2기준 네트워크 설계를 위한 새로운 인코딩 방법을 기반으로 하는 유전자 알고리즘

(A Genetic Algorithm with a New Encoding Method for Bicriteria Network Designs)

김 종 율 † 이 재 욱 †† 현 광 남 †††
(Jong-Ryul Kim) (Jae-Uk Lee) (Mituso Gen)

요 약 인터넷이 발전함에 따라 네트워크 시스템의 토폴로지 설계에 관한 여러 가지 문제들에 관심이 증가하고 있다. 이러한 네트워크의 토폴로지 구조는 서비스 센터, 터미널(사용자), 그리고 연결 케이블로 이뤄져 있으며 네트워크 시스템들은 사용자들로부터의 요구사항이 많아지고 있기에 주로 광케이블로 구축하는 경우가 점차 늘어나고 있다. 하지만, 광케이블의 고비용을 고려하면 네트워크의 구조가 스패닝 트리로 구축되어 지는 것이 바람직하다고 볼 수 있다.

본 논문에서는 연결비용, 평균 메시지 지연, 네트워크 신뢰도를 고려하여, 광케이블로 구성되는 광대역 통신 네트워크의 2기준 네트워크 토폴로지 설계 문제들을 풀기 위한 유전자 알고리즘을 제안한다. 또한, 후보 네트워크 토폴로지 구조를 염색체로 표현하기 위해 Prüfer수(PN)와 클러스터 스트링으로 구성되는 새로운 인코딩 방법도 제안한다. 마지막으로 제안한 유전자 알고리즘이 계산 시간뿐만 아니라 파레토 최적성의 관점에서도 보다 효율적이며 효과적인 방법이라는 것을 수치예를 통해 살펴본다.

키워드 : 유전자 알고리즘, 네트워크 토폴로지 설계, 2기준 최적화 문제

Abstract Increasing attention is being recently devoted to various problems inherent in the topological design of networks systems. The topological structure of these networks can be based on service centers, terminals (users), and connection cable. Lately, these network systems are well designed with fiber optic cable, because the requirements from users become increased. But considering the high cost of the fiber optic cable, it is more desirable that the network architecture is composed of a spanning tree.

In this paper, we present a GA (Genetic Algorithm) for solving bicriteria network topology design problems of wide-band communication networks connected with fiber optic cable, considering the connection cost, average message delay, and the network reliability. We also employ the Prüfer number (PN) and cluster string in order to represent chromosomes. Finally, we get some experiments in order to certify that the proposed GA is the more effective and efficient method in terms of the computation time as well as the Pareto optimality.

Key words : Genetic Algorithms, Network Topology Design, Bicriteria Optimization Problem

1. 서 론

통신 네트워크의 규모가 점차 확대됨에 따라 광대역 통신 네트워크를 위한 토폴로지 설계 문제는 네트워크 설계자, 네트워크 분석가, 네트워크 관리자와 같은 많은 관련 연구자들에 의해 많은 주목을 받고 있다. 고비용의 하드웨어/소프트웨어를 공유하고 원거리로부터의 메인 시스템에 대한 접근을 제공하기 위해 통신 네트워크의 이용은 급속히 늘어가고 있다. 텔레커뮤니케이션, 컴퓨터 네트워킹, 하수도관 시스템, 기름 및 가스관 시스템 등과 같은 많은 실세계의 네트워크 설계 문제에서, 어떤 제약조건들을 만족하면서 목적함수들을 최적화하는 네트워크 시스템을 설계하는 것은 매우 중대한 문제이다. 최근에 사용자로부터의 요구사항들이 늘어나고 있어서

† 비 회 원 : 동서대학교 첨단기술연구센터
xmaskjr@gdsu.dongseo.ac.kr
†† 종신회원 : 동서대학교 인터넷공학부 교수
julee@dongseo.ac.kr
††† 비 회 원 : 일본 와세다대학교 정보생산시스템연구과 교수
gen@waseda.jp
논문접수 : 2004년 8월 5일
심사완료 : 2005년 8월 19일

통신 네트워크 시스템들은 광케이블을 이용하여 주로 구축되고 있다.

일반적으로 대형 통신 네트워크 시스템들은 백본으로 여러 네트워크들을 함께 묶어서 구성된다. 또한 근래에 네트워크를 구축하려고 할 때, 광대역폭(거의 50 Tbps), 적은 신호 감쇠(거의 0.2 dB/km), 적은 신호 왜곡, 낮은 전력 요구 등[1]과 같은 잠재력을 가지고 있어 광케이블이 주로 사용되고 있다. 그러나 광케이블의 비싼 비용을 고려하면 네트워크의 구조가 스패닝 트리로 되는 것이 바람직하다고 볼 수 있다. 따라서 전반적인 네트워크의 구조로 적당한 모델은 고용량의 케이블로 이뤄진 백본 네트워크로 서비스 센터들을 연결하고 분산 네트워크로 그 사용자를 연결하는 것이라고 할 수 있다.

또한, 이러한 통신 네트워크 시스템의 토폴로지를 설계할 때 중요한 것 중에 하나는 구축비용, 평균 메시지 지연시간, 통신량, 네트워크 신뢰도 등과 같은 성능 측도를 최적화하는 최상의 레이아웃을 탐색해 내는 것이다. 이러한 시스템의 성능측도들은 매우 중요하며 네트워크 토폴로지에 의해 좌우되는 경우가 많다. 그리고 클러스터링과 라우팅의 두가지 주요한 항목들도 네트워크 토폴로지 설계 문제에 포함되어 있다. 클러스터링 문제는 얼마나 많은 서비스 센터들(clusters)로 네트워크가 나눠질 수 있는가와 어떻게 사용자들(terminals)이 클러스터에 연결되어 질 것인가와 같은 두 가지 항목들로 이뤄져 있으며 라우팅 문제는 스패닝 트리 토폴로지를 어떻게 상호 연결할 것인가를 결정하는 것이라고 볼 수 있다.

유전자 알고리즘(GA)은 많은 실세계문제들을 풀기 위해 자주 이용되고 있으며 네트워크 최적화 문제들의 최적화 기술로서의 잠재력을 매우 주목받고 있는 기법이다[2,3]. 그리고 다목적 문제를 위한 GA 기법이 많이 연구되어 지고 있고 운송문제[4], 생산프로세스 계획 문제[5], 최소 스패닝 트리 문제[6] 등과 같은 문제들에서도 다목적 문제에 대한 기법으로서 매우 활발한 연구가 이뤄지고 있다.

최적의 네트워크 토폴로지를 찾아내는 것은 매우 복잡한 조합최적화 문제이며 NP-hard 최적화 문제로 분류된다. 전통적인 방법들을 이용하면, 네트워크의 규모가 지수적으로 확대되어지는 경우, 네트워크 토폴로지 설계 문제를 풀이하기가 힘들어 진다. 이 때문에 GA를 기초로 하는 휴리스틱 알고리즘이 주목을 받고 있다.

본 논문에서는 연결비용, 평균 메시지 지연, 그리고 네트워크 신뢰도를 고려한 2기준 네트워크 설계 문제들을 풀기 위해 GA를 제안한다. 본 연구에서 실제 네트워크 구조로 스패닝 트리 토폴로지만이 이용될 수 있다는 조건을 두고 있기 때문에 스패닝 트리를 표현하기 위해서 Prüfer수(PN)와 클러스터 문자열을 이용한 새로운 인코딩 방법을 제안하였다. 제안한 인코딩 방법을 이용하면 n개의 서비스 센터들과 m개의 사용자들로 구성되어지는 문제를 위해 필요한 메모리는 단지 $n+m-2$뿐이다. 이와 달리, Dengiz, Altiparmak, Smith[7,8]에 의해 개발되어진 링크 기반 인코딩은 설계 대상 네트워크의 연결 상태를 나타내는 인접행렬의 상삼각요소 또는 하삼각요소의 정보를 이용한다. 따라서 이들의 인코딩 방법을 이용하면 $(n+m)\times(n+m-1)/2$의 메모리가 필요로 하게 된다. 본 논문에서의 문제들이 두개의 목적함수들을 가지고 있기에 모든 목적함수를 하나의 측도로 나타내기 위해 제안한 GA에서는 가중화 평가법을 이용한다. 마지막으로 수치예를 통해 제안한 GA를 이용하여 얻어진 결과의 효율성과 효과성을 살펴 볼 것이다. 또한, 제안한 인코딩 방법을 이용하는 GA의 결과와 Dengiz, Altiparmak, Smith[7,8]에 의해 제안된 링크 기반의 인코딩 방법을 이용한 GA의 결과를 비교하여 볼 것이다.

2. 2기준 네트워크 설계문제

n개의 서비스 센터들와 m개의 사용자들을 구성되는 네트워크를 생각해 보자. 예를 들어 그림 1과 같이 5개의 서비스 센터와 18개의 사용자들로 구성되는 네트워크를 생각해보자. 사용자들간의 통신 통화량 수요는 사용자 통화량 행렬로 명명되는 $m\times m$ 행렬 $\boldsymbol{U}$로 정의되며 행렬 $\boldsymbol{U}$의 u_{ij}는 사용자에서 사용자로의 데이타 통화량를 의미한다. 사용자간의 통화량은 매우 혼잡한 경우에서 거의 통화량이 없는 경우까지 매우 다양한 형태로 나타난다. 더욱이 이런 통화량은 폭주하거나 일정하게 지속되는 경우도 존재한다. 대부분의 시간에 실제의 통화량은 피크치보다 현저히 적기 때문에 통화량 피크율이 통화량 제약조건보다 과하게 설정되는 경우가 많다. 반면, 평균 통화량으로 제약조건 설정을 하게 된다면 과

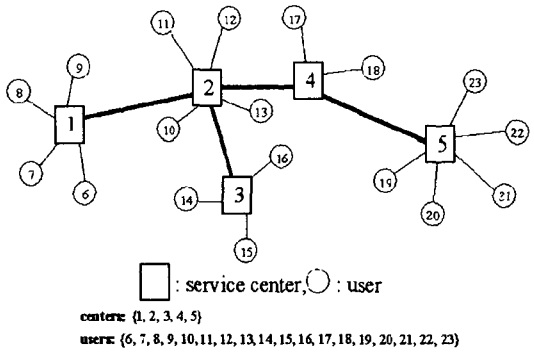


그림 1 네트워크 구조의 예

중한 통화량이 밀려올 때 바람직한 결과를 얻을 수 없게 된다. 본 논문에서는 통화량 특성이 미리 정해져 있으며 사용자 통화량 행렬 $\boldsymbol{U}$에 요약된다.

또한, 서비스 센터들의 연결 형태를 나타내는 $n\times n$서비스 센터 토폴로지 행렬 $\boldsymbol{X_1}$를 정의한다. 행렬 $\boldsymbol{X_1}$의 x_{1ij}는 다음과 같이 정의된다.

$$x_{1ij}=\begin{cases}1, & \text{센터 } i\text{와 센터 } j\text{가 서로 연결되어 있으면 } (i\neq j)\\ 0, & \text{그렇지 않으면}\end{cases}$$

그리고 네트워크가 n개의 세그먼트 (서비스 센터 또는 클러스터)로 나눠질 수 있다고 가정한다. 사용자들은 n개의 서비스 센터들로 분산된다. 사용자가 어떤 센터에 속하는 지를 나타내는 $n\times m$ 클러스터링 행렬 $\boldsymbol{X_2}$를 정의한다. 따라서

$$x_{2ij}=\begin{cases}1, & \text{사용자 } j\text{가 센터 } i\text{에 연결되어 있으면}\\ 0, & \text{그렇지 않으면}\end{cases}$$ 이다.

한 개의 터미널은 반드시 하나의 센터에만 연결된다. $n\times(n+m)$행렬 $\boldsymbol{X}$를 스패닝 트리 행렬($[\boldsymbol{X_1}\ \boldsymbol{X_2}]$)이라고 하고 $n\times n$행렬 $\boldsymbol{T}$는 서비스 센터 통화량 행렬이라고 정의한다. 행렬 $\boldsymbol{T}$의 요소인 t_{ij}는 센터 i에 접속되어 있는 사용자들로부터 센터 j에 접속되어 있는 사용자들에게로 전송되는 통화량를 나타낸다. 즉, $\boldsymbol{T}=\boldsymbol{X_2}^T\boldsymbol{U}\boldsymbol{X_2}$이다.

서비스 센터들은 광케이블에 의해 상호 연결된다. 송신측 센터 i로부터 수신측 센터 j에로의 통화량은 광케이블을 통해 전송된다. 센터 i와 센터 j사이의 경로는 단하나의 케이블로 직접 연결되어 있거나 센터간의 직접연결이 힘든 경우에는 여러 케이블들을 통해 간접적으로 연결되어 있다. 후자의 경우에는 통화량은 연결 경로상의 케이블들과 센터들을 통해 전달되어 진다.

센터는 旣知(기지)의 용량을 갖는 세그먼트로 정의된다. 이 세그먼트는 토큰링, 이더넷 등이 될 수 있으며 각각의 세그먼트의 동작 및 성능은 서로 다르다. 그럼에도 이러한 모든 세그먼트의 평균 지연은 질적으로 같은 형태로 나타난다. 평균 지연은 세그먼트의 통화부하가 체증을 일으키면 증가하게 된다. 통화부하가 세그먼트 용량에 도달하면 지연은 무한대로 증가한다.

본 논문에서는 M/M/1 모델[9]을 하나의 클러스터의 동작을 묘사하기 위해 이용한다. M/M/1 모델에 따르면 센터 i로 전송되는 비트의 평균 지연은 다음과 같이 정의될 수 있다.

$$D_i=\frac{1}{C_i-F_i}$$

여기서, C_i는 센터 i의 통화용량이고 F_i는 센터 i로 전송되는 총통화부하이다.

케이블은 두 센터간을 연결하는 요소이다. 각 센터는 하나의 센터에서 다른 센터로 전송되어야 하는 패킷들을 감지하고 이런 동작은 룩업 테이블을 이용함으로써 이뤄진다. 전송측 센터로부터 전송되어지는 패킷은 룩업 테이블에 의해 수신측 센터가 결정되어 지며 결정된 경로를 통해 전송된다. 룩업 테이블을 따라 수신측 센터로 패킷이 전송이 되어 지면 전송된 패킷은 수신측의 대기열에 넣어진다. 룩업 테이블의 생성 및 수정은 본 논문의 영역에 벗어나는 내용이기에 룩업 테이블의 내용은 변화하지 않는다고 가정한다. 이러한 가정으로부터 각 패킷에 대한 지연은 다음과 같이 두개의 항목으로 구성되어진다.

- 룩업에 기인하는 지연, 즉 룩업 시간은 센터를 통해 패킷이 전송될 때마다 일정하다고 가정될 수 있다.
- 수신측 센터에서의 대기열 지연, 즉 대기열 지연은 통화량은 수신측 센터의 총통화부하의 일부로 볼 수 있기에 수신측 센터의 지연으로 정의될 수 있다.

단지 룩업 시간만이 지연을 일으키는 원인이 되므로, 센터 i와 센터 j간의 케이블에 기인하는 비트 당 지연은 $\beta_{ij}=p_{ij}/l$로 나타낼 수 있다. 여기서, p_{ij}는 패킷당 룩업 지연이며 l은 패킷의 길이이다. $f_{ij}(\mathbf{X})$를 센터 i와 센터 j간의 케이블를 통해 전송되는 총통화량라고 정의한다면 평균지연은 다음과 같다.

$$D_c=\frac{1}{\Gamma}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}\cdot f_{ij}(\boldsymbol{X})\right]$$

여기서 Γ 는 전송되는 총통화량이다.

이상에서의 정의를 통해 네트워크 상의 총 평균 지연은 센터들의 지연 및 패킷들의 지연으로 구성된다고 할 수 있다. 따라서, 네트워크의 평균지연은 다음과 같이 정의된다.

$$D_t=\frac{1}{\Gamma}\left[\sum_{i=1}^{n}\frac{F_i(\boldsymbol{X})}{C_i-F_i(\boldsymbol{X})}+\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}\cdot f_{ij}(\boldsymbol{X})\right]$$

본 논문에서는 2기준 네트워크 토폴로지 설계문제를 정의하기 위해 다음의 기호들을 정의한다.

$$a_{ij}^{k}(\boldsymbol{X})=\begin{cases}1, & \text{센터 } k\text{를 통해 센터 } i\text{에서 센터 } j\text{로}\\ & \text{트래픽이 발생하면}\\ 0, & \text{그렇지 않으면}\end{cases}$$

$$b_{ij}^{(k,l)}(\boldsymbol{X})=\begin{cases}1, & \text{센터 } k\text{와 센터 } l\text{을 연결하는 링크들을 통해}\\ & \text{센터 } i\text{에서 센터 } j\text{로의 트래픽이 존재한다면}\\ 0, & \text{그렇지 않으면}\end{cases}$$

g_i 센터 i의 최대 연결 가능수(사용자)

$R_{\min}$ 네트워크 신뢰도의 최소 요구량

본 논문에서 다룰 2종류의 2기준 네트워크 토폴로지 설계 문제는 다음과 같은 0-1 비선형 계획 모델로 정식화된다.

Model 1:

$$\min \frac{1}{\Gamma}\left[\sum_{i=1}^{n}\frac{F_i(\boldsymbol{X})}{C_i-F_i(\boldsymbol{X})}+\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}\cdot f_{ij}(\boldsymbol{X})\right]$$

$$\min \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{1ij}\cdot x_{1ij}+\sum_{i=1}^{n}\sum_{j=1}^{m} w_{2ij}\cdot x_{2ij}$$

$$s.t.\ R(\boldsymbol{X}) > R_{\min}$$

$$\sum_{i=1}^{n} x_{1ij}=1, \quad j=1,2,\cdots,n$$

$$\sum_{j=1}^{m} x_{2ij}\le g_i, \quad i=1,2,\cdots,n$$

$$\sum_{i=1}^{n} x_{2ij}=1, \quad j=1,2,\cdots,m$$

$$F_i(\boldsymbol{X}) < C_i, \quad i=1,2,\cdots,n$$

Model 2:

$$\max\ R(\boldsymbol{X})$$

$$\min \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{1ij}\cdot x_{1ij}+\sum_{i=1}^{n}\sum_{j=1}^{m} w_{2ij}\cdot x_{2ij}$$

$$s.t.\ \sum_{i=1}^{n} x_{1ij}=1, \quad j=1,2,\cdots,n$$

$$\sum_{j=1}^{m} x_{2ij}\le g_i, \quad i=1,2,\cdots,n$$

$$\sum_{i=1}^{n} x_{2ij}=1, \quad j=1,2,\cdots,m$$

여기서 $R(\boldsymbol{X})$는 이후의 절에서 소개할 네트워크 신뢰도이며, w_{1ij}는 센터 i와 센터 j간의 링크 가중치이고, w_{2ij}는 센터 i와 사용자 j간의 링크 가중치이다. 또한, g_i는 센터 i에 연결할 수 있는 최대수이고 제공되는 총 통화량 Γ 는 다음과 같다.

$$\Gamma=\sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij}$$

센터 k의 총 통화량 $F_k(\boldsymbol{X})$는 다음과 같이 정의된다.

$$F_k(\boldsymbol{X})=\sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij}\cdot a_{ij}^{k}(\boldsymbol{X}), \quad k=1,2,\cdots,n$$

링크 (k, l)의 총통화량 $f_{kl}(\boldsymbol{X})$는 또한 다음과 같이 정의된다.

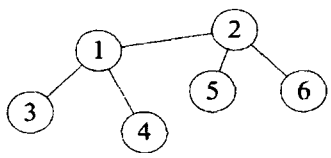
$$f_{kl}(\boldsymbol{X})=\sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij}\cdot b_{ij}^{(k,l)}(\boldsymbol{X}), \quad k=1,2,\cdots,n \quad l=1,2,\cdots,n$$

3. 제안하는 유전자 알고리즘

3.1 유전자 표현 및 초기화

유전자 표현은 코딩 공간에서 문제의 후보해를 표현하기 위한 자료구조를 의미한다. 보통 다른 종류의 문제들은 다른 자료구조 또는 유전자 표현을 사용하게 된다. 스패닝 트리로 나타나지는 실제의 네트워크 구조를 표현하기 위한 방법으로 다음의 두 가지가 있다: 그중 하나는 링크를 기반으로 하는 방법[7,8]이고, 또 다른 하나는 *Prüfer* 수(PN)를 기반으로 하는 방법[10,11]이다. 링크를 기반으로 하는 표현 방법은 n개의 노드에 대하여 $n\times(n-1)/2$의 메모리를 필요로 하지만 PN을 기반하는 표현 방법은 $n-2$개의 메모리만으로도 스패닝 트리를 표현 할 수 있기 때문에 본 논문에서는 PN을 기반으로 하는 유전자 표현법을 이용할 것이다. 본 논문에서 우리는 서비스 센터는 PN으로 나타내고 사용자는 클러스터 문자열로 표현하는 새로운 유전자 표현법을 제안한다.

그래프 열거에 대한 일반적인 정리 중 하나는 k개의 노드를 가진 완전 그래프에는 $k^{(k-2)}$개의 트리가 존재한다는 Cayley의 정리이다. Prüfer는 그러한 트리들과 k개의 숫자들로 이뤄지는 $k-2$길이의 문자열들의 집합사이의 일대일 대응관계를 확증함으로써 Cayley의 정리를 증명하였다[6]. 이것은 k개의 다른 숫자들로 구성되는 $k-2$개의 숫자열로 트리를 표현할 수 있다는 것을 의미한다. 이 숫자열을 PN이라고 한다[12]. 예를 들어 설명하자면 그림 2에서 나타낸 것과 같이 6개의 노드로 구성되는 스패닝트리를 PN(1 1 2 2)으로 표현하는 예를 살펴보자. 상위의 프로시저에 의해 PN이 생성되는 과정을 묘사하면 다음과 같다: 먼저 가장 낮은 라벨을 가진 잎노드를 찾는다. 이 경우에는 노드 3에 해당한다. 트리에서 노드 3는 노드 1에 연결되어 있으므로 PN의 첫숫자로 1을 할당하고 노드 3과 링크 (3, 1)를 트리에서 제거한다. 이런 과정을 링크(2, 6)이 남을 때 까지 반복하면 그림 2의 트리에 대한 PN이 생성된다.



1	1	2	2	Prüfer number

그림 2 트리와 그에 대한 Prüfer수 (PN)

반면에 PN으로부터 트리를 생성하는 디코딩 과정을 살펴 보면, P=(1 1 2 2)에 대해 노드 3, 4, 5, 6가 PN에 나타나지 않는 노드 집합을 P'={3, 4, 5, 6}로 나타낼 수 있다. 노드 3이 제일 낮은 라벨을 가진 노드이고 P의 가장 왼쪽에 있는 숫자가 1이므로 노드 3과 노드 1을 연결한 후 P'에서 노드 3을 PN에서 노드 1을 제거한다. 그 다음으로 노드 4와 P의 가장 왼쪽 숫자 1을 고려하여 노드 4와 노드 1을 연결한 후 각각 제거를 한다. 이때 P에서 숫자 1을 제거할 때 이제 더 이상 숫자 1이 나타나지 않기에 숫자1을 집합 P'에 포함시킨다.

그림, P'는 {1, 5, 6}이 된다. 이러한 과정을 P가 비어지고 P'에 두개의 숫자가 남을때 까지 반복하고 P'에 남은 두 숫자의 노드를 서로 연결하면 그림 2의 트리가 생성된다.

PN 인코딩은 가능한 모든 스패닝 트리를 표현할 수 있을 뿐만 아니라 PN에 노드가 d-1번 나타나면 그 노드의 디그리는 d가 된다는 정보도 나타낼 수 있다. 즉, PN에 노드의 라벨이 d-1번 나타나면 트리에서 그 노드는 d개의 연결을 가지게 된다.

이와 같은 PN은 운송문제, 최소 스패닝 트리 문제 등과 같은 연구 분야에서 스패닝 트리를 표현하기 위한 방법으로 적당하다는 연구들의 보고가 있다[1,2,4,6].

지금까지 PN 인코딩 방법에 대해서 소개하였다. 본 논문에서는 PN의 장점을 살리면서 보다 더 네트워크 토폴로지 설계문제에 적합한 유전자 표현을 위해 다음과 같은 두개의 유전자형을 가지는 인코딩 방법을 제안하였다: 서비스 센터는 PN으로 표현하고 사용자는 클러스터 문자열으로 나타내는 것이다. 따라서 그림 3에 표시된 것과 같이 초기화 과정에서 서비스 센터는 영역 $[1,n]$에서 임의로 $n-2$개의 숫자(PN)로 생성하여 나타내고 사용자는 영역 $[1,n]$에서 임의로 m개 숫자(클러스터 문자열)로 생성한다. 여기서 클러스터 문자열은 어떻게 사용자를 서비스 센터에 할당되는지에 대한 정보를 나타낸 것으로 사용자가 어떤 특정 서비스 센터에 속하는 것을 의미한다. 본 논문에서의 유전자 표현은 $n+m-2$ 길이를 가진다.

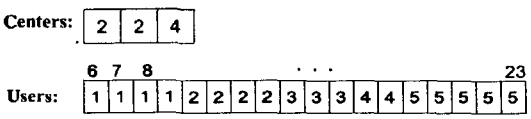


그림 3 그림 1에 대한 유전자 표현

3.2 평가 함수

단일목적 네트워크 토폴로지 설계 문제는 목적 함수치에 따라 각각의 염색체의 적합도를 계산하면 되기 때문에 간단히 다룰 수 있다. 그러나 2기준 네트워크 토폴로지 설계 문제에서는 각각의 목적 함수치는 계산할 수 있으나 실제로 목적함수들이 서로 배타적으로 상충관계를 취할 경우가 많기 때문에 쉽게 적합도를 계산해 내기가 힘들다. 다르게 말하면 절대적인 최적해는 얻을 수 없고 단지 다음과 같은 파레토 최적해들 (nondominated solution)만을 얻을 수 있다. 파레토 최적해들의 개념을 정의하면 다음과 같다.

정의 1. 실현 가능해 집합 $\boldsymbol{S}$에 대해, 다음과 같은 조건을 만족하는 실현 가능해 $\boldsymbol{X}(\boldsymbol{X}\in\boldsymbol{S})$가 존재하지 않는다면 실현 가능해 $\boldsymbol{X'}(\boldsymbol{X'}\in\boldsymbol{S})$는 2기준 네트워크 토폴로지 최적 설계 문제를 위한 파레토 최적해(비우월해)라고 한다.

$f_i(\boldsymbol{X}) \le f_i(\boldsymbol{X'})$, 모든$i(\in\{1,2\})$에 대하여
$f_j(\boldsymbol{X}) < f_j(\boldsymbol{X'})$, 적어도하나의$j(\in\{1,2\})$에 대하여

즉, 만일 하나의 목적 함수치를 희생하지 않으면서 다른 목적 함수치를 최적화 할 수 있는 실행 가능해 $\boldsymbol{X}$가 존재하지 않는다면 실행 가능해 $\boldsymbol{X'}$는 파레토 최적해라고 한다. 2기준 최적해 문제에서의 파레토 최적해와 우월해의 예를 나타내면 다음의 그림 4와 같이 나타낼 수 있다. 여기서 흰점은 우월해들을 검은점은 비우월해(파레토 최적해)들을 의미한다.

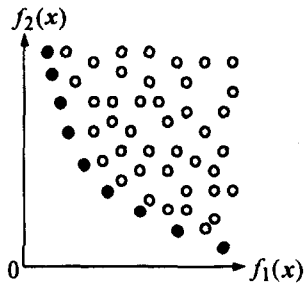


그림 4 파레토 최적해의 예

GA에서 적합도 값을 계산하기 위한 적합도 함수를 정의하기 위해 본 논문에서는 가중화 방법을 이용한다. 따라서 2개의 목적함수들을 평가하고 하나의 적합도 값으로 변환하여 각각의 염색체들을 평가하는 다음의 평가방법을 이용한다.

단계 1. Prüfer수와 클러스터 스트링으로 구성된 염색체를 스패닝 트리 $\boldsymbol{X}$로 변환한다.

단계 2. 목적 함수치($f_i(\boldsymbol{X}), i=1,2$)를 계산한다.

단계 3. 각 목적 함수치에 대하여 최소값 $f_i^{\min}$(또는 최대값 $f_i^{\max}$)을 다음과 같이 결정한다. 이전의 세대에서 저장한 목적 함수치와 현재 세대에서의 목적 함수치들과 비교하여 최선의 목적 함수치를 유지해 나간다.

$$f_i^{\min(t)} = \min_k\{(f_i^{\min(t-1)}, f_i^{(t)}(S_k))|k=1,2,\cdots,chr_size\}, i=1,2$$
$$f_i^{\max(t)} = \max_k\{(f_i^{\max(t-1)}, f_i^{(t)}(S_k))|k=1,2,\cdots,chr_size\}, i=1,2$$

여기서 $f_i^{\min(t)}, f_i^{\max(t)}$는 각각 t세대에서의 i번째 목적 함수치에 대한 최소값과 최대값을 나타내고 $f_i^{(t)}(S_k)$는 t세대의 k번째 염색체의 i번째 목적 함수치를 의미하며 chr_size는 유전자 연산후의 생성된 자손들의 개수와 집단의 크기(pop_size)를 합한 수를 나타낸다.

단계 4. 적합도를 계산하기 위해 필요한 가중치를 계산하기 위해 다음의 수식을 계산한다.

$$\alpha_i = \frac{f_i^{\max(t)} - f_i^{\min(t)}}{f_i^{\max(t)}} \quad \beta_i = \frac{\alpha_i}{\sum_{i=1}^{2}\alpha_i}, \quad i=1,2$$

단계 5. 각 염색체들에 대한 적합도를 다음과 같이 계산한다.

$$eval(S_k) = \sum_{i=1}^{2}\beta_i \cdot d_i(S_k),\ k=1,2,\cdots,chr_size$$

여기서 $d_i(S_k)$는 i번째 목적 함수의 정규화된 값을 의미하고 다음의 수식으로 나타낼 수 있다.

$$d_i(S_k) = \begin{cases} \dfrac{f_i^{(t)}(S_k) - f_i^{\min(t)} + \gamma}{f_i^{\max(t)} - f_i^{\min(t)} + \gamma}, & \text{최대화문제} \\ \dfrac{f_i^{\max(t)} - f_i^{(t)}(S_k) + \gamma}{f_i^{\max(t)} - f_i^{\min(t)} + \gamma}, & \text{최소화문제} \end{cases}$$

γ는 (0,1) 구간내에서 임의로 선택된 양의 실수를 의미하고 위의 수식에서 영으로 나누는 오류를 방지하기 위해 사용된다.

상위의 평가방법에 대한 보다 자세한 정보는 Gen과 Cheng의 책에서 참조할 수 있다[2].

3.3 트리에 기초한 신뢰도 계산

광케이블을 이용하기 때문에 스패닝 트리만을 토폴로지로 구성할 수 있다고 가정하고 있기에, 네트워크의 신뢰도의 평가는 전체 트리를 부분 트리 및 하나의 노드로부터 가능하다[13].

신뢰도 측도로 본 논문에서는 모든 작동 노드들이 연결되어질 확률로서 정의한다. 만일 노드와 링크와 같은 네트워크 구성요소들의 신뢰도가 정해져 있다고 한다면 스패닝 트리는 쉽게 계산될 수 있다. 루트를 가진 트리의 특성을 고려하면서 각 부분트리의 루트에 상태 벡터를 연관시킬 수 있다. 루트노드에 연관된 상태 벡터는 루트노드에 관련된 모든 정보를 포함하고 있다. 따라서 루트노드의 상태벡터가 나타내는 재귀관계는 이미 정해진 부분트리들에 대한 상태 벡터들을 통해 정의된다. 하나의 노드로 구성된 부분트리들에 대한 상태 벡터는 분명하다. 재귀 관계를 이용하여 부분트리들의 상태 벡터는 전체 네트워크의 상태벡터가 얻어질 때까지 더 큰 부분트리에 포함시켜 계산한다[14].

재귀관계를 유도하는 것은 다소 기계적인 경향을 띤다. 그림 5에 나타나 있는 것을 살펴보면 쉽게 유도할 수 있다. 루트 노드 i를 가진 부분트리와 루트 노드 j를 가진 부분트리, 두 부분트리를 살펴보자. 노드 i와 노드 j의 상태 벡터는 알고 있다고 가정하고 링크 (i,j)를 이용하여 노드 j를 노드 i의 부모 노드가 되도록 하여서 노드 j를 루트노드로 하는 새로운 트리의 상태벡터를 계산해 내려고 한다고 하자. 또한, 트리의 재귀관계를 기반으로 하는 신뢰도 계산을 위한 프로시저 기호들을 다음과 같이 정의하자: p_i^f는 노드 i의 노드 고장 확률이고 $p_i^o(=1-p_i^f)$는 노드의 동작확률이다. 그리고 링크 (i,j)에도 유사하게 l_i^f와 l_i^o는 각각 링크가 고장날 확률과 동작할 확률을 의미한다. 또한 각 부분트리들을 위한 상태벡터들을 다음과 같이 정의한다: e_i는 노드 i의 부분트리의 고장확률이며, o_i는 루트노드인 노드 i를 포함하여 부분트리의 동작 노드들 집합들이 서로 연결될 확률이고, r_i는 부분트리의 루트노드는 고장나고 부분트리의 동작노드들 집합이 서로 연결될 확률이다.

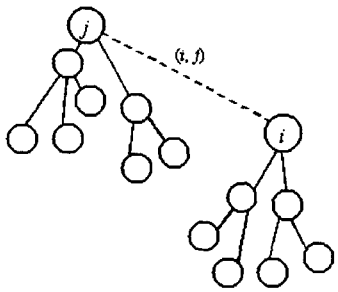


그림 5 트리의 재귀관계

루트노드를 노드 1으로 하고 n개의 노드들 가진 트리에 대해 모든 동작 노드들이 통신할 수 있는 네트워크 신뢰도 $R(\boldsymbol{X})$는 다음과 같이 계산된다:

프로시저: 신뢰도 계산

단계 1. $r_i = 0, o_i = p_i^o, e_i = p_i^f (i=1,2,\cdots,n)$으로 설정한다. 또한, $i=n$으로 설정한 후 단계 2로 이동한다.

단계 2. 노드 i의 부모 노드가 노드 j이라고 하면 다음의 재귀식을 이용하여 r_i, o_i, e_i를 재계산한다.

$$r_j' = r_j e_i + r_i e_j + o_i e_j$$
$$o_j' = o_i o_j l_i^o + o_j e_i$$
$$e_j' = e_i e_j$$

단계 3. $i=i-1$으로 설정한다. 만약 $i=1$이면, 단계 4로 이동하고 그렇지 않다면 단계 2로 이동한다.

단계 4. $r_1 + o_1 + e_1$을 반환한다.

3.4 선택

GA에서 선택연산은 매우 중요한 역할을 한다. 교차 및 돌연변이 같은 유전자 연산들은 해 공간에서의 탐색을 위한 탐험연산이라고 한다면 선택연산은 진화과정을 인도하는 GA를 위한 개척연산이라고 할 수 있다. 본 논문에서 사용된 선택 연산은 자유로이 해 공간을 탐색하도록 하기 위해 룰렛휠법과 엘리티스트(elitist)방법을

혼합한 연산이다. 적합도 비례 방법 중에 하나인 룰렛휠 선택 연산은 임의로 새로운 세대를 생상하기 위하여 이용되고 엘리티스트 방법은 룰렛휠 연산의 오류를 극복하고 가장 좋은 적합도를 가진 염색체를 유지하기 위해 사용된다. 본 논문에서 이용된 선택연산을 다음과 같이 기술한다:

단계 1. 각 염색체 $\boldsymbol{X}_p$, $(p=1,2,\cdots,chr_size)$에 대한 누적 확률 a_p을 계산한다.

단계 2. 구간 [0,1]에서 임의의 실수 r을 생성한다.

단계 3. 만일 $r \le a_1$이면, $\boldsymbol{X}_1$을 선택하고 그렇지 않고 $a_{p-1} < r \le a_p$이라면 p번째 염색체 $\boldsymbol{X}_p$ $(2 \le p \le chr_size)$를 선택한다.

단계 4. 단계 2와 3을 pop_size개의 염색체를 선택하기 위해 pop_size회 반복한다.

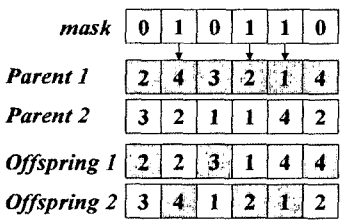
단계 5. 다음 세대에 현재 세대에서의 최선의 염색체가 선택되지 않았다면 최선의 염색체를 새로 생성된 집단에서 임의로 선택된 염색체와 대체하여 최선의 염색체가 다음 세대에 전달되도록 한다.

이런 선택연산을 이용하여 최선의 염색체를 현재의 세대에서 다음세대로 유지할 수 있다.

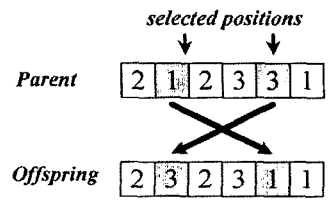
3.5 유전자 연산자들

균등(uniform) 교차로도 불리는 다점(multi-point) 교차를 이용한다. 이러한 종류의 교차는 그림 6에서 보는 바와 같이 두개의 부모 염색체를 임의로 선택하여 하나의 부모염색체에서 임의로 유전자 표현형들을 선택하여 자손염색체를 구성한다.

또한, 본 논문에서의 돌연변이 연산은 하나의 염색체에서 임의로 선택된 두개의 위치에 있는 유전자 표현형들을 교환하는 교환 돌연변이를 사용한다.



(a) 다점 교차 연산



(b) 교환 돌연변이 연산자

그림 6 유전자 연산

3.7 염색체에 대한 수정

각 센터에 연결할 수 있는 최대수에 대한 제약 조건이 존재하기 때문에 임의로 염색체를 생성시키는 초기화 과정에서 또는 교차를 통해 생성되는 자손염색체들에서 센터의 최대 연결 가능수에 대한 제약조건을 위반하는 비실현 가능해가 생성된다. 이러한 상황을 다루는 방법은 페널티 방법과 보수 방법 두 가지가 있다. 조합최적화 문제에서 제약조건의 위반정도를 정량적으로 측정하기 쉽지 않을 때에는 비실현 가능 염색체에 대해 적당한 페널티를 부가하는 것은 실제로 힘들다. 이런 경우, 페널티 방법은 결코 실현 가능해를 생성해 낼 수 없고 탐색과정에서 부적합한 로컬 최적해들에 머물러 있게 된다[15]. 따라서 본 논문에서는 비실현 가능해를 수정하여 실현 가능해로 만드는 방법을 이용한다.

$\overline{G}$를 센터 최대 연결가능 수를 위반하지 않은 센터들의 집합이라고 하자. 만약 센터 i가 최대 연결 가능수 g_i제약조건을 위반하였다면, 이것은 센터 i가 염색체내에 $g_i - 1$이상 존재한다는 것을 의미한다. $\overline{G}$에 있는 센터들 중에 하나를 센터 i와 바꾸어서 g_i미만이 되도록 함으로써 수정할 수 있다.

4. 수치 실험

우선 제안한 GA의 성능을 평가하기 위해 평균 메시지 지연을 최소화하는 단일 목적함수를 가진 네트워크 설계 문제에 대하여 조사해 보았다. 이 실험의 결과는 표 1에 정리되어 있다. 표 1의 두 예제를 위한 데이타 설정은 이후에 설명하는 첫 번째 수학모델의 수치예의 설정(표 2와 표 4)과 같게 입력하여 실험하였다. 표 1에서 볼 수 있듯이 제안한 GA는 참고문헌 [16]에서 제안한 Huffman 트리를 이용한 단순한 GA보다 더 좋은 결과를 나타낸다. 결과에서 본래의 스패닝 트리에서 실현가능한 Huffman 트리를 구하는 데에 오버헤더가 있으며 단순한 유전 연산자를 적용한 단순한 알고리즘보다 본 연구에서 제안하는 유전자 표현 및 다양한 유전자를 결합시키는 유전 연산자를 적용한 본 논문의 결과가 더 우수하다는 것을 알 수 있다.

이제 본 논문에서 제안하는 GA를 2기준 네트워크 토

표 1 평균 메시지 지연만을 고려한 수치예의 결과

	n	m	C_i	β_{ij}	GA*	제안한 GA
예제 1	4	8	50	0.1	0.1933	0.09778
예제 2	6	30	300	0.1	0.0329	0.01784

GA*: 참고문헌[16]에서 제안한 GA

폴로지 설계 문제들에 적용하여 실험한 결과에 대해서 살펴보기로 한다. 제안한 방법을 테스트하기 위해 상기의 두 가지 수학모델에 대해 각각 2가지의 수치예에 적용하여 성능을 알아보기로 한다. 먼저 메시지 지연과 연결비용을 목적함수로 다룬 첫 번째 수학 모델에 대하여 살펴보도록 하자.

수학 모델 1(Delay & Cost): 수치예 1

수치예 1은 4개의 서비스 센터($n=4$), 8개의 사용자($m=8$)를 가지고, g_i는 3으로 설정되며, 센터 i와 센터 j간의 링크 가중치(w_{1ij}) 및 센터 i와 사용자 j의 링크 가중치(w_{2ij})는 표 2에 의해 설정되고 센터 i의 통화 용량(C_i)은 50으로 설정되는 예제이다.

표 2 센터 i와 센터 j 사이의 링크 가중치 ($[w_{1ij}]$) 및 센터 i와 사용자 j 사이의 링크 가중치 ($[w_{2ij}]$)

$$[w_{1ij}]=\begin{bmatrix} - & 125 & 110 & 210 \\ & - & 221 & 134 \\ & & - & 123 \\ & & & - \end{bmatrix}$$

$$[w_{2ij}]=\begin{bmatrix} 34 & 62 & 7 & 46 & 26 & 19 & 31 & 29 \\ 26 & 15 & 16 & 68 & 37 & 15 & 43 & 58 \\ 10 & 49 & 86 & 72 & 36 & 31 & 49 & 78 \\ 28 & 22 & 98 & 80 & 8 & 98 & 15 & 4 \end{bmatrix}$$

표 3 제안한 방법과 링크 기반 인코딩 방법과의 비교

수치예 No.	제안한 인코딩 방법		링크 기반 인코딩 방법#	
	ACT (sec.)	Memory Size $n+m-2$	ACT (sec.)	Memory Size $(n+m)(n+m-1)/2$
1	5.07	10	7.12	66
2	32.29	34	*	630

#: 참고문헌[7,8]에서 제안한 인코딩 방법
ACT: 평균 계산 시간
*: 실행 불가능

또한, 센터들의 동작 확률은 0.95로 설정되며, 사용자의 동작 확률은 0.9로, 센터간의 링크는 0.9로, 센터와 사용자간의 링크는 0.85로 설정되고 $R_{\min}$는 0.9로 설정된다.

또한, 수치예 1에 대한 사용자 통화량 행렬 $\boldsymbol{U}$는 다음과 같이 설정된다:

$$\boldsymbol{U}=\begin{bmatrix} 0 & 3 & 3 & 3 & 0 & 0 & 0 & 1 \\ 3 & 0 & 3 & 3 & 0 & 0 & 1 & 0 \\ 3 & 3 & 0 & 3 & 0 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 3 & 3 & 3 \\ 0 & 0 & 1 & 0 & 3 & 0 & 3 & 3 \\ 0 & 1 & 0 & 0 & 3 & 3 & 0 & 3 \\ 1 & 0 & 0 & 0 & 3 & 3 & 3 & 0 \end{bmatrix}$$

GA를 위한 파라미터들은 다음과 같이 설정하였다: 집단 크기는 100 (*pop_size* = 100), 최대 세대수는 500 (*max_gen* = 500), 교차율은 0.3 (p_C = 0.3), 돌연변이율은 0.7 (p_M = 0.7)로 설정하여 총20번 실험하였다. 파레토 최적해 들이 대부분의 경우에서 발견되었고 그 결과는 그림 7에 표시되었다. 그림 7과 표 3에서 볼 수 있듯이 본 논문에서 제안된 인코딩 방법이 Dengiz, Altiparmak, Smith[7,8]에 의해 제안된 트리상의 링크를 이용하여 이차원배열상의 정보로 구성하는 인코딩 방법보다 더 좋은 결과를 나타낸다는 것을 알 수 있다. 그림 7에서는 더 나은 파레토 해집합을 구할 있다는 것을 알 수 있고 표 3에서 제안한 방법이 수치계산 시간과 사용되는 메모리 공간 면에서 더 나은 효율성을 가진다는 것을 알 수 있다.

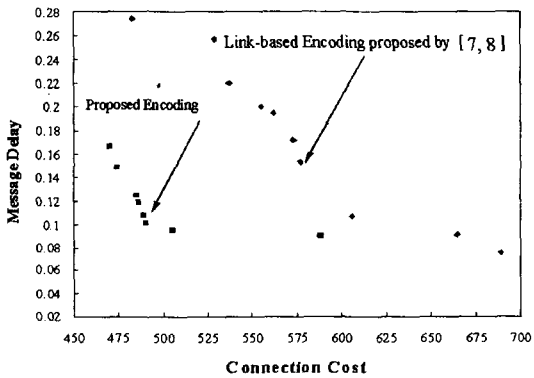


그림 7 수치예 1의 결과(파레토 최적해들)

표 4 센터 i와 센터 j 사이의 링크 가중치 ($[w_{1ij}]$)

$$\begin{bmatrix} - & 125 & 110 & 210 & 221 & 105 \\ & - & 107 & 202 & 154 & 167 \\ & & - & 175 & 163 & 231 \\ & & & - & 217 & 206 \\ & & & & - & 157 \\ & & & & & - \end{bmatrix}$$

수학 모델 1(Delay & Cost): 수치예 2

수치예 2는 6개의 서비스 센터, 30개의 사용자를 가지고, g_i는 10으로 설정되며, 센터 i와 센터 j간의 링크 가중치는 표 4에, 센터 i와 사용자 j의 링크 가중치는 표 5에 의해 설정되고 센터 i의 통화 용량은 300으로 설정되는 예제이다.

또한, 센터들의 동작 확률은 0.95로 설정되며, 사용자의 동작 확률은 0.9로, 센터간의 링크는 0.9로, 센터와 사용자간의 링크는 0.85로 설정되고 $R_{\min}$는 0.9로 설정된다.

표 5 센터 i와 사용자 j 사이의 링크 가중치 ($[w_{2ij}]$)

34	62	7	46	26	19	31	29	26	15	16	68	37	15	100	58	10	49	86	100	36	31	49	78	100	22	98	80	8	98
15	7	80	35	54	85	22	71	81	5	29	46	37	29	79	17	20	39	25	84	5	36	22	12	86	96	9	79	15	54
42	27	25	39	52	90	80	35	58	19	2	34	43	43	51	64	19	36	26	36	16	71	41	51	52	50	13	34	58	73
66	76	84	81	20	45	10	61	34	86	50	18	21	94	25	27	50	61	81	33	54	18	93	7	62	18	75	28	12	37
73	62	34	89	44	85	96	78	7	57	50	43	48	78	25	53	16	45	55	71	11	69	50	93	86	62	18	23	9	73
22	44	24	32	31	3	50	47	76	11	92	63	44	24	30	63	46	66	44	70	23	10	72	7	63	9	17	87	41	64

그리고, 수치예 2에 대한 사용자 통화량 행렬 $\boldsymbol{U}$는 다음과 같이 설정된다:

$$\begin{bmatrix} U_3 & U_1 & O & U_1 & O & U_1 \\ U_1 & U_3 & U_1 & O & U_1 & O \\ O & U_1 & U_3 & U_1 & O & U_1 \\ U_1 & O & U_1 & U_3 & U_1 & O \\ O & U_1 & O & U_1 & U_3 & U_1 \\ U_1 & O & U_1 & O & U_1 & U_3 \end{bmatrix}$$

여기서 O는 영행렬,

$$U_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad U_3 = \begin{bmatrix} 0 & 3 & 3 & 3 & 3 \\ 3 & 0 & 3 & 3 & 3 \\ 3 & 3 & 0 & 3 & 3 \\ 3 & 3 & 3 & 0 & 3 \\ 3 & 3 & 3 & 3 & 0 \end{bmatrix}$$

이다.

GA를 위한 파라미터들은 다음과 같이 설정하였다: 집단 크기는 100, 최대 세대수는 500, 교차율은 0.3, 돌연변이율은 0.7로 설정하여 총20번 실험하였다. 파레토 최적해 들이 대부분의 경우에서 발견되었고 그 결과는 그림 8에 표시되었다.

그림 8과 표 3에서 볼 수 있듯이 본 논문에서 제안된 인코딩 방법이 Dengiz, Altiparmak, Smith[7,8]에 의해 제안된 인코딩 방법보다 더 좋은 결과를 나타내었다는 것을 알 수 있다. 표 3에서 Dengiz, Altiparmak, Smith에 의해 제안된 인코딩 방법은 그 해집합을 찾을 수 없었기 때문에 그림 8에서 비교하여 나타낼 수 없었다. 또한, 그림 9는 그림 8에서 제안한 방법을 통해 구해낸 파레토 해집합들 중에서 최선의 타협 가능한 해의 디코딩한 결과를 나타내었다. 본 논문에서는 Yoon과 Hwang이 제안한 TOPSIS 방법을 이용하여 구해낸 파레토 해들 중에서 최선의 타협해를 찾아내어 표시하였다. TOPSIS는 파레토 해들 중에서 양의 이상적인 해에서 가장 가까우면서 음의 이상적인 해에 가장 먼 해를 찾아서 최선의 타협해를 구하는 방법이다. TOPSIS에 대한 좀더 자세한 정보는 참고문헌 [2]와 [17]에서 찾아볼 수 있다. 그림 9에 표시된 최선의 타협해를 위한 염색체는 다음과 같고 이 염색체의 연결 비용은 1418이며, 평균 메시지 지연은 0.16013이고 네트워크 신뢰도는 0.94758이다.

Centers: 3 3 1 6

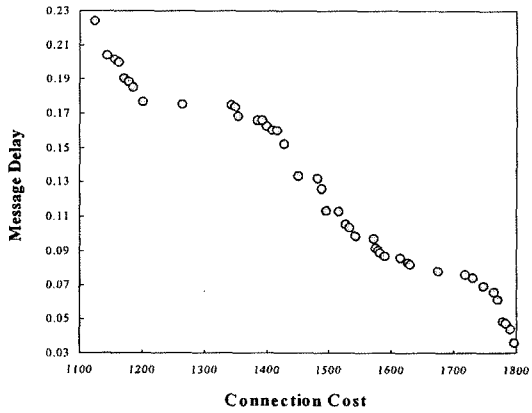


그림 8 수치예 2의 결과(파레토 최적해들)

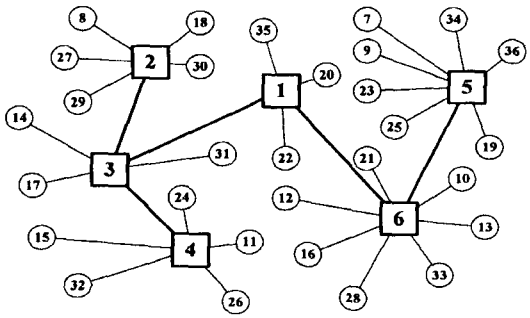


그림 9 수치예 2에 대한 결과 중에 TOPSIS를 찾은 최선의 타협해

Users: 5 2 5 6 4 6 6 3 4 6 3 2 5 1 6 1 5 4 5 4 2 6 2 2 3 4 6 5 1 5

수학 모델 2(Network Reliability & Cost): 수치예 3

수치예 3은 수치예 1과 같이 4개의 서비스 센터, 8개의 사용자를 가지고, g_i는 3으로 설정되며, 센터 i와 센터 j간의 링크 가중치 및 센터 i와 사용자 j의 링크 가중치는 표 2에 의해 설정되는 예제이다.

또한, 센터들의 동작 확률은 0.95로 설정되며, 사용자의 동작 확률은 0.9로, 센터간의 링크는 0.9로, 센터와 사용자간의 링크는 0.85로 설정된다.

GA를 위한 파라미터들은 다음과 같이 설정된다: 집

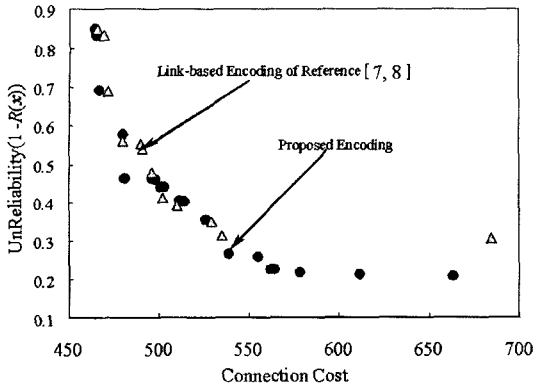


그림 10 수치예 3의 결과(파레토 최적해들)

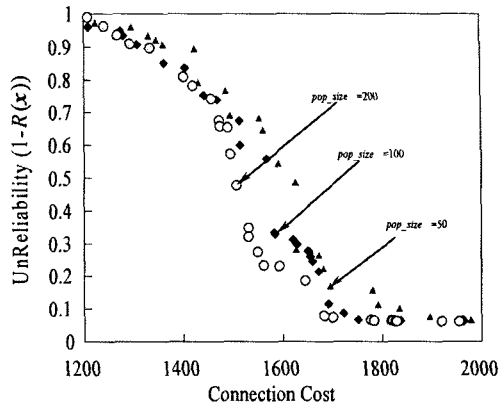


그림 11 수치예 4의 결과(파레토 최적해들)

단 크기는 100, 최대 세대수는 500, 교차율은 0.3, 돌연변이율은 0.7로 설정하여 총20번 실험하였다. 파레토 최적해 들이 대부분의 경우에서 발견되었고 그 결과는 그림 10에 표시하였다.

수치 실험의 결과를 그림 10과 표 6에서 정리하였다. 결과를 살펴 볼 때 본 논문에서 제안된 인코딩 방법이 Dengiz, Altiparmak, Smith[7,8]에 의해 제안된 인코딩 방법보다 더 좋은 결과를 나타내었다.

표 6 제안한 방법과 링크 기반 인코딩 방법과의 비교

수치예 No.	제안한 인코딩 방법		링크 기반 인코딩 방법#	
	ACT (sec.)	Memory Size $n+m-2$	ACT (sec.)	Memory Size $(n+m)(n+m-1)/2$
3	5.16	10	9.37	66
4	33.38	34	*	630

#: 참고문헌[7,8]에서 제안한 인코딩 방법
ACT: 평균 계산 시간
*: 실행 불가능

수학 모델 2(Network Reliability & Cost): 수치예 4

수치예 4는 6개의 서비스 센터, 30개의 사용자를 가지고, g_i는 10으로 설정되며, 센터 i와 센터 j간의 링크 가중치는 표 4에, 센터 i와 사용자 j의 링크 가중치는 표 5에 의해 설정되는 예제이다.

또한, 센터들의 동작 확률은 0.95로 설정되며, 사용자의 동작 확률은 0.9로, 센터간의 링크는 0.9로, 센터와 사용자간의 링크는 0.85로 설정되고 $R_{\min}$는 0.9로 설정된다.

GA를 위한 파라미터들을 다음과 같이 설정한다: 집단 크기는 각각 50, 100, 200으로, 최대 세대수는 500, 교차율은 0.3, 돌연변이율은 0.7로 설정하여 총20번 실험하였다. 파레토 최적해 들이 대부분의 경우에서 발견되었고 그 결과는 그림 11에 표시되었다. 또한, 수치 실험 1의 수치예 2의 결과와 마찬가지로 표 6에서 나타낸 것처럼 트리의 링크 정보를 이용한 이차원 배열로 구성되는 Dengiz, Altiparmak, Smith의 인코딩 방법은 그 결과를 구해 낼 수 없었다.

그림 11에서 집단 크기가 200일 경우 최상의 결과를 얻을 수 있었다. 더 많은 수의 유전 집단을 이용하면 더 다양한 유전 탐색을 할 수 있다고 결과를 통해 알 수 있었다. 또한 TOPSIS를 이용하여 최선의 타협해를 구한 결과는 다음과 같다.

Centers: 6 5 6 2

Users: 5 5 5 6 6 4 6 4 5 2 4 2 1 6 6 2 1 2 5 4 6 5 3 6 3 1 2 5 1 4

이 염색체의 연결 비용은 1595이며 네트워크 신뢰도는 0.79654이다.

5. 결 론

본 논문에서는 2개의 목적함수와 몇몇의 제약조건을 가진 2기준 네트워크 토폴로지 설계 문제를 풀기 위해 PN과 클러스터 문자열로 구성되는 새로운 인코딩 방법을 이용한 유전자 알고리즘을 제안하였다. 또한 파레토 해들을 찾기 위한 방법도 이용하였다. 실험 결과 제안한 알고리즘은 대부분의 경우 2개의 목적을 고려한 파레토 해들을 찾아 낼 수 있었다.

본 논문에서는 통신 네트워크 상의 스패닝 트리를 표현하는 염색체를 나타내기 위해 PN과 클러스터 문자열이 이용 되었으며 타협해들을 얻기 위해 파레토 해들의 저장 기술도 이용되었다. 또한, 얻어진 타협해들로부터 최선의 타협해를 구하기 위해 TOPSIS방법이 이용되었다.

제안한 알고리즘의 효과를 예증하기 위해 본 논문에서는 2종류의 문제에 대해 각각 2개의 예제를 들어 실험을 실시하였다. 또한 제안한 인코딩 방법의 효율성을 나타내기 위해 Dengiz, Altiparmak, Smith에 의한 링

크 정보를 이용한 이차원배열로 구성되는 인코딩 방법과 비교 실험하였다. 표 3과 표 6에서 볼 수 있듯이 링크 기반 인코딩 방법은 작은 크기의 문제에서는 해를 찾을 수 있으나 큰 규모의 문제에서는 그 해를 찾을 수 없었다. 따라서, 제안한 알고리즘은 계산 시간뿐만 아니라 파레토 최적성의 관점에서도 보다 효율적이며 효과적인 방법이라고 할 수 있다.

스패닝 트리를 나타내는 유전자 표현에 대한 최근 연구 보고에 따르면 PN이 유전자 알고리즘의 인코딩으로서 가져야 되는 유전 연산후의 부모 염색체와의 연관성의 약화를 가져와서 결점이 있다고 한다[20]. 따라서 본 논문의 유전자 알고리즘에서 사용하는 PN의 이런 결점을 극복할 수 있는 방향으로 연구를 계속 진행해 나가야 할 것이며 아울러 더 빠른 계산 시간을 위한 궁리도 계속되어야 할 것이다.

참 고 문 헌

[1] Mukherjee, B., *Optical Communication Networks*, McGraw-Hill, Inc., New York, 1997.

[2] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.

[3] Gen, M. and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, 2000.

[4] Gen, M. and Y. Z. Li, "Spanning tree-based genetic algorithm for bicriteria transportation problem," *Proceedings of Japan-China Joint International Workshops on Information Systems*, pp.123-134, 1997.

[5] Zhou, G., and M. Gen, "Evolutionary computation on multicriteria production process planning problem," *Proceedings of IEEE International Corference on Evolutionary Computation*, pp.419-424, 1997.

[6] Zhou, G., and M. Gen, "Approach to degree-constrained minimum spanning tree problem using genetic algorithm," *Engineering Design & Automation*, Vol.3, No.2, pp.157-165, 1997.

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

[8] Dengiz, B., F. Altiparmak, and A. E. Smith, "Efficient optimization of all-terminal reliable networks using evolutionary approach," *IEEE Transaction on Reliability*, Vol.46, No.1, pp.18-26, 1997.

[9] Bertsekas, D. and R. Gallager, *Data Networks*, 2nd ed., Prentice-Hall, New Jersey, 1992.

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

[11] Gen, M., K. Ida, J. R. Kim, and J. U. Lee, "Bicriteria Network Design using Spanning Tree-based Genetic Algorithm," *Proceedings of 3rd International Symposium on Artificial Life and Robotics*, Vol.1, pp.43-46, 1998.

[12] Skiena, S., *Implementing Discrete Mathematics Combinatorics and Graph Theory with Mathematica*, Addison-Wesley, Reading, MA, 1990.

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

[14] Kershenbaum, A. and R. Van Slyke, "Recursive analysis of network reliability," *Networks*, Vol.3, pp.81-94, 1973.

[15] Orvosh, D. and L. Davis, "Using a genetic algorithms to optimize problems with feasibility constraints," *Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE Press, Orlando, Florida, pp.548-552, 1994.

[16] Elbaum, R. and M. Sidi, "Topological design of local-area networks using genetic algorithms," *IEEE/ACM Transactions on Networking*, Vol.4, No.5, pp.766-778, 1996.

[17] Hwang, C. and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Berlin, 1981.

[18] Zhou, G and M. Gen, "A Genetic Algorithm Approach on Tree-like Telecommunication Network Design Problem," *The Journal of the Operational Research Society*, Vol.54, No.3, pp.248-254, 2003.

[19] Gen, M. and R. Cheng, "Evolutionary Network Design: Hybrid Genetic Algorithms Approach," *International Journal of Computational Intelligence and Applications*, Vol.3, No.4, pp.357-380, 2003.

[20] G. R. Raidl and B. A. Julstrom, "Edge Sets: An effective evolutionary coding of spanning trees," *IEEE Transaction on Evolutionary Computation*, Vol.7, No.3, pp.225-239, June, 2003.

김 종 율
정보과학회논문지 : 소프트웨어 및 응용
제 32 권 제 4 호 참조

이 재 욱
정보과학회논문지 : 소프트웨어 및 응용
제 32 권 제 4 호 참조

현 광 남
정보과학회논문지 : 소프트웨어 및 응용
제 32 권 제 4 호 참조