN의 신뢰도라는 하나의 제약조건에 네트워크 설계비용과 같은 하나의 목적함수를 만족시키는 설계방법을 제안하였다. 그러나 네트워크 설계에 있어 하나의 제약조건에 둘 이상의 목적을 만족할 수 있는 즉, 특정 제약조건 하에서 최대 네트워크 신뢰도와 최소 네트워크 설치비용을 동시에 만족시킬 수 있는 설계방법에 대한 연구가 더불어 이루어져야 할 것으로 본다.

참 고 문 헌

1. 국내 문헌

고대식, 서태석, 류명주, 최세하, *데이터 통신 및 LAN 설계*, 도서출판 아진, 2000.

김기화, “유전자 알고리즘”, *전산구조공학*, Vol. 9, No. 2, 1996, pp. 28-35.

김여근, 윤복식, 이상복 공저, *메타 휴리스틱*, 영지문화사, 2004.

문병로, *유전알고리즘*, 다성출판사, 2001.

박우열, “유전자 알고리즘을 이용한 철근공사 최적 물량산출 모델 및 관리 시스템 개발”, *고려대학교 대학원 건축공학과, 박사학위논문*, 2003.

이기혁, 조동성, 김기남 공저, *데이터 네트워크 구축론*, 진한도서, 2000.

이문구, *데이터 통신 시스템*, 도서출판 글로벌, 2000.

염창선, 배상욱, “하이브리드 유전자 접근방법을 이용한 네트워크 확장을 위한 경제적 설계”, *산업경영시스템학회지*, Vol. 28, No. 1, 2005, pp.

8-15.

Melanie Mitchell, *유전자 알고리즘 입문*, 공성곤 외 4인 공역, 진영사, 1997.

2. 국외 문헌

Aggarwal, K. K., Gupta, J. S., and Misra, K. B., "A Simple Method for Reliability Evaluation of a Communication System", IEEE Transactions On Communication System, Vol. 23, No. 5, 1975, pp. 563-566.

Aggarwal, K. K. and Rai, S., " Reliability evolution of computer-communication networks.", IEEE Transactions on Reliability, Vol. 30, No. 1, 1981, pp. 32-35.

Antonisse, J., "A new interpretation of schema notation that overturns the binary encoding constraint", International Conference on Genetic Algorithms, 1989, pp. 86-91.

Arts, E. H. L, and Korst, J. H. M., *Local Search in Combinatorial Optimization*, Chichester, UK, John Wiley & Sons., 1997.

Bäck, T. and H. Schewefel, "An overview of evolution algorithms for parameter optimizations, Evolutionary Computation, Vol. 1, No. 1, 1993, pp. 1-23,

Berry, Michael J. A. and Linoff, Gordon S., *Data mining techniques for*

marketing, sales, and customer support, John Wiley & Sons, Inc., 1997.

Booker, L., Improving search in genetic algorithms, in Davis, L., editor, *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publishers, 1987.

Celli, G., Costamagna, E. and Fanni, A., "Genetic algorithms for telecommunication network optimization", *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, 1995.

Costamagna, E., Fanni, A. and Giacinto, G., "A Simulated Annealing Algorithm for the Optimization of Communication Networks", in : *Proceedings of International Symposium On Signals, Systems and Electronics*, San Francisco, 1995, pp. 405-408.

Davis, L., "Applying adaptive algorithms with epistatic domains", *International Joint Conference on Genetic Artificial Intelligence*, 1985, pp. 162-164.

Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

Deeter, D. L., and Smith, A. E., "Economic Design of Reliable Networks",
IIE Transactions, Vol. 30, No. 12, 1998, pp. 1161-1174.

Dengiz, B., Altiparmak, F. and Smith, A. E., "A genetic algorithm
approach to optimal topology design of all terminal networks", in
Intelligent Engineering Systems Through Artificial Neural Networks,
Vol. 5, 1995, pp. 405-410.

Elbaum, R. and Sidi, M., "Topological Design of Local-Area Networks
using Genetic Algorithms", IEEE/ACM Transactions on
Networking, Vol. 4, No. 5, 1996, pp. 766-778.

Ersoy, C. and Panwar, S. S., "Topological Design of Interconnected
LAN/MAN networks", IEEE Journal on Selected Areas in
Communications, Vol. 11, No. 8, 1993, pp. 1172-1182.

Filho, V. J. M. and Galvão, R. D., "Design of Interconnection of Local Area
Networks", International Transactions in Operational Research, Vol. 5, No.
3, 1998, pp. 211-231.

Fogel, D., *Evolutionary Computation : Toward a New Philosophy of
Machine Intelligence*, IEEE Press, 1995.

Fogel, L., A. Owens, and M. Walsh, *Artificial Intelligence Through*

Simulated Evolution, Wiley, 1966.

Frank, H. and Chou, W., "Topological Optimization of Computer Networks", Proceedings of the IEEE, Vol. 60, No. 11, 1972, pp. 1385-1397.

Gen, M. and Cheng R., "A survey of penalty techniques in genetic algorithms", Proceedings of the IEEE International Conference on Evolutionary Computation, 1996, pp. 804-809.

Gen, M. and Cheng R., *Genetic Algorithm & Engineering Optimization*, John Wiley & Sons, 2000.

Gen, M., IDA, K., and Kim, J. R., "A Spanning Tree-Based Genetic Algorithm for Bicriteria Topological Network Design", Proceedings of IEEE International Conference on Evolutionary Computation, 1998, pp. 15-20.

Goldberg., D. and Lingle., R., "Alleles, loci, and the traveling salesman problem", International Conference on Genetic Algorithms, 1985, pp. 154-159.

Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

Goldberg, D. E., "Real-Coded Genetic Algorithms, Virtual Alphabets and Blocking", *Complex Systems*, Vol. 5, 1991, pp. 139-167.

Holland, J., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975; MIT Press, 1992.

Jan, R. H., Hwang, F. H., and Cheng, S. T., "Topological Optimization of a Communication Network Subject to a Reliability Constraint", *IEEE Transactions of Reliability*, Vol. 42, No. 1, 1993, pp. 63-70.

Kershenbaum, A., *Telecommunication Network Design Algorithms*, McGraw-Hill, Inc., New York, 1993.

Kirkpatrick, S., Gelatt Jr., C. D., and Vecchi, M. P., "Optimization by Simulated Annealing", *Science*, vol. 220, 1983, pp. 671-680.

-a. Kumar, A., Pathak, R. M., and Gupta, Y. P., "Genetic-Algorithm-Based Reliability Optimization for Computer Network Expansion", *IEEE Transactions on Reliability*, Vol. 44, No. 1, 1995, pp. 63-72.

-b. Kumar, A., Pathak, R. M., Gupta, Y. P., and Parsaei, H. R. "A Genetic algorithm for distribute system topology design", *Computers and Industrial Engineering*, Vol. 28, No. 3, 1995, pp.

659-670.

Kumar, A., Pathak, R. M., Gupta, Y. P., and Gupta, Y. P., "Genetic Algorithm Based Approach for Designing Computer Network Topology", Proceedings of the 1993 ACM Conference on Computer Science, 1993, pp. 358-365.

Laszewski, V. G., "Intelligent structural operators for the k-way graph partitioning problem", International Conference on Genetic Algorithms, 1991, pp. 45-52.

Mandal, S., Saha, D., Mukherjee, R., and Roy, A., "An Efficient Algorithm for Designing Optimal Backbone Topology for a Communication Networks", Proceedings of ICCT 2003, Vol. 1, 2003, pp. 103-106.

Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolutionary Programs*, Springer, 1992.

Oliver, I. M., Smith, D. J. and Holland, J. R. C., "A Study of Permutation Crossover Operators on the Traveling Salesman Problem", Proceedings of the 2nd International Conference on Genetic Algorithms, 1987, pp. 224-230.

Orvosh, D. and Davis, L., "Using a genetic algorithm to optimize problems with feasibility constraints", Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 548-552, 1994.

Rao, V. V. B., "Most-vital Edge of a Graph with respect to Spanning Trees", IEEE Transactions on Reliability, Vol. 47, No. 3, 1998, pp. 6-7.

Schwefel, H., *Evolution and Optimum Seeking*, wiley, 1995.

Satyanarayana, A., "A Unified Formula for Analysis of Some Network Reliability Problems", IEEE Transactions on Reliability, Vol. R-31, No. 1, 1982, pp. 16-22.

Shao, F. M., and Zhao, L. C., "Topological Optimization of Computer Network Expansion With Reliability Constraint", Computers Math. Applic., Vol. 35, No. 11, 1998, pp. 17-26.

Syswerda, G., "Uniform crossover in genetic algorithms", International Conference on Genetic Algorithms, 1991, pp. 69-76.

Tate, D. M. and Smith, A. E., "A Genetic Approach to the Quadratic Assignment Problem", Computers and Operations Research, Vol 22, No. 1, 1995, pp. 73-83.

Walters, G. A. and Smith., D. K., "Evolutionary Design Algorithm for Optimal Layout of Tree Networks", Engineering Optimization, Vol. 24, 1995, pp. 261-281.

Wilkov, R. S., "Analysis and Design of Reliable Computer Networks", IEEE Transactions on Communications, Vol. COM-20, 1972 Jun, pp. 660-678.

Van Slyke, R. M., and Frank, H., "Network Reliability Analysis-Part I", Networks, Vol. 1, No. 3, 1972, pp. 49-64.

Economic Design of Networks Using Genetic Algorithms

Lee Han-Jin

*Department of Business Administration, Graduate School,
Pukyong National University*

Abstract

Economic network design problems have attracted many companies' attentions in order to share expensive hardware and software resources and provide access to main systems from distant locations.

Generally, the type of networks can be classified into backbone and LAN network. The purpose of the backbone network is to facilitate fast and easy information exchange and resources sharing among all the interconnected departmental points and to provide good connectivity to the outside world. LANs are commonly used as a communication infrastructure that meets the demands of users in a local environment.

Network reliability which is defined as the probability that every pair of nodes can communicate with each other and network design cost are important parameters in

the design of a computer communication network.

These economic network topology design problems are known to be NP-hard combinatorial optimization problems.

Solution approaches for the optimal network design problems considering network reliability can be classified as follows : (1) an enumeration-based approach, (2) a heuristic-based approach, or (3) a genetic algorithm-based approach.

Recently, genetic algorithm-based approaches have been receiving increasing attention as a new solution method for the optimal design of networks considering reliability by many researchers. genetic algorithms are inspired from the idea of evolution. Therefore, it is natural to expect that the adaptation is used not only for finding solutions to a given problem, but also for tuning genetic algorithms to the particular problem such as network designs.

In this study, we have applied the genetic algorithms(GA) for the design problem of communication networks, i.e., backbone network expansion problems, LAN topology design problems. The economic design problem of networks is defined as finding the network minimizing cost subject to reliability constraint or finding the network maximizing reliability subject to cost constraint.

In the economic backbone network expansion problems, the existing network can be expanded with addition of new nodes and multiple choices of link type for each possible link. To efficiently solve the problems, a genetic approach is suggested. The k-nery encoding to present backbone network topology is used as a chromosome for the GA. To exactly calculate the backbone network reliability, we proposed network

reliability calculation method. Numerical experiments are provided to evaluate the proposed backbone network design method using the GA. Simulated annealing(SA) and random greedy search are used to compare the effectiveness of the GA and sensitive analysis is carried out for the implement evaluation of the proposed network design according to several different backbone network reliability or cost constraints.

In economic LAN topology design problems, the design problems include issues such as multiple choices of link type for each possible link, multiple choices of hub type for each hub, and allocation of the users to the hubs. To efficiently solve these problems, a genetic approach is suggested. The k-ary encoding to present LAN topology is used as a chromosome for the GA. To exactly calculate the LAN reliability, we proposed LAN reliability calculation method. Numerical experiments are provided to evaluate the proposed LAN design method using the GA. SA and random greedy search are used to compare the effectiveness of the GA and sensitive analysis is carried out for the implement evaluation of the proposed network design according to several different LAN reliability or cost constraints.

According to the experiments, the proposed approaches improve search performance, because our proposed approaches have the ability to find good solutions or near optimal solutions within short generations.

감사의 글

지난 4년간의 박사과정을 통한 결과를 부족하나마 이 한편의 논문으로 마무리하면서 그 동안 저를 아껴주시고 많은 도움을 주신 분들께 깊은 감사의 마음을 전하고자 합니다.

먼저, 부족한 제가 4년간 연구자의 길로 들어섬을 묵묵히 보아주시고 기다려 주신 부모님께 감사드립니다. 특히, 제가 어려울 때마다 기도로써 세상과 타협함을 붙잡아 주신 어머님께 머리 숙여 감사드립니다.

본 논문이 마무리되기까지 늘 부족한 저에게 연구자의 자세를 일깨워주시고 학문의 시야를 넓혀주신 염창선 교수님의 은혜에 진심으로 감사드립니다.

그리고 심사위원으로써 기꺼이 본 논문을 검토해주신 심사위원들께 감사드립니다. 주심으로써 논문의 미흡한 부분을 채워주시고 지도해주신 어윤양 교수님, 논문의 완성을 위해 좋은 의견을 많이 주신 한재호 교수님, 자칫 간과하기 쉬운 부분에 대해 세심하게 검토해주신 김하균 교수님, 외부 심사위원으로써 어려운 시간을 내셔서 논문을 검토해 주시고 좋은 의견을 주신 동의대학교 조재균 교수님께 감사드립니다. 그리고 논문에 대해 많은 관심과 의견을 주신 성덕현 교수님과 논문을 완성하기까지 관심과 격려를 아끼지 않으신 국제통상학부의 김창완 교수님, 경제학부의 류장수 교수님께도 깊이 감사드립니다.

항상 깊은 관심을 주시고 격려해주신 경영학부 여러 교수님들께도 늘 감사의 마음을 담고 있습니다.

대학원 박사과정을 수학하는 동안 어렵고 힘들 때마다 격려를 아끼지 않았던

경영학부 선·후배 여러분들께도 감사의 마음을 전해드립니다. 박사과정 동안 어려울 때마다 형님으로써 격려해주던 박영식, 이남수 선배, 후배로써 마지막까지 응원해주던 최종현, 신경엽군, 조교로써 많은 도움을 준 김성재군, 멀리 학문의 시야를 넓히고자 미국에서 어렵게 수학하고 있을 신상현, 오동엽, 이주형군과 어학연수를 하고 있는 장인호군, 선배라기 보단 오빠로써 지지해 주던 허혜선양, 이제는 사회인으로써 자기 맡은바 책무를 잘 해내고 있는 이상원, 천정인, 김지원, 서봉원군에게도 깊은 감사의 마음을 전합니다. 이들이 없었다면 자칫 힘들었을 대학원 생활을 무사히 마칠 수 있었을까 하는 생각이 앞섭니다.

마지막으로 연구라는 보잘 것 없는 핑계로 소홀했음에도 불구하고 항상 가까이에서 언제나 용기와 격려를 아끼지 않았던 사랑하는 윤경이와 이 기쁨을 함께 나눌 수 있음에 감사드립니다.

2006年 1月

李 漢 珍