

Rural Postman Problem 해법을 위한 휴리스틱 알고리즘

강 명 주[†] · 한 치 근^{††}

요 약

본 논문에서는 Rural Postman Problem(RPP) 해법으로 2가지의 휴리스틱 알고리즘을 제안한다. 첫 번째 휴리스틱 알고리즘으로 냉각 스케줄을 향상시킨 Simulated Annealing(SA) 알고리즘을 제안하였고, 두 번째로는 문제의 특성인 주어진 에지들을 모두 나타낼 수 있는 염색체 구성 방법을 포함한 유전자 알고리즘을 제안하였다. 실험 계산을 통하여 제안된 두 방법이 기존의 방법보다 우수함을 보였다.

Heuristic Algorithms for Rural Postman Problems

Myung-Ju Kang[†] · Chi-Geun Han^{††}

ABSTRACT

This paper proposes two kinds of heuristic algorithms for Rural Postman Problems (RPPs). One is a Simulated Annealing(SA) algorithm for RPPs. In SA, we propose a new cooling schedule which affects the performance of SA. The other is a Genetic Algorithm(GA) for RPPs. In GA, we propose a chromosome structure for RPPs which are edge-oriented problems. In simulations, we compared the proposed methods with the existing methods and the results show that the proposed methods produced better results than the existing methods.

1. 서 론

그래프 및 네트워크 문제는 실생활의 다양한 문제들을 풀기 위해 적용되는 가장 기본적인 문제이다. 그중 여러 분야에서 응용되는 라우팅 문제들은 Chinese Postman Problem(CPP)이나 Traveling Salesman Problem(TSP), Vehicle Routing Problem(VRP) 그리고 Rural Postman Problem(RPP)등이 있다.

CPP는 모든 에지를 한 번 이상 경유하여 처음 위치

로 되돌아오는 최소 비용 경로를 구하는 문제이며, TSP는 모든 노드를 반드시 한번만 경유하여 처음 위치로 되돌아오는 최소 비용 경로를 구하는 문제이다[3, 6, 7]. 그리고, VRP는 속성이 다른 노드들로 구성된 집합들을 용량이 한정된 최소한의 차량으로 한번씩만 모두 경유하여 처음 위치로 되돌아오는 최소 비용 경로 문제이며[1, 3, 11], RPP는 주어진 특정 집합에 속한 에지들을 각각 한번 이상 경유하여 처음 위치로 되돌아오는 최소 비용 경로 문제이다[3, 10].

CPP는 다항함수시간(Polynomial Time) 내에 문제를 풀 수 있으나, TSP, VRP, 그리고 RPP는 NP-Complete

[†] 정 회 원 : 경희대학교 전자정보학부 교수
^{††} 정 회 원 : 경희대학교 전자정보학부 교수
논문접수 : 1999년 1월 29일, 심사완료 : 1999년 8월 5일

문제로 알려져 있다[3]. RPP는 우편배달 경로 설정 문제, 버스 노선 설정 및 스케줄링 문제, 통신에서의 라우팅 문제, 그리고 지리정보시스템(GIS)에서의 라우팅 최적화 문제 등과 같은 많은 실생활 문제에 적용이 가능하다. 따라서, RPP는 이러한 실생활의 여러 가지 문제에 응용될 수 있는 중요한 문제로서 좋은 효율의 해법을 위한 알고리즘 연구가 필요하다.

본 논문에서는 RPP의 해법으로 전역 최적화(Global Optimization) 문제에 많이 사용되는 2가지의 휴리스틱 알고리즘들을 제안하고, 각 알고리즘의 실험 결과로부터 알고리즘의 성능을 비교 분석한다.

본 논문의 구성은 1장 서론에 이어 2장에서는 RPP 문제의 정의와 적용 문제들을 살펴보고, 3장에서는 SA를 이용한 RPP 해법을 제안하고, SA의 성능에 큰 영향을 주는 냉각 스케줄을 제안한다. 그리고 4장에서는 유전자 알고리즘을 이용한 RPP 해법을 제안하고, 유전자 알고리즘에서 사용되는 새로운 염색체 구성 방법을 제안하며, 5장에서는 실험을 통한 각 알고리즘의 성능을 비교 분석한 후 결론을 맺도록 한다.

2. Rural Postman 문제 정의 및 적용 문제

RPP는 주어진 특정 집합에 속한 에지들을 각각 한번 이상 경유하여 처음 위치로 되돌아오는 최소 비용 경로 문제이다. 즉, 노드의 집합 V 와 에지의 집합 E , 그리고 반드시 한번 이상 경유해야 하는 에지의 집합 $E'(\subseteq E)$ 으로 구성된 무방향 그래프 $G=(V, E, E')$ 에서 에지의 집합 E' 을 반드시 한번 이상 모두 거치는 최소 비용의 경로를 구하는 문제로서, NP-Complete 문제로 알려져 있다[3, 7].

RPP는 노드 중심의 문제인 TSP와는 달리 에지 중심의 문제이다. 즉, 실제 도로상에서 교차점을 중심으로 라우팅이 이루어지는 것이 TSP가 되며, 도로를 중심으로 라우팅이 이루어지는 것이 RPP가 된다.

다음은 RPP를 위한 매개변수 및 수식을 정의한 것이다.

매개변수 :

- $e_i = (e_i^1, e_i^2) (\in E') : E'$ 의 i 번째 에지.
- $e_i^1 (\in V) : i$ 번째 에지($\in E'$)의 라우팅 시작노드.
- $e_i^2 (\in V) : i$ 번째 에지($\in E'$)의 라우팅 끝노드.
- $d_{e_i, e_{i+1}} : e_i^2$ 에서 e_{i+1}^1 까지의 최소 비용. 즉, 디코딩

명(SA와 GA에서의 스트링을 해석하여 해를 얻어내는 과정)에서 생성된 라우팅에서 i 번째 에지의 끝 노드와 $(i+1)$ 번째 에지의 시작노드 사이에 경유되는 중간 경로의 최소 비용.

- $c_i (e_i \in E') : i$ 번째 에지의 비용, 즉, 디코딩에서 생성된 라우팅에서 E' 에 속하는 i 번째 에지의 시작노드와 끝노드 사이의 비용.

$C :$ 라우팅 비용 목적함수.

RRP의 정의에 의해, 방문하는 각 에지에서의 비용과 다음에 방문하는 에지까지의 최소 비용을 모두 구함으로써 RPP의 최소 비용을 구할 수 있다. (식 1)은 RPP의 최소 라우팅 비용을 구하기 위한 목적함수이다.

목적함수 :

$$\text{Minimize } C = \sum_{i=1}^n (c_{e_i} + d_{e_i, e_{i+1}}) \quad (\text{식 1})$$

여기서, i 는 방문하는 에지의 순서를 나타내며, $n=|E'|$ 이다. 그리고 만일 $i=n$ 이면 $i+1=1$ 로 한다. 즉, 시작 지점에서 출발하여 시작 지점에서 종료한다.

3. Simulated Annealing 알고리즘을 이용한 문제 해법

3.1 Simulated Annealing 알고리즘

Simulated Annealing(SA) 알고리즘은 금속의 담금질로부터 프로그램의 알고리즘으로 적용된 것이다[12]. 금속의 담금질과 비슷하게 SA 알고리즘에서는 초기 온도를 높은 온도로 설정하여 불안정한 상태에서 해 공간 탐색이 이루어지고, 점차 온도를 낮추면서 평형 상태를 유지하여 최적 해를 찾는 알고리즘이다. 즉, SA 알고리즘은 온도와 에너지 변화량에 대한 특정 함수로 이루어진 Metropolis 알고리즘에 따른 확률적 방법으로 지역 최소해를 탈피하는 방법 중의 하나이다. 또한, SA 알고리즘은 냉각 스케줄에서 온도를 어떻게 변화시키느냐에 따라 해의 결과와 수렴 속도에 영향을 미친다. 따라서, 본 논문에서는 새로운 냉각 스케줄의 온도 변화식을 제안하고, 기존에 많이 사용하는 냉각 스케줄과 제안한 냉각 스케줄을 이용한 SA 알고리즘의 성능을 분석한다.

3.2 벡터 구성

SA 알고리즘에서 해를 구성하는 벡터는 에지의 집합 E 와 노드의 집합 V , 그리고 반드시 경유해야 하는 에지의 집합 $E'(\subseteq E)$ 으로 구성된 무방향 그래프 $G = (V, E, E')$ 에서 E' 의 모든 원소들로 이루어진다. 이때, 해를 구성하는 벡터에는 서로 중복이 없고, 또한, 한 번씩만 나타나도록 하여 모든 E' 의 에지들을 최소로 경유할 수 있도록 한다. 구성된 벡터의 특징은 벡터를 구성하는 원소들의 순서가 라우팅의 순서가 되고, 벡터의 각 원소(에지 정보)들 사이의 중간 라우팅 경로 및 비용은 각 원소를 구성하는 노드들 간의 최단 경로와 최소 비용을 할당하게 된다.

3.3 냉각 스케줄

냉각 스케줄은 SA 알고리즘에서의 해 공간 탐색에 있어서 유한회의 탐색을 가능하게 하기 위한 것이다. 일반적으로 초기 온도를 높게 설정하여 서서히 냉각 스케줄에 따라 감소시킴으로써, SA의 초기 상태에서는 이웃해로의 진이가 가능한 많이 이루어지도록 하여 지역 최소해를 탈피하도록 하며, 상태가 진행됨에 따라 낮은 온도를 유지함으로써 안정된 상태로 진행된다. 그러나 적용되는 문제에 따라 어떤 냉각 스케줄을 사용하느냐가 SA 알고리즘의 성능을 결정하게 되며, 적합한 냉각 스케줄의 선택은 실험적으로 얻을 수 있다 [8].

본 논문에서는 기존에 많이 사용되는 2개의 냉각 스케줄과 본 논문에서 제안하는 1개의 냉각 스케줄을 이용한 SA를 10개의 RPP 문제에 적용하여, 각 냉각 스케줄에 대해 해의 수렴을 살펴보았다. (식 2)~(식 4)는 SA에 적용된 냉각 스케줄의 함수들을 나타내고 있다. 여기서 T_0 는 초기 온도를 그리고 $T(k)$ 는 k 번째 상태의 온도를 나타낸다.

$$T(k) = 0.9 \times T(k-1) \quad (\text{식 } 2)$$

$$T(k) = \frac{T_0}{\log(1+k)} \quad (\text{식 } 3)$$

$$T(k) = \frac{T_0}{1 + \sqrt{0.1 \times k}} \quad (\text{식 } 4)$$

일반적으로 기존의 SA 알고리즘에서는 냉각 스케줄을 위한 함수로 (식 2)와 (식 3)을 많이 사용하며 [8, 12], 본 연구에서는 RPP를 위한 냉각 스케줄의 온도 변화식으로 (식 4)를 제안한다. (식 2)는 (식 4)에 비해 낮은 온도 상태로 너무 일찍 내려가서 안정된 상태를

유지함으로써 조기 수렴(Premature Convergence)이 될 가능성이 높고, (식 3)은 (식 4)에 비해 높은 온도에서 시작되고 또한 보다 일찍 안정된 상태를 유지하지만 일정한 상태 이후에는 (식 4)가 더 낮은 온도에서 안정된 상태를 유지하게 된다.

3.4 이웃해 생성 방법

본 논문에서는 이웃해 생성 방법으로서 삽입 방법을 사용한다. TSP와 같은 노드 중심의 문제나, 혹은, CPP와 RPP와 같은 에지 중심의 문제에서는 해의 구성이 정수로 이루어지기 때문에 일반적인 이웃해 생성 방법인 Flip 방법을 사용할 수 없다. 즉, 이진 스트링에서와 같이 선택된 코드를 반전시키는 방법은 사용할 수 없다. 따라서, 본 연구에서는 삽입 방법을 이웃해 생성 방법으로 사용한다.

4. 유전자 알고리즘을 이용한 문제 해법

4.1 유전자 알고리즘

유전자 알고리즘은 다윈의 진화론에 근거한 알고리즘으로 자연의 법칙과 적자생존의 원리를 이용한 것이다. 유전자 알고리즘에서는 모집단(Population)이라는 개체들의 집합을 구성하여 이 집합의 개체들로부터 우성의 성질을 갖는 개체들에 대해 유전 연산자를 통해 다음 세대로 진화시켜 나가는 방법이다. 유전자 알고리즘에서의 각 염색체들은 문제의 완전한 해의 구조를 갖는 스트링(String)으로 표현되며, 이 스트링에 대한 해석을 통해 그 염색체의 우성과 열성의 성질이 결정된다. 즉, 문제의 해로부터 그 염색체의 적합(Fitness) 정도를 측정하기 위한 함수를 통해 그 염색체의 상대적인 우성 정도를 판별하게 된다. 적합 함수(Fitness Function)에 따라 우성의 성질을 갖는 염색체들은 다음 세대로 진화할 가능성이 높으며, 반면, 열성의 성질을 갖는 염색체들은 도태되어 더 이상 다음 세대로 진화가 이루어지지 않는다. 이런 방법을 통해 최적화 문제에서는 좋은 해를 갖는 스트링들을 계속 진화시켜 나감으로써 최적해를 찾는다. 본 논문에서는 RPP를 위한 염색체 구성 방법을 제안하고, 기존의 유전자 알고리즘과 비교하여 성능을 분석한다.

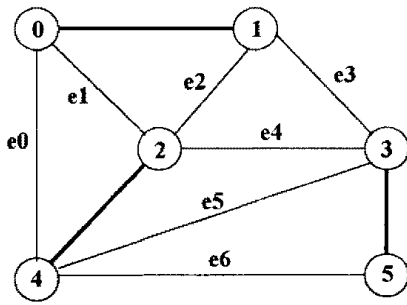
4.2 염색체 구성 방법

4.2.1 기존의 염색체 구성 방법

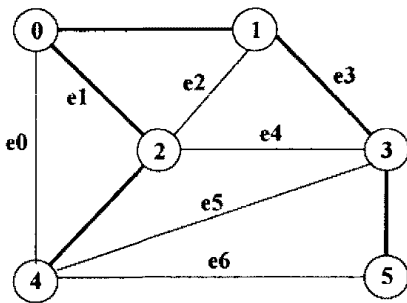
기존의 염색체 구성 방법은 노드의 집합 V , 에지의

집합 E' 반드시 경유해야 하는 에지의 집합 E' 으로 구성된 그래프 $G=(V, E, E')$ 에서 $E-E'$ 에 속한 에지들의 스트링으로 구성한다[2]. 탐색체에 속한 각 요소는 E' 에 속하지 않는 모든 에지들로 이루어지며, 그 에지들이 라우팅에 포함되는지의 여부에 따라 0과 1이 할당된다. 즉, RPP 라우팅을 위해 중간 에지로 포함되면 1로 할당하고, 그렇지 않으면 0을 할당하게 된다.

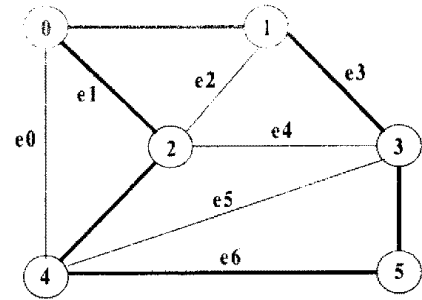
(그림 1)은 RPP 그래프를 나타낸 것으로 굵은 선으로 표시된 에지들이 E' 에 속하는 에지들이며 나머지 에지들은 $E-E'$ 에 속하는 에지들이다. (그림 2)는 (그림 1)에서 $E-E'$ 에 속한 에지들로 구성된 탐색체 스트링의 결과가 (0 1 0 1 0 0 0)인 경우를 나타낸 것이다. 따라서, 탐색체 스트링 중에 1로 할당된 에지들을 RPP의 중간 경유 에지로 사용하는 것이다. 그러나, 이와 같이 중간 경유 에지가 RPP에 포함되더라도 (그림 2)에서처럼 RPP을 위한 완전한 라우팅이 이루어지지 않는다. 따라서, 보정 연산에 의해 (그림 3)과 같이 완전한 라우팅이 이루어지도록 해야하는 단점이 있다. 기존 방법에서의 보정 연산은 E' 과 탐색체 스트링에서 1로 할당된 에지들에 대해 오일러 그래프가 될 수 있도록 각 노드의 차수를 짝수로 만들어 줌으로써 가능하다.



(그림 1) RPP 그래프



(그림 2) 탐색체 스트링이 (0 1 0 1 0 0 0)으로 구성된 경우에 RPP 그래프에 중간 경유 에지들이 추가된 그래프



(그림 3) 보정 연산이 수행된 후의 그래프

4.2.2 제안 탐색체 구성 방법

본 논문에서는 그래프 변환에 의한 탐색체 구성 방법을 제안한다. 기존의 탐색체 구성 방법에서는 두 가지 정보, 즉, 에지 정보와 방향정보를 이용하여 탐색체를 구성하기 때문에 탐색체의 크기가 $2 \times |E'|$ 가 된다. 또한, 에지 정보의 표현은 E' 에 속한 에지들의 번호로 이루어져야 하고, 또한, 방향 정보의 표현은 2진 코드로 이루어진다. 따라서, 유전 연산자를 적절하게 적용해야 하는 어려움이 있다. 본 논문에서는 이러한 단점을 극복하고 단일 정보만을 적용하기 위해 RPP 그래프를 완전연결 그래프로 변환한 후 탐색체를 구성하는 방안을 제안한다.

RPP에서 반드시 한번 이상 지나야 하는 에지의 집합 $E'(\subseteq E)$ 을 갖는 그래프 $G=(V, E, E')$ 가 주어지고, 완전연결 그래프에서의 노드집합, V^H , 에지의 집합, E^H 로 구성된 그래프 $G^H=(V^H, E^H)$ 라고 할 때, 다음은 그래프 변환에 의한 탐색체 코딩 과정을 나타내고 있다. 본 논문에서는 그래프 변환에 의한 탐색체 구성 방법을 GCS (Graph-Transformation Chromosome Structure)라고 부르기로 한다.

단계 1

E' 에 속하는 모든 에지 e_i 상에 우편함의 위치를 나타내는 노드 v_i^H 를 설정한다. 여기서, v_i^H 는 V^H 에 속하는 하나의 노드이다.

단계 2

완전연결 그래프의 노드 집합인 V^H 에 속하는 노드, v_i^H 와 v_j^H ($i \neq j$)에 대해, 완전연결 그래프의 에지(v_i^H, v_j^H)($\in E^H$)를 생성한다. 이 때, v_i^H 와 v_j^H 사이의 에지와 실제 RPP 그래프에서의 Path와의 매핑 (Mapping) 정보는 최종해를 구하기 위해 임시 테이블

에 저장한다.

단계 3

완전연결 그래프의 노드 집합, V^H 의 각 노드 정보를 이용하여 염색체를 구성한다.

4.3 유전 연산자

4.3.1 교차연산

교차연산은 실제로 선택된 스트링 한 쌍에 대한 실질적인 진화 조작이다. 교차연산시 일반적인 교차연산 방법을 본 문제에 그대로 사용하는 것은 의미가 없다. 왜냐하면 성질이 다른 스트링간에 교차연산을 허용할 경우 스트링 구조가 깨지기 때문이다. 따라서 본 논문에서는 이러한 짐을 해결하기 위해서 Partially Matched Crossover (PMX)를 이용하였다[4, 5, 9]. 이 방법에서 스트링의 교차연산은 상대 스트링과의 부분적 대응(Match)을 이용하여 자신의 스트링에서만 내부적으로 노드들의 위치를 바꾸어 나감으로써 스트링구조를 만족하면서 스트링을 변화시켜 나간다.

4.3.2 돌연변이연산

돌연변이연산은 교차연산과는 달리 2개의 스트링을 이용해서 새로운 2개의 스트링을 만드는 것이 아니라 하나의 스트링을 부분적으로 조작하여 새로운 스트링을 만들어 내는 조작이다. 즉, 돌연변이연산은 교차연산이 실질적인 진화 동작을 하는 것과는 달리 하나의 스트링 상에서 변화를 줌으로써 교차연산시 지나칠 수도 있는 상태를 만들어 내는 오퍼레이터이다. 본 연구에서는 두 가지 돌연변이연산 방법을 사용하고있다. 첫 번째는 교환 방법(Exchange Method)으로써, 선택된 염색체의 스트링 중 두 노드를 선택해서 맞바꾸는 것이고(mutation1), 두 번째는 반전 방법(Inversion Method)으로써, 선택된 염색체의 스트링 중 두 곳의 지점을 택해 두 지점 사이의 스트링을 반전하는 것이다(mutation2).

4.4 적합 함수

본 연구에서는 새로운 세대의 모집단을 생성하기 위한 선택 기법으로 Roulette Wheel 방법을 사용한다[4]. 선택 연산을 수행하기 위해서는 각 염색체에 대한 적합함수(Fitness Function) 값이 필요한데, 본 연구에서는 제 2장에서 설명한 RPP의 목적 함수를 이용하여 (식 5)와 같은 적합함수 값을 구하고, (식 6)에 의해

염색체 선택이 이루어진다.

RPP를 위한 적합 함수 $F(k)$:

$$F(k) = \left(\frac{1}{C(k)}\right)^m \quad (\text{식 5})$$

여기서, $C(k)$ 는 2장에서 설명한 k 번째 염색체의 목적함수를 나타내고, m 은 스케일링(Scaling)인자로서 0보다 큰 정수이다. 이 스케일링 인자는 우성과 열성의 차이를 크게 해 줌으로써 우성인자를 갖는 염색체의 선택 확률을 높여준다.

(식 6)은 Roulette Wheel 선택 방법에서 사용하기 위한 각 염색체의 선택확률을 계산하기 위한 수식이다.

선택 확률 $P(k)$:

$$P(k) = \frac{F(k)}{\sum_{i=1}^n F(i)} \quad (\text{식 6})$$

여기서, $P(k)$ 는 k 번째 개체의 상대적인 적합함수 값으로서, 염색체의 상대적인 우성 정도를 나타낸다.

5. 실험 결과

5.1 실험 환경

서로 다른 알고리즘들의 성능 분석 및 비교를 위해서는 동일한 실험 환경이 중요하다. 따라서, 본 연구에서는 IBM PC Pentium Pro 200에서 실험이 수행되었으며, 적용된 RPP 문제의 총수는 10개이다. 그리고, SA 알고리즘에서의 이웃해 생성 방법은 삼입 방법을 사용하였으며, 냉각 스케줄은 (식 2), (식 3), 그리고 (식 4)를 적용하였다. 알고리즘의 종료조건은 100,000회의 총 비용 계산 횟수로 한정하였다. 유전자 알고리즘에서는 염색체 구성 방법으로 GCS 방법을 적용하고, 교차 연산으로 PMX를 0.6, 0.7, 그리고 0.8의 비율로 적용하였다. 그리고 돌연변이 연산으로 교환과 반전을 각각 0.03과 0.04의 비율로 적용하였으며, 선택 방법으로 Roulette Wheel 방법을 사용하였다. 그리고 알고리즘 종료 조건은 SA와 동일하도록 설정하였다.

5.2 SA 알고리즘의 실험 결과

<표 1>은 10개의 RPP 문제에 대해 일반적으로 냉각 스케줄에 많이 사용되는 온도 계산 방법(식 2)와 (식

3)에 의한 SA 알고리즘과 본 논문에서 제안한 온도 계산 방법(식 4)에 의한 SA 알고리즘의 근사최적해 및 Count를 비교한 것이다. 여기서, Count는 처음 근사최적해를 찾을 때까지의 목적함수 계산 횟수를 나타낸다.

<표 1>에서도 알 수 있듯이 (식 2)를 적용한 냉각 스케줄을 사용한 경우에는 지역 최소해를 탈피할 수 없음을 알 수 있다. 이는 (식 2)를 적용한 냉각 스케줄이 너무 일찍 그리고 너무 낮은 온도에서 안정된 상태를 유지하기 때문에 Metropolis 알고리즘을 적용할 기회가 적어짐으로 인해 이웃해로의 전이가 거의 이루어지지 않기 때문이다. 반면, (식 3)을 적용한 냉각 스케줄을 사용한 경우에는 (식 4)를 이용한 냉각 스케줄과 비교해 볼 때, 초기 상태에서는 (식 4)보다 낮은 온도를 유지하지만 상태가 진행됨에 따라 오히려 높은 온도를 유지한다. 따라서, (식 3)을 적용할 경우에는 (식 4)를 적용한 경우보다 불안정한 상태가 되어 최소해로 탐색이 이루어지기보다는 이웃해로의 전이가 많이 이루어지기 때문에 (식 4)를 적용한 냉각 스케줄이 대체로 좋은 결과를 보이고 있다.

<표 1> 기존 방법과 제안한 방법의 근사최적해 비교

문 제	냉각 스케줄 유형					
	식 2		식 3		식 4	
	근사 최적해	Count	근사 최적해	Count	근사 최적해	Count
1	295	499	290	11,030	290	34,300
2	295	8,201	318	93,800	292	58,913
3	264	64,680	291	72,969	257	99,911
4	277	16,205	286	54,733	262	76,606
5	307	14,890	306	24,695	301	63,795
6	329	1,751	337	35,541	325	86,980
7	284	1,893	291	97,240	284	115,921
8	405	81,671	501	56,979	421	113,921
9	358	6,119	424	88,859	341	96,279
10	385	47,169	438	55,835	401	40,753

5.3 유전자 알고리즘의 실험 결과

<표 2>는 기존의 탐색체 구성 방법을 이용한 유전자 알고리즘과 본 연구에서 제안한 탐색체 구성 방법(GCS)을 이용한 유전자 알고리즘의 결과를 비교한 것으로서, 각 방법에 대해 36가지의 매개변수 조합으로부터 생성된 해들 중 최적인 해를 비교한 것이다.

<표 2>의 결과에서도 알 수 있듯이 본 논문에서 제안한 탐색체 구성 방법인 GCS를 이용한 유전자 알고리즘의 결과가 기존의 탐색체 구성 방법을 이용한 유

전자 알고리즘의 결과보다 많은 경우에 있어서 좋은 결과를 얻고 있음을 알 수 있다. 이는 기존의 방법에서는 EE에 속하는 에지들로 탐색체를 구성하기 때문에 탐색체의 크기가 GCS 방법을 이용한 탐색체의 크기보다 크고, 또한, 해 공간의 크기가 커지기 때문에 탐색 범위가 넓어진다. 따라서, 최적해의 탐색 가능성이 그만큼 줄어든다. 그리고, 탐색체의 구성 요소에서 RPP 라우팅을 위한 중간 경유 에지로 선택되는 에지에 따라 오일러 그래프의 결과가 달라지기 때문에 GCS 방법에 비해 효율적으로 탐색하는 것이 어렵다.

<표 2> 기존 탐색체 구성 방법을 이용한 유전자 알고리즘과 GCS 방법을 이용한 유전자 알고리즘의 결과 비교

문제	기존 방법	GCS
1	290	290
2	284	287
3	262	255
4	267	265
5	306	299
6	327	331
7	287	297
8	521	420
9	452	369
10	477	396

5.4 알고리즘 비교 분석

본 연구에서는 실험에 적용된 10개의 RPP에 대해 제안한 냉각 스케줄을 이용한 SA 알고리즘, 그리고 GCS를 이용한 유전자 알고리즘에 대해 근사최적해를 비교 분석하였다.

<표 3>은 각 알고리즘들에 대한 근사최적해를 비교한 것이다. 표에서도 알 수 있듯이 유전자 알고리즘은 5개의 문제에 대해, SA 알고리즘은 4개의 문제에 대해 가장 좋은 해를 찾고 있으며, 나머지 1개는 동일한 해를 찾고 있다. <표 3>에서 SA 알고리즘에서의 괄호 안에 있는 수치는 근사최적해를 처음으로 찾았을 때의 루프 횟수를 나타내며, 유전자 알고리즘에서의 괄호 안의 수치는 근사최적해를 처음 찾은 세대를 나타내고 있다. 그리고 유전자 알고리즘에서의 모집단 크기는 100이므로, 근사최적해가 계산될 때까지의 비용 계산 횟수는 다음의 범위 안에 들어간다.

$$(세대수-1) \times 100 < \text{비용 계산 횟수} < \text{세대수} \times 100$$

〈표 3〉 RPP에 대한 휴리스틱 알고리즘의 비교

문제	라우팅 비용	
	SA결과(Count)	GA결과(Generation)
1	290(34,300)	290(347)
2	292(58,913)	287(540)
3	257(99,911)	255(320)
4	262(76,606)	265(614)
5	301(63,795)	299(401)
6	325(86,980)	331(986)
7	284(43,465)	297(783)
8	421(93,921)	420(940)
9	341(96,279)	369(946)
10	401(40,753)	396(831)

6. 결 론

본 논문에서는 최적화 문제에 많이 적용되는 휴리스틱 알고리즘 중, SA 알고리즘과 유전자 알고리즘을 NP-Complete 문제로 알려진 RPP 문제의 해법으로 적용하였다.

SA 알고리즘에서는 알고리즘의 성능에 중요한 요소가 되는 냉각 스케줄을 제안하였으며, 또한, 기존에 많이 사용되는 냉각 스케줄과 본 연구에서 제안한 냉각 스케줄 방법과의 성능을 비교 분석하였다. 실험 결과에서 초기 상태에서는 기존 방법 보다 많은 횟수의 Metropolis 알고리즘이 적용되어 이웃해 탐색이 많이 이루어지고, 일정한 상태 이후에는 안정된 상태에서 보다 좋은 근사최적해를 탐색하도록 하여, RPP에 대해 제안한 SA 알고리즘이 좋은 결과를 얻을 수 있었다.

유전자 알고리즘에서는 RPP를 위한 염색체 구성 방법을 제안하였으며, 기존 방법을 구현하여 제안한 알고리즘과 성능을 비교 분석하였다. 실험 결과에서, RPP에 대해 GCS가 기존 방법에 비해 좋은 결과를 얻을 수 있었다.

본 논문에서는 적용된 휴리스틱 알고리즘 중에서 제안한 냉각 스케줄을 이용한 SA 알고리즘, 그리고 GCS를 이용한 유전자 알고리즘에 대한 성능 분석이 이루어졌다. 본 논문에서의 실험환경으로, 각 알고리즘의 종료조건은 목적함수의 계산횟수를 동일하게 주어 종료하도록 하였다. 실험 결과를 통해 RPP 문제에 대해서는 GCS를 이용한 유전자 알고리즘이 SA 알고리즘에 비해 좋은 결과를 얻을 수 있었다.

향후 연구 방향으로서는 본 논문에서 제안한 알고리즘들을 이용하여 유향 그래프 및 아크와 에지가 혼재된

혼합 네트워크에 대해 적용하고, 또한, 지리정보시스템(GIS)과 같은 실제 응용 시스템과의 연동 방안을 연구하는 것이다. 그리고, 실제 도로 및 통신 네트워크 상에서 다양한 제약 조건 등을 고려하여 본 논문에서 제안한 휴리스틱 알고리즘을 적용하여 그 성능을 분석하고자 한다.

참 고 문 헌

- [1] Chambers, L., *Practical Handbook of Genetic Algorithms*, CRC Press, 1995.
- [2] Cook, C., Schoenefeld, D. A. and Wainwright, R. L., Finding Rural Postman Tours, *Proceedings of the 1998 ACM Symposium on Applied Computing*, pp.318-326, 1998.
- [3] Garey, M. R. and Johnson, D. S., *Computers and Intractability - A Guide to the Theory of NP-Completeness*, pp.213, FREEMAN, 1979.
- [4] Goldberg, D. E., *Genetic Algorithm in Search, Optimization & Machine Learning*, Addison-Wesley, 1989.
- [5] Ladd, S. R., *Genetic Algorithms in C++*, M&T Books, 1995.
- [6] Laporte, G., Asef-Vaziri, A. and Sriskandarajah, C., Some Applications of the Generalized Traveling Salesman Problem, *Journal of the Operational Research Society*, Vol.47, pp.1461-1467, 1996.
- [7] Lenstra, J. K. and Rinnooy Kan, A. H. G., On General Routing Problems., *Networks*, Vol.6, pp. 273-280, 1976.
- [8] Lin, Chin-Teng and Lee, C. S. George, *Neural Fuzzy Systems*, Prentice Hall, 1996.
- [9] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Second, Extended Edition, Springer-Verlag, 1994.
- [10] Pearn, W. L., and Wu, T. C., Algorithms for The Rural Postman Problem, *Computers Ops Res*, Vol.22, No.8, pp.819-828, 1995.
- [11] Thangiah, S. R., *Vehicle Routing with Time Windows using Genetic Algorithms*, Practical Handbook of GENETIC Vol.2, ALGORITHMS, CRC Press, 1995.

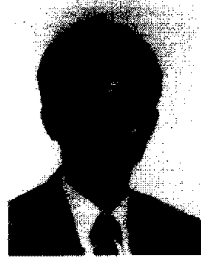
[12] 김대수, *신경망 이론과 응용* (1), 하이테크정보, 1992.



강 명 주

e-mail : mjkgang@nms.kyunghee.ac.kr
1993년 경희대학교 전자계산공학과 졸업(학사)
1995년 경희대학교 대학원 전자계산공학과(공학석사)
1998년 경희대학교 대학원 전자계산공학과(공학박사)

1999년~현재 경희대학교 전자정보학부 강의전임
관심분야 : Metaheuristics 알고리즘, 인터넷 & 멀티미디어, 라우팅 알고리즘 등



한 치 근

e-mail : cghan@nms.kyunghee.ac.kr
1983년 서울대학교 산업공학과 졸업(학사)
1985년 서울대학교 대학원 산업공학과(공학석사)
1988년 펜실베니아주립대학 전산학과(이학석사)

1991년 펜실베니아주립대학 전산학과(이학박사)
1991년 9월~1992년 2월 수원대학교 전자계산학과 조교수
1992년 3월~1995년 9월 경희대학교 전자계산공학과 조교수
1995년 10월~현재 경희대학교 전자정보학부 부교수
관심분야 : 정보통신망 알고리즘, 문서관리, 멀티미디어, 계산이론 등