

## 외판원 문제의 다항시간 알고리즘

이 상 운\*

# A Polynomial Time Algorithm of a Traveling Salesman Problem

Sang-Un, Lee \*

### 요 약

본 논문은 NP-완전으로 다항시간 알고리즘이 존재하지 않는 대규모 외판원 문제의 최적 해를  $O(n^2)$ 의 다항시간으로 구하는 알고리즘을 제안하였다. 대규모 외판원 문제에서 가장 큰 문제는 처리될 데이터가  $n \times n$ 으로  $n$ 이 커질수록 기하급수적으로 증가한다. 본 논문에서는 먼저, 데이터의 양을 약  $n/2$ 의 크기로 축소시킨다. 다음으로 임의의 정점에서 시작하여 양방향으로 경로를 탐색하는 방법을 적용하였다. 제안된 알고리즘을 26개의 유럽 도시들을 방문하는 TSP-1과 46개 미국 도시들을 방문하는 TSP-2에 적용한 결과 모두 최적 해를  $O(n^2)$  수행 복잡도로 빠르게 구하는데 성공하였다. 따라서 제안된 알고리즘은 TSP의 일반화된 알고리즘으로 적용할 수 있을 것이다.

▶ Keywords : 외판원 문제, 전수 탐색 방법, 간선 교환 방법, 데이터 축소

### Abstract

This paper proposes a  $O(n^2)$  polynomial time algorithm to obtain optimal solution for Traveling Salesman problem that is a NP-complete because polynomial time algorithm has been not known yet. The biggest problem in a large-scale Traveling Salesman problem is the fact that the amount of data to be processed is  $n \times n$ , and thus as  $n$  increases, the data increases by multifold. Therefore, this paper proposes a methodology by which the data amount is first reduced to approximately  $n/2$ . Then, it seeks a bi-directional route at a random point. The proposed algorithm has proved to be successful in obtaining the optimal solutions with  $O(n^2)$  time complexity when applied to TSP-1 with 26 European cities and TSP-2 with 46 cities of the USA. It could therefore be applied as a generalized algorithm for TSP.

▶ Keywords : CTraveling Salesman Problem, Exhaustive Search Method, Edge Exchange Method, Data Reduction

•제1저자 : 이상운

•투고일 : 2013. 09. 03. 심사일 : 2013. 09. 21. 게재확정일 : 2013. 10. 05.

\* 강릉원주대학교 멀티미디어공학과 (Dept. of Multimedia Eng., Gangneung-Wonju National University)

## I. 서론

본 논문에서는 외판원 문제 (Traveling Salesman Problem, TSP)의 최적 해를 찾는 알고리즘을 제안한다. TSP는  $n$ 개의 도시가 있는 경우, 한 도시 (노드)에서 출발하여  $n-1$ 개의 도시를 단지 한번 씩 방문하고 출발 도시로 다시 되돌아오는 해밀턴 사이클 (Hamiltonian Cycle, HC) 문제로 경로 길이의 합이 최소가 되는 조건을 만족시켜야 한다[1].

TSP는 계산수학 (Computational Mathematics) 분야에서 가장 집중적으로 연구 중인 문제로 전수 탐색 방법 이외에는 효율적인 해법이 아직 알려지지 않은 NP-완전 (Non-deterministic polynomial complete) 난제로 남아 있다[2-3].

TSP의 해법 (Solution Method)은 정확한 방법 (Exact Method)과 근사 방법 (Approximate Method)으로 분류된다[1,4].

정확한 방법은 최적 해 (Optimal Solution)를 얻음을 보장하기 위해  $n$ 개의 도시를 연결하는 도로 (간선)에 대해 전수 탐색 (Exhaustive Search)을 수행해야 한다. 전수 탐색은 대칭행렬 (Symmetric Matrix)에 대해  $(n-1)!/2$ 회 수행된다[5]. 결국,  $n$ 이 크면 컴퓨터를 활용하여도 원하는 시간에 해답을 얻지 못한다.

현실세계의 많은 문제들은 정확한 방법으로 해결하기에는 너무 큰 경향이 있으며, 일반적으로 빠른 결론을 요구한다. 따라서 좋은 발견적 방법을 적용하면 양질의 해법을 찾을 수 있으며, 매우 빠른 장점을 갖고 있다. 반면에, 근사 방법은 발견적 방법 (Heuristic Algorithms or Heuristics)으로 최적 해를 찾음을 보장하지는 못하지만 적당히 좋은 (Reasonable Good) 해법을 빠르게 찾기 위해 일반적으로 사용된다[4].

발견적 방법은 초기 해를 구하는 구성 (Constructive) 단계, 초기 해를 개선하는 지역 개선 (Local Improvement) 단계, 지역적으로 최적화된 경로를 개선하는 확장 (Extensions) 단계로 수행된다. 구성단계에 적용되는 알고리즘에는 최단 인접 이웃 탐색 (Nearest Neighbor Search, NN), 양측 최단인접 이웃 탐색 (Double-sided NN), 다중파편 (Multiple Fragment, MF), 삽입 탐색 (Insertion Search, IS) 등이 있다. 지역개선단계에 적용되는 알고리즘에는 간선 교환 ( $k$ -edge Exchange,  $k$ -opt)이 있으며, 확장단계에 적용되는 알고리즘에는 Tabu 탐색 알고리즘, 모의실험 담금질 (Simulated Annealing)과 유전자 알고리즘 (Genetic Algorithm), 신경망 (Neural Networks), 진화 알고리즘 (Evolutionary Algorithms), 변종 NN (Variable Neighborhood Search, VNN), 개미집단 최적화 (Ant Colony Optimization) 등 다양한 방법들이 있다[4].

본 논문은 각 정점의 최소 인접 정점 2개씩을 이웃으로 하

여, 에르되시 수 (Erdős number)가 2인 탐색 범위에서 최단경로를 양방향으로 탐색하는 발견적 방법으로 TSP의 최적 해를  $O(n^2)$ 의 다항시간으로 구할 수 있는 알고리즘을 제안한다. 제안되는 알고리즘은 대규모 TSP의 처리될 데이터 양을 약  $n/2$ 로 축소시키고 각 정점의  $1^{st} Min$ 과  $2^{nd} Min$ 만을 대상으로 양방향 최단 경로를 탐색하여 선형시간  $O(n^2)$ 으로 최적 해를 도출하는 방법이다.

2장에서는 대규모 TSP에서 최적 해를 도출할 경우 발생하는 문제를 고찰한다. 3장에서는 알고리즘을 제안하고, 4장에서는  $n=26$ 과  $n=42$ 인 실제 데이터에 적용하여 제안된 알고리즘의 적합성을 검증한다.

## II. 관련 연구와 문제점

TSP에서 다루는 데이터는 대칭행렬 (Symmetric Matrix)을 대상으로 한다. 즉, 두 도시  $i, j$  간의 거리  $w\{i, j\}$ 는  $w\{j, i\}$ 와 같으며 양방향 또는 무방향이다.

TSP의 초기 해법을 얻는 일반적인 방법에는 NN, DNN, MF, IS 등이 있다. 이들 방법은 모두  $n \times n$ 의 거리계산 데이터를 활용한다. 만약,  $n=100$ 인 대칭행렬인 경우, 계산될 거리 데이터 개수는 5,000개이다. 따라서 대규모의 TSP에 대해서는 초기 해를 구하기 위해  $n \times n$ 의 모든 노드들 간의 거리를 계산하는데 많은 시간이 소요되며, 모든 노드들 간의 거리가 TSP의 최적 해 간선들로 선택되지 않아 최적 해를 구하는데 불필요한 시간만 소요된다. 결국, 대규모의 TSP를 빠르게 해결하기 위해서는 필요한 데이터의 양을 현격히 축소시킬 필요가 있다.

만약, 데이터를 임의로 축소시킬 경우 TSP의 초기 해법을 얻는 NN, DNN 등은 하나의 경로를 형성하는데 실패할 수 있는 문제가 발생할 수 있다. 따라서 이러한 문제를 해결할 수 있도록 데이터 축소 양을 결정하는 방법을 연구해야 한다.

초기 해에 대해 최적 해를 얻는 방법은 지역개선과 확장 등 많은 방법이 제시되어 있다. 그러나 적용하기에 가장 쉬운 방법이 지역개선 방법이다. 지역 개선 알고리즘인  $k$ -opt는 Stougie[6]와 Chuin[7] 등이 제안하고 있으며,  $k=1, 2, 3, 4, \dots$ 로 무한히 확장 가능하다. 그러나 3-opt와 4-opt를 적용하려면 2-opt에 비해 알고리즘 수행횟수가 기하급수적으로 증가한다. 결국, 실제적으로는 2-opt와 3-opt들이 주종을 이루고 있다. Lee[8]는 확장된  $k$ -opt를 적용하여 TSP를 풀고자 하였다.

3장에서는 축소된 데이터들을 대상으로 각 정점의  $1^{st} Min$ 과  $2^{nd} Min$ 만을 대상으로 양방향 최적 해를 도출하는 방법을 제안하여, 대규모 TSP를 해결할 수 있는 실용적인 알고리즘으로 활용하고자 한다.

### III. 대규모 TSP의 빠른 최적 해 도출 알고리즘

본 장에서는 먼저, 데이터를 축소시키고 TSP를 양방향으로 빠르게 찾는 알고리즘을 제안한다. 제안된 알고리즘은 먼저, 식 (1)을 적용하여 데이터를 약 절반으로 감소시켰다.

$$x \leq \left( \frac{Min + Max}{2} \right) \tag{1}$$

두 번째로 각 정점에서 1<sup>st</sup> Min과 2<sup>nd</sup> Min의 2개 간선을 선택한다. 왜냐하면 TSP는 해밀턴 사이클로 한 정점의 간선은 2개가 되기 때문이다. 각 정점에서 선택된 1<sup>st</sup> Min과 2<sup>nd</sup> Min을 대상으로 양 방향으로 최단경로를 결정하였다.

- 
1. 각 정점  $v_i$ 의 거리 값에서 하위 50%만 선택.  

$$x \leq \left( \frac{Min + Max}{2} \right)$$
  2. 각 정점  $v$ 에서 최소값 2개 (1<sup>st</sup>, 2<sup>nd</sup> Min) 선택.  
 $\{x, y\} = \{y, x\}$  도 선택된 것으로 취급.
  3. 임의의 정점  $v_i$ 에서 시작하여 양방향으로 탐색.  
 Do While  $|V| \neq \emptyset$ 
    - (1) 현 정점  $v_i$ 의 1<sup>st</sup>, 2<sup>nd</sup> Min인 인접정점  $N_G(v_i)$  선택.  
 if  $N_G(v_i) = \emptyset$  then Min 값 4개를  $N_G(v_i)$ 로 선택.
    - (2)  $N_G(v_i)$ 인  $v_j$  정점의 1<sup>st</sup>, 2<sup>nd</sup> Min 인접정점  $N_G(v_j)$ 인  $v_k$  정점 선택.
    - (3)  $v_j$  모두를 거치고  $v_k$ 로 향하는 최단 경로 선택.  
 if  $N_G(v_j)$ 로 최단 경로 결정 불가시  $v_i$ 의 3<sup>rd</sup> Min과  $N_G(v_j)$ 의 1<sup>st</sup>, 2<sup>nd</sup> Min 정점 선택.
    - (4)  $v_{i1}$ 와  $v_j$  경로 선택,  $v_{i2}$ 와 3<sup>rd</sup> Min 경로 선택.
 End While.
- 

그림 1. TSP의 다항시간 알고리즘  
 Fig. 1. Polynomial time algorithm for TSP

최단 경로 결정 방법은 현재 위치한 정점  $v_i$ 의 1<sup>st</sup> Min과 2<sup>nd</sup> Min 인접 정점  $v_j$ 와  $v_j$ 의 1<sup>st</sup> Min과 2<sup>nd</sup> Min 인접 정점  $v_k$  또는  $v_i$ 의 3<sup>rd</sup> Min 정점을 대상으로 이웃의 이웃까지 탐색 영역을 대상으로 하는 탐욕 알고리즘 (greedy algorithm)이다. 제안된 알고리즘은 그림 1에 제시하였다.

### IV. 알고리즘 적용 및 분석

본 장에서는 그림 2의 TSP-1과 TSP-2 데이터를 대상으로 제안된 알고리즘의 적합성을 평가해 본다.

#### 4.1 실험 데이터

TSP-1은 Pleines[5]가 제안한 데이터로, 유럽의 26개 도시를 여행하는 경우로 편의상 각 도시를 번호로 표기하였다. 참고로, 1 (포르투갈 리스본), 2(핀란드 헬싱키), 3(스페인 마드리드), 4(터키 이스탄불), 5(그리스 아테네), 6(루마니아 부다페스트), 7(불가리아 소피아), 8(스웨덴 스톡홀름), 9(노르웨이 오슬로), 10(세르비아 베오그라드), 11(헝가리 부다페스트), 12(덴마크 코펜하겐), 13(이탈리아 로마), 14(폴란드 바르샤바), 15(오스트리아 빈), 16(독일 베를린), 17(네덜란드 암스테르담), 18(영국 런던), 19(벨기에 브뤼셀), 20(체코 프라하), 21(이탈리아 밀라노), 22(스위스 취리히), 23(스페인 바르셀로나), 24(스위스 제네바), 25(독일 프랑크푸르트), 26(프랑스 파리)이다. 전수탐색은  $7.7556 \times 10^{24}$ 회 수행한다.

TSP-2는 Dantzig, Fulkerson과 Johnson[9]이 제시한 데이터로 미국의 49개 도시를 방문하는 문제로, 7개 도시는 40과 41번 도시사이에 존재하여 실제 데이터는 42개이다. 본 데이터의 단위는 마일이며,  $1/17(d_{ij} - 11)$ 로 정규화시켜 정수로 표현되었다. TSP-1과 TSP-2의 각 정점에서  $x \leq (Min + Max)/2$ 와 1<sup>st</sup> Min, 2<sup>nd</sup> Min을 선택한 결과는 그림 3에 제시되어 있다.

#### 4.2 TSP-1 데이터

TSP-1의 최적 해는 1-3-13-21-22-25-20-15-11-10-7-5-4-6-14-2-8-9-12-16-17-19-18-26-24-23-1=16,189 Km이다.

정점 ①에서 시작하여 양방향으로 최단경로를 탐색한 결과는 그림 4와 같다. 실험 결과 26개 정점을 단지 7회의 탐색으로 TSP를 얻을 수 있었다. 첫 번째로, 정점 1을 탐색한 결과 (a)와 같이 13-3-1-23-24의 경로를 얻었다.







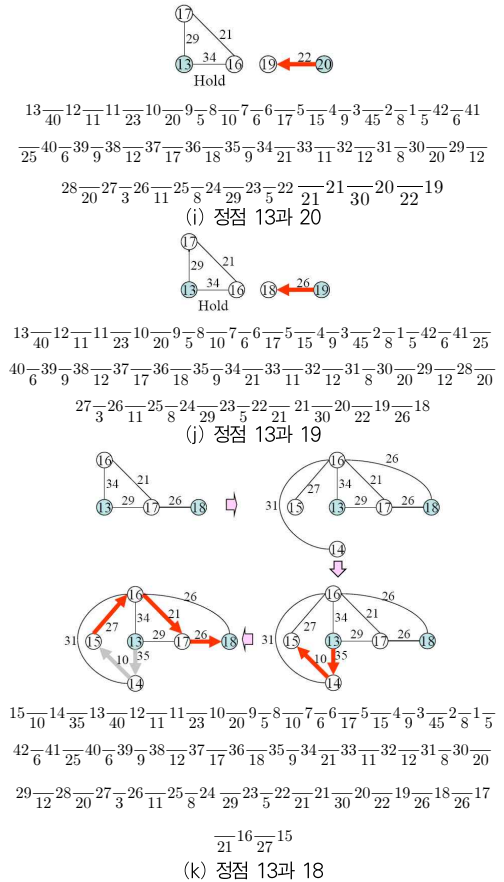
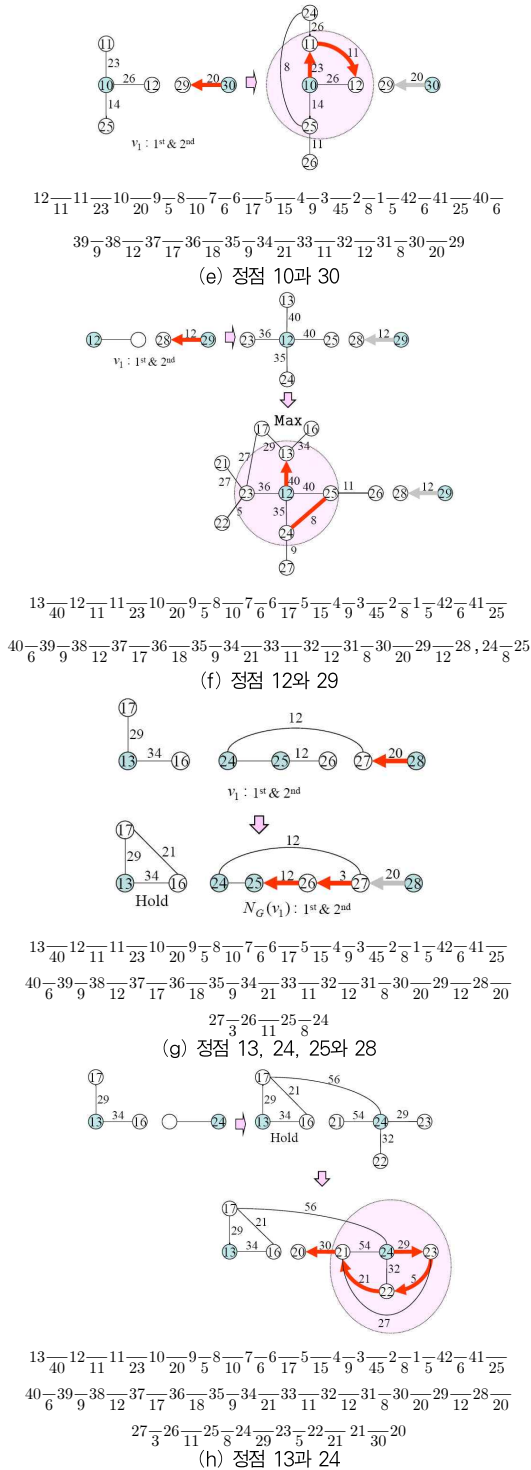


그림 5. TSP-2의 양방향 최단경로 탐색  
Fig. 5. Two-sided shortest path search for TSP-2

TSP-2는 정점 1 탐색 결과 (a)와 같이 2-1-42-41-40 경로를 얻었다. (a)에서 최좌측 2와 최우측 40 정점 탐색 결과 (b)와 같이 6-5-4-3-2-1-42-41-40-39-38-37 경로를 얻었다. (b)의 결과에서 6과 37 정점 탐색 결과 (c)와 같이 9-8-7-6-5-4-3-2-1-42-41-40-39-38-37-36-35-34 경로를 얻었다. (c) 결과에서 9와 34 정점 탐색 결과 (d)와 같이 10-9-8-7-6-5-4-3-2-1-42-41-40-39-38-37-36-3 5-34-33-32-31-30 경로를 얻었다. (d)에서 10과 30 정점 탐색 결과 (e)와 같이 12-11-0-9-8-7-6-5-4-3-2-1-42-41-40-39-38-37-36-35-34-33-32-31-30-2 9 경로를 얻었다. 이와 같이 (f)의 12와 29, (g)의 13, 24, 25와 28, (g)의 13과 24, (h)의 13과 20, (i)의 13과 18 정점 탐색을 수행한 결과 15-14-13-12-11-0-9-8-7-6-5-4-3-2-1-42-41-40-39-38-37-36-35-34-33-32-31-30-29-28-27-26-25-24-23-22-21-20-19-18-17-16-15로 15 정점에서 만나는 해밀턴 사이클을 얻었다.

## V. 결론

본 논문은 NP-완전으로 다항시간 알고리즘이 존재하지 않는 대규모 외관원 문제에 대한 최적 해를  $O(n^2)$ 의 다항시간으로 도출할 수 있는 알고리즘을 제안하였다.

제안된 방법은 먼저, 알고리즘 수행에 적용되는 데이터 양을 약  $n/2$ 로 축소시키고 각 정점에서  $1^{st}$   $Mfn$ 과  $2^{nd}$   $Mfn$  인 집 정점들만을 선택하였다. 이 과정은  $O(n)$ 의 선형 시간이 소요된다.  $n=26$ 인 TSP-1과  $n=42$ 인 TSP-2 데이터에 대해  $1^{st}$   $Mfn$ 과  $2^{nd}$   $Mfn$  정점들을 대상으로 양 방향으로 탐색하는 방법을 적용하였다. TSP-1은 26개 데이터에 대해 7회만에 TSP를 구할 수 있었으며, TSP-2는 42개 데이터에 대해 11회만에 TSP를 정확히 구하였다.

결국, 본 논문은 대규모 TSP에 대해 발견적 알고리즘으로 최적 해를 빠르게 구할 수 있음을 검증하였으며, TSP의 최적 해를 구하는 일반적인 알고리즘으로 적용할 수 있을 것이다.

## 참고문헌

- [1] D. S. Johnson and L. A. McGeoch, "The Traveling Salesman Problem and Its Variations," Kluwer Academic Publishers, pp. 369-443, 2002.
- [2] A. Likas and V. T. Paschos, "A Note on a New Greedy-solution Representation and a New Greedy Parallelizable Heuristic for the Traveling Salesman Problem," *Chaos, Solitons and Fractals*, Vol. 13, pp. 71-78, 2002.
- [3] A. Schrijver, "On the History of Combinatorial Optimization (till 1960)," in *Handbook of Discrete Optimization* (K. Aardal, G.L. Nemhauser, R. Weismantel, eds.), Elsevier, Amsterdam, pp. 1-68, 2005.
- [4] E. Charniak and M. Herlihy, "CSC 751 Computational Complexity: Local Search Heuristics," Department of Computer Science, Brown University, 2008.
- [5] J. Pleines, "ZIP-Methode: ein Kombinatorischer Ansatz zur Optimalen Lösung Allgemeiner Traveling-Salesman-Problem (TSP)," *Können bekannte Lösungen nicht nur auf*

Gesamtgrphen sondern auf Teilgraphen angewandt werden, so bringt die ZIP-Methode den entscheidenden Quantensprung der rechentechnischen Vereinfachung, <http://www.jochen-pleines.de/download/ZIP2006.pdf>, 2006.

- [6] L. Stougie, "2P350: Optimiseringsmethoden," <http://www.win.tue.nl/~leen/OW/2P350/Week8/week8.pdf>, College Wordt ggeven op vinjdagmiddag, 2001.
- [7] L. H. Chuin, "IS 703: Decision Support and Optimization," School of Information Systems, Department of Computer Science, Brown University, 2008.
- [8] S. U. Lee, "The Extended k-opt Algorithm for Traveling Salesman Problem," *Journal of Korea Society of Computer Information*, Vol. 17, No. 10, pp. 155-165, Oct. 2012.
- [9] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a Large-scale Traveling-Salesman Problem," The Rand Corporation, <http://www.cse.wustl.edu/~chen/7102/TSP.pdf>, 1954.

## 저자 소개



이 상 운(Sang-Un, Lee)

1983년 ~ 1987년 :

한국항공대학교 항공전자공학과 (학사)

1995년 ~ 1997년 :

경상대학교 컴퓨터학과 (석사)

1998년 ~ 2001년 :

경상대학교 컴퓨터학과 (박사)

2003.3 ~ 현재 :

강릉원주대학교 멀티미디어공학과 부교수

관심분야 : 소프트웨어 프로젝트 관리,

소프트웨어 개발 방법론,

소프트웨어 신뢰성,

그래프 알고리즘

e-mail : [sulee@gwmu.ac.kr](mailto:sulee@gwmu.ac.kr)