

유전알고리즘을 이용한 소형케도차량 선로네트워크 설계*

원진명

워털루대학교 전자컴퓨터공학부
(jinmyung.won@gmail.com)

본 논문에서는 연결성, 신뢰성, 교통량 제약조건을 만족하는 최소 비용의 소형케도차량 선로네트워크를 설계하기 위한 유전알고리즘을 제안한다. 소형케도차량 시스템은 다수의 자동화된 차량들이 공중에 설치된 선로네트워크를 따라 움직이는 신개념 교통 시스템이다. 주어진 역의 위치와 역간 교통량 수요에 대해 최적의 선로네트워크를 구하는 문제는 소형케도차량 시스템에 관련된 가장 중요한 문제 가운데 하나이다. 본 논문에서는 선로네트워크를 방향성 링크를 갖는 그래프로 표현하고 그 비용과 연결성, 신뢰성, 교통량을 수식화하였다. 이렇게 주어진 네트워크 성능지표를 바탕으로 선로네트워크 설계 문제에 적절한 연산자들로 구성된 유전알고리즘을 제안한다. 이 연산자들은 안정상태 선택 연산자, 수리 알고리즘, 방향성이 있는 돌연변이 연산자를 포함한다. 제안된 유전알고리즘의 적절한 변수를 결정하고 그 성능을 타 알고리즘과 비교하기 위한 실험을 수행하였다. 최대 210개의 링크를 갖는 선로네트워크에 대해 수행된 실험결과는 제안된 유전알고리즘이 적절한 시간 내에 만족스러운 해를 구할 수 있음을 보인다.

논문접수일 : 2007년 01월

게재확정일 : 2007년 07월

교신저자 : 원진명

1. 서론

네트워크 접속형태(network topology) 최적화 문제는 컴퓨터 네트워크, 텔레커뮤니케이션, 자원 분배, 교통 네트워크 설계 등 네트워크에 기반을 둔 모든 공학 분야에서 중요하게 다루어져 왔다. 일반적인 네트워크 최적화 문제의 목적은 주어진 노드의 집합 N 과 설치 가능한 링크의 집합 L 에 대해 비용을 최소화하면서 제약조건을 만족하는 L 의 부분집합을 구하는 것이다. 널리 사용되는 네트워크 제약조건의 예로는 연결성(connectivity),

신뢰성(reliability), 전송 지연(transmission delay), 용량(capacity) 등이 있다.

네트워크 최적화 문제는 NP-완전(NP complete)하다[1]. n 개의 노드로 이루어진 방향성 네트워크(directed network)는 $n(n-1)$ 개의 링크를 가질 수 있으며 가능한 모든 링크 부분집합의 조합은 $2^{n(n-1)}$ 가지이다. 이는 n 에 대해 지수함수 이상의 정도로 증가하며 네트워크 최적화 문제를 다항함수 연산시간 이내에 풀 수 있는 알고리즘은 존재하지 않는다. 네트워크 최적화 문제를 풀기 위한 다양한 방법이 제안되어 왔으나, 그 NP-완

* 이 논문은 한국과학재단의 해외 Post-doc. 연수지원에 의하여 연구되었음.

전성으로 인해 주로 발견적 방법(heuristic method)이 사용되어져 왔다. 지금까지 소개된 대표적인 발견적 방법으로는 greedy heuristic[2], 분기 한정법[3], tabu search[4], simulated annealing [5, 6, 7] 등이 있다.

보다 최근에는 유전알고리즘을 사용하여 네트워크 최적화 문제를 해결하려는 노력이 활발하게 진행되어 왔다. 유전알고리즘은 네트워크 최적화 문제의 규모와 형식에 대해 강건한 특성을 가지며 적절한 연산시간 내에 최적에 근사한 비용을 갖는 네트워크를 산출해 준다. 또한, 표준 유전알고리즘의 이진 인코딩 방식은 네트워크 해 후보와 일대일 대응이 가능하다는 장점이 있다.

본 연구 이전에는 노드간 거리(network diameter)와 신뢰성을 고려한 분산 컴퓨터 네트워크를 설계하기 위한 유전알고리즘이 개발된 바 있으며 [8], 신뢰성 제약조건을 만족하면서 비용을 최소화하는 통신 네트워크를 설계하기 위한 로컬탐색 유전알고리즘도 소개되었다[9]. 또한, 지연과 신뢰성 제약조건을 만족하는 패킷 스위치 네트워크를 최적화하기 위해 유전알고리즘이 사용되기도 하였다[10]. 교통 네트워크에 관해서는 고정비용 수송 문제(fixed charge transportation problem)와 최단거리탐색 문제가 포함된 네트워크 최적화 문제를 해결하기 위한 유전알고리즘에 대한 연구가 이루어진 바 있다[11]. 네트워크 최적화 문제를 풀기 위한 유전알고리즘의 변수들이 그 성능에 미치는 영향을 조사하여 유전알고리즘을 조율하는데 사용되기도 하였다[12]. 한편, 본 논문의 저자는 소형케도차량 시스템의 비용과 교통량을 동시에 최소화하는 파레토 최적집합을 구하기 위해 다목적 유전알고리즘을 적용한 바 있다[13].

소형케도차량 시스템은 컴퓨터가 다수의 소형 차량들을 선로네트워크 상에서 운전하는 미래형

교통시스템이다[14]. 도시에 위치한 다수의 소규모 역들을 연결하는 선로네트워크는 공중에 설치되어 다른 교통흐름을 방해하지 않으며 타 교통수단에 맞먹는 수송능력을 제공할 수 있다. 또한, 측선과 본선으로 구성되는 역 구조로 인해 차량이 역을 통과하는 동안 정차할 필요가 없다. 이로 인해 승객이 목적지에 도달하기까지 정차나 환승 과정을 거치지 않는다. 최대 4인 정도가 탑승할 수 있는 차량 규모는 사생활 보호와 운전 비용 측면에서도 유리하다. 타 대중교통수단과 차별화되는 이런 다양한 장점으로 인해 상당수의 소형케도차량 시스템이 현재 전세계적으로 연구, 개발되고 있다[15]. 소형케도차량 시스템은 방향성 네트워크로 표현될 수 있는 선로네트워크에 기반을 두고 있으며 차량 흐름의 완전한 관측 및 제어가 가능하다는 측면에서 흥미로운 연구주제들을 제공한다. 그 중에서도 선로네트워크 접속형태 최적화 문제는 서비스의 질과 시스템 운전 비용에 직결되는 중요한 최적화 문제 가운데 하나다.

본 논문에서는 저자의 기존 연구결과를 보강하여 소형케도차량 시스템의 선로네트워크를 최적화하기 위한 유전알고리즘을 제안하고자 한다. 우선, 기존 연구에서 배제되었던 네트워크 신뢰성 제약조건을 고려하여, 어떤 노드나 링크가 사고 또는 고장으로 사용이 불가능할 때에도 연결성을 보장하는 네트워크를 산출할 수 있게 한다. 또한 주어진 문제의 특성에 알맞은 유전알고리즘을 구현하기 위해 다음과 같은 연산자들을 제안한다. 첫째, 안정상태 선택 연산자(steady state selection operator)를 사용하여, 매 세대마다 두 개의 자손을 생성하여 부모 개체군과 경쟁시킨다. 안정상태 유전알고리즘은 개체의 신뢰성 연산에 많은 시간을 소요해야 하는 선로네트워크 최적화 문제에 적절한 방식이다. 둘째, 안정상태 유전알고리즘의

빠른 수렴 속도를 제어하기 위해, 순위 기반의 로그합수 선택(rank based logarithm selection) 알고리즘을 고안한다. 셋째, 연결성과 신뢰성에 대해 유효한 개체군을 확보하기 위해, 새로 생성된 자손이 연결성이나 신뢰성을 만족하지 않을 경우 이를 수리하는 알고리즘(repair algorithm)을 사용하였다. 넷째, 지역 최적해 주변의 해가 빠르게 수렴할 수 있도록 돌연변이 연산자에 방향성을 부가하였다. 또한, 제안된 유전알고리즘의 적절한 변수 값을 결정하고 그 효용성을 검증하기 위해 다양한 선로네트워크 최적화 문제에 대한 실험을 수행하고 그 결과를 분석하였다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 방향성 네트워크 표현 방식에 기반을 두고 선로네트워크 최적화 문제를 수식화한다. 제 3장에서는 제안된 유전알고리즘을 상세히 설명하고, 제 4장에서 실험결과를 소개한다. 제 5장에서는 본 논문을 마무리하고 소개된 연구결과를 실제로 적용하기 위해 필요한 장래의 연구방향을 제시한다.

2. 문제 설정

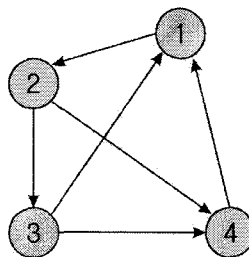
소형케도차량 선로네트워크는 다음과 같은 방법을 통해 방향성 네트워크로 표현된다. 선로네트워크 상의 n 개의 역들을 역 노드 집합 $N = 1, \dots, n$ 으로 나타낸다. 이들 사이를 연결하는 $n(n-1)$ 개의 가능한 모든 선로 링크들을 링크 집합 $L = \{(i, j) | i, j \in N, i \neq j\}$ 으로 나타낸다. 이 때, 선로네트워크 최적화 문제의 해 후보는 완전 방향성 네트워크 $G = (N, L)$ 의 부분 네트워크가 된다. 출발 노드와 도착 노드가 같은 루프 형태의 링크는 존재하지 않으며, 노드 i 에서 j 방향으로의 링크는 단 하나만 존재할 수 있다고 가정한다. 역 i

에서 j 방향으로 향하는 링크를 둘 이상 설치하는 것을 허용하면, 시스템 신뢰성은 향상시킬 수 있으나 제 2.5절에서 설명되는 바와 같이 수송능력에는 도움이 되지 않는다.

주어진 노드 집합 N 에 대해, 선로네트워크 최적화 문제의 해 후보는 이진벡터 $x \in \{0, 1\}^{n(n-1)}$ 로 나타낼 수 있다. 역 노드의 순서쌍 (i, j) 에 대응되는 x 의 원소 $x_{i,j} \in \{0, 1\}$ 는 링크 (i, j) 의 존재 여부를 나타낸다. 이 값이 1이면 링크가 존재하고 0이면 링크가 존재하지 않는다. 예를 들어, [그림 1]과 같이 표현되는 선로네트워크는

$x = [100011101100]$ 로 인코딩 될 수 있다.

선로네트워크 상의 차량 이동은 멀티호핑 방식으로 이루어진다. 즉, 한 노드를 출발한 차량은 최종 노드에 도달하기 위해 하나 이상의 노드를 거칠 수 있다. 예를 들어, [그림 1]에서 노드 1을 출발한 차량이 노드 3 또는 노드 4에 도달하기 위해서는 노드 2를 거쳐야 한다.



[그림 1] 선로네트워크 모델의 예

선로네트워크 최적화 문제를 수식화 하기 위해 다음과 같은 변수들을 정의한다. 0을 포함하는 양의 실수 집합을 \mathcal{R}_+ , 0을 포함하는 양의 정수 집합을 \mathcal{N}_+ , 링크 (i, j) 의 길이를 $d_{i,j} \in \mathcal{R}_+$, 노드 i 에서 j 로의 피크 시간대 시간당 교통량 수요를 $t_{i,j} \in \mathcal{R}_+$, 시스템 수명 동안의 전체 교통량 수요를 $T_{i,j} \in \mathcal{N}_+$,

단위 길이당 링크 건설 및 유지비용을 $c_L \in \mathcal{R}^+$, 선로네트워크 x 상에서의 노드 i 에서 j 로의 차량 한 대당 운전비용을 $c_{i,j}(x) \in \mathcal{R}^+$, 선로 링크의 시간당 차량수송능력을 $t_L \in \mathcal{R}$ 로 정의하자. $t_{i,j}$ 의 단위는 시간당 차량 수인 반면, $T_{i,j}$ 의 단위는 차량 수이다. 이렇게 정의된 변수들로부터 선로네트워크 후보 x 의 전체 비용 $C(x)$ 와 연결성, 신뢰성, 교통량 제약조건에 대한 유효성이 평가될 수 있다. 선로네트워크 최적화 문제의 목적은 이 세가지 제약조건들을 만족하면서 $C(x)$ 를 최소화하는 선로네트워크 x^* 를 구하는 것이다. 본 절에서는 선로네트워크 후보의 비용 및 연결성, 신뢰성, 교통량을 평가하는 방법을 상세히 기술한다.

2.1 경로 선택

선로네트워크 후보의 비용 및 성능지표를 결정하기 위해서는 한 노드로부터 다른 노드로의 차량 이동 경로가 결정되어야 한다. 소형케도차량 시스템 운영의 중요한 원칙은 차량이 역 구역을 제외한 모든 부분에서 일정한 속력으로 움직인다는 것이다. 이 등속운행 특성 때문에 두 노드 간의 최단 거리 경로는 최단 시간 경로이자 최소 에너지 소비 경로이다.

노드 쌍들간의 최단경로가 어떤 특정한 노드를 지나치는 회수가 많아지면 그 노드의 교통량이 그 노드의 수송능력을 초과할 수도 있다. 이 경우에는 차량의 이동 경로를 결정할 때, 최단 거리 기준을 희생하면서 차선의 경로를 선택해야 할 것이다. 그러나 본 논문에서는 차량의 경로는 항상 노드 간 최단 거리 경로로 결정되며, 차량들이 최단 경로를 따라 이동함에 따라 발생하는 최악의 교통량 수요에 대해 교통량 제약조건이 만족되게끔 선로네트워크를 설계한다고 가정한다. 선로네트워

크가 설계된 후, 예상치 못한 교통량 수요에 대해 차량의 경로를 지능적으로 결정하는 방법은 본 논문에서 다루지 않는다.

주어진 선로네트워크 후보의 노드 쌍들 간 최단경로는 Floyd의 최단경로 생성 알고리즘을 통해 구할 수 있다[16]. Floyd 알고리즘의 계산 복잡도는 $\Theta(n^3)$ 이나 타 알고리즘들보다 빠른 시간 내에 모든 노드 쌍들 간의 최단경로를 구해준다. 어떤 선로네트워크 x 상에서 노드 i 로부터 j 로의 최단 경로는 n 차원의 벡터 $r_{i,j}(x) \in \{0, 1\}^n$ 으로 나타낼 수 있다. 만약 i 로부터 j 로의 최단 경로가 어떤 노드 k 를 통과하면 $r_{i,j}(x)$ 의 k 번째 성분 $r_{i,j,k}(x)$ 의 값은 1이 되고, 그렇지 않으면 0이 된다. $i = j$ 이면 $r_{i,j,k}(x)$ 는 k 값에 관계없이 0이라고 가정하고, 노드 i 로부터 j 로의 최단 경로 길이는 $l_{i,j}(x) \in \mathcal{R}$ 로 정의한다.

2.2 시스템 비용

선로네트워크의 비용 $C(x)$ 는 링크 건설 및 유지비용 $C_L(x)$ 와 차량 운전비용 $C_V(x)$ 의 합으로 주어진다. $C_L(x)$ 와 $C_V(x)$ 의 식은 각각 식 (1), 식 (2)와 같다.

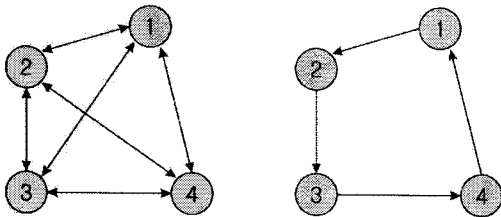
$$C_L(x) := c_L \sum_{i,j=1}^n d_{i,j} x_{i,j} \quad (1)$$

$$C_V(x) := \sum_{i,j=1}^n c_{i,j}(x) T_{i,j} \quad (2)$$

$C_L(x)$ 는 선로 링크 전체 길이에 비례하며, $C_V(x)$ 는 시스템 수명 동안의 모든 차량 운전비용의 합으로 주어진다. 노드 i 로부터 j 로의 차량 한 대 운전비용 $c_{i,j}(x)$ 는 $l_{i,j}(x)$ 에 대해 단조증가의 관계가 있다.

이러한 관계로 인해, 일반적으로 선로네트워크 상에 더 많은 링크가 존재하면 $C_L(x)$ 는 증가하는

반면 $C_V(x)$ 는 감소하는 경향이 있다. 반대로 선로 네트워크 상에 더 적은 링크가 존재하면 $C_L(x)$ 는 감소하는 반면 $C_V(x)$ 는 증가하는 경향이 있다. 예를 들어, [그림 2]의 두 선로네트워크들을 고려해 보자. 왼쪽의 선로네트워크는 모든 링크가 존재하는 완전 선로네트워크로 $C_L(x)$ 는 최대인 반면, 모든 노드 순서쌍에 대해 최단거리 경로를 제공하므로 $C_V(x)$ 는 최소가 된다. 오른쪽의 선로네트워크는 최소한의 링크가 존재하는 루프 형태로서 $C_L(x)$ 는 최소가 되는 대신 $C_V(x)$ 는 최대가 된다.



[그림 2] 완전 선로네트워크와 루프 형태의 선로네트워크

2.3 연결성

어떤 선로네트워크 x 의 연결성은 x 상에서 모든 노드 쌍들간의 경로가 존재함을 의미한다. 연결성은 선로네트워크 상의 모든 다른 노드로 이동하고자 하는 승객의 요구에 의해 주어지는 필수적인 제약조건이다. 선로네트워크 후보의 연결성은 그 네트워크에 Floyd의 최단경로 생성 알고리즘을 적용하는 동안 밝혀진다. 만약 Floyd 알고리즘이 x 에 적용된 후에도 $l_{ij}(x)$ 가 Floyd 알고리즘의 초기값으로 남아 있다면 노드 i 로부터 j 로의 경로는 존재하지 않음을 의미한다.

2.4 신뢰성

연결성은 네트워크 상의 어떤 노드나 링크가 끊어지면 파괴될 수 있다. 선로네트워크가 어떤

노드나 링크의 끊김에 대해서 얼마나 강건하게 연결성을 보장할 수 있는지를 나타내는 척도가 네트워크 신뢰성이다. 네트워크 신뢰성의 기준으로는 크게 두 단계 연결성[10]과 전 노드 간 신뢰성[9]의 두 가지 기준이 주로 사용된다. 두 단계 연결성은 어떤 선로네트워크가 하나의 노드 또는 링크의 고장에 대해서도 여전히 연결성을 유지하는지를 나타내며 모든 터미널 간 신뢰성은 각 노드와 링크에 대해 다르게 주어지는 끊김 확률에 대해 전체 네트워크가 연결성을 유지할 확률이 얼마나 되는지를 수치로 나타낸다.

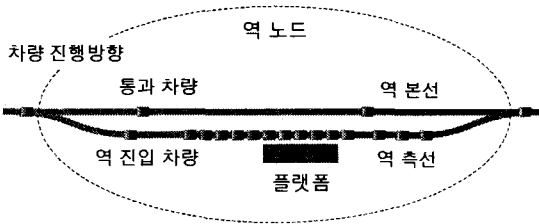
본 논문에서는 두 단계 연결성을 신뢰성의 기준으로 사용한다. 두 단계 연결성을 만족하는 선로네트워크 상에서 어떤 노드 쌍 간에는 두 개 이상의 구별된 경로가 존재한다. 어떤 선로네트워크가 두 단계 연결성을 만족하는지 확인하려면 네트워크 내의 모든 노드와 링크를 하나씩 삭제한 후 남은 네트워크가 연결성을 만족하는지 확인해야 한다. 즉, 두 단계 연결성 확인을 위한 최악의 계산 복잡도는 $\Theta(n^5)$ 이다.

세 단계 혹은 그 이상의 연결성도 고려할 수 있으나 본 논문은 두 가지 이유로 인해 두 단계 연결성을 사용한다. 첫째, 세 단계 이상의 연결성을 제공하는 선로네트워크 상의 모든 노드는 셋 이상의 진입 링크와 진출 링크를 갖는다. 이 제약조건은 선로네트워크의 연결 형태를 복잡하게 만든다. 둘째, 세 단계 이상의 연결성을 확인하는 데는 지나치게 많은 시간이 소요된다. 따라서 매 세대마다 새로이 생성되는 선로네트워크 후보들의 신뢰성을 확인해야 하는 유전알고리즘 등에 적용하기에는 적절치 않다.

2.5 교통량

소형궤도차량 시스템의 교통량 예측 방법은 저자의 이전 논문에서 좀 더 자세하게 설명되어 있다

[17]. [그림 3]에 묘사된 바와 같이, 역 노드는 축선과 본선으로 구성된다. 역의 입구에서는 선로의 분기가 발생하며 역의 출구에서는 선로의 합류가 발생한다. 역의 입구에서 선로가 분기되기 직전에는 그 역으로 향하는 모든 링크들이 역 입구로 합류하게 되며, 역의 출구에서 축선과 본선이 합류한 직후에는 그 역으로부터 시작되는 모든 링크들이 분기한다. 따라서 어떤 선로네트워크 상의 링크 교통량 최대값은 노드 출입구 교통량의 최대값보다는 항상 작거나 같다. 따라서, 노드 출입구 교통량의 최대값이 선로 라인의 수송능력보다 작거나 같으면, 그 선로네트워크는 교통량 제약조건을 만족한다.



[그림 3] 완전 선로네트워크와 루프 형태의 선로네트워크

노드 i 의 피크 시간대 출구 교통량 $\tau_i(\mathbf{x})$ 는 식 (3)과 같이 유도된다.

$$\tau_i(\mathbf{x}) = \sum_{j=1}^n t_{i,j} + \sum_{j=1}^n \sum_{k=1}^n r_{j,k,i}(\mathbf{x}) t_{j,k} \quad (3)$$

우변의 첫 번째 항은 노드 i 로부터 출발하는 피크 시간대 교통량을 나타내며, 두 번째 항은 모든 역간 피크 시간대 교통량 가운데 노드 i 를 통과하는 교통량의 합이다. 피크 시간대에 발생할 수 있는 교통 체증을 피하기 위해서는 링크의 시간당 차량수송능력 t_L 에 대해 (4)가 만족되어야 한다.

$$\max_{i \in \{1, \dots, n\}} \tau_i(\mathbf{x}) \leq t_L \quad (4)$$

t_L 은 링크 상의 차량 운행속도 v_L 과 차량 하나가 차지하는 슬롯의 최소 길이 s_L 에 대해 식 (5)와 같이 주어진다.

$$t_L = \frac{v_L}{s_L} \quad (5)$$

즉, 차량 운행속도를 증가시키고 차량 슬롯 길이를 줄이는 등, 시스템의 안전성을 희생할수록 더 나은 수송능력을 확보할 수 있다.

3. 유전알고리즘

본 절에서는 제 2장에서 기술된 선로네트워크 설계 문제에 알맞은 특수한 연산자들을 갖춘 맞춤형 유전알고리즘을 제안한다. 이 유전알고리즘은 링크 인코딩(link encoding), 제약조건 처리를 위한 벌점 함수(penalty function), 안정상태 선택 연산자(steady-state selection operator), 수리 알고리즘(repair algorithm), 선로네트워크 설계 문제에 알맞은 유전 연산자들(genetic operators)을 포함한다. 제안된 유전알고리즘의 세대 인덱스를 g , 개체군을 $P(g)$, 자손 생성을 위해 선택된 두 부모 개체를 x_1, x_2 , 두 자손을 x_3, x_4 라 하면 의사코드는 다음과 같이 주어진다.

```

g = 0;
Initialize(P(g));
Evaluate(P(g));
DO
    x1, x2 = Select_parents(P(g));
    x3, x4 = Crossover(x1, x2);
    Repair(x3, x4);
    
```

```

Mutate(x3, x4) ;
Evaluate(x3, x4) ;
P(g+1) = Select({x3, x4} ∪ P(g)) ;
g = g + 1 ;
WHILE NOT (termination condition met)
    
```

3.1 인코딩

선로네트워크 후보를 인코딩하는 방법은 유전 알고리즘의 성능에 직접적으로 영향을 미치는 중요한 요소이다. 좋은 인코딩 방법은 모든 선로네트워크 후보를 균일하게 표현할 수 있어야 한다. 또한, 부호 상의 변화가 선로네트워크 상에서도 의미 있는 변화를 야기할 수 있는 것이 바람직하다. 만약, 인코딩 방법이 제약조건을 만족하는 해들만 나타낼 수 있다면 유전알고리즘이 별개의 제약조건 관리 알고리즘 없이도 제약조건을 만족하는 해들만으로 개체군을 진화시켜 나갈 수 있는 장점이 있다.

네트워크 설계 문제에 자주 사용되는 두 가지 인코딩 방법은 Prüfer 인코딩과 링크 인코딩이다 [12]. Prüfer 인코딩은 n 개의 노드를 갖는 모든 경우의 확장 트리(spanning tree)를 $n-2$ 의 크기를 갖는 1에서 n 사이의 숫자 배열로 일대일 대응시킬 수 있는 특성을 갖는다. Prüfer 부호 상의 작은 변화는 대응되는 네트워크 상에서 큰 변화를 일으키므로, 최소 확장 트리(minimum spanning tree)에 기반을 둔 특수한 네트워크 설계 문제를 해결하는데 알맞다. 링크 인코딩은 각 링크의 존재 여부를 직접 유전자 성분으로 나타내는 방식으로서 네트워크 최적화를 위한 유전알고리즘에서 주로 사용되고 있다. 본 논문에서는 제 2장의 전반부에서 설명된 바와 같이 링크 인코딩에 기반을 둔 이진 벡터 x 로 선로네트워크 후보를 표현한다.

3.2 개체군 초기화

제안된 유전알고리즘은 초기화 과정 동안 $|P|$ 개의 네트워크 후보를 생성해 $P(0)$ 에 추가한다. 초기 개체 x 의 각 비트는 0.5의 확률로 0 또는 1의 값을 가지게 한다. 초기화 과정 동안 수리 알고리즘은 적용되지 않으므로 초기 개체군은 제약조건을 만족하지 않는 개체를 포함할 수 있다.

3.3 목적함수

선로네트워크 설계 문제의 연결성, 신뢰성, 교통량 제약조건을 다루기 위해 유전알고리즘의 목적함수에 벌점 성분을 추가한다. 즉, 선로네트워크 x 의 비용함수 $C(x)$ 에 연결성 및 신뢰성 벌점함수 $Z_N(x)$ 와 교통량 벌점함수 $Z_P(x)$ 를 부가하여 (6)과 같이 목적함수 $Z(x)$ 를 정의한다.

$$Z(x) = (C(x) + Z_N(x))Z_P(x) \quad (6)$$

$Z_N(x)$ 를 구현하기 위해 x 상에서의 각 노드 쌍들에 대한 연결성 및 신뢰성 정도를 $\gamma_{i,j}(x) \in [0, 1]$ 로 정의한다. 연결성이 낮은 노드 쌍 i, j 에 대해서는 식 (7)과 같이 링크 길이 $d_{i,j}$ 에 비례하는 벌점을 부가한다.

$$Z_N(x) := c_L \sum_{i,j=1}^n d_{i,j} (1 - \gamma_{i,j}(x)) \quad (7)$$

만약, 연결성 제약조건만 주어진다면 $\gamma_{i,j}(x)$ 는 단순하게 노드 i 로부터 j 로의 도달 가능 여부에 따라 0(도달 가능하지 않으면) 또는 1(도달 가능하면)의 값을 가질 수 있다. 만약, 두 단계 연결성이 제약조건으로 주어지면 노드 i 로부터 j 로의 연결성을 파괴할 수 있는 노드나 링크의 비율로 그 신뢰성 정도를 정의할 수 있다. 즉, $a_{i,j}(x)$ 를 x 내에 존재하는 모든 노드와 링크의 개수로 정의하고

$b_{ij}(x)$ 를 i 로부터 j 로의 연결성을 보장하기 위해 필수적인 노드와 링크의 개수라 하자. x 상에서 노드 i 로부터 j 로 도달 가능하면 $\gamma_{i,j}(x)$ 를 식 (8)과 같이 정의하고 도달 불가능하면 0으로 정의한다.

$$\gamma_{i,j}(x) := \frac{a_{i,j}(x) - b_{i,j}(x)}{a_{i,j}(x) - 2} \quad (8)$$

위 정의에 따라 연결성 또는 신뢰성을 만족하는 선로네트워크는 그렇지 않은 선로네트워크보다 항상 더 적은 목적함수 값을 갖는다.

교통량 벌점 함수는 선로네트워크 상의 최대 노드 교통량이 링크의 시간당 차량수송능력보다 크면 목적함수 값이 그 비율에 의해 증가되도록 설계할 수 있다. 본 논문에서는 식 (9) 형태의 교통량 벌점 함수를 사용한다.

$$Z_P(x) = \max \left[1, \left[\frac{\max_{i \in \{1, \dots, n\}} \tau_i(x)}{t_L} \right]^\alpha \right]. \quad (9)$$

α 는 1보다 큰 실수로 선택될 수 있으며 본 논문에서는 $\alpha = 4$ 를 사용한다.

3.4 안정상태 선택 연산자

안정상태 유전알고리즘은 세대 기반 유전알고리즘(generational genetic algorithm)과는 달리, 한 세대에 소수(주로 하나 또는 둘)의 자손만을 생성한다. 생성된 자손 개체들은 현재 개체군 내의 부모 개체들과 경쟁한 후 다음 세대로의 생존 여부가 결정된다. 안정상태 유전알고리즘은 각 개체의 목적함수를 계산하고 제약조건 만족 여부를 판단하는데 많은 시간이 소요되는 선로네트워크 설계 문제에 적합한 방식이다.

본 논문에서 제안하는 안정상태 선택 연산자는

매 세대마다 두 부모 개체 x_1 과 x_2 를 선택하여 유전 연산자들을 차례로 적용한 후 두 자손 x_3 과 x_4 를 생성한다. 제안된 유전알고리즘은 현재 개체군으로부터 x_1 과 x_2 를 선택하기 위해 순위 기반의 로그함수 선택 연산자를 사용한다. 이 연산자는 우선 개체군 내의 모든 개체를 목적함수의 오름차순으로 정리한다. 그리고 주어진 디자인 변수 $k \in \mathcal{R}$ 와 개체군 크기 $|P|$ 에 대해, $[0, \ln(|P|/k+1))$ 범위 내의 균일한 분포를 갖는 두 난수 u_1 과 u_2 를 생성한 후, $([k(e^{u_1} - 1)] + 1)$ 번째와 $([k(e^{u_2} - 1)] + 1)$ 번째 개체를 x_1 과 x_2 로 선택한다. 따라서 k 가 작을수록 선택 압력(selection pressure)이 증가하고 k 가 클수록 선택 압력이 줄어든다.

안정상태 선택 연산자의 중요한 특징은 지역 최적해(local minimum)로의 수렴 속도를 빠르게 한다는 것이다. 이를 보상하기 위해 제안된 유전알고리즘은 큰 k 값을 사용하고 개체군 내에 중복된 개체의 존재를 허용하지 않는다. 후자를 구현하기 위해 생성된 자손이 개체군 내의 어떤 개체와 동일할 경우, 그 자손을 부모 개체들과 경쟁시키지 않고 버린다.

3.5 교차 연산자

제안된 유전알고리즘은 p_C 의 확률로 활성화되는 표준 균일 교차 연산자(uniform crossover operator)를 사용한다. 즉, 두 자손은 p_C 의 확률로 서로 같은 위치의 유전자를 0.5의 확률로 교환하거나, $1 - p_C$ 의 확률로 변화 없이 제 3.6절에서 설명되는 수리 알고리즘을 거치게 된다.

3.6 수리 알고리즘

비록 원래 x_1 과 x_2 이 연결성을 만족하더라도 교차

연산자에 의해 그 연결성이 파괴될 수 있다. 수리 알고리즘은 교차 연산을 거친 선로네트워크 후보가 연결성 또는 신뢰성 제약조건을 만족하게끔 각 노드의 진출 링크 수와 진입 링크 수를 수리한다. 즉, 연결성 제약조건이 주어지면 모든 노드가 하나 이상의 진입 링크와 진출 링크를 갖게 하고, 신뢰성 제약조건이 주어지면 모든 노드가 둘 이상의 진입 링크와 진출 링크를 갖게 한다. 수리 알고리즘은 임의의 순서로 각 노드가 필요한 수 이상의 진출입 링크를 가지고 있는지 체크한다. 만약 그렇지 않으면 그 노드로부터 연결되지 않은 노드 가운데 가장 가까운 노드 사이에 링크를 추가한다.

노드들이 충분한 수의 진출입 링크들을 갖고 있다는 것이 연결성을 보장하지는 않으며, 위와 같은 수리 알고리즘도 최종 네트워크의 연결성을 보장하지는 않는다. 하지만 제안된 알고리즘은 두 가지 이유로 인해 노드의 진출입 링크 수 만을 수리한다. 첫째, 위 수리 알고리즘은 간단하며 적은 계산량을 필요로 한다. 둘째, 제약조건을 만족하는 해 영역의 경계에 있는 최적 네트워크를 찾기 위해서는 제약조건을 만족하지 않는 해도 개체군에 포함시킬 필요가 있다.

3.7 돌연변이 연산자

수리된 자손 개체는 돌연변이 연산을 거치게 된다. 제안된 유전알고리즘은 돌연변이 연산자에 방향성을 부여함으로써 수렴 속도를 개선한다. 만약 수리된 자손 개체가 주어진 제약조건을 모두 만족하면 개체 내에 불필요한 링크가 추가되어 있을 수 있으므로, p_M 의 확률로 링크 제거 돌연변이를 수행한다. 반대로 자손 개체가 주어진 제약조건 중 하나 이상을 만족하지 못하면, p_M 의 확률로 링크 추가 돌연변이를 수행한다. 링크 제거 돌연

변이는 개체 내에 존재하는 임의의 링크를 하나 선택하여 이를 제거하며, 링크 추가 돌연변이는 개체 내에 존재하지 않는 임의의 링크를 하나 선택하여 이를 개체에 추가한다.

4. 실험결과

본 절에서는 제안된 안정상태 유전알고리즘을 여러 가지 선로네트워크 설계 문제에 적용한 실험 결과를 소개한다. 실험을 위한 모든 알고리즘은 C++언어로 구현되었으며 Intel Pentium M Processor 1.6GHz의 CPU 상에서 테스트되었다. <표 1>과 같이 $C(x)$ 의 두 가지 옵션과 제약조건 두 가지 옵션을 조합하여 네 가지 선로네트워크 설계 목적을 고려한다. OBJ1과 OBJ2는 차량 운전비용을 배제한다. OBJ1과 OBJ3은 연결성 제약조건만을, OBJ2와 OBJ4는 두 단계 연결성과 교통량 제약조건을 모두 고려한다. OBJ4가 가장 현실적인 선로네트워크 설계 목적이거나 다른 목적들도 중요한 의미를 가진다. 예를 들어, OBJ1을 갖는 선로네트워크 설계 문제는 이동 세일즈맨 문제[18]로 간주될 수 있다.

<표 1> 네 가지 선로네트워크 설계 목적

이름	비용 $C(x)$	제약조건
OBJ1	$C_L(x)$	연결성
OBJ2	$C_L(x)$	두 단계 연결성 및 교통량
OBJ3	$C_L(x) + C_V(x)$	연결성
OBJ4	$C_L(x) + C_V(x)$	두 단계 연결성 및 교통량

4.1 유전알고리즘 변수

본 논문에서는 제안된 유전알고리즘의 다음 다섯 가지 변수 값들이 탐색된 선로네트워크의 성

능에 미치는 영향을 이해하기 위해 분산 분석법 (analysis of variance)을 사용하였다.

- (1) 개체군 크기 : $|P| \in \{50, 100\}$
- (2) 선택 압력 : $k \in \{10, \infty\}$
- (3) 수리 알고리즘 사용 여부 : $r \in \{\text{false}, \text{true}\}$
- (4) 교차 연산 확률 : $p_C \in \{0.5, 1\}$
- (5) 돌연변이 연산 확률 : $p_M \in \{0.5, 1\}$

위처럼 다섯 가지 독립 변수들이 각각 두 가지 값을 가질 수 있다는 가정하에, 서로 다른 독립 변수 조합을 갖는 32가지 유전알고리즘이 구성될 수 있다. 이 32개의 유전알고리즘을 30개의 선로네트워크 설계 문제들로 평가하여, 총 960개의 데이터 샘플들을 추출하였다. 선로네트워크 설계 문제는 다섯 가지 노드 수($n = 11, \dots, 15$)와 두 가지 목적(Obj1, Obj4)에 대해 생성되었다(각 노드 수와 목적에 대해 3개씩). 목적 함수로는 식 (6)을 사용하였다. 선로네트워크의 각 노드 위치는 $[0, 100] \times [0, 100]$ 의 영역 내에서 임의로 결정되었으며, $d_{i,j}$ 는 두 노드 간의 거리(Euclidean distance)로 가정되었다. $t_{i,j}$ 와 $T_{i,j}$ 는 각각 $\{20, \dots, 150\}$ 과 $\{20000, \dots, 150000\}$ 범위 내에서 임의로 결정되었다. c_L 과 t_L 은 각각 1과 2880으로 주어졌다. 차량 운전비용의 식은 다음과 같이 주어졌다.

$$c_{i,j}(x) = 310^{-6} \cdot l_{i,j}(x) + 10^{-5}$$

유전알고리즘의 수렴이 거의 이뤄지는 10000세대 의 개체군 내에서 가장 뛰어난 개체의 $Z(x)$ 값을 분산 분석법의 대상으로 삼았다.

분산 분석법의 결과가 <표 2>~<표 5>에 정리되었다. <표 2>는 다섯 개의 독립 변수와 이들 쌍에 대한 $Z(x)$ 의 P-값들을 보여주고 있다. 각각의 독립 변수가 0.002이하의 P-값으로 $Z(x)$ 에 의미 있는 영향을 미침을 알 수 있다. <표 3>에서

<표 2> $Z(x)$ 에 대한 분산 분석법 결과

Source	SS	df	MS	P 값
$ P $	244673.2	1	244673.2	<0.001
k	554689.4	1	554689.4	<0.001
r	240286.8	1	240286.8	<0.001
p_C	386564.3	1	386564.3	<0.001
p_M	54692.2	1	54692.2	0.002
$ P \times k$	6636.0	1	6636.0	0.268
$ P \times r$	14915.3	1	14915.3	0.098
$ P \times p_C$	62210.4	1	62210.4	<0.001
$ P \times p_M$	31304.5	1	31304.5	0.016
$k \times r$	382.5	1	382.5	0.268
$k \times p_C$	15990.3	1	15990.3	0.086
$k \times p_M$	56980.0	1	56980.0	0.001
$r \times p_C$	13969.0	1	13969.0	0.109
$r \times p_M$	4524.0	1	4524.0	0.361
$p_C \times p_M$	14045.4	1	14045.4	0.108
error	4867138.1	899	5414.0	
subject	2961349295.6	29		
within group	2966216433.7	928		
total	2968004501.0	959		

는 각 독립 변수 값에 대한 $Z(x)$ 의 평균값을 나열하였다. 이로부터 $|P| = 100, k = \infty, r = \text{true}, p_C = 1$ 에 속한 그룹들이 상대 그룹들보다 더 적합한 선로네트워크를 탐색한다는 결론을 내릴 수 있다. 분산 분석법의 결과는 두 독립 변수 쌍, $|P|$

<표 3> 각 독립 변수 값에 따른 $Z(x)$ 의 평균값

독립 변수	값	$Z(x)$ 의 평균값
$ P $	50	2089.88
	100	2057.95
k	10	2097.95
	∞	2049.88
r	false	2089.74
	true	2058.09
p_C	0.5	2093.98
	1	2053.85
p_M	0.5	2081.46
	1	2066.37

와 p_C , k 와 p_M 사이에도 의미 있는 관계가 있음을 보여준다. <표 4>로부터 $|P|=100$ 과 $p_C=1$ 의 조합이 $Z(x)$ 에 미치는 영향이 각각의 영향보다 월등함을 알 수 있다. 또한 <표 5>로부터 $k=\infty$ 이기만 하면 p_M 이 $Z(x)$ 에 미치는 영향이 약해짐을 알 수 있다.

<표 4> $|P|$ 와 p_C 값에 따른 $Z(x)$ 의 평균값

	$ P =50$	$ P =100$
$p_C=0.5$	2101.90	2086.07
$p_C=1$	2077.86	2029.83

<표 5> k 와 p_M 값에 따른 $Z(x)$ 의 평균값

	$k=50$	$k=100$
$p_M=0.5$	2113.20	2049.72
$p_M=1$	2082.70	2050.03

분산 분석법의 결과는 유전알고리즘의 선택 압력을 낮추고 자손의 다양성을 확보하며 수리 알고리즘을 사용하는 것이 선로네트워크 설계 문제 해결에 도움이 됨을 나타낸다. 이는 선로네트워크 설계 문제가 많은 지역 최적해를 가지고 있으며, 지역적 탐색, 수리 알고리즘이 문제 해결에 도움이 될 수 있음을 암시한다. 다음 절에서 이어지는 실험에서는 분산 분석법의 결과로부터 유추 되는대로 $|P|=100$, $k=\infty$, $r=true$, $p_C=1$, $p_C=1$ 을 사용한다.

4.2 타 알고리즘과의 비교

제안된 유전알고리즘의 성능을 이전에 제안된 세 가지 타 알고리즘들과 실험을 통해 비교해 보았다. 분기 및 제한 알고리즘[19]과 두 가지 유전 알고리즘[9,20]이 비교에 사용되었다. 편의상 각

알고리즘을 BnB, GA1, GA2로 표기하기로 한다.

BnB는 분기 및 제한 알고리즘(branch and bound heuristic)의 약자로 순수 나열형 알고리즘보다는 빠른 시간 내에 최적의 해를 구해준다. BnB는 주어진 선로네트워크 설계 문제를 최적해 내에 존재하는 링크의 수를 기준으로 하나 이상의 하위 문제들로 분류한다. 각각의 하위 문제들에 대해 BnB는 모든 해 후보를 표현할 수 있는 트리 구조를 설정한다. 순서 개념을 포함하는 이 트리 구조는 트리의 제한적인 부분만 검색해 봄으로써 하위 문제의 최적해를 구할 수 있게 한다. 그러나 이 제한적인 부분 역시 링크의 수에 대해 기하급수적으로 증가하므로, 네트워크의 규모가 커지면 BnB는 적절한 시간 내에 해를 구할 수 없다.

GA1과 GA2는 Deeter and Smith and Dengiz에 의해 제안된 세대 기반 네트워크 설계용 유전알고리즘으로서 매 세대마다 일정한 수의 자손 개체를 생성하여 부모 개체군을 치환한다. 모든 자손들에 대해 비용 및 제약조건 평가가 이루어져야 하므로, GA1과 GA2는 제안된 유전알고리즘보다 많은 연산 시간을 필요로 함을 예측할 수 있다. GA1은 수리 알고리즘을 사용하지 않는 반면, GA2는 수리 알고리즘을 사용한다. GA1과 GA2의 보다 상세한 내용은 부록에 제시하였다. 공정한 비교를 위해 원래 논문에서 사용된 BnB, GA1, GA2의 변수들을 그대로 사용하였다.

본 연구에서는 제안된 유전알고리즘을 SSGA로 표시하고 네 알고리즘들을 제 4.1절에서와 같은 방식으로 생성된 24가지 선로네트워크 설계 문제들에 적용하였다. 이 문제들은 여섯 가지 노드 수($n=7, 8, 9, 10, 12, 15$)와 네 가지 목적들을 조합하여 생성되었다. 각 테스트에 대해 $Z(x)$ 값과 연산시간이 기록되었다. BnB는 최적해를 찾는 즉시 종료되었고, 유전알고리즘들은 10000세대가 지

<표 6> OBJ1을 갖는 선로네트워크 설계 문제에 대한 SSGA, BnB, GA1, GA2의 Z(x) 평균값

문제		Z(x) 평균값			
목적	n	SSGA	BnB	GA1	GA2
OBJ1	7	274.4	271.8	388.2	299.5
	8	298.0	-	494.2	347.7
	9	236.2	-	447.0	288.1
	10	251.4	-	535.3	320.8
	12	292.3	-	1019.2	435.1
	15	427.3	-	2207.2	809.5

주) BnB는 n > 7에 대해 8시간 내에 해를 찾는데 실패함.

<표 7> OBJ1을 갖는 선로네트워크 설계 문제에 대한 SSGA, BnB, GA1, GA2의 연산시간 평균값

문제		연산시간(초)			
목적	n	SSGA	BnB	GA1	GA2
OBJ1	7	12.5	4298.0	210.7	223.2
	8	16.8	-	290.0	294.8
	9	21.7	-	372.2	376.2
	10	26.2	-	442.3	472.4
	12	38.9	-	645.8	858.0
	15	66.1	-	1415.7	1428.4

난 후에 종료되었다. 아래의 모든 결과는 25번의 시행에 대한 평균값이다.

OBJ1에 대한 실험결과는 <표 6>과 <표 7>에 정리되었다. BnB는 평균 한 시간을 초과하는 연산시간 후에 7노드, 42링크의 선로네트워크 문제의 최적해를 구했다. 그러나 그 이상의 노드 수를 갖는 문제에 대해서는 8시간 이내에 최적해를 구하지 못했다. 반면, SSGA는 짧은 연산 시간 내에 적절한 해를 구해 내었다. GA1과 GA2의 연산시간은 SSGA보다 상당히 길었으며, 그들의 Z(x) 값 역시 SSGA의 그것보다 큰 경향을 보여주었다.

<표 8> OBJ2, OBJ3, OBJ4를 갖는 선로네트워크 설계 문제에 대한 SSGA, BnB, GA1, GA2의 Z(x) 및 연산시간 평균값

문제	n	SSGA		GA1		GA2	
		Z(x)	시간(초)	Z(x)	시간(초)	Z(x)	시간(초)
OBJ2	10	487	59	820	732	567	934
	12	704	107	1445	1426	841	1599
	15	1874	271	4223	4034	2246	4011
OBJ3	10	1634	29	1865	390	1668	504
	12	3022	43	3621	680	3082	889
	15	4817	76	6528	1116	4971	1258
OBJ4	10	1733	59	2044	637	1745	1043
	12	3234	114	3960	1179	3258	1740
	15	5808	281	7939	2588	6071	4003

실제로 대부분의 시도에 대해 GA1과 GA2의 해 후보는 1000세대 이전에 지역 최적해 근처로 수렴하여 거의 빠져 나오지 못했다.

세 가지 유전알고리즘을 OBJ2, OBJ3, OBJ4를 갖는 선로네트워크 설계 문제에 적용한 결과는 표 8에 정리하였다. BnB는 여전히 8시간의 연산시간으로도 해를 구하지 못했다. 한편, SSGA는 여전히 GA1이나 GA2에 비해 더 나은 성능을 보임을 알 수 있다. OBJ3과 OBJ4에 대해서는 SSGA와 GA2의 Z(x) 값에 큰 차이가 없었다. 이것은 운전비용이 목적함수에 추가되면 선로 링크비용과 차량 운전비용 간의 상충관계 때문에 Z(x) 값의 상대적 차이가 줄어들을 의미한다. OBJ3과 OBJ4에 대한 알고리즘 간의 성능 차이를 확인하기 위해, 150개의 샘플에 대해 쌍 t-테스트(paired t-test)를 수행하였다. <표 9>에 정리된 쌍 t-테스트 결과에 따르면 SSGA가 GA2보다 의미 있게 좋은 성능을 보임을 알 수 있다. 15개의 노드와 210개의 링크를 갖는 문제와 OBJ4에 대해 GA1과 GA2가 대체로 한 시간 정도의 연산시간을 필요로

<표 9> OBJ3과 OBJ4를 갖는 선로네트워크 설계 문제에 대한 SSGA와 GA2의 Z(x) 값 간의 t-테스트 결과

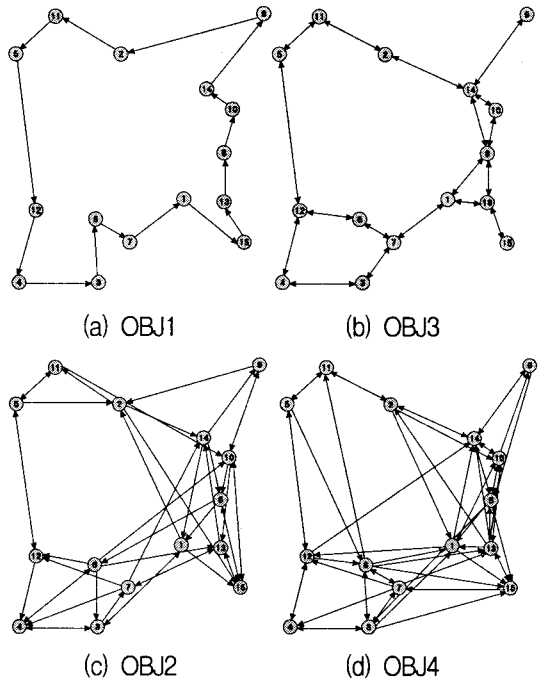
알고리즘	평균	표준편차	T-값	P-값
SSGA	3374.60	1526.32	9.58	<0.0001
GA2	3465.83	1610.84		

하는 반면, 제안된 유전알고리즘은 5분 이내에 종료되었다. 이는 제안된 알고리즘이 수 백여 개의 링크를 갖는 선로네트워크 문제에 대해서도 적절한 시간 내에 해를 구할 수 있음을 암시한다. 실제 선로네트워크 문제는 같은 수의 노드에 대해 완전 방향성 그래프보다는 훨씬 적은 수의 후보 링크를 포함하므로 수 십여 또는 수 백여 개의 노드를 갖는 선로네트워크 문제에도 제안된 유전알고리즘이 적절한 연산시간 내에 해를 탐색할 수 있을 것이다.

[그림 4]는 제안된 유전알고리즘이 $n = 15$ 인 문제에 대해 구해낸 최적의 네트워크들을 보여주고 있다. OBJ1이 주어졌을 경우, 유전알고리즘은 예상대로 [그림 4] (a)와 같은 루프 형태의 네트워크들을 구해냈다. OBJ3이 주어졌을 경우, 유전알고리즘은 [그림 4] (b)와 같이 거미줄 형태의 네트워크들을 구해냈다. 이 경우, 전체 비용에서 차량 운전비용의 비율이 증가할수록 네트워크 내의 링크 수는 증가할 것이다. OBJ2와 OBJ4가 주어졌을 경우, 유전알고리즘은 [그림 4] (c), (d)와 같은 복잡한 형태의 네트워크들을 구해냈다. 이 네트워크들이 실제 최적 네트워크에 얼마나 근접한지 알 수는 없으나, 모든 노드들의 교통량이 t_L 에 가까움을 확인할 수 있었다.

5. 결론

본 논문에서는 소형케도차량 시스템의 선로네



[그림 4] $n = 15$ 인 문제에 대해 제안된 유전알고리즘이 구한 최적 네트워크들

트워크를 최적화하기 위한 유전알고리즘을 제안하였다. 우선 선로네트워크를 방향성 네트워크로 모델링 한 후, 건설 및 유지비용, 차량 운전비용, 연결성, 신뢰성, 교통량을 예측하는 방법을 제시하였다. 이 선로네트워크 모델을 기반으로 선로네트워크 최적화 문제에 적절한 연산자로 구성된 안정상태 유전알고리즘을 개발하였다. 제약조건을 다루기 위해 벌점 함수를, 유전알고리즘의 선택 압력을 제어하기 위해 순위 기반의 로그함수 선택 연산자를, 매 세대마다 생성된 자손 개체의 유효성을 확보하기 위해 수리 알고리즘을, 지역 최적해 주변에서의 수렴 속도를 증가시키기 위해 방향성을 갖는 돌연변이 연산자를 사용하였다. 실험을 통해 선로네트워크 최적화 문제에 알맞은 유전알고리즘 변수를 결정하였고, 이전에 제안된 타 알

고리증과의 비교를 통해 제안된 유전알고리즘의 성능을 검증하였다.

본 연구의 결과를 현실적인 문제에 직접 적용하기 위해서는 몇 가지 해결해야 할 과제가 있다. 첫째, 수 백여 개에 달하는 역을 연결하는 네트워크를 적절한 시간 내에 설계하기 위해서는 연결성, 신뢰성 및 교통량 제약조건에 대한 주어진 선로네트워크의 유효성 여부를 빠른 시간 안에 판단할 수 있는 방법이 필요하다. 선로네트워크 후보를 하나 이상의 루프로 모델링 하고 루프들의 접점들에 대해 연결성, 신뢰성 및 교통량을 예측하는 것이 한 방법이 될 수 있을 것이다. 둘째, 한 링크의 형태는 일정 수준 이상의 회전 반경을 가지며 주변 지형지물의 영향을 받게 되므로, 그 길이를 본 논문에서처럼 예측하기 어렵다. 게다가 실제 선로네트워크 상에는 역이 아닌 노드가 존재할 수 있으므로 그 위치를 어떻게 결정하여 어떻게 선로네트워크 설계 문제에 포함하느냐 하는 것도 중요한 이슈이다. 셋째, 본 논문에서는 빈 차의 교통량이 $t_{i,j}$ 와 $T_{i,j}$ 에 포함되어 있다고 가정하였으나, 주어진 승객 교통량에 대해 빈 차 경로와 교통량을 결정하는 것은 또 다른 최적화 문제로 선형 프로그래밍 기법 등을 통해 해결되어야 한다.

<부록> 비교에 사용된 타 알고리즘

1. 유전알고리즘 1[20]

유전알고리즘 1(GA1)은 Deeter와 Smith에 의해 제안된 네트워크 최적화 유전알고리즘을 일부 수정해서 얻어졌다. GA1은 링크 인코딩을 사용하며 40개의 초기 개체를 난수 발생을 통해 생성한다. 매 세대마다 GA1은 현재 개체군 내에 존재하는 가장 적합도가 높은 개체를 다음 세대 개체군

에 우선적으로 복사한다. 그 후 다음 세대 개체군을 구성할 나머지 개체들을 생성하기 위해 다음과 같은 방식을 사용한다. 우선 이차 순위기반 선택 연산에 의해 현재 개체군 내에서 두 개의 부모 개체를 선택한다. 확률 1을 갖는 균일 교차 연산과 확률 0.25를 갖는 돌연변이 연산을 두 부모 개체에 적용함으로써 두 자손 개체를 생성하여 다음 세대 개체군에 추가한다. 이 과정을 다음 세대 개체군이 완성될 때까지 반복한다.

2. 유전알고리즘 2[9]

유전알고리즘 2(GA2)는 전 노드 간 신뢰성 제약조건을 만족하는 최소 비용 통신 네트워크를 설계하기 위해 제안되었다. 원래 알고리즘의 부분 중 그대로 취할 수 없는 부분에 대해서는 본 논문에서 제안된 연산자를 적용하였다. GA2는 가변길이 정수 인코딩을 사용하여 50개의 초기 개체를 난수로 생성한다. 이렇게 생성된 초기 개체들은 수리 알고리즘을 거쳐 모든 노드가 주어진 연결성 또는 신뢰성 조건을 만족하기에 충분한 수의 진출입 링크를 갖도록 수리된다. 매 세대마다 GA2는 다음 세대 개체군에 49개의 자손이 새로 생성되어 추가될 때까지 다음 과정을 반복한다.

- (1) $k = 10$ 인 로그함수 선택 연산을 통해 두 부모 개체 x_1 과 x_2 를 현재 세대 개체군으로부터 선택한다.
- (2) 0.7의 확률로 다음과 같은 교차 및 수리 연산을 통해 두 자손 개체 x_3 과 x_4 를 생성한다. 머지 0.3의 확률로는 x_1, x_2 를 그대로 x_3, x_4 에 복사한다.
A : x_1 과 x_2 에 공통적으로 존재하는 링크들을 두 자손 개체 x_3 과 x_4 에 복사한다.

B : 만약 x_3 이 연결성을 만족하면 D로 이동한다.

C : x_3 가 연결성을 만족할 때까지 x_1 에는 존재하지만 x_3 에 존재하지 않는 링크를 x_3 에 하나씩 추가한다.

D : B, C를 x_4 에 대해서도 반복한다.

E : x_1 의 어떤 링크가 x_3 에 존재하지 않는지 확인하고 이 링크들을 x_4 에 추가한다.

F : x_2 의 어떤 링크가 x_4 에 존재하지 않는지 확인하고 이 링크들을 x_3 에 추가한다.

- (3) 0.3의 확률로 돌연변이 연산과 수리 과정을 거친 x_3 과 x_4 를 다음 개체군에 복사한다. 만약 다음 개체군에 48개의 개체가 존재하면 x_3 만을 복사한다.

다음 세대 개체군의 마지막 한 자리는 현재 세대 개체 가운데 가장 적합도가 뛰어난 개체가 차지한다. 교차 및 돌연변이 연산의 확률은 원 논문에서 구한 값을 그대로 사용하였다.

참고문헌

- [1] 전영재, 김재철, “시뮬레이티드 어닐링을 이용한 배전 계통 재구성”, 한국지능정보시스템학회 학술대회지(1999), 195~202.
- [2] 원진명, “소형자동케도차량 시스템의 그래프 모델 기반 수송능력 추정”, 제어자동화시스템공학 논문지, 게재확정.
- [3] Garey, M. R. and D. S. Johnson, *Computers and Intractability: A Guide to the Theory*, Freeman, San Francisco, 1979.
- [4] Aggarwal, K. K., Y. C. Chopra, and J. S. Bajwa, “Topological layout of links for optimising the overall reliability in a computer communication system”, *Microelectronics and Reliability*, Vol.22, No.3(1982), 347~351.
- [5] Ventetsanopoulos, A. N. and I. Singh, “Topological optimization of communication networks subject to reliability constraints”, *Problem of Contr. Inform. Theory*, Vol.15 (1986), 63~78.
- [6] Glover, F., M. Lee, and J. Ryan, “Least-cost network topology design for a new service: An application of a tabu search”, *Annals of Operations Research*, Vol.33(1991), 351~362.
- [7] Pierre, S., M.-A. Hyppolite, J.-M. Bourjolly, and O. Dioume, “Topological design of computer communication networks using simulated annealing”, *Engineering Applications of Artificial Intelligence*, Vol.8(1995), 61~69.
- [8] Randall, M., G. McMahon, and S. Sugden, “A simulated annealing approach to communication network design”, *Journal of Combinatorial Optimization*, Vol.6(2002), 55~65.
- [9] Kumar, A., R. M. Pathak, and Y. P. Gupta, “Genetic algorithm based reliability optimization for computer network expansion”, *IEEE Transactions on Reliability*, Vol.44 (1995), 63~72.
- [10] Dengiz, B., F. Altıparmak, and A. E. Smith, “Local search genetic algorithm for optimal design of reliable networks”, *IEEE Transactions on Evolutionary Computation*, Vol.1(1997), 179~188.
- [11] Pierre, S. and G. Legault, “A genetic algorithm for designing distributed computer network topologies”, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol.28, No.2(1998), 249~258.
- [12] Gen, M., R. Cheng, and S. S. Oren, “Network design techniques using adapted genetic

- algorithms", *Advances in Engineering Software*, Vol.32, No.9(2001), 731~744.
- [13] Chou, H., G. Premkumar, and C.-H. Chu, "Genetic algorithms for communications network design-an empirical study of the factors that influence performances", *IEEE Transactions on Evolutionary Computation*, Vol.5, No.3(2001).
- [14] Won, Jin-Myung, F. Karray, K.-M. Lee, and J. S. Lee, "Guideway network design of personal rapid transit system: a multiobjective genetic algorithm approach", in *Proc. 2006. IEEE Cong. Evol. Comput., Vancouver BC, Jul(2006)*, 7979~7985.
- [15] Anderson, J. E., "Fundamentals of personal rapid transit", in *Proc. 4th Int. Conf. Automated People Movers*, (1993), 516~524.
- [16] Anspach, G., "PRT 2000 operational characteristics and modeling", in *Proc. 6th Int. Conf. Automated People Movers*, (1997), 497~507.
- [17] Floyd, R. W., "Algorithm 97: shortest path", *Communications of ACM*, Vol.5(1962), No.6, 345~1962.
- [18] Lawler, E., J. Lenstra, A. Rinnooy Kan, and D. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, New York: Wiley, 1985.
- [19] Jan, R.-H., F.-J. Hwuang, and S.-T. Chen, "Topological optimization of a communication network subject to a reliability constraint", *IEEE Transactions on Reliability*, Vol.42 No.1(1993), 63~70.
- [20] Deeter, D. K. and A. E. Smith, "Heuristic optimization of network design considering all-terminal reliability", in *Proc. Reliability and Maintainability Symp.*, (1997), 194~199.

Abstract

A Genetic Algorithm for Guideway Network Design of Personal Rapid Transit

Jin-Myung Won*

In this paper, we propose a customized genetic algorithm (GA) to find the minimum-cost guideway network (GN) of personal rapid transit (PRT) subject to connectivity, reliability, and traffic capacity constraints. PRT is a novel transportation concept, where a number of automated taxi-sized vehicles run on an elevated GN. One of the most important problems regarding PRT is how to design its GN topology for given station locations and the associated inter-station traffic demands. We model the GN as a directed graph, where its cost, connectivity, reliability, and node traffics are formulated. Based on this formulation, we develop the GA with special genetic operators well suited for the GN design problem. Such operators include steady state selection, repair algorithm, and directed mutation. We perform numerical experiments to determine the adequate GA parameters and compare its performance to other optimization algorithms previously reported. The experimental results verify the effectiveness and efficiency of the proposed approach for the GN design problem having up to 210 links.

Key words : Personal Rapid Transit, Network Optimization Problem, Genetic Algorithm



* Department of Electrical and Computer Engineering, University of Waterloo