論文
6-3-16

PS-NC GA를 이용한 최적 LAN 설계
Optimal LAN Design Using a Pareto Stratum-Niche Cubicle Genetic Algorithm

최강희(Choi Kang Hee)[1] 정경희(Jung Kyoung Hee)[2]

요 약

본 논문은 연결비용 및 메시지 지연시간을 최소화하는 최적 LAN 설계를 위한 파레토 계층-적소영역 유전자 알고리즘의 응용이다. 이러한 연구과제를 수행하기 위하여 LAN 설계를 위한 적합한 가능해의 표현 방법을 제안하고, LAN 설계 시 선택되는 여러 위상(topology) 형태는 지역적이고, 내선망에 가장 보편적으로 쓰이는 스패닝 트리 형태에 한정하여 유전자 알고리즘을 적용한다. 스패닝 트리를 초기 모집단의 염색체로 표현하기 위해 Prüfer 수 개념을 이용한다. Prüfer 수는 간결하면서도 스패닝 트리의 특징을 잘 반영하며 염색체 표현에 적합하다. 유전자 알고리즘에 의해 이중 목적에 맞는 적응도 평가함수의 산출하고, 파레토 계층-적소영역 선별 알고리즘을 사용하며, 적합한 유전연산자를 적용함으로써 다양한 해공간을 탐색하여 최적 LAN을 하여, 제안되었던 알고리즘이 꽤 짧은 시간에 다목적 LAN 설계 문제의 좋은 해답들을 제공할 수 있는 것으로 나타났다.

ABSTRACT

The spanning tree, which is being used the most widely in indoor wiring network, is chosen for the network topology of the optimal LAN design. To apply a spanning tree to GA, the concept of prüfer numbers is used. Prüfer numbers can express he spanning tree in an efficient and brief way, and also can properly represent the characteristics of spanning trees.

This paper uses Pareto Stratum-Niche Cubicle(PS-NC) GA by complementing the defect of the same priority allowance in non-dominated solutions of pareto genetic algorithm(PGA). By applying the PS-NC GA to the LAN design areas, the optimal LAN topology design in terms of minimizing both message delay time and connection-cost could be accomplished in a relatively short time. Numerical analysis has been done for a hypothetical data set. The results show that the proposed algorithm could provide better or good solutions for the multi-objective LAN design problem in a fairly short time.

논문접수 : 2005. 7. 1.
심사완료 : 2005. 8. 1.

1) 정회원 : 대원과학대학 정보처리전공 교수

2) 정회원 : 대원과학대학 겸임교수

1. 서 론

새로운 네트워크를 구축하거나 확장, 재구성할 때 당면하는 과제는 노드간의 최단거리를 찾아 연결 설치비용을 최소화하거나, 사용자가 네트워크를 사용함에 있어 허브나 라우터를 통해 서버와 접속할 수 있는 지연시간을 최소화하는 문제 등으로 요약될 수 있다.

연구 과제를 수행하기 위하여 LAN 설계를 위한 적합한 가능해의 표현 방법을 제안하고, LAN 설계 시 선택되는 여러 위상(topology) 형태는 지역적이고, 내선망에 가장 보편적으로 쓰이는 스패닝 트리 형태에 한정하여 유전자 알고리즘을 적용한다. 유전자 알고리즘은 네트워크 최적화 설계문제를 위한 최적화 기술로서 알고리즘의 가능성에 관해서 많은 주목을 받았고, 많은 실세계 문제를 해결하는데 자주 사용되어 왔다. 최적 LAN 위상의 설계는 매우 복잡한 조합 최적화 문제이며, 그 위상 설계 문제가 두 가지 목적을 가질 때에는 LAN의 설계가 더 복잡해진다.

본 논문의 목적은 단일문제가 아닌 다중 문제에서의 최적해를 찾고자 함에 있다. 서로 상충(trade-off)되어지는 다목적 문제에서는 여러 개의 해가 존재할 수 있는데, 다목적 해를 찾기 위한 여러 가지 방법 중에서 파레토 계층-적소영역(Pareto Stratum-Niche Cubicle) 개념의 유전자 알고리즘을 사용한다.

기존의 파레토 알고리즘은 비 지배되는 최적해들의 우선 순위를 같게 한 데 반하여, 파레토 계층-적소 영역 유전자 알고리즘은 파레토 해들의 계층별 적소영역의 밀도에 따라 우선 순위를 모두 다르게 줌으로써 다음 세대에 다양한 해공간을 탐색하여 보다 좋은 해를 얻을 수 있다. 본 논문에서 응용한 파레토 적소 개념의 유전자 알고리즘은 Hyun과 Kim이 '혼합 모델 라인 균형화 해법'에 적용하여 다중 목적 최소화 문제를 해결한 바 있다[1].

다중 목적 해결을 위한 파레토 적소영역 유전자 알고리즘은 기존의 파레토 알고리즘을 적용하였을 때 보다 좋은 결과를 보여주고 있으며, 이 알고리즘을 최적 LAN 설계에 적용하고자 한다.

2. 유전자 알고리즘

유전자 알고리즘(genetic algorithm)은 생물의 진화과정, 즉 자연선별(natural selection)과 유전법칙을 모방한 확률적 탐색기법이다. 이 알고리즘은 1975년 홀랜드(Holland)의 논문 "Adaptation in Natural and Artificial Systems"에서 처음으로 소개되었다. 그는 자연 시스템의 한 메커니즘으로 생물의 진화과정을 추상화하여 인공시스템을 설계하고자 했다. 그 후 20여 년 동안 유전 알고리즘의 이론과 응용에 관하여 활발한 연구가 이루어져 왔다[2][3-5].

유전자 알고리즘은 좋은 해의 이용능력과 해 공간의 탐색능력이 적절히 조화되어야 한다. 좋은 해의 이용만을 강조하면 조기 수렴하여 부분 최적에 빠질 수 있고 해 공간의 탐색만을 강조하면 임의 탐색에 가까워 좋은 해를 찾아가지 못하게 된다. 유전자 알고리즘은 매 세대마다 모집단을 운용함으로써 내재된 병렬성(parallelism)을 갖는 기법이다.

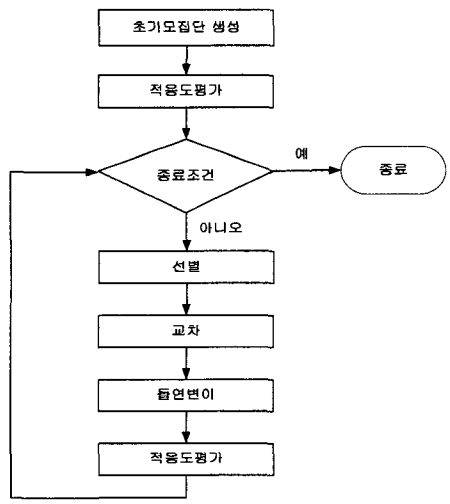
특히 유전자 알고리즘은 복잡한 해 공간의 탐색성능이 우수하여 변수와 제약이 많은 대형 수리문제를 푸는데 적합한 기법이고, 또한 모형에 대한 유연성이 높아 제약 첨가나 목적함수의 변경이 용이하다는 장점을 갖는다[2][3-5].

유전자 알고리즘은 자연계에 있는 생물의 진화과정을 공학적인 접근방법으로 모형화하여 복잡한 실세계의 문제를 해결하고자 하는 방법이다. 이 방법은 적응적 탐색과 학습 및 최적화를 통한 공학적인 문제의 해결에 많이 이용되고 있다. 유전자 알고리즘은 확률적 계산모형으로서 전역적(global)인 최적해를 병렬적으로 탐색하는 기법이다. 유전자 알고리즘에서는 대상이 되는 문제의 후보해를 문자나 기호들의

배열인 염색체(chromosome)로 표현한다. 이 절차를 코드화(encoding)라고 하고 염색체를 구성하는 문자나 기호들을 유전자(element or gene)라고 한다. 염색체를 개체(individual)라 언급하며, 이들 개체들의 적당한 크기(population size)로 구성되는 모집단(population)을 생성하여, 이들을 점차적으로 갱신함으로써 전역적인 최적해를 탐색하는 것이다. 모집단의 생성과정을 초기화라 하며 모집단의 갱신은 유전연산자(genetic operators)에 해당되는 선별(selection), 교차(crossover), 돌연변이(mutation)의 세 가지 과정을 통하여 이루어지고 이를 1세대라고 한다.

유전자 알고리즘은 해의 탐색과정에서 모집단을 이용한 병렬처리를 수행하기 때문에 탐색 소요시간이 매우 짧다. 모집단의 갱신과정에서 개체의 수를 유지하므로 다목적 최적화 문제의 다양한 파레토 최적해를 탐색하기에 매우 적절한 기법이며 유전자 알고리즘은 [그림 1]과 같다.

초기화되어진 모집단의 각 개체에 대해 적응도(fitness)가 평가되어지며, 이는 각 개체의 우수성을 판정하기 위한 것이다.



[그림 1] 유전자 알고리즘

선별(selection)이란 현재의 모집단을 갱신시키기 위한 준비단계로서, 지정된 전략에 따라 우수한 개체를 다음 세대로 복제(reproduction)시키거나 교배를 위한 개체의 쌍을 만들기 위해 현재의 모집단을 재편성하는 과정이다. 교차(crossover)는 두 부모가 갖는 유전자를 조합하여 자손을 생산하는 과정이다. 교차는 좋은 해를 이용하는 역할을 한다. 이를 위해서 교차는 부모의 좋은 형질이 가능한 파괴되지 않고 자손에게 상속될 수 있어야 한다. 교배율이 높으면 계산 시간이 많이 걸리고 불필요한 해의 공간을 반복해서 탐색하게 된다. 반면에 교배율이 낮으면 충분한 공간을 탐색하지 못하여 최적해를 찾지 못하게 된다. 돌연변이(mutation)는 미지의 해의 공간을 탐색하는 것으로 이는 국부적인 최적해로 수렴하는 것을 방지하는 역할을 한다. 돌연변이는 완전히 새로운 유전자를 생성하기 때문에 이를 '진화의 연료'라 하기도 한다. 돌연변이는 대부분 해로운 것이다. 해로운 돌연변이는 자손들의 적응을 감퇴시켜 자연선별에서 제거된다. 그러나 때때로 유익한 돌연변이가 일어나며, 이런 돌연변이에 의해 개체는 환경에 잘 적응하여 유사한 자손을 번식시킨다. 유전자 알고리즘에서 돌연변이 연산자의 역할은 해의 다양성을 증대시켜 준다. 돌연변이율(mutation rate)이 낮으면 유용한 해의 공간을 탐색하지 않아 국부 최적해로 빠질 수 있고, 너무 높으면 부모세대의 장점을 다음 세대에서 흡수하지 못하여 기존에 탐색한 영역에서 좋은 해를 손실할 수 있다. 그러므로 적정한 교배율과 돌연변이율의 선정이 유전자 알고리즘의 성능에 민감한 영향을 미친다.

가. 2.1 스패닝 트리

스패닝 트리는 망의 링크들 중 일부만을 선택하여 이루어진 트리 구조체로써 루프가 없는 그래프 구조이고, 두 라우터간에는 단지 하나의 연결 경로만 존재하는 형태이다.

스패닝 트리가 구축되면 라우터는 패킷을 수신한 링크만을 제외한 스패닝 트리에 속하는 다른 모든 링크로 해당 패킷을 전진시키기만 하

면 된다. 스패닝 트리의 구축을 통하여 루프없이 모든 라우터들에게 패킷을 전송할 수 있다. 스패닝 트리 알고리즘은 네트워크 루프에 따른 문제를 해결하는 IEEE802.1d로 표준화 되어있다[6]. 데이터를 전송하기 전에 특수한 연결 패킷을 이용하여 네트워크 상에 루프가 존재하는지 여부를 검색한다. 만약 루프가 있다면, 루프를 형성하는 하나 또는 두개의 링크를 논리적으로 블록상태로 설정하여 패킷을 전송하지 못하도록 한다. 만약 스패닝 트리 알고리즘을 사용하는 연결 구성 네트워크 상에서 어느 한 링크의 중단으로 인하여 데이터 전송이 중단되는 경우가 발생하면, 자동적으로 블록화 되어있는 링크가 전송할 수 있는 상태로 변경된다. 스패닝 트리 알고리즘에서의 링크 블록은 해당 링크에서 연결 서비스를 이용하여 데이터를 전송하지 못하는 것이지, 물리적으로 커넥션이 끊어진 상태를 의미하는 것은 아니다. 따라서 IP또는 IPX 라우팅 환경에서는 여전히 블록화된 링크를 사용할 수 있다.

나. 2.2 Prüfer 수의 표현

유전자 알고리즘을 위한 해의 암호화(encoding)는 표현된 암호를 통해 해의 최적성 및 가능해 여부를 충분히 평가할 수 있도록 해야 함과 동시에 유전자 알고리즘에 효율적이도록 간결해야 한다. 본 논문에서는 네트워크의 노드들의 경로를 표현하기 위해 Prüfer가 소개한 Prüfer 수를 사용한다. Prüfer 수는 기존 연구들에서 스패닝 트리를 암호화하는데 사용되어 왔다. Prüfer 수는 n개의 노드를 가지는 트리를 나타내는데 $(n-2)$자리의 일반 문자열을 필요로 한다. Prüfer는 0과 $(n-1)$사이의 정수로 이루어진 $(n-2)$자리의 Prüfer 수와 n개의 노드를 가지는 트리 간에 일대일 대응관계가 있음을 증명하였으며 이것은 각 노드번호가 0과 $(n-1)$사이의 정수인 n개의 노드를 가지는 트리를 유일하게 표현하는 $(n-2)$자리의 Prüfer 수가 존재함을 뜻한다[7-8]. Zhou는 차수제한이 있는 최소 스패닝 트리 문제에서 스패닝 트리를 암호화하는데 Prüfer 수를 사용하였다. Prüfer 수가 모든 가능한 트리를 나타낼 수 있으며 특히 차수에 대한 정보를 가지며 차수의 확인 방법 또한 간단하기 때문에 차수제한이 있는 스패닝 트리 문제에 매우 유용하다. 실험 결과도 전통적인 발견적 해법에 견줄 만큼 우수하다고 언급하고 있다[9]. Prüfer 수는 수송문제, 최소 스패닝 트리문제 등과 같은 분야에 더욱 적당하다. Gen은 그의 논문에서 prüfer 수의 우수함을 증명하였다[10].

스패닝 트리를 Prüfer 수로 표현하기 위한 인코딩(encoding) 알고리즘은 [그림 2]와 같다.

단계 1 : 트리 T에서 가장 작은 잎 노드를 i로 놓자.

단계 2 : 인코딩의 처음 자리 수를 j로 놓자.

노드 j는 노드 i를 유일하게 결정지어 준다.

인코딩은 왼쪽에서 오른쪽으로 생성되며, 읽혀진다.

단계 3 : 노드 i와 노드 i에서 j로의 링크를 삭제하면, 트리의 수가 $n-1$ 개가 된다.

단계 4 : 하나의 링크가 남을 때까지 단계 1에서 단계 3을 반복한다. Prüfer 수는 1에서 n사이에서 $n-2$개의 자리 수만큼 인코딩된다.

[그림 2] 인코딩 알고리즘

Prüfer 수로 표현된 것을 다시 트리 형태로 표현하기 위한 디코딩(decoding)알고리즘은 그림 3과 같다.

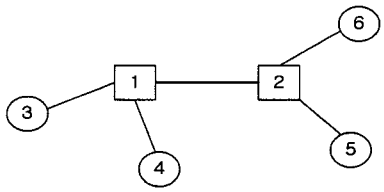
단계 1 : Prüfer 수로 표현된 집합을 P로, 트리를 구성할 수 P에 포함되지 않은 모든 노드들을 P로 놓자.

단계 2 : P의 노드들 중에서 순서번호가 가장 작은 노드를 i라 하고, P에서 가장 왼쪽에 위치하는 노드를 j라 하자.

단계 3 : i와 j를 연결하는 노드를 생성하고, P의 i와 P에서의 j를 제거한다. P에 더 이상 남아 있는 노드가 없을 때까지 [단계 3]을 반복한다.

단계 4 : 만약 P에 더 이상 남아있는 노드 i가 없을 경우,P에 현재 P의 노드 j를 추가하고, P의 링크는 $k-1$로 갱신한다.

[그림 3] 디코딩 알고리즘



센터 = { 1, 2 }

사용자 = { 3, 4, 5, 6 }

Prüfer 수 = | 1 | 1 | 2 | 2 |

[그림 4] 스패닝 트리와 Prüfer 수

그림 4의 예에 디코딩 알고리즘을 적용하여 살펴보면, [단계 1]에서 Püfer 수는 P = {1, 1, 2, 2} 이고, 노드 Ṕ = {3, 4, 5, 6}이다.

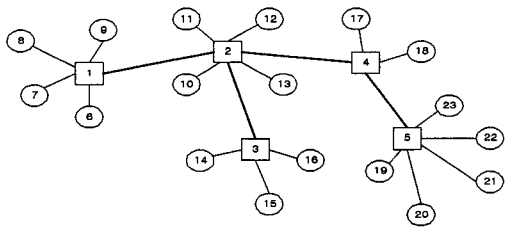
[단계 2]에서 Ṕ의 가장 작은 노드는 3이고, P의 가장 왼쪽에 위치하는 노드는 1이다. [단계 3]에서 {3, 1}을 연결하는 노드를 생성하고, Ṕ와 P에서 각각 3과 1을 제거한다. 남아있는 P의 노드는 {1, 2, 2} 이다.

Ṕ에서 가장 작은 노드는 4이고, P에서 가장 왼쪽에 위치하는 노드는 1이다. 트리에 {4, 1}을 연결하는 노드를 생성하고, Ṕ와 P에서 각각 4와 1을 제거한다. 남아있는 P의 노드는 {2, 2}이다. P에 남아있는 노드 1이 더 이상 없으므로, [단계 4]에 의해 Ṕ에 노드 1을 추가한다. 이제 Ṕ = {1, 5, 6}이다.

Ṕ에 가장 작은 노드는 1이고, P에서 왼쪽에 위치하는 노드는 2이다. {1, 2}를 연결하는 노드를 생성하고, Ṕ와 P에서 각각 1과 2를 제거한다. P에 남아있는 노드는 {2}이다.

[단계 3]을 반복하면, Ṕ의 5와 P의 2의 노드를 생성하고 각각을 Ṕ와 P에서 제거한다. [단계 4]에 의해 P에 더 이상의 2노드가 없으므로 Ṕ에 2를 추가한다.

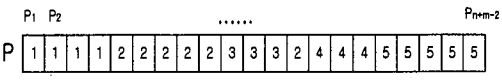
따라서, Ṕ = {2, 6} 이고, 2와 6노드를 트리에 추가한다. 위의 디코딩 과정이 끝나면 스패닝 트리가 형성된다. 좀 더 복잡한 스패닝 트리형태는 그림 5와 같다.



(□ : 센터, ○ : 최종 사용자)

[그림 5] 스패닝 트리 형태의 예

그림 5의 스패닝 트리를 Püfer수로 표현한 것이 그림 6이다.



$P_i \in \{ 1, 2, 3, 4, 5 \}$, $i = 1, 2, ..., n+m-2$

Ṕ = { 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 }

[그림 6] 스패닝 트리[그림 5]의 Prüfer 수

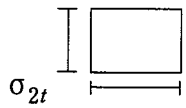
prüfer 수의 모든 숫자는 센터의 번호를 나타냄을 알 수 있다.

염색체의 초기화는 $n+m-2$ 크기만큼 [1, n]의 범위 안에서 임의로 생성되므로 사용자는 Prüfer 수에 나타나지 않고 단지 고정된 집합 Ṕ의 요소로 사용되었으며, Prüfer 수는 센터의 번호로 대표되었다. 센터들은 1에서 n까지의 수로 표현하고, 사용자들은 $n+1$에서 $n+m$까지의 수로 표현된다.

다. 2.3 파레토 계층-적소영역 유전자알고리즘

파레토 유전자 알고리즘에서는 순위가 동일한 개체들이 동일한 적응도를 갖기 때문에 다양한 파레토 최적해를 구하는데 한계가 있다. 이러한 문제를 해결하기 위하여 파레토 유전자 알고리즘을 보완하여 개체간의 지배관계에 대한 순위를 파레토 계층-적소영역(Pareto Stratum-Niche Cubicle)개념을 도입한 Hyun과

Kim의 알고리즘을 이용하였다[1]. 이하 이 알고리즘을 PS-NC 유전자 알고리즘이라고 부른다. 각 파레토 해의 개체에 적소반경(niche radius)을 이용하여 적소영역의 크기를 다음과 같이 정하여 준다.



$$\sigma_{it} = \frac{MAX_{it} - MIN_{it}}{\sqrt[n]{Pop_size}}$$

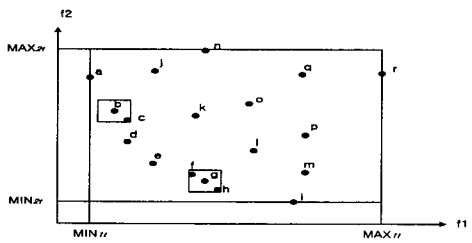
여기서, Pop_size : 모집단크기

MAX_{it} : t세대의 i번째 최대값

MIN_{it} : t세대의 i번째 최소값

n : 목적 개수

문제에서는 n개의 목적을 가지고 있다고 생각한다. 본 논문에서는 설치비용과 메시지 지연시간 등 두 가지 목적이므로 $n = 2$이다. MAX_{it}와 MIN_{it}는 t세대의 i번째의 최대값과 최소값이다. 적소크기는 먼저 한 세대에서 모든 개체들을 대해 구성된다. 하나의 개체의 적소 크기는 해를 중심으로 하는 직사각형의 영역이다. 그림 7과 같이 파레토 해 a, b, c, d, e, f, g, h, i를 중심에 위치시켜 적소 크기 반경 내에 해당하는 해의 수에 따라 순위가 정해진다. 적소 크기에 따라 적소 반경 내에 위치하는 파레토 해의 수가 적을수록 우선 순위는 높고, 다음 세대에 유전될 확률도 따라서 높아진다.



[그림 7] 적소영역을 적용한 모습

이상 9개의 파레토 해의 순위가 정하여 지면, 그 해들을 제외한 나머지 파레토 해들에 대해 다시 적소영역을 정하고 연속된 순위를 정하여 준다. 파레토 계층-적소영역 알고리즘에서는 비 지배적 순위가 높은 개체의 생존확률을 높게 부여하고, 높은 생존확률을 가짐으로써 다양하고 좋은 해를 효율적으로 탐색할 수 있게 한다.

본 논문에서 사용한 PS-NC 선별 알고리즘은 [그림8]과 같다.

단계 1 : 현재 모집단의 모든 개체들에 대해 적소영역 크기를 계산한다.

단계 2 : 현재 모집단에서 비지배되는 개체들을 하나의 그룹으로 둔다

단계 3 : 그룹내의 비지배되는 모든 개체에 대해 적소영역 내의 개체수를 구하고, 적소영역내의 개체수가 가장 작은 개체부터 차례로 나열하여 순위를 1, 2, ..., N_1로 부여한다.

단계 4 : 이 그룹을 제외한 후, 다시 모집단에서 비지배되는 개체들 그룹을 찾는다.

단계 5 : 다시 적소영역내의 개체 수를 구하고, 적소영역내의 개체수가 적은 개체로부터 나열하여 순위를 N_1+1, N_1+2, ..., N_2로 부여한다. 이러한 과정을 모집단의 모든 개체들에 순위가 부여될 때까지 반복한다.

단계 6 : 각 개체가 선별될 확률은 다음과 같이 기하분포로준다.

$$\Pr ob(rank) = q(1-q)^{rank-1}$$
$$rank = 1, 2, 3, \ldots, N_p$$

q는 선별 파라메터로써, 범위는 $0 < q < 1$ 이다.

[그림 8] 파레토 계층-적소영역 알고리즘

3. PS-NC 유전자 알고리즘의 구현

본 논문에서는 메시지 지연시간과 연결비용을 기준으로 한 스패닝 트리 LAN을 공식화하기

위한 기호 및 수식을 다음과 같이 정의한다.

n 서비스 센터의 수

m 연결된 사용자의 수

U=[u_{ij}] m×m 사용자 통신 트래픽 배열

T=[t_{ij}] n×n 센터 i에서 j로의 서비스 센터 트래픽 배열

X_1 n×n 서비스 센터 위상 행렬

x_{1ij}=⎧ 1 , 센터i에 센터j가 연결되어 있으면
⎨
⎩ 0 , 그렇지 않으면

X_2 n×m 클러스터링 행렬

x_{2ij}=⎧ 1 , 센터i에 사용자j가 연결되어 있으면
⎨
⎩ 0 , 그렇지 않으면

X n×(n+m) 스패닝트리 행렬 ([X_1, X_2])

$a_{ij}^{k}(X)$=

⎧1 , 센터i에서k를 통과하여 센터j로의 트래픽 이 존재하면
⎨
⎩ 0 , 그렇지 않으면

$b_{ij}^{(k,l)}$ =

⎧1 , 센터 i에서 k와 l을 통과하여 센터 j로의 트래픽이 존재하면
⎨
⎩ 0 , 그렇지 않으면

C_k 센터 k의 트래픽 수용량(capacity)

β_{ij} 센터i와 j사이의 연결된 비트당 지연시간

g_i 센터i에 연결될 수 있는 최대 사용자 수

라. 3.1 구현 과정

(1) 초기화

초기 모집단은 임의로 생성한다. 이러한 임의적인 생성에 의해 해 공간이 전체영역에 대해 고르게 샘플링되는 효과를 기대할 수 있다. 모집단의 염색체 초기화는 [$n+m-2$] 자리 수만큼 [1, n]범위의 숫자를 임의로 생성하여 초기해로 이용한다.

(2) 적응도 평가

인코딩된 해는 적절한 해석 및 평가과정을 통해 그 우수성을 평가한다. 해의 해석을 통해 개체해의 가용성을 검사하고, 적합도를 평가하기 위한 적절한 적합도 함수를 설계해야 한다. 본 논문에서 사용할 적응도 평가(fitness) 함수로는 센터 당 메시지 지연시간을 최소화하는 함수식과 설치비용을 최소화하는 함수식을 사용한다[10, 11].

$$\min\ c_1(X) = \frac{1}{\Gamma}\left[\sum_{i}^{n}\frac{l_i(X)}{C_i - l_i} + \sum_{i=1}^{n}\sum_{j=1}^{n}\beta_{ij}\cdot f_{ij}(X)\right]$$

Γ : 센터와 사용자, 사용자와 사용자간의 통신 트래픽의 전체 합

C_i : 센터 i의 트래픽 수용량

l_i : 현재 센터 i의 통신 트래픽의 합

β_{ij} : 센터 i와 센터 j사이의 메시지 지연시간

$$\min\ c_2(X) = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{1ij}\cdot x_{1ij} + \sum_{i=1}^{n}\sum_{j=1}^{m} w_{2ij}\cdot x_{2ij}$$

w_{1ij} : 센터와 센터사이의 설치비용

w_{2ij} : 센터와 사용자사이의 설치비용

x_{1ij} : 센터와 센터사이의 연결상태

x_{2ij} : 센터와 사용자간의 연결상태이다.

(3) 선별

유전자 알고리즘은 현 단계의 모집단으로부터 다음 단계의 모집단을 구성하기 위하여 현 단계의 모집단에서 적합도에 근거한 평가함수에 따라 유전자 조합과정에 참여할 부모해를 확률적으로 선별한다. 이러한 부모해의 선별은 해의 우수성에 확률적으로 비례하는 샘플링의 시도를 의미한다. 유전자 알고리즘에서 모집단을 선별하는 샘플링 기법은 여러 가지 방법이 알려져 있다. 일반적으로 비교적 단순한 형태의 추계적 복원 샘플링(stochastic sampling with replacement)기법과 추계적 비복원 샘플링(stochastic sampling without replacement)기법이 사용되고 있다. 추계적 복원 샘플링 기법의 대표적인 방법이 확률바퀴(roulette wheel) 방법이다[8].

(4) 유전연산자

세대 교체를 위해 확률적 샘플링에 의해 선발된 부모해는 다음 세대를 준비하기 위해 복제 및 유전자 재 조합과정을 거쳐 자손해를 구성한다. 일반적으로 부모해를 결합하여 유전 정보를 재 조합하기 위하여 복제, 상호교차 및 돌연변이 연산자를 사용하고 있다. 또한, 유전자 알고리즘에서는 암호화 기법에 따라 유전자의 재 조합과정에 사용되는 연산자가 서로 달라질 수 있는데, 이는 동일한 재조합 연산자라 하더라도 각 암호화 기법의 특성에 따라 다른 탐색 및 수렴성능을 가지기 때문이다[8].

① 교차 연산자

교차의 종류에는 일점(one-point)교차, 이점(two-point)교차, 순서(order)교차 등이 있으며, 본 논문에서는 이점교차(two-point crossover)를 선택하였다.

일점 교차가 하나의 절단점을 갖는데 반하여 이점교차는 두 개의 절단점을 갖는다.

원소 사이의 두 절단점은 임의로 잡는다. 이점교차는 두 절단점에 의하여 나누어진 앞부분, 중간부분, 뒷부분에 있는 원소 중 앞부분과 뒷부분은 부모-1로부터 상속받고 중간부분의 원소는 부모-2로부터 상속받는 것이다[2].

이점교차에 의해 두 부모 P1, P2가 자손 O_1을 생산하는 과정은 그림 9와 같다.

P_1 = (0 1 1 | 0 0 1 0 1 0 1 | 0 0)

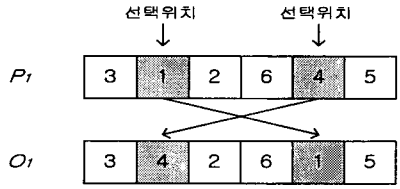
O_1 = (0 1 1 1 0 0 1 0 0 1 0 0)

P_2 = (1 1 0 | 1 0 0 1 0 0 1 | 1 1)

[그림 9] 이점교차 연산자에 의한 자손 O_1의 생성 예

② 돌연변이 연산자

돌연변이 연산자로는 역전(inversion), 삽입(insertion), 변위(displacement), 상호교환(reciprocal exchange) 등이 있는데 본 논문에서는 단순히 두 개의 위치를 임의로 선택하여 바꾸어 주는 교환(swap) 연산자를 사용한다[10, 12]. 교환 연산자에 의해 부모로부터 자손 O_1생산되는 과정은 그림 10과 같다.



[그림 10] 교환 돌연변이 연산자에 의한 자손 생성 예

4. 실험 및 결과

마. 4.1. 실험

본 논문에서 제안한 유전자 알고리즘의 유효성을 검증하기 위하여 Gen의 논문[10]에서 사용된 6개의 센터와 30개의 사용자로 구성된 모의 네트워크를 구성하여 실험하였다.

센 터 : { 1, 2, 3, 4, 5, 6 }
사용자 : { 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 }

한 센터에 클러스터 될 수 있는 최대 사용자 수는 10개(g_i = 10)로 하였으며, 한 센터 당 수용할 수 있는 최대 교통량은 300(C_i = 300), 센터와 센터 사이의 메시지 지연시간은 0.1초(β_{ij} = 0.1), 그리고 사용자간의 교통량 U 는 다음에 따른다.

$$U = \begin{pmatrix} u_3 & u_1 & 0 & u_1 & 0 & u_1 \\ u_1 & u_3 & u_1 & 0 & u_1 & 0 \\ 0 & u_1 & u_3 & u_1 & 0 & u_1 \\ u_1 & 0 & u_1 & u_3 & u_1 & 0 \\ 0 & u_1 & 0 & u_1 & u_3 & u_1 \\ u_1 & 0 & u_1 & 0 & u_1 & u_3 \end{pmatrix}$$

$$u_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$u_3 = \begin{pmatrix} 0 & 3 & 3 & 3 & 3 \\ 3 & 0 & 3 & 3 & 3 \\ 3 & 3 & 0 & 3 & 3 \\ 3 & 3 & 3 & 0 & 3 \\ 3 & 3 & 3 & 3 & 0 \end{pmatrix}$$

각 센터와 센터들간의 연결된 비용은 w_{1ij}의 범위 내에서, 센터와 사용자들 연결된 비용은 w_{2ij} 범위 내에서 임의로 발생시켜 주었다.

$w_{1ij} \in [100, 200]$

$w_{2ij} \in [\ 1, 100]$

유전자 알고리즘 내에서 발생되어지는 각 변수들의 수치는 여러 차례의 준비실험을 통해 다음과 같이 설정하였다.

초기 모집단 크기 $pop_size = 20$

최대 반복 세대 수

$\max_generation = 50$

교차율 $p_C = 0.4$

돌연변이율 $p_M = 0.1$

초기화에서 발생하는 임의의 수치폭을 고려하여 실험 반복 회수를 50번 이상으로 한다.

LAN 설계를 위한 PS-NC 유사 프로그램 코드는 다음과 같다.

```
main( )
{
 w1_ij( );
 // 센터와 센터간의 weight 값을 임의발생
  w2_ij( );
// 센터와 사용자간의 weight 값을 임의발생

  for (i=0; i<Pop_size; i++)
      init_pop( );   //초기 모집단 발생

  Fitness( );     // 적응도 평가

  while (i<max_gen)
//종료조건에 만족할 때까지 반복
    {
    selection( );   // 선별과정
    two_point_crossover( );
 // 이점교차 연산자 적용
    swap_mutation( );
// 교환돌연변이 연산자 적용
    Fitness( );     // 적응도 평가
    max_gen++;   // 세대 증가
     }
}
```

[그림 11] PS-NC GA의 유사 프로그램 코드

main 부분의 적응도 평가를 위한 Fitness 함수는 다음과 같은 서브 함수들을 포함하고 있으며 [그림12]와 같다.

```
Fitness( )
{
x_1();
// 센터와 센터간의 연결된 상태 배열
x_2();
 // 센터와 사용자간의 연결된 상태 배열
a_matrix();
// 센터 i에서 센터 k를 통과하여 센터 j로의 트래
픽이 존재하는 배열
f_kl_matrix();
// 센터 i에서 센터간 링크 (k,l)을 통과하는 센터 j
로의 트래픽이 존재하는 배열
t_matrix();   /   센터와 센터간의 연결 트래픽 값
l_k();          // 센터 k에 연결된 트래픽 값
gamma();
// 센터와 사용자 전체에 대한 트래픽 값
f_1(); // 목적함수 1(메시지 지연시간)을 계산
f_2(); // 목적함수 2(설치비용)를 계산
}
```

[그림12] PS-NC GA의 적응도 평가

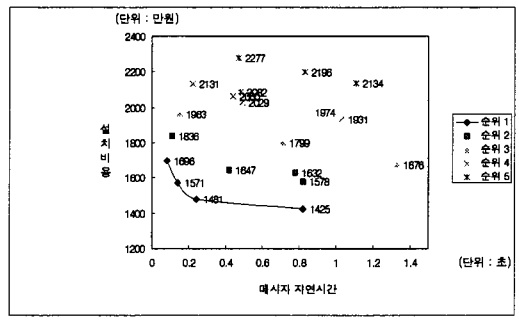
선별부분은 pareto_set( )함수에서 모든 개체들에 대한 파레토 계층을 그룹화하고, cubicle( ) 함수에서 개체들의 최대, 최소값을 찾아 적소영역 크기를 산출하며, 적소영역 크기내의 개체수에 따라 ranking( )함수에서 순위를 부여한다. 부여된 순위에 따라 ps_nc_selection( )함수에서 다음 세대로 유전될 모집단들을 선별하며, selection( ) 함수는 [그림 13]과 같다.

```
selection( )
{
    pareto_set( );  // 파레토 계층 그룹화
    cubicle( );
            // 파레토계층에 대한 적소영역 적용
    ranking( );
            // 모든 개체에 대해 순위부여
    ps_nc_selection( );
            // 제안한 파레토계층-적소영역
               알고리즘에 의해 선별
}
```

[그림 13] PS-NC GA의 선별과정

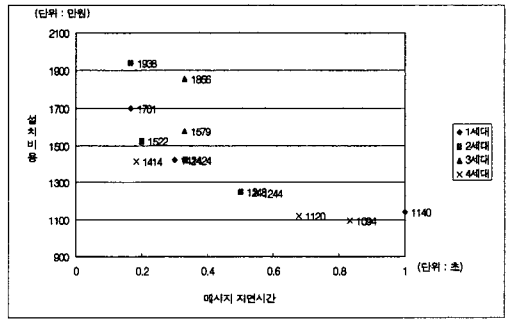
4.. 결과

초기 모집단들은 init_pop 함수에 의해 무작위로 넓게 분포하며, 개체들은 매 세대별 목적함수에 의해 적응도가 높은 모집단이 다음 세대에 선별되므로 모집단내의 개체들은 점점 파레토 최적해들로 수렴한다. 초기 모집단의 1세대 변환후의 개체들에 대한 분포도는 그림 14와 같으며, [순위1]로 표시된 해들이 현 세대에서의 파레토 최적해들(pareto frontier)이다.



[그림 14] 1세대 파레토 해

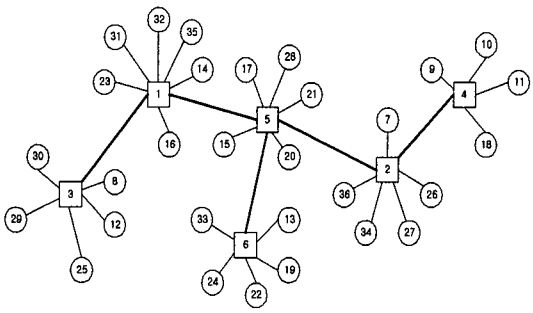
최적 LAN 설계를 위한 PS-NC GA는 평균 10-15세대만에 최적해에 도달하였으며, 그때의 파레토 최적해들로 구성된 결과는 그림 15와 같다.



[그림 15] 최종 파레토 해 집합

이 때 해당되는 모집단은 {2, 3, 4, 4, 4, 3, 6, 1, 5, 1, 5, 4, 2, 6, 5, 5, 6, 1, 6, 3, 2, 2, 5, 3, 3, 1, 1, 1, 6, 5, 2, 1, 5, 2}이고, prüfer수로 표

현된 모집단을 디코딩 알고리즘을 사용하여 트리 형태로 나타내면 그림 16과 같다.



[그림 16] 최적 네트워크 구조

센터와 사용자수를 늘리거나, 초기모집단의 수를 늘려서 더 복잡한 네트워크를 구성할 때에도 파레토 유전자 알고리즘보다 파레토 계층-적소영역 유전자 알고리즘을 적용한다면 빠른 시간내에서 최적의 네트워크를 구성할 수 있는 값들을 찾아 낼 수 있을 것이다.

5. 결 론

본 논문에서는 설치비용과 메시지 지연시간을 동시에 최소화할 수 있는 최적 LAN 설계를 다목적 유전자 알고리즘을 이용하여 구축하였다. LAN 구축 시 메시지 지연시간 및 설치비용을 최소화하는 문제는 NP-hard한 경우가 대부분인데, 병렬적인 탐색기법으로 비교적 짧은 시간 내에 만족한 해를 얻을 수 있는 유전자 알고리즘을 적용하였다. 최적 LAN 설계를 위한 네트워크 위상은 내선망에서 가장 보편적으로 이용되고 있는 스패닝 트리 형태를 선택하였다. 스패닝 트리를 유전자 알고리즘에 적용하기 위하여 Prüfer 수 개념을 사용하였다. Prüfer 수는 스패닝 트리를 효율적이고 간결하게 표현 가능하며, 스패닝 트리의 특징도 잘 반영할 수 있는 표현방법이다. 파레토 계층-적소영역 유전자 알고리즘은 두 가지 목적이 서로 상충되어지는 해를 효과적으로 찾을수 있다. 본 논문에서는 파레토 계층-적소영역 알고리즘을 LAN 설계 분야에 적용함으로써, 100세대만에 최적 해들을 얻은 파레토 유전자 알고리즘과 비슷한 값들을 평균 10세대에서 15세대 얻을 수 있었다. 센터와 사용자수를 늘리거나, 초기모집단의 수를 늘려서 더 복잡한 네트워크를 구성할 때에도 파레토 유전자 알고리즘보다 파레토 계층-적소영역 유전자 알고리즘이 빠른 시간 내에 최적의 네트워크를 구성할 수 있는 값들을 찾아 낼 수 있을 것이다.

참 고 문 헌

[1] Hyun, C. J., Kim, Y. H. and Kim, Y. K., "A Genetic Algorithm for Multiple Objective Sequencing Problems in Mixed Model Assembly Lines", Computers Ops. Res., vol. 25, No. 7-8, pp. 675-690, 1998.

[2] 김여근, 윤복식, 이상복 공저, "메타 휴리스틱", 영지문화사, 1999.

조성배 역, "GA의 기초이론, 공학응용 및 인공생명 유전자 알고리즘", 대청컴퓨터월드, 1996.

[3] 조성배 역, "GA의 기초이론, 공학응용 및 인공생명 유전자 알고리즘", 대청컴퓨터월드, 1996.

[4] Goldberg, D. E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addson-wesely, 1989.

[5] Gen, M., and Cheng, Runwei, "Genetic Algorithms and Engineering Design", John wiley & sons, 1997.

[6] 김태윤, "데이터통신과 컴퓨터 통신", 집문당, 1990.

[7] Prüfer, H., "Neuer beweis eines satzes über permutationen", Arch. Math. Phys., Vol. 27, pp. 742-744, 1918.

[8] 이재승, "Prüfer수를 이용한 외판원문제의 유전해법", 석사학위논문, 한양대학교, 1996.

[9] Zhou, G., Gen, M., Wu, T., "An Approach

to the Degree-Constrained Minimum Spanning Tree Problem Using Genetic Algorithm", Proceedings of 20th International Conference on Computers & Industrial Engineering, 1996.

[10] Gen, M., kenichi IDA and Kim, J. Y., "A Spanning Tree-Based Genetic Algorithm for Bicriteria Topological Network Design", Proceedings of the IEEE International Conference on Evolutionary Computation , 1998.

[11] Kim, J. R., Gen, M., Mutsuo Y., "A Reliable Network Topology Design using Hierarchical Genetic Algorithm", The Second Asia-Pacific Conference on Industrial Engineering and Management Sys. (APIEMS'99), pp. 275-278, 1999.

[12] Kim, J. R. and Gen, M., "Genetic Algorithm for Solving Bicriteria Network Topology Design Problem", Proceedings of the 1999 Congress on Evolutionary Computation , Vol. 3 , pp. 2272-2279 , 1999.

최강희

1988.3-1993.2 관동대학교 정보처리과 이학사

1993.3-1995.2 관동대학교 전자계산공학과 공학석사

1966-2000.2 관동대학교 전자계산공학과 공학박사

2001.3-2005.8 현재 대원과학대학 정보처리전공 교수

관심분야 :Mobile DB,전자상거래

정경희

1994.3-1998.2 관동대학교 전자계산공학과 공학사

1998.9-2000.8 관동대학교 전자계산공학과 공학석사

2000.9.-2003.8 관동대학교 전자계산공학과 박사과정수료

2005.8 현재 대원과학대학 겸임교수

관심분야: 유전자알고리즘,멀티미디어,그래픽스