

# 높은 신뢰도의 네트워크 설계를 위한 진화 연산에 기초한 알고리즘

## (An Algorithm based on Evolutionary Computation for a Highly Reliable Network Design)

김 종 율 <sup>†</sup>      이 재 욱 <sup>\*\*</sup>      현 광 남 <sup>\*\*\*</sup>  
 (Jong-Ryul Kim)    (Jae-Uk Lee)      (Mituso Gen)

**요 약** 일반적으로 네트워크 설계 문제는 네트워크의 크기가 늘어남에 따라 지수적으로 복잡도가 증가하여 전통적인 방법으로는 풀이하기 힘든 NP-hard 조합 최적화 문제 중의 하나로 분류될 수 있다. 본 논문에서는 네트워크 신뢰도 제약을 고려하면서 네트워크 구축비용을 효과적으로 최소화하는, 높은 신뢰도의 네트워크 토폴로지 설계 문제를 풀기 위해 스패닝 트리를 효율적으로 표현할 수 있는 Prüfer수(PN) 기반의 진화 연산법과 2-연결성을 고려하는 휴리스틱 방법으로 구성된 두 단계의 효율적인 해법을 제안한다. 즉, 먼저 스패닝 트리를 찾아내기 위해 진화 연산법 중에 보편적으로 널리 알려져 있는 유전자 알고리즘(GA)을 이용하고 그 다음으로 첫 번째 단계에서 발견한 스패닝 트리에 대해 최적의 네트워크 토폴로지를 찾기 위해서 2-연결성을 고려한 휴리스틱 방법을 적용한다. 마지막으로 수치예의 결과를 통해 제안한 해법의 성능에 대해서 살펴보도록 한다.

**키워드** : 진화 연산, 네트워크 신뢰도, 조합 최적화 문제, 통신 네트워크

**Abstract** Generally, the network topology design problem is characterized as a kind of NP-hard combinatorial optimization problem, which is difficult to solve with the classical method because it has exponentially increasing complexity with the augmented network size. In this paper, we propose the efficient approach with two phase that is comprised of evolutionary computation approach based on Prüfer number(PN), which can efficiently represent the spanning tree, and a heuristic method considering 2-connectivity, to solve the highly reliable network topology design problem minimizing the construction cost subject to network reliability: firstly, to find the spanning tree, genetic algorithm that is the most widely known type of evolutionary computation approach, is used; secondly, a heuristic method is employed, in order to search the optimal network topology based on the spanning tree obtained in the first phase, considering 2-connectivity. Lastly, the performance of our approach is provided from the results of numerical examples.

**Key words** : Evolutionary Computation, Network Reliability, Combinatorial Optimization, Communication Network

### 1. 서 론

1980년대 이후로 컴퓨터 네트워크는 폭발적인 성장을 해 왔다. 이러한 컴퓨터 네트워크는 자동 응답 서비스,

항공기 예약, 전자 메일, 전자 게시판 등을 비롯한 인터넷 서비스를 통해 매일의 일상생활에 두루 영향을 끼치고 있다. 정보화 시대에 있는 현재의 일상생활에서 컴퓨터 네트워크 시스템은 더욱더 성장해 나갈 것이고 고비용의 하드웨어/소프트웨어를 공유하고 원거리에서 메인 시스템에 접근을 제공하기 위해 통신 네트워크의 이용은 급속히 증가하게 될 것이다. 이렇게 컴퓨터 통신 네트워크 시스템의 규모가 점차 확대되어 감에 따라 컴퓨터 통신 네트워크를 위한 토폴로지 설계 문제는 네트워크 설계자, 네트워크 분석가와 같은 많은 관련 연구자들에 의해 많은 주목을 받고 있다. 텔레커뮤니케이션, 컴

<sup>†</sup> 비 회 원 : 한국과학기술원 테크노경영연구소 정보통신경영연구실 연수연구원  
 xmaskjr@kgs.m.kaist.ac.kr

<sup>\*\*</sup> 정 회 원 : 동서대학교 인터넷공학부 교수  
 julee@dongseo.ac.kr

<sup>\*\*\*</sup> 비 회 원 : Waseda University Grad. School of Inform., Prod. & Sys. Professor  
 gen@waseda.jp

논문접수 : 2004년 11월 10일  
 심사완료 : 2005년 2월 16일

퓨터 네트워킹, 하수도관 시스템, 기름 및 가스관 시스템 등과 같은 많은 실세계 문제에서, 어떤 제약조건들을 만족하면서 목적함수들을 최적화하는 네트워크 시스템을 설계하는 것은 매우 중대한 문제이다. 특히, 네트워크 신뢰도를 고려하는 네트워크 토폴로지 설계는 보통 NP-hard 문제에 속하며 조합 최적화 문제로 정식화 되고 컴퓨터 통신 산업에서 매우 중요하게 여겨지고 있다.

지난 수십 년간 이러한 네트워크 설계 문제들을 풀기 위한 많은 아이디어와 방법들이 제안되었다. 최근에 컴퓨터 네트워크에 관련된 여러 문제들에 유전자 알고리즘을 비롯한 진화 연산에 기반을 둔 접근법들을 적용하려는 시도가 점점 증가하고 있다. 네트워크 신뢰도를 제약조건으로 고려한 네트워크 최적 설계문제들에 대하여 열거(enumeration)에 기반을 둔 해법, 휴리스틱에 기반을 둔 해법, 유전자 알고리즘에 기초한 해법 등의 해법들이 제안되었다.

열거에 기반을 둔 해법들은 규모가 작은 네트워크 설계 문제에만 적용이 가능하며 최소경로(minpaths) 또는 최소컷(mincuts)과 같은 상태들의 열거를 통하여 해를 발견하는 접근법들이다. 대체로 이러한 방법들은 상태들의 열거 및 제거하는 기술들을 어떤 조합으로 구성하여 이뤄져 있다. 열거에 기반을 둔 해법들은 신뢰도 측도에 대하여 서로 배타적이면서 부분의 합이 전체를 구성하는(MECE - Mutually Exclusive, Collectively Exhaustive) 확률적인 사건들의 집합을 열거하는 방법들이다. Jan, Hwang와 Chen은 최소 네트워크 신뢰도 제약을 가지는 링크 연결 비용을 최소화하는 분기 한정법을 기초한 분해 방법을 이용하는 알고리즘을 제안하였다[1]. Aggarwal과 Rai는 컴퓨터 네트워크의 신뢰도를 평가하기 위해 스페닝 트리에 기반을 둔 방법들을 제안하였고[2] Wilkov는 새로운 신뢰 측도를 이용하여 컴퓨터 네트워크를 설계하는 방법을 제안하였다[3].

타부 탐색법, 시뮬레이티드 어닐링 등과 같은 휴리스틱에 기반을 두는 방법들은 최적성을 보장하지 못하지만 비교적 규모가 큰 네트워크 설계문제들에 적용할 수 있도록 개발되었다. Aggarwal, Chopra와 Bajwa는 전체 네트워크 신뢰도를 고려한 컴퓨터 네트워크 설계 문제들에 탐욕적인 휴리스틱 접근법을 적용하였으며[4] Chopra, Sohi, Tiwari와 Aggarwal은 네트워크 신뢰도를 최대화하는 네트워크 토폴로지의 설계를 위한 탐욕적인 휴리스틱 접근법을 제안하였다[5]. Venetsanopoulos와 Singh는 신뢰도를 제약조건으로 하는 연결비용을 최소화하는 2 단계 휴리스틱을 개발하였고[6] Glover, Lee와 Ryan은 최소 비용의 네트워크 토폴로지를 설계하기 위한 타부 탐색법을 제안하였다[7]. Koh는 타부 탐색법을 신뢰도를 고려한 광케이블 통신 네트워크를 설계하

는 문제에 적용하였다[8]. Atiqullah와 Rao는 통신 네트워크의 신뢰도를 최적화하기 위해 시뮬레이티드 어닐링을 이용한 방법을 제안하였으며[9] Fetterlof 등은 LAN-WAN의 통합 설계를 최적화하는 문제에 시뮬레이티드 어닐링을 적용하였다[10]. Pierre 등은 시뮬레이티드 어닐링을 이용하여 컴퓨터 통신 네트워크의 토폴로지 설계를 위한 알고리즘을 개발하였다[11].

최근에 메타 휴리스틱 방법의 하나로 주목 받고 있는 유전자 알고리즘에 기초한 해법은 신뢰도를 고려한 네트워크의 최적 설계를 위한 해법의 도구로서 주목 받고 있으며 많은 실세계 문제들을 풀기 위해 많이 사용되고 있다[12,13]. Kumar 등은 분산 시스템의 토폴로지를 설계하기 위한 유전자 알고리즘을 제안하였으며[14] Kumar, Pathak와 Gupta는 그래프의 직경, 거리 및 신뢰도를 고려하는 컴퓨터 네트워크의 설계 및 확장을 위한 문제를 풀기 위해 유전자 알고리즘을 개발하였다[15]. Deeter와 Smith는 다른 비용과 신뢰도를 가진 여러 종류의 링크 케이블 및 네트워크 신뢰도를 고려하여 네트워크를 설계하는 문제에 유전자 알고리즘을 적용하였다[16]. Dengiz, Altiparmak와 Smith는 자신들의 유전자 표현을 위해 인접행렬의 상삼각 요소를 이용하는 표현법을 이용하여 네트워크 신뢰도를 고려한 네트워크 최적 설계문제에 유전자 알고리즘을 적용할 수 있도록 시도하였고[17] 네트워크 신뢰도를 고려한 큰 백본 통신 네트워크 설계문제에 최적화를 위해 효과적이고 효율적으로 유전자 알고리즘을 적용할 수 있는 해법을 제공하였다[18].

본 논문에서 네트워크 신뢰도를 고려하여 네트워크 구축비용을 최소화시키는 네트워크 토폴로지 설계 문제에 대한 하나의 해법을 제시한다. 제안 방법은 두 단계로 구성된다. 즉, 첫 번째 단계에서는 스페닝 트리를 찾기 위해 진화 연산법들 중에 잘 알려진 GA를 이용하고 두 번째 단계에서는 2-연결성을 이용하여 탐욕적으로 최적의 네트워크 토폴로지를 찾아내도록 한다. 또한, 스페닝 트리를 나타내기 위하여 GA의 표현방법으로 PN 인코딩 방법을 이용한다[12]. PN 인코딩 방법을 이용하면  $n$ 개의 정점을 가지는 네트워크 설계문제에 대하여  $n-2$ 의 메모리만을 필요로 한다. 네트워크 설계 문제의 목적함수는 네트워크 구축비용을 최소화하는 것이고 제약 조건으로는 구축되는 네트워크의 신뢰도가 기준이 되는 신뢰도 이상이 되어야 하는 것이다. 기존의 유전자 알고리즘에 기반을 둔 해법들은 단순히 신뢰도를 고려한 네트워크 설계 문제의 해법으로만 이용되어 해를 구해낸다. 본 논문에서는 유전자 알고리즘을 이용하여 최적해의 뼈대를 구성할 가능성이 높은 스페닝 트리를 탐색해 내고 그러한 트리를 바탕으로 최적해를 탐색해 내는 두 단계로 구성되어 있다. 또한, 본 논문에서는 후보

네트워크 토폴로지(x)에 대하여 네트워크의 신뢰도를 계산하기 위해 백트래킹 알고리즘을 이용한다[19]. 이 백트래킹 알고리즘은 시스템의 신뢰도(1-R(x))를 정확하게 계산하기 위해 사용된다. 논문의 말미에 제안한 방법을 이용하여 얻어진 네트워크 토폴로지의 평가를 위해 수치 실험한 결과를 다른 접근 방법과의 비교를 나타내었다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2절에서는 신뢰도를 고려한 네트워크 토폴로지 설계 문제의 수학적 모델과 문제의 정의를 나타낸다. 3절에서는 제안한 접근법에 대하여 설명한다. 4절은 수치 예와 그 결과를 나타내고 제안한 해법을 일련의 테스트 문제들에 적용하는 것에 대한 토의를 한다.

## 2. 높은 신뢰도의 네트워크 토폴로지 설계 문제

통신 네트워크는 그림 1에서 살펴 볼 수 있듯이 컴퓨터 사이트를 의미하는 정점(노드) 집합 N과 통신 케이블을 의미하는 간선(링크) 집합 E로 구성되는 무방향 그래프 G=(N, E)로 나타낼 수 있다. 임의의 그래프 G에서 모든 정점들을 연결 하는 스패닝 트리를 구성하는 데에 n개의 정점에 대하여 n-1개의 간선이 필요하며 가능한 간선의 수는 (n(n-1))/2개이다. 네트워크 신뢰도를 고려하는 최적의 네트워크 설계 문제를 정의하기 위해 다음의 기호들을 정의한다.

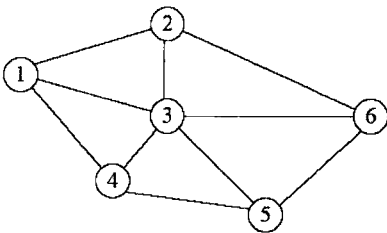


그림 1 네트워크 구성

- n: 정점의 개수
- $x_{ij}(\in \{0, 1\})$ : 정점 i와 정점 j사이의 간선을 나타내는 의사 결정 변수
- $\mathbf{x}(\{x_{12}, x_{13}, \dots, x_{n-1,n}\})$ : 네트워크 토폴로지
- $\mathbf{x}^*$ : 찾아낸 최선해(best solution)
- p: 간선의 신뢰도, q: 간선의 불(不) 신뢰도(p+q=1)
- R(x): 네트워크 토폴로지 x의 신뢰도
- $R_{min}$ : 네트워크 신뢰도 제약
- $c_{ij}$ : 정점 i와 정점 j사이의 간선 비용
- $\delta$ : 만일  $R(x) < R_{min}$ 이면 1의 값을, 아니면 0의 값을 가지는 변수
- $E'$ : 통신이 가능한 간선의 집합( $E' \subseteq E$ )

$\Omega$ : 통신이 가능한 간선의 집합( $E'$ )으로 구성되는 상태들 또한, 다음과 같은 가정을 한다. 각 정점의 위치는 주어지고 정점들은 완전히 신뢰할 수 있으며 각  $c_{ij}$ 와 p는 고정되어 있고 미리 정해져 있으며 각 간선은 쌍방향이고 네트워크에서 간선의 중복 사용은 없다. 간선들은 통신이 가능하거나 불가능한 상태를 가지며 간선들의 통신 불가능한 상태는 상호 독립적이고, 간선의 보수(repair)는 없다.

본 논문에서 다루는 높은 신뢰도를 가지는 네트워크 토폴로지 설계 문제는 다음과 같이 정식화된다. 정식화되는 높은 신뢰도를 가지는 네트워크 토폴로지 설계 문제의 목적함수는 네트워크 구축비용을 최소화하는 것이고 구축되는 네트워크의 신뢰도가 어느 기준( $R_{min}$ ) 이상 되어야 하는 제약조건을 가진다.

$$\min Z(\mathbf{x}) = \sum_{i=1}^n \sum_{j=i+1}^n c_{ij} \cdot x_{ij}$$

s. t.  $R(\mathbf{x}) \geq R_{min}$

임의의 시각에 그래프 G에서 일부분의 간선들만 통신이 가능하기 때문에 그래프 G의 통신이 가능한 상태는 그래프 G의 부분 그래프  $G'(\subseteq (N, E'))$ 로 나타낼 수 있다. 통신이 가능한 상태 그래프 G'에 관한 네트워크 신뢰도는 다음과 같이 나타낼 수 있다.

$$\sum_{\mathbf{x}} \left[ \prod_{e \in E} p_e \right] \left[ \prod_{e \in (E \setminus E')} q_e \right]$$

이상에서 나타낸 네트워크 설계 문제는 일반적으로 다음 두 종류의 신뢰도 중 하나를 고려하여 최적화하는 문제로 정식화된다. 본 논문에서는 첫 번째로 설명되는 전체 네트워크 신뢰도를 고려하는 최적화 문제를 정식화하였다.

**전체 터미널(all-terminal or overall) 네트워크 신뢰도(또는 전체 네트워크 신뢰도):** 모든 정점이 복수개로 분할되지 않고 하나의 네트워크를 구성하고 있을 확률, 즉 어떤 시각에 네트워크의 모든 정점이 상호 연결되어 있어서 서로 통신 가능할 확률을 의미한다.

**소스-싱크(source-sink) 네트워크 신뢰도:** 어떤 시각에 데이터를 전송하는 소스가 데이터의 목적지(싱크)와 연결되어 있어서 통신이 가능할 확률, 즉 어떤 시각에 소스와 싱크만 어떤 경로를 통해서든 연결되어 통신이 가능할 확률을 의미한다.

## 3. 제안 방법: 두 단계 접근 방법

### 3.1 유전자 알고리즘: 첫 번째 단계

실세계 문제들 중에서 특히 생산 시스템, 네트워크 시스템에 관련된 많은 문제들은 본래 매우 복잡하고 고전적인 최적화 기술들로는 풀기 매우 어렵다. 1960년대 이후부터 그러한 어려운 최적화 문제들을 풀기위해 자연

생물들을 모방하는 것에 차츰 관심들을 가져왔었고 그러한 관심들을 통해 자연의 시스템을 모방하여 어려운 실세계 문제들에 적용하였을 경우 고전적인 최적화 방법들보다 성능이 좋은 진화연산법이라고 불리는 확률론적인 최적화 기술들이 제안되었다. 이와 관련된 연구들은 다음의 4가지 주요한 영역으로 나뉘어져서 진행되고 있다: 유전자 알고리즘(genetic algorithms), 진화 프로그래밍(evolutionary programming), 진화 전략(evolution strategies), 유전자 프로그래밍(genetic programming). 이 중에서 유전자 알고리즘은 오늘날의 진화 연산법으로 가장 널리 알려진 형태이다[12,13].

최근에 유전자 알고리즘은 복잡한 문제들을 위한 최적화 기법으로 그 잠재력에 대하여 상당한 주목을 받고 있으며 스케줄링, 시퀀싱, 신뢰성 설계, 운송수단의 라우팅 및 스케줄링, 시설 배치 설계 문제, 네트워크 설계 및 라우팅 문제 등과 같은 실세계의 많은 공학적 문제에 성공적으로 적용되고 있다[12,13].

유전자 알고리즘은 두드러지는 특징 중에 하나는 여러 개의 해들을 이용하여 동시에 집단적으로 탐색해 나간다는 것이다. 이때 해들은 유전자로 표현되는데 이렇게 실제 문제의 후보해를 유전자 알고리즘의 유전자로 표현하는 방법들을 유전자 표현법이라고 하고 이 유전자 표현법에 따라 일련의 유전자 알고리즘내의 유전과정에서 이용되는 교차와 돌연 변이와 같은 유전 연산자들에 많은 영향을 끼치므로 잘 설계되어야 하고 유전자 표현법에 문제에 대한 다양한 정보를 잘 표현할 수 있도록 설계되어야 한다. 유전자 표현법이 정해지면 해를 어떻게 평가할 것인가의 기준을 정하는 것이 중요한데 대체로 최적화 문제들에서는 목적 함수치를 이용하여 평가기준을 정한다. 그리고 평가된 해들 중에서 좋은 해들을 선택하여 다음 세대로 전달하여 계속 탐색을 진행하기 위해 필요한 선택법을 어떻게 결정하느냐를 정하여야 한다. 이러한 유전자 알고리즘의 구성요소들이 결정되면 다음과 같은 순서로 유전자 알고리즘을 진행시킬 수가 있다.

**begin**

$t \leftarrow 0$

$p(t)$ 의 초기화(초기 모집단 생성)

$p(t)$ 의 적응도 평가

**while** (종료조건이 만족되지 않으면) **do**

**begin**

$t \leftarrow t+1$

$p(t-1)$ 로부터  $p(t)$ 를 선택

$p(t)$ 의 유전연산(교차와 돌연변이)

$p(t)$ 의 적응도 평가

**end**

**end**

여기서  $p(t)$ 는  $t$ 세대의 집단, 즉 유전자 표현법을 통해 염색체로 표현된 해집합을 의미하고 일반적으로 종료조건으로 최대세대수를 지정하여 최대 세대수 동안만 while문을 수행한다.

본 논문에서는 이러한 유전자 알고리즘을 신뢰도를 고려한 네트워크 설계 문제를 풀기 위한 방법의 첫 번째 단계의 최적화 기법으로 이용한다.

### 3.1.1 유전자 표현

유전자 표현은 유전자 탐색 공간에서 문제의 후보 해를 표현하기 위한 자료구조를 의미한다. 보통 다른 종류의 문제들은 다른 자료구조 또는 유전자 표현을 사용하여야 유전자 알고리즘의 탐색 능력을 훼손하지 않으면서 준 최적해 또는 최적해를 빠르게 탐색해 나갈 수 있다. 본 논문에서 제안하는 해법의 첫 번째 단계인 유전자 알고리즘이 주어진 네트워크 설계 문제 환경에서 스페닝 트리를 찾기 위해서 이용되므로 유전자 표현으로 스페닝 트리를 나타내는 자료구조를 잘 고안해야 한다.

그래프 이론에서 오래된 정리들 중 하나는  $k$ 개의 정점으로 이뤄진 완전 그래프에서  $k^{k-2}$ 개의 구별되는 트리들이 존재한다는 Cayley 정리이며 Prüfer는 그러한 트리들과  $k$ 개의 숫자들로 이뤄지는  $k-2$ 길이의 문자열들의 집합사이의 일대일 대응관계를 확증하였다[12]. 이것은  $k$ 개의 다른 숫자들로 구성되는  $k-2$ 개의 숫자 조합의 열로 트리를 표현할 수 있다는 것을 의미한다. 왜냐하면 임의의 트리에 대해 적어도 2개의 잎(leaf) 정점이 존재하기 때문이다[20]. 이러한 숫자 조합의 열을 Prüfer 수(PN)이라고 한다[12,13]. 이러한 아이디어를 기초한 다음과 같은 프로시저를 통해 트리에서 PN을 구할 수 있다:

#### Procedure: Encoding of Prüfer Number

**Step 1.** 정점의 레이블이 있는 트리(labeled tree)에서 가장 적은 숫자로 레이블된 잎 정점을  $i$ 정점이라고 하고 정점  $i$ 에 인접한 정점이  $j$ 라고 하면 PN의 오른쪽에 숫자  $j$ 를 추가한다.

**Step 2.** 정점  $i$ 와 간선  $(i, j)$ 을 트리에서 제거한다.

**Step 3.** 위의 두 Step들을 하나의 간선이 남을 때까지 반복한다. 반복 후 얻어지는 숫자 조합의 열은 1에서  $k$ 사이의  $k-2$ 개의 숫자들로 구성되는 PN이다.

또한, 다음의 프로시저를 통해서 PN으로부터 트리를 생성할 수 있다:

#### Procedure: Decoding of Prüfer Number

**Step 1.**  $P$ 를 PN이라고 하고  $P'$ 를  $P$ 에 속하지 않는 정점들의 집합이라고 둔다.

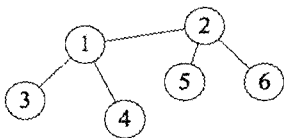
**Step 2.**  $P'$ 에서 가장 작은 숫자(가장 작은 숫자로 레이블된 정점)를  $i$ 라고 하고  $P$ 의 가장 왼쪽에 있

는 숫자를  $j$ 라고 하자. 그러면  $i$ 로부터  $j$ 에로의 간선  $(i, j)$ 을 트리에 추가한다. 추가한 후  $P'$ 로부터는  $i$ 를,  $P$ 로부터는  $j$ 를 제거한다. 만약  $P$ 로부터는 삭제되는  $j$ 가 삭제된 후  $P$ 에  $j$ 가 더 이상 나타나지 않으면  $j$ 를  $P'$ 에 입력한다.

**Step 3.**  $P$ 에 숫자가 남지 않을 때까지 Step 2를 반복한다.  $P$ 에 숫자가 남지 않으면  $P'$ 에 숫자 두개, 즉  $r$ 과  $s$ 가 남는다. 그러면 간선  $(r, s)$ 을 트리에 추가하면 최종적인 트리가 생성된다.

그림 2에 6개의 정점으로 구성되는 스패닝 트리를 PN(1 1 2 2)으로 표현하는 예를 나타내었다. 상위의 프로시저에 의해 PN이 생성되는 과정을 묘사하면 다음과 같다: 먼저 가장 낮은 라벨을 가진 일 정점을 찾는다. 이 경우에는 정점 3에 해당한다. 트리에서 정점 3은 정점 1에 연결되어 있으므로 PN의 첫 숫자로 1을 할당하고 정점 3과 간선(3, 1)을 트리에서 제거한다. 이런 과정을 간선(2, 6)이 남을 때까지 반복하면 그림 1의 트리에 대한 PN이 생성된다.

반면에 PN으로부터 트리를 생성하는 디코딩 과정을 살펴보면,  $P=(1\ 1\ 2\ 2)$ 에 대해 정점 3, 4, 5, 6이 PN에 나타나지 않는 정점 집합을  $P'=\{3, 4, 5, 6\}$ 으로 나타낼 수 있다. 정점 3이 제일 낮은 라벨을 가진 정점이고  $P$ 의 가장 왼쪽에 있는 숫자가 1이므로 정점 3과 정점 1을 연결한 후  $P'$ 에서 정점 3을 PN에서 정점 1을 제거한다. 그 다음으로 정점 4와  $P$ 의 가장 왼쪽 숫자 1을 고려하여 정점 4와 정점 1을 연결한 후 각각 제거를 한다. 이때  $P$ 에서 숫자 1을 제거할 때 이제 더 이상 숫자 1이 나타나지 않기에 숫자 1을 집합  $P'$ 에 포함시킨다. 그럼,  $P'$ 는  $\{1, 5, 6\}$ 이 된다. 이러한 과정을  $P$ 가 비어지고  $P'$ 에 두개의 숫자가 남을 때까지 반복하고  $P'$ 에 남은 두 숫자의 정점을 서로 연결하면 그림 2의 트리가 생성된다.



1 1 2 2 Prüfer number

그림 2 트리와 그에 대한 Prüfer수(PN)

PN을 이용한 인코딩 방법은 가능한 모든 스패닝 트리를 표현할 수 있을 뿐만 아니라 PN에 정점이  $d-1$ 번 나타나면 그 정점의 차수는  $d$ 가 된다는 정보도 나타낼 수 있다. 즉, PN에 정점의 라벨이  $d-1$ 번 나타나면 트리

에서 그 정점은  $d$ 개의 연결을 가지게 된다. PN은 운송 문제, 최소 스패닝 트리 문제 등과 같은 연구 분야에서 스패닝 트리를 표현하기 위한 방법으로 적합하다는 연구들의 보고가 있다[12].

그림 2로부터 PN은 일 정점이 아닌 정점들의 숫자로 구성된다는 것을 알 수 있다. 본 논문에서는  $n$ 개의 정점으로 구성된 네트워크를 설계하는 문제에 대하여 임의로  $n-2$ 개의 숫자를  $[1, n]$ 의 범위에서 생성하여 염색체들을 초기화한다. 염색체에는 일 정점이 아닌 정점들만 나타나고 일 정점은 염색체에 표시되지 않고 다만 집합  $P'$ 의 원소로 사용된다.

### 3.1.2 평가법

높은 신뢰도를 가지는 네트워크의 토폴로지를 설계하는 문제를 위한 평가함수  $eval(v_k)$ 의 값을 평가하기 위해 우선 인코딩화 된 염색체  $v_k$ 를 인접행렬  $x_k$ 로 디코딩한다. 그런 후 목적함수 값을 구해 내고 평가함수  $eval(v_k)$ 을 다음과 같이 계산한다.

여기서  $BIG$ 는 매우 큰 수를 나타내며  $\delta$ 는 만일  $R(x_k) < R_{min}$ 이면 0이고 그렇지 않으면 1로 설정된다.

$$eval(v_k) = \begin{cases} \frac{1}{Z(x_k)}, & \text{만일 } \delta = 1; \\ 1/BIG, & \text{그렇지 않으면} \end{cases}$$

그리고 각 세대마다 최고 큰 적합도를 가지는 최선의 염색체  $v^*$ 는 다음의 수식을 통해 유지해 나간다.

$$v^* = \text{argmax}(eval(v_k) \mid k = 1, 2, \dots, \text{pop\_size})$$

여기에서 argmax는 최대값을 가지는 인수(argument of the maximum)를 의미한다.

### 3.1.3 네트워크 신뢰도 계산

네트워크 설계에 관련한 활발한 연구 분야 중에 하나는 어떻게 네트워크 신뢰도를 계산 또는 추정할 것인가에 관련한 분야이다. 이러한 연구 분야에 대한 해법으로는 분석적 방법을 통한 정확한 계산, 몬테카를로 모의실험법을 이용한 추정의 2가지 부류로 나누어서 설명할 수 있다.

전체 네트워크 신뢰도 계산을 위한 모든 분석적 방법들은 최악의 경우 네트워크 크기가 증가함에 따라 계산 시간도 지수적으로 증가하는 경향이 있다. 따라서 복잡한 네트워크 설계에 대해서는 추천하지 못할 방법이다. Ball과 Van Slyke는 무방향 그래프의 전반적인 신뢰도를 계산하기 위해 백트래킹을 이용하여 네트워크 단절(network-cut)을 형성하는 커트집합(cutset)들을 열거하는 방법을 제안하였고[19] Hanzhong와 Dongkui는 커트 방법을 이용하여 복잡한 네트워크 신뢰도를 계산하

는 새로운 해법을 제안하였다[21]. Mandalsis와 Kontolen은 계층적 라우팅 전략을 이용하는 컴퓨터 네트워크의 전체 신뢰도를 계산하기 위한 방법을 연구하였고 [22] Liu 등은 네트워크 전체의 신뢰도 계산을 위한 해법을 제안하였다[23].

네트워크가 완전 연결된 것에 가까운 상태일 경우 일반적으로 효율성을 잃어버리는 경향이 있기 때문에 전체 네트워크 신뢰도 문제에 모의실험 방법을 효율적으로 적용하기는 힘들지만 모의실험 방법은 네트워크의 규모가 선형적으로 증가하는 경우에는 계산 시간이 다소 더 빠른 경향을 나타내기 때문에 큰 규모의 네트워크에 적합하다는 보고가 있다. Cancela와 Khadiri는 통신 네트워크의 신뢰도를 추정하기 위해 몬테카를로 방법을 이용한 재귀적 분산 제거 알고리즘을 이용하였다 [24]. Fishman은 네트워크 신뢰도 추정을 위한 몬테카를로 샘플링 계획에 대한 연구를 하였고 송신정점과 수신정점이 연결되어지는 확률을 추정하기 위해 4가지 몬테카를로 방법들에 대한 분석을 하였다[25]. Kamat와 Riley는 이벤트에 기반을 둔 몬테카를로 모의실험을 이용하는 신뢰도 결정 방법을 연구하였다[26]. Kumamoto 등은 몬테카를로 방법을 이용하여 시스템의 신뢰도를 효율적인 평가 방법을 제안하였다[27]. Yeh 등은 네트워크 신뢰도를 추정하기 위해서 새로운 몬테카를로 방법을 제안하였다[28].

본 논문에서는 네트워크의 신뢰도( $1-R(x)$ )를 정확하게 계산하기 위해서 백트래킹 알고리즘을 이용한다. 백트래킹 알고리즘을 요약하여 나타내면 다음과 같다:

#### Step 0. 초기화

모든 간선들을 프리(free)한 상태로 마크하고 빈 스택  $S$ 를 생성한다.

#### Step 1. 수정된 커트집합을 생성

**Step 1-1.** 모든 비동작(inoperative) 간선들과 함께 네트워크 단절을 형성하는 프리한 간선 집합을 찾아낸다.

**Step 1-2.** Step 1-1에서 찾아낸 모든 간선들을 비동작으로 표시하고 그것들을 스택에 입력한다(Push).

**Step 1-3.** 스택에 있는 간선들의 집합이 수정된 커트집합이다. 스택의 확률을 계산하여 누적 합을 구한다.

#### Step 2. 백트랙(Backtrack)

**Step 2-1.** 만일 스택이 비어 있으면 Step 3로 이동하고 그렇지 않으면 Step 2-2로 이동한다.

**Step 2-2.** 스택의 정상(top)에 있는 간선을 뽑아낸다(pop).

**Step 2-3.** 만일 뽑아낸 간선이 비동작이고 뽑아낸 간선을 동작(operative) 상태로 만들면 동작 상태의

간선들로 이뤄지는 스페닝 트리가 존재한다고 하면 뽑아낸 간선을 프리한 상태로 마크하고 Step 2-1로 이동한다.

**Step 2-4.** 만일 뽑아낸 간선이 비동작 상태이고 Step 2-3에서 조사되는 조건이 만족하지 않으면 뽑아낸 간선을 동작 상태로 만들고 뽑아낸 간선을 스택에 다시 입력(push)한 후 Step 1로 이동한다.

**Step 2-5.** 만일 뽑아낸 간선이 동작 상태이면 뽑아낸 간선을 프리한 상태로 마크하고 Step 2-1로 이동한다.

**Step 3.** 네트워크의 신뢰도( $1-R(x)$ )를 반환하고 계산 과정을 끝낸다.

여기서 스택에 있는 모든 간선들의 확률은 모든 비동작 상태의 간선들의 실패확률  $q_e$ 의 곱과 모든 동작 상태의 간선들의 성공 확률  $p_e(1 - q_e)$ 의 곱을 곱한 것을 의미한다.

### 3.1.4 선택법

GA에서 선택연산은 매우 중요한 역할을 한다. 교차 및 돌연변이 같은 유전자 연산들은 해공간에서의 탐색을 위한 탐험(exploration) 연산이라고 한다면 선택연산은 진화탐색 과정을 인도하는 GA를 위한 개척(exploitation) 연산이라 할 수 있다. 본 논문에서 사용된 선택연산은 자유로이 해 공간을 탐색하도록 하기 위해 룰렛 휠법과 엘리트리스트(elitist) 방법을 혼합한 연산이다. 적합도 비례 방법 중에 하나인 룰렛휠 선택 연산은 임의로 새로운 세대를 생성하기 위하여 이용되고 엘리트리스트 방법은 룰렛휠 연산의 오류를 극복하고 가장 좋은 적합도를 가진 염색체를 유지하기 위해 사용된다. 본 논문에서 이용하는 선택법은 다음과 같은 프로시저로 나타낼 수 있다:

**Step 1.** 각 염색체  $v_k$ 에 대한 누적 확률  $a_k$ 을 계산한다( $k = 1, 2, \dots, chr\_size$ ). 여기서  $chr\_size$ 는 한 세대의 집단의 크기인  $pop\_size$ 에 교차와 돌연변이 연산을 통해 얻은 염색체의 개수를 더한 값이다.

**Step 2.** 영역  $[0, 1]$ 에서 임의의 실수(real number)  $r$ 를 생성한다.

**Step 3.** 만일  $r \leq a_1$ 이면 첫 번째 염색체  $v_1$ 을 선택하고 그렇지 않으면  $a_{p-1} < r \leq a_p$ 인 경우에 염색체  $v_p$ 를 선택한다.

**Step 4.** Step 2와 3을  $pop\_size$ 번 반복하여  $pop\_size$ 개의 염색체를 구한다.

**Step 5.** 만일 구해진  $pop\_size$ 개의 염색체내에 가장 좋은 적합도를 가진 염색체가 선택되어 있지 않으면 구해진  $pop\_size$ 개의 염색체 중에 임의의 하나와 바꾸어 다음 세대에 현재 세대의 가장 좋은 적

합도를 가진 염색체가 유지되도록 한다.

3.1.5 교차

그림 3에서와 같이 균등교차(uniform crossover)이라고도 불리는 다점 교차(multi-point crossover)를 이용한다. 이런 형태의 교차 연산은 임의로 두 부모 염색체를 선택한 후 두 부모 염색체 중에 임의의 위치에 있는 유전자를 자식 염색체의 유전자로 복사하여 자식 염색체를 생성해 나간다. 이러한 균등교차는 우선 교차를 위해 임의로 0 또는 1의 마스크 열을 염색체의 길이만큼 생성한 후 한쪽 부모 염색체로(Parent1)부터 마스크가 0인 유전인자를, 다른 쪽 부모 염색체(Parent2)로부터는 마스크가 1인 유전인자를 복사해 하나의 자식 염색체(Offspring1)를 생성하고 다른 자식 염색체(Offspring2)는 앞서와 반대로 하여 생성하는 유전 연산자이다.

mask	0	1	0	1	1	0
Parent 1	2	4	3	2	1	4
Parent 2	3	2	1	1	4	2
Offspring 1	2	2	3	1	4	4
Offspring 2	3	4	1	2	1	2

그림 3 균등교차 연산자

3.1.6 돌연변이

본 논문에서의 돌연변이 연산은 그림 4에서 나타낸 것과 같이 하나의 염색체에서 임의로 선택된 두개의 위치에 있는 유전자 들을 교환하는 교환 돌연변이를 사용한다. 즉, 교환 돌연변이는 먼저 염색체 길이의 범위 내에서 임의의 두 위치를 선택한 후, 선택되지 않은 나머지 위치의 유전자들은 부모 염색체로부터 그대로 복사하고 선택된 두 위치의 서로 위치를 바꾸어 복사하여 자식 염색체를 생성하도록 하는 연산자이다.

	selected positions					
Parent	2	1	2	3	3	1
Offspring	2	3	2	3	1	1

그림 4 교환 돌연변이 연산자

3.2 휴리스틱 방법: 두 번째 단계

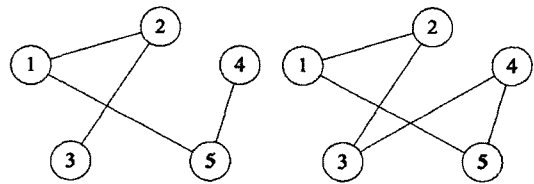
본 논문에서 제안한 방법 중 두 번째 단계에서는 첫 번째 단계에서 얻어진 최선의 염색체에 대하여 우선 디코딩을 한다. 그런 후 각 정점의 차수를 계산하여 2-연

결성을 만족하는지 아닌지를 조사한다(만일 하나 또는 그 이상의 정점의 차수가 2보다 작으면 2-연결성은 만족되지 않는다). 첫 번째 단계에서 찾아낸 최선의 염색체는 스패닝 트리이므로 2-연결성 조사에서 만족하지 않을 것이다. 2-연결성을 만족하지 않을 경우 다음의 원칙에 따라 2-연결성을 만족하도록 탐욕적으로 인접행렬을 수정한다. 첫 번째 원칙은 정점의 차수가 2보다 크면 정점에 연결된 간선들 중에 가장 연결 비용이 높은 간선을 제거하여 정점의 차수가 2에 근접하도록 하는 것이고 두 번째 원칙은 정점의 차수가 2보다 적은 경우 그 정점에 연결 가능한 간선들 중에 연결 비용이 제일 작은 간선을 연결하여 정점의 차수가 2가 되도록 하는 것이다.

앞에서 나타낸 방법을 예를 들어 설명하면 다음과 같다. 5개의 정점을 가지고 있고 연결 간선 비용이 다음과 같은 네트워크를 생각하자.

$$[c_{ij}] = \begin{bmatrix} - & 32 & 54 & 62 & 25 \\ & - & 34 & 58 & 45 \\ & & - & 36 & 52 \\ & & & - & 29 \\ & & & & - \end{bmatrix}$$

그림 5에서 정점 3과 정점 4가 2-연결성을 만족하지 않기 때문에 연결성을 수정해야 한다. 우선 정점 3을 고려하면 정점 3에 연결된 가장 적은 비용의 간선을 찾는다. 즉, 정점 3에 연결된 간선들 (1,3), (3,4), (3,5) 중에 간선 (3,4)이 제일 적은 비용을 가지고 있으므로, 간선 (3,4)을 네트워크 토폴로지에 추가를 하면 그림 5(b)와 같이 수정된 토폴로지를 얻을 수 있다.



(a) 스패닝 트리 (b) 수정된 토폴로지

그림 5 휴리스틱 방법으로 수정된 토폴로지

4. 수치 실험

수치 실험한 결과는 표 1에 요약해 놓았다. 수치 실험한 예제들은 완전 연결된 네트워크이고 정점의 개수는 5에서 25로 설정하였으며 간선의 개수는 10에서 300으로 설정하였다. 또한, 간선들의 연결 비용은 [1,100]의 구간에서 임의로 생성하였다. 다만, 예제 3,4,5의 연결 간선 비용은 [75,250]의 구간에서 임의로 생성하였다. 각 문제에 있어서 수치 실험은 10씩 실행하였다. 제안한 결

표 1 수치 실험 결과

Problem	n	N <sub>e</sub>	p <sub>e</sub>	R <sub>min</sub>	B&B	GA				Proposed Method			
					Obtained Cost	Best Cost	Worst Cost	Mean Cost	STD	Best Cost	Worst Cost	Mean Cost	STD
1	5	10	0.80	0.90	255	201	252	215.2	19.2	156	156	156.0	0
2	5	10	0.90	0.95	201	156	218	189.0	23.4	156	156	156.0	0
3	7	21	0.90	0.90	720	630	800	703.0	51.9	630	755	680.5	46.1
4	7	21	0.90	0.95	845	630	815	702.0	63.1	630	655	646.5	8.8
5	7	21	0.95	0.95	630	670	775	708.5	36.3	630	680	652.0	12.1
6	8	28	0.90	0.90	208	173	231	194.3	17.3	151	186	176.6	13.7
7	8	28	0.90	0.95	247	173	243	192.1	20.8	179	184	181.0	2.6
8	8	28	0.95	0.95	179	179	212	192.4	12.1	151	184	177.4	9.5
9	9	36	0.90	0.90	239	209	324	253.3	38.5	207	247	237.3	12.6
10	9	36	0.90	0.95	286	211	291	255.7	27.1	209	248	233.7	11.2
11	9	36	0.95	0.95	209	209	276	249.0	23.5	207	261	233.0	18.4
12	10	45	0.90	0.90	154	144	217	169.8	25.8	144	192	163.2	15.7
13	10	45	0.90	0.95	197	144	239	166.7	35.4	146	176	159.0	8.7
14	10	45	0.95	0.95	136	144	248	188.1	39.0	136	176	148.0	12.4
15	15	105	0.90	0.95	--	--	--	--	--	183	266	218.3	26.1
16	20	190	0.95	0.95	--	--	--	--	--	157	281	201.7	43.2
17	25	300	0.95	0.90	--	--	--	--	--	308	567	453.3	80.1

N<sub>e</sub>: 간선의 수

GA: Dengiz 등에 의해 제안된 GA 방법[17]

--: 해를 찾을 수 없었음.

B&B: Jan 등에 의해 제안된 분기 한정법[1]

STD: 표준편차

과와 함께 Jan 등에 의해 제안된 분기 한정법(Branch-and-Bound: B&B)[1]과 Dengiz 등에 의한 GA 방법 [17]를 이용한 결과도 표 1에 표시하였다. 예제 15, 16, 17과 같이 규모가 큰 문제에 있어서 문제의 크기가 커짐에 따라 계산량이 매우 많이 늘어나서 Jan 등이 제안한 분기 한정법과 Dengiz 등이 제안한 GA 방법은 실제로 결과를 얻을 수가 없었다. 그리고 표 1에서 살펴 볼 수 있듯이 본 논문에서 제안한 방법이 다른 두 방법보다 그 결과가 좋다는 것을 알 수 있다. Dengiz 등이 제안한 GA 방법은 계산할 때 필요한 메모리양이 많아서 예제 15, 16, 17과 같이 규모가 큰 문제에는 적용하기 힘들다는 것을 알 수 있다. 또한, 수치 실험의 결과에 대한 표준편차를 살펴보면 제안하는 알고리즘이 Dengiz 등에 의한 GA 방법보다 적은 표준편차를 나타내기 때문에 더 강건하다(robust)라고 볼 수 있다.

그리고 예제 3,4,5에 대한 네트워크의 간선 비용은 다음과 같다:

$$[c_{ij}] = \begin{bmatrix} - & 125 & 150 & 125 & 150 & 150 & 130 \\ & - & 75 & 100 & 150 & 200 & 250 \\ & & - & 75 & 90 & 250 & 200 \\ & & & - & 75 & 100 & 150 \\ & & & & - & 75 & 100 \\ & & & & & - & 75 \\ & & & & & & - \end{bmatrix}$$

특히, 예제 5에 대해 제안하는 해법을 통해 구해진 결과 네트워크 토폴로지와 [8]에서 제안된 GA 방법을 이용하여 구한 결과 토폴로지를 다음의 인접행렬들로 나타낼 수 있다. 제안하는 해법을 이용하여 얻은 최선의

네트워크 토폴로지의 인접행렬은 다음과 같고 그에 대한 연결 비용은 630(=125+130+5×75)이다.

$$[x_{ij}] = \begin{bmatrix} - & 1 & 0 & 0 & 0 & 0 & 1 \\ & - & 1 & 0 & 0 & 0 & 0 \\ & & - & 1 & 0 & 0 & 0 \\ & & & - & 1 & 0 & 0 \\ & & & & - & 1 & 0 \\ & & & & & - & 1 \\ & & & & & & - \end{bmatrix}$$

또한, [8]에서 제안된 GA 방법을 이용하여 구한 최선의 결과 토폴로지의 인접행렬은 다음과 같고 그 연결 비용은 670(=125+130+100+90+3×75)이다.

$$[x_{ij}] = \begin{bmatrix} - & 0 & 0 & 1 & 0 & 0 & 1 \\ & - & 1 & 1 & 0 & 0 & 0 \\ & & - & 0 & 1 & 0 & 0 \\ & & & - & 0 & 0 & 0 \\ & & & & - & 1 & 0 \\ & & & & & - & 1 \\ & & & & & & - \end{bmatrix}$$

표 2는 한번 실행(pop\_size×max\_gen)동안에 Dengiz 등이 제안한 GA 방법 및 본 논문에서 채택하고 있는 PN을 이용하는 GA 방법에 의해 실제로 탐색해야 하는 탐색 공간과 실제 해공간의 크기를 비교하고 두 방법을 이용해서 얻은 네트워크 신뢰도의 평균을 비교 정리한 것이다. 여기서, 집단의 개수 pop\_size는 20이고 max\_gen은 최대 세대수이고 예제 문제의 크기에 따라 30-20000의 범위에서 설정하였다. 또한, 교차 비율 p<sub>c</sub>는 0.9로 돌연변이 비율 p<sub>m</sub>은 0.1로 설정하였다. 표 2에서 살펴 볼 수 있듯이 제안한 해법은 Dengiz 등에 의해 제안된 GA 방법과 같은 탐색 노력으로 보다 평균적으로



표 2 탐색 공간에 대한 비교

Problem	Search Space	Solutions Searched	Fraction Searched	$R_{min}$	GA $R(x)$ (Mean)	Proposed Method $R(x)$ (Mean)
1	1.02 E03	6.00 E02	5.86 E-01	0.90	0.9653	0.9379
2	1.02 E03	6.00 E02	5.86 E-01	0.95	0.9884	0.9877
3	2.10 E06	1.50 E04	7.14 E-03	0.90	0.9879	0.9875
4	2.10 E06	1.50 E04	7.14 E-03	0.95	0.9878	0.9869
5	2.10 E06	1.50 E04	7.14 E-03	0.95	0.9972	0.9971
6	2.68 E08	2.00 E04	7.46 E-05	0.90	0.9884	0.9896
7	2.68 E08	2.00 E04	7.46 E-05	0.95	0.9884	0.9888
8	2.68 E08	2.00 E04	7.46 E-05	0.95	0.9973	0.9974
9	6.87 E10	4.00 E04	5.82 E-07	0.90	0.9881	0.9888
10	6.87 E10	4.00 E04	5.82 E-07	0.95	0.9881	0.9888
11	6.87 E10	4.00 E04	5.82 E-07	0.95	0.9973	0.9974
12	3.52 E13	8.00 E04	2.27 E-09	0.90	0.9881	0.9889
13	3.52 E13	8.00 E04	2.27 E-09	0.95	0.9885	0.9890
14	3.52 E13	8.00 E04	2.27 E-09	0.95	0.9973	0.9974
15	4.06 E31	1.40 E05	3.45 E-27	0.95	--	0.9976
16	1.57 E57	2.00 E05	1.27 E-52	0.95	--	0.9977
17	2.04 E90	4.00 E05	1.96 E-85	0.90	--	0.9981

GA: Dengiz 등에 의해 제안된 GA 방법[8]

--: 해를 찾을 수 없었음.

높은 신뢰도를 가지 해들을 탐색해 낼 수 있다는 것을 알 수 있다.

표 3에는 제안한 해법과 Dengiz 등에 의해 제안된 GA 방법에서 계산에 필요한 메모리량과 계산 시간에 대한 비교를 나타내었다. 표 3을 통해 제안한 해법이 Dengiz 등이 제안한 GA 방법보다 더 적은 메모리를 이용하면서 더 빠르게 최적해 또는 준 최적해(near-optimal)를 구해 낼 수 있다는 것을 살펴볼 수 있다.

표 3 메모리 공간과 계산 시간 비교

Problem	$n$	GA		Proposed Method	
		Memory $(n \times (n-1))/2$	ACT (sec.)	Memory $n-2$	ACT (sec.)
1	5	10	0.2172	3	0.0841
2	5	10	0.1831	3	0.0790
3	7	21	10.1074	5	7.0963
4	7	21	9.8602	5	7.5468
5	7	21	12.1495	5	8.7879
6	8	28	41.2264	6	12.9225
7	8	28	41.0972	6	11.0289
8	8	28	40.1506	6	11.1440
9	9	36	230.9028	7	26.2449
10	9	36	317.4627	7	35.7372
11	9	36	308.8991	7	37.9236
12	10	45	1834.2825	8	93.1180
13	10	45	2191.8337	8	98.2712
14	10	45	2172.2625	8	64.9963
15	15	105	--	13	2324.4823
16	20	190	--	18	6895.8018
17	25	300	--	23	10322.8720

ACT: Average Computation Time

GA: Dengiz 등에 의해 제안된 GA 방법[17]

--: 해를 찾을 수 없었음.

### 5. 결론

본 논문에서는 네트워크 신뢰도를 고려하면서 네트워크 구축비용을 최소화시키는 높은 신뢰도의 네트워크 토폴로지 설계 문제에 대한 하나의 해법을 제시하였다. 제안하는 방법은 두 단계로 구성되어 있어, 첫 번째 단계는 스페닝 트리를 찾기 위해 스페닝 트리를 기반으로 하는 GA를 이용하고 두 번째 단계에서는 2-연결성을 이용하는 휴리스틱 방법을 이용하여 최적의 네트워크 토폴로지를 찾아내도록 하였다. 또한, 스페닝 트리를 나타내기 위하여 GA의 표현방법으로 PN 인코딩 방법을 이용하였다. PN 인코딩 방법을 이용하면  $n$ 개의 정점을 가지는 네트워크 설계문제에 대하여  $n-2$ 의 메모리만을 필요로 한다. 본 논문에서 다루는 네트워크 설계 문제의 목적함수는 네트워크 구축비용을 최소화하는 것이고 제약 조건으로는 구축되는 네트워크의 신뢰도가 기준이 되는 신뢰도 이상이 되어야 하는 것이다.

또한, 후보 네트워크 토폴로지에 대하여 네트워크의 신뢰도를 계산하기 위해 백트래킹 알고리즘을 이용하였다. 이 백트래킹 알고리즘은 시스템의 신뢰도( $1-R(x)$ )를 정확하게 계산하기 위해 사용되었다.

그리고, 여러 수치 실험을 통하여 제안한 방법의 효율 성과 얻어낸 해의 우월함을 나타내었고, Jan 등에 의해 제안된 분기 한정법과 Dengiz 등이 제안한 GA 방법을 이용한 결과를 제안한 방법의 결과와 비교함으로써 제안한 방법이 구해낸 해가 최적해 또는 그에 더 가까운 해를 구해내는 데에 필요한 메모리량 및 계산 시간의 면에서 더 효과적이고 효율적이라는 것을 나타내었다.

수치 실험의 결과를 통해 알 수 있듯이 본 논문에서 제안한 두 단계로 구성된 유전자 알고리즘 및 탐욕적인 휴리스틱을 이용한 진화연산에 기초한 알고리즘은 기존의 다른 접근법들보다 더 좋은 결과를 더 효과적이면서도 효율적으로 찾아낸다.

스패닝 트리를 나타내는 유전자 표현에 대한 최근 연구[29]에 따르면 PN이 유전자 알고리즘의 인코딩으로서 가져야 되는 유전 연산후의 부모 염색체와의 연관성의 약화를 가져와서 결점이 있다고 한다. 따라서 본 논문의 첫 번째 단계의 유전자 알고리즘에서 이런 결점을 극복할 수 있는 방향으로 연구를 계속 진행해 나가야 할 것이다.

### 참 고 문 헌

- [1] R.H. Jan, F.J. Hwang, and S.T. Chen, Topological optimization of a communication network subject to a reliable constraint, *IEEE Trans. Reliability*, vol.42, no.1, pp.63-70, 1994.
- [2] K.K. Aggarwal and S. Rai, Reliability evaluation of computer-communication networks, *IEEE Trans. Reliability*, vol.R-30, no.1, pp.32-35, 1981.
- [3] R.S. Wilkov, Design of computer networks based on a new reliability measure, *Symposium on Computer-Communications Networks and Teletraffic*, Ploytechnic Institute of Brooklyn, pp.371-384, 1972.
- [4] K.K. Aggarwal, Y.C. Chopra, and J.S. Bajwa, Topological layout of links for optimising the overall reliability in a computer communication, *Microelectronics & Reliability*, vol.22, no.3, pp.347-351, 1982.
- [5] Y.C. Chopra, B.S. Sohi, R.K. Tiwari, and K.K. Aggarwal, Network topology for maximizing the terminal reliability in a computer communication networks, *Microelectronics & Reliability*, vol.24, pp.911-913, 1984.
- [6] A.N. Venetsanopoulos and I. Singh, Topological optimization of communication networks subject to reliability constraints, *Problem of Control and Information Theory*, vol.15, pp.63-78, 1986.
- [7] F. Glover, M. Lee, and J. Ryan, Least-cost network topology design for a new service: an application of an tabu search, *Annals of Operations Research*, vol.33, pp.351-362, 1991.
- [8] S.J. Koh and C.Y. Lee, A tabu search for the survivable fiber optic communication network design, *Computer and Industrial Engineering*, vol.28, no.4, pp.689-700, 1995.
- [9] M.M. Atiqullah and S.S. Rao, Reliability optimization of communication networks using simulated annealing, *Microelectronics & Reliability*, vol.33, no.9, pp.1303-1319, 1993.
- [10] P.C. Fetterlof and G. Anandalingam, Optimal design of LAN-WAN internetworks: an approach using simulated annealing, *Annals of Operations Research*, vol.36, pp.275-298, 1992.
- [11] S. Pierre, M. Hyppolite, J. Bourjolly, and O. Dioume, Topological design of computer communication networks using simulated annealing, *Engineering Applications of Artificial Intelligence*, vol.8, no.1, pp.61-69, 1995.
- [12] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.
- [13] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, 2000.
- [14] A. Kumar, R.M. Pathak, Y.P. Gupta, and H.R. Parsaei, A genetic algorithm for distributed system topology design, *Computer and Industrial Engineering*, vol.28, no. 3, pp.659-670, 1995.
- [15] A. Kumar, R.M. Pathak, and Y.P. Gupta, Genetic-algorithm-based reliability optimization for computer network expansion, *IEEE Trans. Reliability*, vol.44, no.1, pp.63-72, 1995.
- [16] D.L. Deeter and A.E. Smith, Economic design of reliable network, *IIE Transaction*, vol.30, pp.-1161-1174, 1999.
- [17] B. Dengiz, F. Altiparmak, and A.E. Smith, Efficient optimization of all-terminal reliable networks using evolutionary approach, *IEEE Trans. Reliability*, vol.46, no.1, pp.18-26, 1997.
- [18] B. Dengiz, F. Altiparmak, and A.E. Smith, Local search genetic algorithm for optimal design of reliable networks, *IEEE Trans. Evolutionary Computation*, vol.1, no.3, pp.179-188, 1997.
- [19] M. Ball and R.M. Van Slyke, Backtracking algorithms for network reliability analysis, *Annals of Discrete Mathematics*, vol.1, pp.49-64, 1977.
- [20] S. Skiena, *Implementing Discrete Mathematics Combinatorics and Graph Theory with Mathematica*, Addison-Wesley, Reading, MA, 1990.
- [21] C. Hanzhong and L. Dongkui, A new algorithm for computing the reliability of complex networks by the cut method, *Microelectronics & Reliability*, vol.34, no.1, pp.175-177, 1994.
- [22] D. Mandaltsis and J.M. Kontolen, Overall reliability determination of computer networks with hierarchical routing strategies, *Microelectronics & Reliability*, vol.27, no.1, pp.129-143, 1987.
- [23] C. Liu, M. Dai, X.Y. Wu, and W.K. Chen, A network overall reliability algorithm, *Proc. Neural, Parallel & Scientific Computations*, pp.287-292, Atlanta, 1995.
- [24] H. Cancela and M.E. Khadiri, A recursive variance-reduction algorithm for estimating communication-network reliability, *IEEE Trans. Reliability*, vol.44, no.4, pp.595-602, 1995.

[25] G.S. Fishman, A comparison of four monte carlo methods for estimating the probability of s-t connectedness, *IEEE Trans. Reliability*, vol.R-35, no.2, pp.145-155, 1986.

[26] S.J. Kamat and M.W. Riley, Determination of reliability using event-based monte carlo simulation, *IEEE Trans. Reliability*, vol.R-24, no.1, pp.73-75, 1975.

[27] H. Kumamoto, K. Tanaka, and K. Inoue, Efficient evaluation of system reliability by monte carlo method, *IEEE Trans. Reliability*, vol.R-26, no.5, pp.311-315, 1977.

[28] M.S. Yeh, J.S. Lin, and W.C. Yeh, A new monte carlo method for estimating network reliability, *Proc. 16th Inter. Conf. on Computer & Industrial Engineering*, pp.723-726, 1994.

[29] G. R. Raidl and B. A. Julstrom, Edge Sets: An effective evolutionary coding of spanning trees, *IEEE Trans. on Evol. Compu.*, Vol.7, No.3, pp.225-239, June, 2003.

간. 2000년 Genetic Algorithms & Engineering Optimization, John Wiley & Sons 출간. 관심분야는 진화연산, 지능시스템, 정보기술



김 종 울

1996년 동서대학교 컴퓨터공학과(학사)  
1998년 일본 아시카가공대 경영공학과(석사). 2000년 일본 아시카가공대 경영정보공학과(박사). 2000년~2001년 (주)정보지식연구소 연구원. 2001년~2004년 (주)비트웍 책임연구원. 2004년~현재 카이스트 테크노경영 연구소 연수연구원. 관심분야는 진화 연산, 멀티미디어 전송, 오버레이 멀티캐스트



이 재 욱

1975년 경북대학교 전장공학과(학사). 1981년 동아대학교 전자공학과(석사). 1989년 동아대학교 전자공학과(박사). 1987년~1988년 일본 아시카가공대 방문교수  
1992년~현재 동서대학교 인터넷공학부 교수. 관심분야는 진화연산, 멀티미디어,

유무선 통합



현 광 남

1969년 일본 공학원대학교 전자공학과(학사). 1974년 일본 동대학교 대학원(박사). 1974년~1980년 일본 아시카가공대 경영공학과 강사. 1982년~1983년 미국 네브라스카 링컨대학교 객원조교수  
1982년~1987년 동대학 조교수. 1987년~2003년 동대학 교수. 1999년~2000년 미국 캘리포니아 대학(버클리) 방문교수. 2003년~현재 일본 와세다대학교 대학원 정보생산시스템연구과 교수. 1997년 Genetic Algorithms & Engineering Design, John Wiley & Sons 출