

## 무향 Rural Postman Problem 해법을 위한 유전 알고리즘에서 그래프 변환에 의한 디코딩 알고리즘

강 명 주\*

### A Decoding Algorithm Using Graph Transformation in A Genetic Algorithm for Undirected Rural Postman Problems

Kang, Myung-Ju \*

#### 요 약

무향 Rural Postman Problem(URPP)은 주어진 네트워크에서 특정 아크를 적어도 한 번씩 경유하는 최단 경로를 찾는 문제이다. URPP는 실생활의 다양한 문제를 풀기 위한 기본적인 네트워크 문제 중에 하나이며 NP-Complete 문제로 알려져 있다. URPP는 아크 중심의 문제로 아크의 라우팅 방향을 고려하여야 하며, 노드 중심의 문제인 TSP(Traveling Salesman Problem) 해법을 그대로 적용하는 것은 힘들다. 본 논문에서는 URPP 해법을 위한 유전 알고리즘에서 그래프 변환에 의한 디코딩 방법을 제안한다. 즉, 아크 중심의 그래프를 노드 중심의 그래프로 변환함으로써 아크의 방향에 상관없이 전체 라우팅 경로를 구할 수 있도록 하였다. 실험을 통해 제안 알고리즘과 기존 알고리즘의 성능을 비교하였다. 실험 결과에서 제안 알고리즘은 기존 알고리즘보다 좋은 결과를 얻을 수 있음을 알 수 있었다.

#### Abstract

Undirected Rural Postman Problem(URPP) is a problem that finds a shortest tour traversing the given arcs at least once in a given network. The URPP is one of the basic network problems used in solving the various real-world problems. And it is known as NP-Complete. URPP is an arc-oriented problem that the direction of a tour in an arc has to be considered. Hence, in URPP, it is difficult to use the algorithm for Traveling Salesman Problem (TSP), which is a node-oriented problem, directly. This paper proposes the decoding algorithm using graph transformation in the genetic algorithm for URPP. That is, you can find the entire tour traversing without considering the direction of arcs by transforming the arc-oriented graph into the node-oriented graph. This

---

• 제1저자 : 강명주  
• 접수일 : 2007.3.14, 심사일 : 2007.4.11, 심사완료일 : 2007. 5.10.  
\* 청강문화산업대학 컴퓨터게임과 교수

paper compares the performances of the proposed algorithm with an existing algorithm. In the simulation results, the proposed algorithm obtained better than the existing algorithm

▶ Keyword : Undirected Rural Postman Problem, Genetic Algorithm, Decoding

## I. 서론

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

CPP는 모든 에지를 한 번 이상 경유하여 처음 위치로 되돌아오는 최소 비용 경로를 구하는 문제이며, TSP는 모든 노드를 반드시 한번만 경유하여 처음 위치로 되돌아오는 최소 비용 경로를 구하는 문제이다[1, 2, 3]. 그리고 VRP는 속성이 다른 노드들로 구성된 집합들을 용량이 한정된 최소한의 차량으로 한 번씩만 모두 경유하여 처음 위치로 되돌아오는 최소 비용 경로 문제이며[1, 4, 5], RPP는 주어진 특정 아크 또는 에지의 집합을 반드시 한번 이상 경유하여 처음 위치로 되돌아오는 최소 비용 경로 문제이다[1, 6].

특히, RPP는 우편배달 경로 설정 문제, 버스 노선 설정 및 스케줄링 문제, 통신에서의 라우팅 문제, 그리고 지리정보 시스템(GIS)에서의 라우팅 최적화 문제 등과 같은 실생활 문제에 적용이 가능하다. 또한, RPP는 NP-Complete 문제로 알려져 있으며 문제의 크기가 커질수록 해 공간이 폭발적으로 커짐에 따라 다항함수시간(Polynomial Time) 내에 최적 해를 탐색하는 것이 어렵다. 일반적으로 NP-Complete 문제는 휴리스틱 알고리즘을 이용한 해법이 많이 연구되고 있다.

따라서 본 논문에서는 휴리스틱 알고리즘들 중 유전 알고리즘을 이용한 무향 그래프에서의 RPP 문제 해법을 제안한다. 노드 중심의 TSP 문제와는 달리 아크 중심의 RPP 문제에서는 아크의 라우팅 방향을 고려해야 하기 때문에 TSP에 적용되는 유전 알고리즘으로는 해를 구하기 힘들다. 이를 위해 본 논문에서는 그래프 변환을 이용한 디코딩 알고리즘을 제안하고, 실험 결과를 통해 알고리즘의 성능을 분석한다.

## II. 무향 RPP 및 목적함수

### 2.1 무향 RPP

무향 RPP (URPP : Undirected Rural Postman Problem)은 노드의 집합  $V$ 와 아크의 집합  $A$ , 그리고  $A' (\subseteq A)$ 로 구성된 무향 그래프  $G = (V, A, A')$ 에서  $A'$ 의 모든 아크들을 적어도 한 번 이상 경유하는 최소 비용 경로를 구하는 문제이다.

그림 1에서  $a-a'$ ,  $b-b'$  그리고  $c-c'$ 이  $A'$ 에 속하는 아크들이고,  $a' \sim b$ ,  $b' \sim c$ ,  $c' \sim a$ 는  $A-A'$ 에 속하는 아크들로써 아직 알려지지 않은 경로의 일부를 나타내며  $A'$ 의 에지들을 경유하는데 사용되는 중간 경로들을 나타낸다.

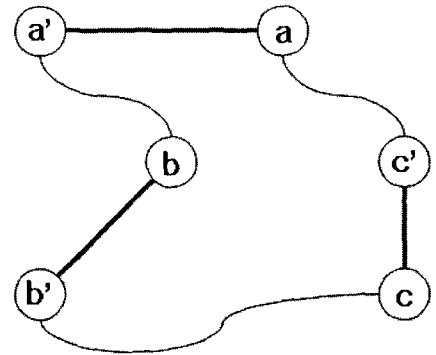


그림 1. 무향 RPP 그래프의 해  
Fig. 1. Solution of URPP

### 2.2 URPP를 위한 목적함수

다음은 URPP를 위한 매개변수 및 목적함수를 정의한 것이다.

매개변수 :

$$a_i = (a_i^1, a_i^2) (\in A') : A'의 i 번째 아크.$$

$a_i^1 (\in V)$  :  $i$  번째 아크( $\in A'$ )의 시작노드.

$a_i^2 (\in V)$  :  $i$  번째 아크( $\in A'$ )의 끝 노드.

$d_{a_i^2, a_{i+1}^1}$  :  $a_i^2$  에서  $a_{i+1}^1$  까지의 최소 비용. 즉, 유전알고리즘의 염색체 디코딩에서 생성된 라우팅에서  $i$  번째 에지의 끝 노드와  $(i+1)$  번째 아크의 시작노드 사이에 경유되는 중간 경로의 최소 비용.

$c_{a_i}$  ( $a_i \in A'$ ) :  $i$  번째 아크의 라우팅 비용, 즉, 유전알고리즘의 염색체 디코딩에서 생성된 라우팅에서  $A'$ 에 속하는  $i$  번째 아크의 시작노드와 끝 노드 사이의 비용.

$C$  : 라우팅 비용 목적함수.

URRP의 정의에 의해, 방문하는 각 아크에서의 비용과 다음에 방문하는 아크까지의 최소 비용을 모두 구함으로써 URPP의 최소 비용을 구할 수 있다. (식 1)은 URPP의 최소 라우팅 비용을 구하기 위한 목적함수이다.

목적함수 :

$$\text{Minimize } C = \sum_{i=1}^n (c_{a_i} + d_{a_i^2, a_{i+1}^1}) \dots \text{(식 1)}$$

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

### III. URPP를 위한 유전 알고리즘

#### 3.1 유전 알고리즘

그림 2는 일반적인 유전 알고리즘을 나타내고 있다. 그림 2에서 모집단은 한 세대를 구성하는 개체들의 집합을 나타낸다. 단계 (2)는 초기 세대의 모집단을 구성하는 단계로서, 랜덤하게 생성된다. 단계 (3)과 단계 (7)은 초기 세대의 모집단에 대한 각 염색체의 우성과 열성 정도를 판별하기 위한 평가 부분이다. 그리고, 단계 (4)~(8)은 유전자 연산의 반복 처리에 의한 진화과정을 나타내는 부분으로서, 자연 개체에서의 교배에 해당되는 교차연산과 돌연변이연산이 이루어진다. 교차연산은 선택된 두 염색체 사이에 조작이 이루어지며 이 과정에서 실제 염색체 교배가 이루어진다. 그리고 돌연변이연산은 선택된 하나의 염색체에 대해 조작

이 이루어지는 과정으로서, 교차연산에 의해 생성할 수 없는 상태를 만들기 위한 연산 과정이다. 단계 (5)는 현 세대의 모집단에서 다음 세대로 진화할 염색체들을 선택하여 새로운 모집단을 구성하는 과정으로서, 상대적으로 우성인 개체들이 선택될 가능성이 높도록 처리된다.

```

(1) Begin
(2) Initialize population;
(3) Evaluate population;
(4) While not maxgeneration
(5)   Select chromosomes for next population;
(6)   Crossover and Mutation;
(7)   Evaluate population;
(8) end while
(9) end Begin
    
```

그림 2. 유전자 알고리즘  
Fig. 2. Genetic Algorithm

#### 3.2 염색체 인코딩

##### 3.2.1 기존의 염색체 인코딩 방법

기존의 염색체 구성 방법은 노드의 집합  $V$ , 아크의 집합  $A$ , 반드시 경유해야 하는 아크의 집합  $A'$ 으로 구성된 그래프  $G=(V, A, A')$ 에서  $A-A'$ 에 속한 에지들에 대해 비트(Bit) 스트링으로 구성한다(7). 염색체에 속한 각 요소는  $A'$ 에 속하지 않는 모든 에지들로 이루어지며, 그 에지들이 라우팅에 포함되는지의 여부에 따라 0과 1이 할당된다. 즉, URPP 라우팅을 위해 중간 아크에 포함되면 1로 할당하고, 그렇지 않으면 0을 할당하게 된다.

그림 3은 URPP 그래프를 나타낸 것으로 굵은 선으로 표시된 아크들이  $A'$ 에 속하는 아크들이며 나머지 아크들은  $A-A'$ 에 속하는 에지들이다. 그림 4는 그림 3에서  $A-A'$ 에 속한 에지들로 구성된 염색체 스트링의 결과가 (0 1 0 1 0 0 0)인 경우를 나타낸 것이다. 따라서, 염색체 스트링 중에 1로 할당된 아크들을 RPP의 중간 경유 아크로 사용하는 것이다. 그러나 이와 같이 중간 경유 아크가 URPP에 포함되더라도 그림 4에서처럼 URPP를 위한 완전한 라우팅이 이루어지지 않는다. 따라서 보정 연산에 의해 그림 5와 같이 완전한 라우팅이 이루어지도록 해야 하는 단점이 있다. 기존 방법에서의 보정 연산은  $A'$ 와 염색체 스트링에서 1로 할당된 에지들에 대해 오일러 그래프가 될 수 있도록 각 노드의 차수를 짝수로 만들어 줌으로써 가능하다.

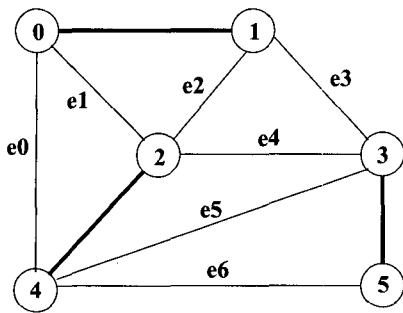


그림 3. URPP 그래프  
Fig. 3. URPP Graph

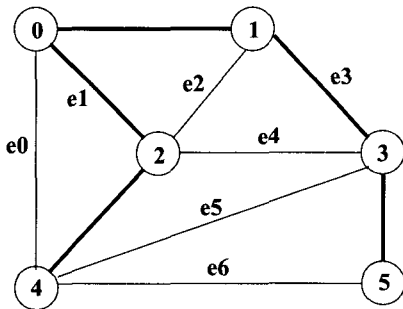


그림 4. 염색체 스트링이 (0 1 0 1 0 0 0)인 URPP 그래프  
Fig. 4. URPP Graph that the chromosome string is (0 1 0 1 0 0 0)

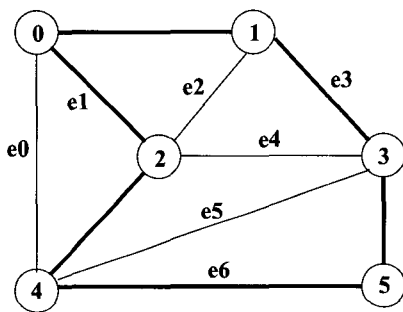


그림 5. 보정 연산 수행 후 그래프  
Fig. 5. Compensated URPP Graph

### 3.2.2 제안 염색체 인코딩 방법

기존의 염색체 인코딩 방법은 A-A'의 요소들을 염색체로 구성하는 반면, 본 논문에서 제안한 방법은 A'의 요소들을 염색체로 구성한다. 즉, 노드의 집합 V, 아크의 집합 A, 반드시 경유해야 하는 아크의 집합 A'으로 구성된 무향 그래

프  $G=(V, A, A')$ 에서 A'에 속한 아크들의 스트링으로 구성한다. 즉, URPP 그래프를 나타내는 그림 6에서 굵은 선으로 표시된 A'에 속하는 아크들인 {a, b, c}를 요소로 하는 스트링으로 구성한다.

예를 들어 그림 6의 그래프에서 염색체가  $p=(b\ c\ a)$ 로 구성되었다고 가정하면, 전체 라우팅 경로는  $b\sim(b\text{와 }c\text{의 최단경로})\sim c\sim(c\text{와 }a\text{의 최단경로})\sim a$ 가 된다.

전체 라우팅 경로는 본 논문에서 제안하는 디코딩 알고리즘에 의해 구하게 된다. 따라서 기존 연구에서의 단점, 즉, 보정 연산을 수행할 필요 없이 URPP의 해를 구할 수 있다. 본 논문에서 제안한 디코딩 알고리즘은 3.4절에서 설명하고 있다.

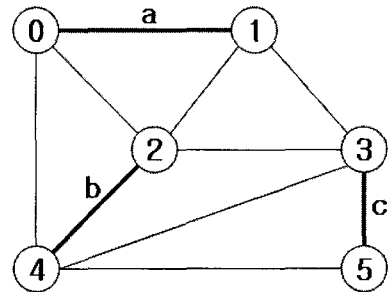


그림 6. URPP 그래프  
Fig. 6. URPP Graph

## 3.3 유전 연산자

### 3.3.1 교차연산

교차연산은 실제로 선택된 스트링 한 쌍에 대한 실질적인 진화 조작이다. 교차연산 시 일반적인 교차연산 방법을 본 문제에 그대로 사용하는 것은 의미가 없다. 왜냐하면 성질이 다른 스트링 간에 교차연산을 허용할 경우 스트링 구조가 깨지기 때문이다. 따라서 본 논문에서는 이러한 점을 해결하기 위해서 PMX(Partially Matched Crossover)를 이용하였다[8, 9, 10]. 이 방법에서 스트링의 교차연산은 상대 스트링과의 부분적 대응(Match)을 이용하여 자신의 스트링에서만 내부적으로 노드들의 위치를 바꾸어 나감으로써 스트링구조를 만족하면서 스트링을 변화시켜 나간다.

### 3.3.2 돌연변이 연산

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 하

는 것과는 달리 하나의 스트링 상에서 변화를 줌으로써 교차연산 시 지나칠 수도 있는 상태를 만들어 내는 오퍼레이터이다. 본 논문에서는 돌연변이연산 방법으로 교환방법(Exchange Method)과 반전방법(Inversion Method)을 사용하고 있다. 교환방법은 선택된 염색체의 스트링 중 두 노드를 선택해서 맞바꾸는 방법이고, 반전방법은 염색체의 스트링 중 두 곳의 지점을 택해 두 지점 사이의 스트링을 반전시키는 방법이다.

### 3.4 그래프 변환에 의한 염색체 디코딩

염색체 디코딩은 스트링으로 구성된 염색체를 해석하는 과정이다. 노드 중심의 문제인 TSP에서는 염색체를 구성하는 스트링의 순서가 방문하는 노드의 순서가 되므로 디코딩이 간단하다. 그러나 아크 중심의 문제인 URPP에서는 아크에서의 라우팅 방향이 결정되어 있지 않기 때문에 TSP와 같은 디코딩 방법을 사용할 수 없다. 따라서 본 논문에서는 그래프 변환에 의한 URPP의 염색체 디코딩 방법을 제안한다.

노드의 집합  $V$ 와 아크의 집합  $A$ 로 구성된 URPP 그래프  $G=(V, A, A'(\subseteq A))$ 에서는 디코딩 과정을 통해  $A'$ 의 요소로 구성된 염색체를 해석하여 전체 라우팅 경로를 구하게 된다. 본 논문에서 제안한 염색체 디코딩 방법은 다음과 같다.

$A' = \{a(0,1), b(2,4), c(3,5)\}$ 로 구성된 그림과 같은 그래프에서 염색체 스트링  $p = "bca"$ 라고 가정하면, 첫 번째 디코딩 단계에서는 아크  $b$ 와 아크  $c$  사이의 최단 경로를 구한다. 두 번째 디코딩 단계에서는 아크  $c$ 의 도착 노드에서 아크  $a$  까지의 최단 경로를 구한 후, 마지막으로 아크  $a$ 의 도착 노드에서 아크  $b$ 의 시작 노드까지의 최단 경로를 구한다. 그림 6과 같은 URPP 문제가 주어졌을 때, 그림 7은 아크와 아크 사이의 최단 경로를 구하기 위한 중간 단계를 나타내고 있고, 그림 8은 아크를 가상의 노드로 변환하고 그림 7에서 구한 최단 경로를 가상의 아크로 구성한 것을 나타내고 있다. 그림 9는 최종 디코딩 결과를 보여주고 있다.

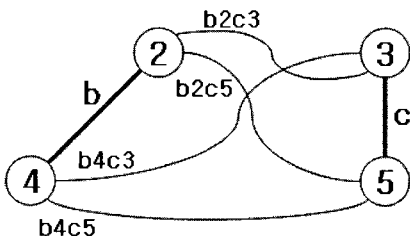


그림 7. 아크  $b$ 와 아크  $c$  사이의 최단 경로를 구하기 위한 중간 단계  
Fig. 7. The intermediate step to search the tour between arc  $b$  and arc  $c$ .

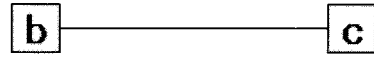


그림 8. 아크  $b$ 와 아크  $c$  사이의 최단 경로  
Fig. 8. The shortest path between arc  $b$  and arc  $c$

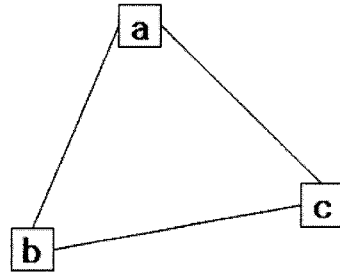


그림 9. 디코딩 결과  
Fig. 9. The result of decoding

### 3.5 적합 함수

본 논문에서는 새로운 세대의 모집단을 생성하기 위한 선택 기법으로 Roulette Wheel 방법을 사용한다. 선택 연산을 수행하기 위해서는 각 염색체에 대한 적합함수(Fitness Function) 값이 필요한데, 본 논문에서는 제 2장에서 설명한 목적 함수를 이용하여 (식 2)와 같은 적합함수 값을 구하고, (식 3)에 의해 염색체 선택이 이루어진다.

URPP를 위한 적합 함수  $F(k)$ :

$$F(k) = \left(\frac{1}{C(k)}\right)^m \dots\dots\dots (식 2)$$

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 3)은 Roulette Wheel 선택 방법에서 사용하기 위한 각 염색체의 선택확률을 계산하기 위한 수식이다.

선택 확률  $P(k)$ :

$$P(k) = \frac{F(k)}{\sum_{i=1}^n F(i)} \dots\dots\dots (식 3)$$

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

표 1. URPP 라우팅 비용  
Table 1. URPP Tour Costs

문제	V	E	E'	Pop Size (200)			Pop Size (500)		
				0.6	0.7	0.8	0.6	0.7	0.8
1	50	63	12	192	188	188	188	188	188
2	50	63	12	197	192	196	192	197	197
3	50	63	12	137	130	130	135	130	130
4	50	100	20	147	151	147	149	161	148
5	50	100	20	166	168	170	166	169	171
6	50	100	20	166	156	161	167	175	161
7	75	94	18	250	271	263	272	261	262
8	75	94	18	231	209	215	215	228	215
9	75	94	18	302	304	290	290	292	290
10	75	150	30	300	312	290	290	292	308

표 2. 기존 방법과 제안 방법의 라우팅 비용 비교  
Table 2. Comparison the proposed algorithm with the existing algorithm

문제	V	E	E'	Pop Size (200)		Pop Size (500)	
				기존방법	제안방법	기존방법	제안방법
1	50	63	12	188	188	188	188
2	50	63	12	200	192	197	192
3	50	63	12	130	130	142	130
4	50	100	20	146	147	146	148
5	50	100	20	179	166	168	166
6	50	100	20	156	156	160	161
7	75	94	18	260	250	253	261
8	75	94	18	209	209	209	215
9	75	94	18	303	290	290	290
10	75	150	30	294	290	284	290

#### IV. 실험 결과

##### 4.1 실험 환경

본 논문에서는 IBM PC Pentium 4에서 실험이 수행되었으며, 적용된 URPP 문제는 [7]에서 사용한 10개의 문제를 적용하였다. 유전알고리즘에 적용된 교차 연산 방법은 PMX를 사용하였고, 돌연변이 연산은 교환방법 및 Inverse 방법을 사용하였으며, 선택연산은 Roulette Wheel 방법을 사용하였다. 교차연산률은 0.6, 0.7, 그리고 0.8의 비율로 적용하였으며, 돌연변이 연산률은 0.03의 비율로 적용하였고, 모든 문제에 대해 총 진화세대는 500으로 하였다.

##### 4.2 실험 결과

표 1은 10개의 URPP 문제와 각 파라미터에 따른 유전 알고리즘의 결과를 나타내고 있다. 모집단의 크기가 200인 경우 교차연산률이 0.7과 0.8인 경우에 좋은 결과를 얻을 수 있었고, 모집단의 크기가 500인 경우에는 교차연산률이 0.6인 경우에 좋은 결과를 얻을 수 있었다. 즉, 모집단의 크기가 작은 경우에는 교차연산을 많이 적용함으로써 지역 최소해(Local Minima)를 탈피할 수 있고, 반면 모집단의 크기가 큰 경우에는 교차연산을 많이 적용함에 따라 동일한 염색체들을 많이 가지게 되어 오히려 지역 최소 해를 탈피하지 못하는 경향을 보이고 있다.

표 2는 기존 방법[7]의 결과와 본 논문에서 제안한 방법의 근사 최적 해를 비교한 결과이다. 모집단의 크기가 200인 경우 기존 방법에서는 5개의 문제에서 근사 최적 해를

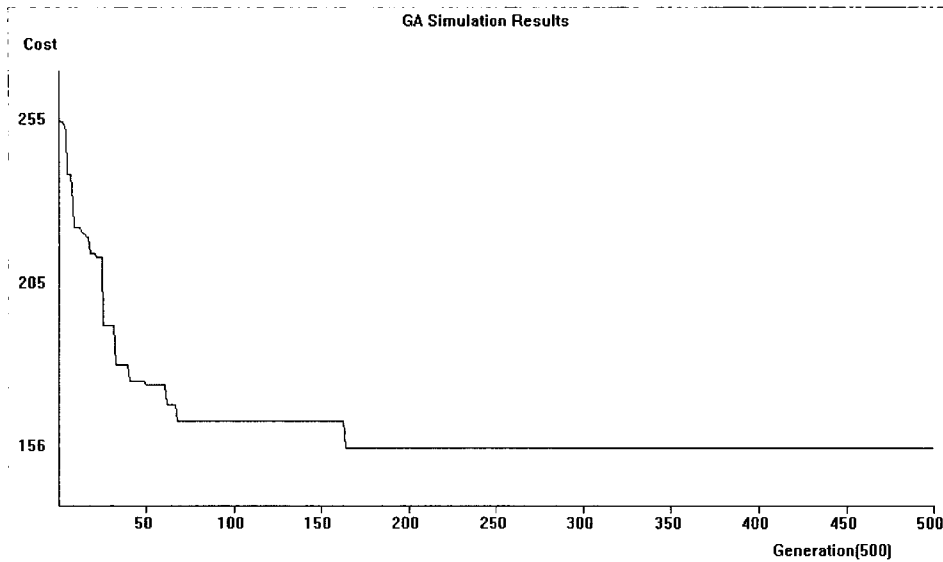


그림 10. 진화과정  
Fig. 10. Evolution Graph

얻었고 본 논문에서 제안한 방법에서는 8개의 문제에서 근사 최적 해를 얻을 수 있었다. 모집단의 크기가 500인 경우에는 기존 방법과 본 논문에서 제안한 방법 모두에서 5개의 문제에 대해 최적 값을 얻을 수 있었다.

그림 10은 문제 6에 대한 진화 과정을 그래프로 나타낸 것이다. 총 500 세대 중 180 세대 이전에 최적

값을 찾고 있음을 알 수 있다. 이런 결과는 대부분의 문제에 대해 적용됨을 실험을 통해 알 수 있었다.

## V. 결론

무향 RPP는 노드의 집합  $V$ 와 아크의 집합  $A$ , 그리고  $A' (\subseteq A)$ 로 구성된 무향 그래프  $G = (V, A, A')$ 에서  $A'$ 의 모든 아크들을 적어도 한 번 이상 경유하는 최소 비용 경로를 구하는 문제로 NP-Complete 문제로 알려져 있다. 따라서 무향 RPP 문제에서는  $A'$ 의 크기가 커짐에 따라 해 공간이 기하급수적으로 늘어나기 때문에 다항함수시간 (Polynomial Time) 내에 최적 해를 구하기 힘들다. 본 논문에서는 NP-Complete 문제에 많이 적용되는 휴리스틱 알고리즘 중 유전 알고리즘을 적용한 무향 RPP의 해법을 제안하였다. 특히, 아크 중심의 RPP 문제는 노드 중심의 TSP와는 달리 아크의 라우팅 방향에 대한 정보가 필요하므로 탐색체 구성이나 디코딩 과정이 복잡하다. 기존 연구[7]

에서는  $A-A'$ 의 아크들에 대한 비트(bit) 스트링으로 탐색체를 구성하는 방법을 제안하고 있으나, 비트 스트링으로 구성된 탐색체를 사용할 경우 라우팅 경로가 중간에 연결되지 않는 경우가 발생하여 디코딩 과정에서 보정 연산을 수행해야 하는 단점이 있다.

본 논문에서 그래프 변환에 의한 디코딩 알고리즘을 제안하였으며, 기존 연구의 문제점을 해결할 수 있었다. 실험은 기존 방법[7]에서 적용한 10개의 RPP 문제에 대해 본 논문에서 제안한 알고리즘을 적용하여 실시하였다. 실험 결과로부터 모집단의 크기가 200이고 교차연산률이 0.7과 0.8인 경우와 모집단의 크기가 500이고 교차연산률이 0.6인 경우에 좋은 결과를 얻을 수 있었다. 또한 기존 연구와 비교해 볼 때 모집단의 크기가 200인 경우 기존 방법에서는 5개의 문제에서 최적 값을 얻었고 본 논문에서 제안한 방법에서는 8개의 문제에서 최적 값을 얻을 수 있었다. 모집단의 크기가 500인 경우에는 기존 방법과 본 논문에서 제안한 방법 모두에서 5개의 문제에 대해 최적 값을 얻을 수 있었다.

향후 연구 방향으로서는 본 논문에서 제안한 알고리즘을 이용하여 유향 그래프에 적용하고, 또한 지리정보시스템 (GIS)과 같은 실제 응용 시스템과의 연동 방안을 연구하는 것이다. 또한, 실제 도로 및 통신 네트워크 상에서 다양한 제약 조건을 고려한 다중목적 최적화 문제에 본 논문에서 제안한 알고리즘을 적용하고 그 성능을 분석하고자 한다.

## 참고문헌

- [1] Garey, M. R. and Johnson, D. S., Computers and Intractability - A Guide to the Theory of NP-Completeness, pp.213, FREEMAN, 1979.
- [2] 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.
- [3] Lenstra, J. K. and Rinnooy Kan, A. H. G., On General Routing Problems., Networks, Vol. 6, pp. 273-280, 1976.
- [4] Chambers, L., Practical Handbook of Genetic Algorithms, CRC Press, 1995.
- [5] Thangiah, S. R., Vehicle Routing with Time Windows using Genetic Algorithms, Practical Handbook of GENETIC vol.2, ALGORITHMS, CRC Press, 1995.
- [6] Pearn, W. L., and Wu, T. C., Algorithms for The Rural Postman Problem, Computers Ops Res, Vol. 22, No. 8, pp. 819-828, 1995.
- [7] 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.
- [8] Goldberg, D. E., Genetic Algorithm in Search, Optimization & Machine Learning, Addison-Wesley, 1989.
- [9] Ladd, S. R., Genetic Algorithms in C++, M&T Books, 1995.
- [10] Michalewicz Z., Genetic Algorithms + Data Structures = Evolution Programs, Second, Extended Edition, Springer-Verlag, 1994.

## 저자소개



### 강명주

1993년 경희대학교 전자계산공학과  
학사

1995년 경희대학교 전자계산공학과  
석사

1998년 경희대학교 전자계산공학과  
박사

2000년 ~ 현재 청강문화산업대학  
컴퓨터게임과 교수

〈관심분야〉 게임알고리즘, 메타휴리스틱 알고리즘, 네트워크 디자인