

외판원 문제를 위한 효율적인 분산 최근접 휴리스틱 알고리즘

김 정숙*, 이희영**

*김포대학 컴퓨터 계열 소프트웨어 개발 전공

**동국대학교 컴퓨터공학과

e-mail:{kimjs@kimpo.ac.kr, klfamily@chollian.net

An Efficient Distributed Nearest Neighbor Heuristic for the Traveling Salesman Problem

Jung-Sook Kim*, Hee-Young Lee**

*Div. of Computer Science, Kimpo College

**Dept of Computer Engineering, Dongguk University

요약

외판원 문제(Traveling Salesman Problem)는 주어진 n 개의 도시들과 그 도시들간의 거리 비용이 주어졌을 때, 처음 출발도시에서부터 정확히 한 도시는 한 번씩만 방문하여 다시 출발도시로 돌아오면서 방문한 도시들을 연결하는 최소의 비용이 드는 경로를 찾는 문제로 최적해(optimal value)를 구하는 것은 전형적인 NP-완전 문제중의 하나이다[2, 4, 5, 8]. 따라서 이들의 수행시간을 줄이고자 하는 연구가 많이 진행된다. 본 논문에서는 외판원 문제의 최적의 해를 구하는데, 휴리스틱 알고리즘인 최근접 휴리스틱을 이용한다. 물론 수행 시간을 줄이고자 최적화 문제에서 좋은 성능을 보이는 유전 알고리즘(Genetic Algorithm)으로 얻은 근사해(near optimal)를 초기 분기 함수로 사용하고, 근거리 통신망(Local Area Network)에 기반한 분산 처리 환경에서 여러 프로세서에 분산시켜 병렬성을 살린다.

1. 서론

외판원 문제는 주어진 n 개의 도시들과 그 도시들간의 거리 비용이 주어졌을 때, 처음 출발도시에서부터 정확히 한 도시는 한 번씩만 방문하여 다시 출발도시로 돌아오면서 방문한 도시들을 연결하는 최소의 비용이 드는 경로를 찾는 문제로 최적해를 구하는 것은 전형적인 NP-완전 문제중의 하나이다[2, 4, 5, 8]. 출발도시에서 시작해서 나머지 $(n-1)$ 개의 도시를 한 번씩 거치는 경로의 총수는 $(n-1)!$ 만큼 존재하게 되어 계산 복잡도는 지수시간이 된다. 그리고 이 문제는 n 개의 도시 방문 순서를 일부분 바꾸더라도 경로의 합이 증가하게 되는 국소 최소점이 많이 있는 문제이다. 이렇게 국소 최소점이 많은 울퉁불퉁한 표면(rugged landscape)에서 전체 최적해를 찾는 문제는 변수 공간 전체에 대한 검색을 하지 않는 한 불가능하다.

그러나 이렇게 복잡한 외판원 문제가 실제 다른

최적화 문제들, 예를 들어 VLSI 반도체 설계에서 연결선의 길이를 최소화하는 경우에 응용이 될 뿐만 아니라, NP-완전 문제들을 현실적으로 해결하려는 노력의 일환이 된다. 따라서 외판원 문제를 해결하기 위한 다양한 연구들이 시도되고 있다. 분기 한정법(Branch-and-Bound) 알고리즘[8]이나 동적 프로그래밍(Dynamic Programming) 알고리즘[8] 방법과 같이 최적해를 구하는 방법이 있고, 확률적 탐색 휴리스틱(probability search heuristic)에 근거해서 근사해(near optimal value)를 구하는 유전 알고리즘 및 최근접 휴리스틱등 다양한 방법들이 있다. 그런데 최적해를 구하는 알고리즘들은 변수 공간 전체에 대한 검색을 하므로 최악의 경우 계산 복잡도가 지수 시간이 되어 많은 수행시간이 걸린다. 따라서 이들의 수행시간을 줄이고자 하는 연구가 많이 진행되고 있다. 특히 최적해를 구하는 방법들을 분산이나 병렬 환경에서 수행하는 알고리즘을 개발하거나, 또

는 최적의 해를 구하는 알고리즘을 단독으로 사용하는 것 보다 여러 가지 다양한 알고리즘을 서로 조합한 하이브리드 알고리즘을 작성하여 수행 시간을 줄이고자 하는 방안들도 활발히 연구되고 있다. 지역 탐색(Local Search)과 분기 한정법의 조합 및 유전 알고리즘과 분기 한정법의 조합이 하이브리드 알고리즘의 예이다.

본 논문에서는 외판원 문제의 최적의 해를 구하는데, 휴리스틱 알고리즘인 최근접 휴리스틱을 이용한다. 물론 수행 시간을 줄이고자 최적화 문제에서 좋은 성능을 보이는 유전 알고리즘으로 얻은 근사해를 분기함수로 사용하고, 근거리 통신망(Local Area Network)에 기반한 분산 처리 환경에서 여러 프로세서에 분산시켜 병렬성을 살린다.

논문의 구성은 먼저 2장에서 유전 알고리즘과 최근접 휴리스틱 알고리즘으로 외판원 문제를 해결하는 연구를 살펴보고, 3장에서는 본 논문에서 설계한 효율적인 분산 최근접 휴리스틱 알고리즘을 기술한다. 그리고 4장에서 실험한 내용과 결과를 설명하고 마지막 5장에서 결론을 내리고 향후 연구과제를 제시한다.

2. 외판원 문제를 위한 알고리즘들

외판원 문제를 위한 유전 알고리즘들은 다양한 후보해(candidate population)의 표현과 유전 연산을 제시하고 있다. 후보해의 표현 방법은 정수 표현 방법을 사용하고 있으며, 유전 연산자는 주로 교잡 연산과 돌연변이 연산이 사용되고 있다. 따라서 외판원 문제의 입력자료로서 개체는 모든 도시들을 정수로 매핑하여 n 개의 도시들을 임의로 생성 가능한 도시들의 배열로 입력한다. 이렇게 생성된 개체의 예를 들어 도시의 수가 17개인 경우 다음과 같다. (1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17, V)는 도시 "1"에서 출발하여 도시 2를 방문하고 다음 도시 3을 방문하고 등으로 하여 도시 17을 방문하는 경로를 나타내며, V는 목적 함수 값으로 도시들을 방문하면서 걸린 비용의 합을 나타낸다.

초기 후보 집단은 이렇게 임의의 수들의 배열로 생성한 후 생성된 개체들에 대한 비용을 계산한다. 그 비용들을 정렬한 후 비용이 적은 우수 인자들을 이용하여 교잡 연산자와 돌연 변이 및 역순 연산자를 수행하였다. 교잡 연산자는 순서 교잡(Order Crossover, OX)를 사용했으며, 돌연변이는 상호 교

환 돌연변이(Reciprocal Exchange Mutation)을 이용하였다. 역순 연산자는 한 개체내에서 임의의 위치를 정하여 그를 기점으로 도시의 배열 순서를 역으로 바꾸는 것이다.

그리고 외판원 문제의 목적 함수는 간단한 함수를 적용하는데, 여행 경로중에 걸린 비용의 합을 목적 함수 값으로 정하여 이들에 대해 평가한다. 또한 우수한 개체들을 생성하기 위한 선택 방법은 순위 선택 방법으로 목적 함수값을 정렬하여 좋은 값을 가지는 개체들을 선택하여 일정한 비율은 다음 세대로 재생산(reproduction)하고 나머지 개체에서 유전 연산들을 행하여 새로운 자손을 만든다. 자손의 세대는 항상 일정하게 유지한다.

최근접 휴리스틱 알고리즘은 처음 시작에서부터 점차 한 단계씩 진행하면서 해를 구해 추가하고, 다시 다음 한 단계 진행하고 하는 방법으로 해를 만들어 가는 알고리즘이다. 외판원 문제를 해결하기 위해 처음 임의의 시작 도시에서부터 다음 방문해야 할 도시를 선정하는데 처음 도시에서 갈 수 있는 다른 모든 도시들의 비용을 비교하여 가장 적은 도시를 방문하도록 한다. 같은 방법으로 현재 방문한 그 도시에서 다음 방문할 도시를 찾는데, 그 도시에서 갈 수 있는 도시들 중에서 지금까지 방문하지 않은 도시이면서 가장 비용이 적은 도시를 선정하여 다음 방문 도시로 한다. 이러한 방법으로 계속 반복하다 주어진 모든 도시들을 방문하여 외판원이 다시 시작 도시로 오면 멈추는 알고리즘이다.

그런데 이 알고리즘은 시작도시를 어디로 하느냐에 따라 얻어진 결과에 많은 차이가 있음을 알 수 있다. 이 알고리즘에서 최상의 결과를 얻기 위해 처음 시작도시를 어디로 해야 할지를 찾는 것이 중요한 일이다. 그러나 그 해를 구하기가 쉽지 않으므로 보통은 처음 시작 도시를 문제의 도시 순서에서 반 정도번째 되는 도시, 예를 들면 문제의 도시수가 17개라면 대략 8번째 도시나 9번째 정도의 도시를 처음 시작도시로 하여 방문을 시작한다[5]. 이를 뒷받침하는 이론이 Rosenkrantz, Stearns and Lewis(1977)[5]에 의해 발표되었다. 그 이론은 모든 $r > 1$ 이고, 임의의 큰 n 이 주어졌을 때, 최근접 휴리스틱으로 도시수가 n 개인 외판원 문제의 최적의 해를 구하기 위해서는 적어도 r 번 이상 수행해야 한다는 것이다. 본 논문에서는 모든 각 도시를 시작도시로 하여 최적의 해를 구한다.

3. 효율적인 분산 최근접 휴리스틱

분산 최근접 휴리스틱 알고리즘은 PVM 환경에서 마스터/슬레이브 모델을 사용한다. 하나의 프로세서가 마스터가 되고 여러개의 슬레이브 프로세서가 있다. 먼저 마스터에서 초기의 후보해를 생성하여 유전 알고리즘을 간단하게 수행하여 근사해를 구한다. 구해진 근사해와 해결해야 할 시작도시들을 pvm_send() 함수를 이용하여 각 슬레이브에게 전달한다. 근사해와 주어진 시작도시들을 받은 각 슬레이브들은 자신에게 주어진 시작도시들 중 순서대로 하나의 도시를 시작도시로 하면서 모든 해를 구한다. 해를 구하는 과정에서 마스터에서 보내준 근사해를 초기 분기함수로 사용하여, 만약 지금까지 구해진 해가 분기함수보다 많은 비용이 걸린다면 그 생성 경로는 더 이상 생성하지 않고 분기하여 다른 시작도시로 해를 같은 방법으로 구한다. 이러한 과정을 반복하여 주어진 시작도시들로부터 다 시작하여 얻어진 결과 중 가장 최상의 결과를 마스터에게 돌려준다. 그러면 마스터는 각 슬레이브로부터 온 결과들을 비교하여 최적의 해를 구한다. 다음 알고리즘 1은 본 논문에서 제안한 분산 최근접 휴리스틱 알고리즘이다.

```

Procedure Dis_Nnh( )
{
/* 유전 알고리즘 수행 */
NearOptimal = Genetic_Alg( );

/* 슬레이브에게 근사값과 방문해야 할 시작도시들을 나누어 전달함 */
for all Pi where 1 <= i <= n do
    PVM_send(NearOptimal, Partial_StrCities);
endfor

/* 여러 슬레이브 프로세서에서 분산 최근접 휴리스틱 수행 */
for all Pi where 1 <= i <= n do
    EachValue[i] = Dis_EffNnhi(NearOptimal, Partial_StrCities);
endfor

/* 각 슬레이브 프로세서에서 리턴한 값들 중 가장 최상의 결과를 선택함 */

```

```

for all Pi where 1 <= i <= n do
    optimal = Select(EachValue[i]);
endfor

```

알고리즘 1. 분산 최근접 휴리스틱 알고리즘

4. 실험 방법 및 결과

본 논문에서 제시한 내용들을 C 언어로 구현했으며, 실험 환경은 Sun 워크스테이션을 이더넷으로 연결하여 사용한다. 유전 알고리즘에서 사용한 파라미터는 후보해의 크기 : 10, 자손의 세대 : 50, 교잡 연산자의 비율 60%, 재생산 연산자의 비율 20%, 돌연변이 연산자의 비율 : 10%, 역순 연산자의 비율 : 10%이다. 그리고 외관원 문제에 대한 거리의 비용은 외관원 문제를 연구하는 많은 연구자들이 인용하고 있는 TSPLIB[7]에 있는 값들을 사용하였다.

실험은 여러 프로세서에서 최근접 휴리스틱 방법과 본 논문에서 제시한 효율적인 최근접 휴리스틱 방법으로 수행할 때의 수행 시간을 비교하였다. 그리고 분산 분기 한정법으로 수행할 때의 시간을 측정하여 비교하였다. 도시에 대한 비용을 ftv33.atsp와 ry48p.atsp 사용하였을때 결과는 다음 표 1, 2와 같다.

	수행 시간 비율
최근접 휴리스틱	1
제안한 알고리즘	0.93

표 1. 수행 결과

	수행 시간 비율
분산 분기 한정법	1
제안한 알고리즘	0.95

표 2. 분기 한정법 결과와 비교

5. 결론 및 향후 연구과제

외판원 문제(Traveling Salesman Problem)는 주어진 n 개의 도시들과 그 도시들간의 거리 비용이 주어졌을 때, 처음 출발도시에서부터 정확히 한 도시는 한 번씩만 방문하여 다시 출발도시로 돌아오면서 방문한 도시들을 연결하는 최소의 비용이 드는 경로를 찾는 문제로 최적해를 구하는 것은 전형적인 NP-완전 문제중의 하나이다. 본 논문에서는 외판원 문제의 최적의 해를 반드시 구할 수 있는 효율적인 분산 최근접 휴리스틱 알고리즘을 설계하고 실험하였다. 먼저 최근접 휴리스틱 알고리즘을 수행하면서 유전 알고리즘으로 얻은 근사해를 초기 분기 함수값으로 사용하여 수행 시간을 줄였으며, PVM 환경에서 마스터/슬레이브 모델을 사용하여 분산시켜 수행함으로써 병렬성을 얻을 수 있었다. 앞으로의 연구과제는 최근접 휴리스틱 알고리즘을 수행하는 도중 생성되는 경로의 정보를 이용하여 좀 더 빨리 최적해를 구하는 방법을 연구하는 일이다.

참고문헌

- [1] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderam, "PVM : Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing", *The MIT Press*, 1992
- [2] D. E. Goldberg, Genetic Algorithms : in Search and Optimization, *Addison-Wesley*, pp. 1-125, 1989.
- [3] J. Kim, Y. Hong, "A Distributed Hybrid Algorithm for the Traveling Salesman Problem", *IASTED, AI'99*, Feb. 1999.
- [4] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, *Springer-verlag*, pp. 209-237, 1995.
- [5] G. Reinelt, The Traveling Salesman Computational Solutions for TSP Applications, *Springer-Verlag*, 1994.
- [6] T.C. Sapountzis, "The Traveling Salesman Problem", *IEEE Computing Futures*, pp. 60-64, Spring, 1991.
- [7] TSPLIB,
<http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/ATSP.html>.
- [8] E. Horowitz, S. Sahni, Computer Algorithms, *Computer Science*, pp. 370-421, 1978.