

Rural Postman Problem 해법을 위한 Iterative Local Search 알고리즘

강 명 주*

An Iterative Local Search Algorithm for Rural Postman Problems

Myung-Ju Kang*

요 약

본 논문에서는 Rural Postman Problem(RPP) 해법을 위한 Iterative Local Search(ILS) 알고리즘을 제안한다. ILS 알고리즘은 초기해를 여러 번 생성하여 탐색 시작점을 달리하는 방법으로, 초기해의 설정 방법에 따라 알고리즘의 성능이 크게 좌우되는 Local Search(LS) 알고리즘의 단점을 보완할 수 있다. 본 논문에서는 LS 알고리즘과 ILS 알고리즘을 18개의 RPP 문제에 적용하고 그 성능을 분석한다. 실험 결과에서는 ILS 알고리즘이 각각 다른 탐색 시작점에서 해 공간을 탐색함으로써 LS에 비해 좋은 해를 찾을 수 있음을 알 수 있었다.

Abstract

This paper proposes an iterative Local Search (ILS) algorithm for Rural Postman Problems (RPPs). LS searches neighbors from an initial solution in solution space and obtains a near-optimal solution which can be a local-minima. As an extension of LS, the ILS algorithm is a method that uses various initial solutions for LS. Hence, ILS can overcome the defect of LS. This paper proposes LS and ILS methods for 18 RPPs and analyzes the results of LS and ILS. In the simulation results, the ILS method obtained the better results than the LS method.

I. 서론

그래프 및 네트워크 문제는 실생활의 다양한 문제에 적용되는 가장 기본적인 문제이다. 일반적으로 네트워크 문제에 적용되는 문제들로는 Traveling Salesman Problem (TSP), Chinese Postman Problem (CPP), Vehicle Routing Problem (RPP), 그리고 Rural Postman Problem (RPP) 등이 있다.

특히, RPP 문제는 우편배달 경로 설정 문제, 쓰레기 수거 경로 문제, 버스노선 설정 및 라우팅 설정 문제, 통신의 라우팅 문제, 지리정보시스템(GIS)에서의 최적 경로 문제 등에서 많이 적용되는 것으로 NP-Complete 문제로 알려져 있다.

RPP 해법에 관한 기존의 방법으로는 1994년 Coberan에 의해 제안된 Polyhedral 접근 방법[1]과 1997년 Wainwright에 의해 제안된 유전자 알고리즘을 이용한 해법[2]이 있다. Polyhedral 접근 방법은 E' 내의 에지들의 수 뿐만 아니라 E' 내의 구성 요소들이 얼마나 서로 연결되어 있는지에 따라 효율이 크게 좌우되는 단점이 있다. Wainwright에 의해 제안된 유전자 알고리즘 해법은 오일러 그래프 변환 방법을 이용한 것으로, $E-E'$ 의 에지들에 대한 중간 경유 에지들의 포함 여부를 판단하기 위해 이진 코드로 염색체를 구성하였다. 그러나, 이 방법은 에지의 연결도가 커짐에 따라 염색체의 크기가 커지고, 또한, RPP의 가능해(Feasible Solution)를 위한 보정연산이 필요하다는 단점이 있다. 따라서, 본 논문에서는 RPP의 해법으로 Local Search(LS) 알고리즘의 단점을 보완한 Iterative Local Search (ILS) 알고리즘을 사용한다.

LS는 휴리스틱 알고리즘으로는 가장 간단한 방법 중의 하나로 하나의 초기해로부터 이웃해를 탐색하는 방법이다[4, 6]. 그러나 LS 알고리즘은 초기해의 설정 방법에 따라 알고리즘의 성능이 크게 좌우된다는 단점이 있다. 이러한 LS의 단점을 보완하기 위한 방법으로서, 초기해를 여러번 생성하여 탐색 시작점을 달리하는 방법인 ILS 알고리즘이 있다. ILS는 각각 다른 탐색 시작점에서

해 공간을 탐색함으로써 LS에 비해 좋은 해를 찾을 수 있다.

본 논문에서는 RPP를 위한 ILS의 이용 방안을 제안한다. 실험에서는 LS와 ILS 알고리즘에 각각 18개의 RPP 문제를 적용하고 그 성능을 분석한다.

II. Rural Postman Problem

2.1 문제정의

RPP는 주어진 특정 에지의 집합을 반드시 한번 이상 경유하는 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 문제로 NP-Complete 문제로 알려져 있다[3, 5].

2.2 Rural Postman Problem을 위한 수식 표현

노드 V 와 에지 E , 그리고 반드시 한 번 이상 경유해야 하는 에지 $E'(\subseteq E)$ 으로 구성된 무방향 그래프 $G=(V, E, E')$ 가 주어질 때, RPP는 E' 을 한번 이상 경유하면서 전체 라우팅 비용을 최소화하는 문제이다. 따라서, 전체 라우팅 비용은 E' 에 속하는 에지들의 비용과 E' 내의 에지들 간의 중간 경로들의 경유 비용을 모두 합한 것이 전체 라우팅 비용이 된다.

다음은 RPP를 위한 수식 파라미터(Parameter) 및 수식을 정의한 것이다.

파라미터 :

$e_i = (e_i^1, e_i^2) (\in E')$: E' 의 i 번째 에지.

$e_i^1 (\in V)$: i 번째 에지($\in E'$)의 시작노드.

$e_i^2 (\in V)$: i 번째 에지($\in E'$)의 끝노드.

$d_{e_i^2, e_{i+1}^1}$: e_i^2 에서 e_{i+1}^1 까지의 최소 비용. 즉, 디코딩에서 생성된 라우팅에서 i 번째 에지의 끝노드와 $(i+1)$ 번째 에지의 시작노드 사이에

경유되는 중간 경로의 최소 비용.

c_{e_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 목적함수 있다.

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.

2.3 적용되는 Rural Postman Problem 문제

[표 1]은 18개의 문제들에 대해 노드의 집합 V 와 에지의 집합 E 그리고 반드시 경유해야 하는 에지의 집합 E' ($\subseteq E$)로 구성된 무방향 그래프 $G=(V, E, E')$ 에서 노드의 크기($|V|$, 에지의 크기($|E|$) 그리고 반드시 경유해야 하는 에지의 크기($|E'|$)을 나타내고 있다. 적용된 18개의 문제는 [3]에서 적용된 문제들이다.

표 1. Rural Postman Problem 문제
Table 1. Rural Postman Problems

문제	$ V $	$ E $	$ E' $
1	50	63	12
2	50	63	12
3	50	63	12
4	50	100	20
5	50	100	20
6	50	100	20
7	75	94	18
8	75	94	18
9	75	94	18
10	75	150	30
11	75	150	30
12	75	150	30
13	100	125	25
14	100	125	25
15	100	125	25
16	100	200	40
17	100	200	40
18	100	200	40

III. Rural Postman Problem을 위한 Iterative Local Search 알고리즘

3.1 Iterative Local Search 알고리즘

LS 알고리즘은 하나의 초기해로부터 이웃하는 해를 탐색하기 때문에 초기해를 어떻게 설정하느냐에 따라 근사 최적해 결과가 달라질 수 있다는 단점이 있다. 이를 극복하기 위한 방법이 ILS 알고리즘으로써, ILS 알고리즘은 초기해를 바꾸어가면서 LS를 여러번 반복 수행한다. 즉, 탐색 시작점을 바꾸면서 LS 알고리즘을 수행하기 때문에 다양한 해 공간을 탐색할 수 있다는 장점이 있다.

ILS 알고리즘에서는 전체 해 공간(Solution Space)이 알고리즘이 반복되면서 초기해로 설정되면 전역 최적해를 찾을 수 있다. 그러나, NP-Complete 문제에서는 전체 해 공간을 탐색하는 것이 어렵기 때문에 일반적으로 일정한 횟수만큼 알고리즘을 반복하여 실행하고 종료하게 된다.

[그림 1]은 ILS 알고리즘을 설명하고 있다. [그림 1]에서, 단계 (3)은 초기해를 설정하는 부분으로서 여러번 실행되어 탐색 시작점이 여러 곳이 되기 때문에 LS 보다 다양한 탐색이 가능하도록 해주는 부분이다. 단계 (4) ~ 단계 (9)는 LS 알고리즘이 수행되는 부분이다.

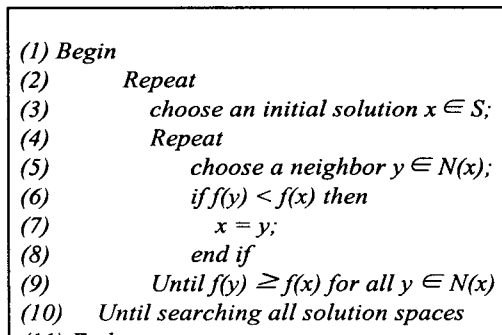


그림 2. Iterative Local Search 알고리즘
Fig 1. Iterative Local Search Algorithm

3.2 초기해 설정 방법

LS와 ILS 알고리즘에서는 초기해를 어떻게 설정하느냐에 따라 근사 최적해의 결과가 달라질 수 있다. 본 연구에서는 LS와 ILS 알고리즘을 위한 해 구성은 반드시 경유해야 하는 에지들로 구성된다. 즉, 에지의 집합 E와 노드의 집합 V, 그리고 반드시 경유해야 하는 에지의 집합 E' ($E' \subseteq E$)으로 구성된 무방향 그래프 $G = (V, E, E')$ 에서 해는 E'의 모든 원소들로 이루어진다. 이 때, 해를 구성하는 스트링에는 서로 중복이 없고, 또한, 한 번씩만 나타나도록 하여 모든 E'의 에지들을 최소로 경유할 수 있도록 한다.

구성된 해의 특징은 해를 구성하는 원소들의 순서가 라우팅의 순서가 되고, 해의 각 원소(에지 정보)들 사이의 중간 라우팅 경로 및 비용은 각 원소를 구성하는 노드들 간의 경로들 중 최단 경로와 최소 비용을 할당하게 된다.

3.3 이웃해 생성 방법

본 논문에서는 이웃해 생성 방법으로서 삽입(Insertion) 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. 따라서, 본 논문에서는 삽입 방법을 이웃해 생성 방법으로 사용한다.

[그림 2]는 삽입 방법을 이용한 이웃해 생성 방법을 나타내고 있다. 즉, 현재 상태의 벡터의 요소 중 여섯번째 요소인 e7을 선택하여 벡터의 세번째 위치로 삽입함으로써 새로운 상태의 벡터를 생성하는 방법을 나타내고 있다. 이 방법을 사용함으로써, 현재 상태의 해로부터 최소한의 변화를 주어 이웃해를 구하게 되며, 또한, 해의 구조가 깨지는 것을 막을 수 있다.

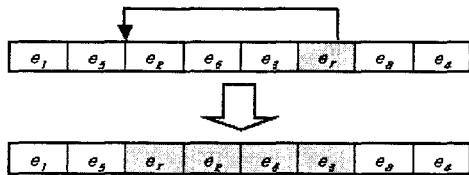


그림 2. 삽입(Insertion) 방법을 이용한 새로운 상태의 벡터 생성 방법
Fig 2. Insertion method

IV. 실험 결과

4.1 실험환경

본 연구에서는 18개의 RPP 문제에 대해 다음과 같은 실험 환경하에서 LS 알고리즘과 ILS 알고리즘의 근사 최적해를 구하고, 이를 바탕으로 두 알고리즘의 성능을 평가한다.

- 시스템 : IBM PC Pentium III
- 운영체제 : Windows 2000 Server
- 적용된 총 문제 수 : 18개.
- 이웃해 생성 방법 : 삽입(Insertion) 방법.
- 총 비용 계산 횟수(알고리즘 종료조건) : 100,000 회
- LS : Local Search 반복 (100,000)
- ILS : 초기해 생성 (100) × Local Search 반복(1,000)
- 비용 계산식 :

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

4.2 실험결과

[표 2]는 18개의 RPP 문제에 대해 동일한 실험 환경에서 구한 근사 최적해와 CPU 시간, 그리고 LS 알고리즘에 대한 ILS 알고리즘의 근사최적해의 향상 정도 및 탐색 시간을 비교한 것이다. LS 알고리즘에 대한 ILS 알고리즘의 근사 최적해 및 탐색 시간을 (식 2)와 (식 3)에 의해 %로 나타내었다. 대부분의 문제에 대해 ILS 알고리즘의 근사 최적해가 좋지만 근사 최적해 수렴 속도는 LS 알고리즘이 빠르다. 이는 LS 알고리즘은 하나의 초기해를 이용하여 수렴해 가기 때문에 해 공간을 탐색하는 범위가 작아서 근사 최적해는 ILS에 비해 좋지 않지만 빠른 수렴을 보이고 있다. 반면, ILS는 여러개의 초기해를 사용하기 때문에 해 공간을 탐색하는 범위가 커서 좋은 근사 최적해를 얻고 있지만, 많은 해 공간을 탐색하는 만큼 수렴 속도가 느리다는 것을 알 수 있다.

표 2. LS와 ILS의 근사 최적해 비교
Table 2. Comparison of near-optimal solution between LS and ILS

문제	리우팅 비용					
	Local Search		Iterative Local Search		LS에 대한 ILS의 근사최적해 (%)	LS에 대한 ILS의 탐색시간(%)
	근사최적해	CPU	근사최적해	CPU		
1	188	140	188	10,376	0	-7,311.42
2	197	234	192	21,540	2.53	-9,105.12
3	130	153	130	11,350	0	-7,318.30
4	165	815	147	15,699	10.9	-1,826.25
5	172	2,163	163	55,716	5.23	-2,475.86
6	167	851	157	39,960	5.98	-4,595.65
7	265	793	250	9,920	5.66	-1,150.94
8	232	1,102	209	22,441	9.91	-1,936.38
9	312	369	290	33,778	7.05	-9,053.92
10	310	4,036	305	70,931	1.61	-1,657.45
11	270	5,560	273	51,928	-1.11	-833.95
12	278	4,109	276	30,851	0.71	-650.81
13	328	3,446	315	24,760	3.96	-618.51
14	364	740	339	31,657	6.86	-4,177.97
15	307	3,088	301	36,775	1.95	-1,090.90
16	435	4,984	470	7,936	-8.04	-59.22
17	381	8,102	384	49,924	-0.78	-516.19
18	411	8,591	428	11,780	-4.13	-37.12

$$\frac{LS의\ 근사최적해 - ILS\ 근사최적해}{LS의\ 근사최적해} \times 100 \quad (식\ 2),$$

$$\frac{LS의\ 탐색시간 - ILS의\ 탐색시간}{LS의\ 탐색시간} \times 100 \quad (식\ 3).$$

초기해에 대한 근사 최적해의 수렴 정도는 (식 4)를 이용하여 구하였다.

$$\frac{초기해 - 근사최적해}{근사최적해} \times 100 \quad (식\ 4).$$

표 3. 초기해에 대한 근사 최적해의 수렴 정도
Table 3. Degree of convergence

문제	근사 최적해 수렴 정도(%)	
	Local Search	Iterative Local Search
1	42	42
2	50	54
3	65	65
4	78	100
5	58	67
6	74	85
7	92	104
8	60	77
9	25	34
10	73	76
11	90	88
12	102	103
13	97	106
14	70	83
15	126	131
16	79	66
17	67	65
18	72	65

V. 결론

본 논문에서는 LS 알고리즘과 ILS 알고리즘을 이용한 RPP 문제의 해법을 제안하였다. LS 알고리즘은 초기해로부터 이웃하는 좋은 해들을 탐색하면서 근사 최적해로 수렴해 가는 알고리즘이다. 그리고, ILS 알고리즘은 LS의 해 공간의 탐색 범위를 넓혀서 좀 더 좋은 근사 최적해를 구하기 위한 방법이다. 즉, 해 공간에서의 탐색 지점을 여러 곳에서 시작함으로써 LS의 단점을 보완할 수 있다.

실험 결과로부터 대부분의 문제에 대해 LS 알고리즘이 ILS 알고리즘에 비해 근사 최적해의 결과는 나쁘지만 수렴 속도가 빠르다는 것을 알 수 있고, 반면, ILS 알고

[표 3]은 동일한 초기해에 대해 LS 알고리즘과 ILS 알고리즘의 근사 최적해의 수렴 정도를 나타내고 있다.

리즘은 LS 알고리즘에 비해 근사 최적해가 좋지만 수렴 속도가 느리다는 것을 알 수 있다. 이는 LS 알고리즘이 하나의 초기해로부터 근사 최적해로 수렴해가는 반면, ILS 알고리즘은 근사 최적해로 수렴해 가는 과정에서 여러개의 초기해로부터 수렴하기 때문이다.

향후 연구 과제로는 RPP 문제를 GIS의 최적경로 탐색 문제와 네트워크에서의 라우팅 문제에 적용하고, 또한, 시간 제약이 있는 RPP 문제에서도 본 논문에서 제안한 알고리즘을 적용하는 것이다.

참고문헌

- [1] Corberàn, A. and Sanchis, J. M., "A Polyhedral Approach to the Rural Postman Problem", *European Journal of Operational Research* 79, pp.95-114, 1994.
- [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] Hjorring, C. A., *The Vehicle Routing Problem and Local Search Metaheuristics*, Ph.D. thesis, 1995
- [5] Lenstra, J. K. and Rinnooy Kan, A. H. G., *On General Routing Problems.*, *Networks*, Vol. 6, pp. 273-280, 1976.
- [6] Migkikh, V. V., Topchy, A. P., Kureichik, V. M., Tetelbaum, A. Y., *Combined Genetic and Local Search Algorithm for The Quadratic Assignment Problem.*, —.

저자 소개



강 명 주

1993. 2 경희대학교 전자계산
공학과 졸업(공학사)

1995. 2 경희대학교 전자계산
공학과 (공학석사)

1998. 8 경희대학교 전자계산
공학과 (공학박사)

현재 청강문화산업대학 컴퓨터
소프트웨어과 교수

관심분야 : 운영체제, 네트워크
디자인, 메타휴리스틱 알고
리즘, 유전자 알고리즘 등