

외판원 문제를 위한 다중 에이전트를 이용한 분산 유전 알고리즘의 이주 정책

김정숙, 이회영
김포대학 컴퓨터계열, 동국대학교 컴퓨터공학과

Migration Policy in Distributed Genetic Algorithm Using Multi-Agent for the Traveling Salesman Problem

Jung-Sook Kim, Hee-Young Lee
Dept. of Computer Science, Kimpo College,
Dept. of Computer Engineering, Dongguk University

요 약

분산 유전 알고리즘은 외판원 문제를 해결하는데 효율적이고 적용하기 쉬운 알고리즘이다. 그러나 다중 후보해를 가진 분산 유전 알고리즘을 수행할 때, 효율성과 정확성에 영향을 주는 많은 요소들이 고려되어야 한다. 후보해의 크기를 얼마로 할 것인지, 이주의 비율 및 횟수는 어떻게 할 것인지와 그리고 어떤 개체들을 선택해서, 어떤 후보해 개체와 교환할 것인가가 중요하게 고려되어야 한다. 따라서 본 논문에서는 이주해야 할 개체의 크기를 동적으로 변경하면서 교환하는 방법과 또한 개체들이 이주되어야 할 위치를 결정하는 이주 정책을 개발하고 실험하였다.

1. 서론

외판원 문제는 주어진 n 개의 도시들과 그 도시들 간의 거리 비용이 주어졌을 때, 처음 출발도시에서부터 정확히 한 도시는 한번씩만 방문하여 다시 출발도시로 돌아오면서 방문한 도시들을 연결하는 최소의 비용이 드는 경로를 찾는 문제로 최적해를 구하는 것은 전형적인 NP-완전 문제중의 하나이다[5,6]. 출발도시에서 시작해서 나머지 $(n-1)$ 개의 도시를 한 번씩 거치는 경로의 총수는 $(n-1)!$ 만큼 존재하게 되어 계산복잡도는 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 문제들의 응용이 될 뿐만 아니라, NP-완전 문제를 현실적

으로 해결하려는 노력의 일환이 된다. 따라서 외판원 문제를 해결하기 위한 다양한 연구들이 시도되고 있다. 분기 한정법(Branch-and-Bound) 알고리즘이나 동적 프로그래밍(Dynamic Programming) 알고리즘 방법과 같이 최적해를 구하는 연구들[5]이 있고, 확률적 탐색 휴리스틱(probability search heuristic)에 근거해서 근사해(near optimal)를 구하는 유전 알고리즘[3] 등 다양한 방법들이 개발되었다. 더불어 요즘은 통신망과 가격이 저렴하면서 성능 좋은 분산 시스템의 발달로 이러한 분산 환경에서 수행시간을 단축시키면서 성능 향상을 가져올 수 있는 분산 알고리즘들이 많이 개발되고 있다. 특히 분산 인공지능이라는 분야에서 비롯된 에이전트는 네트워크의 부하를 줄이고 비동기적이고 자율적으로 실행되며, 동적으로 적응할 수 있고, 플랫폼에 독립적이라는 장점을 가지며, 네트워크 상에 분산되어 존재하는 호스트들로 구성된 가상의 분산 컴퓨팅 환경을 구축할 수 있다[2]. 이러한 분산 환경에서 다중 후보해를 가진 분산 유전 알고리

즘에서 성능에 영향을 많이 미치는 몇 가지 요소들이 고려되어야 한다[1]. 가장 먼저 노드간에 이동해야 할 후보해의 크기가 결정되어야 하고, 그 선택된 후보해가 어떤 노드의 후보해와 교환되어야 할 것인가가 결정되어야 한다. 더불어 교환해야 할 후보해는 어떻게 선정되며 어떤 개체를 대체할 것인가와, 얼마나 자주 개체 교환이 발생하여야 하는지, 그리고 후보해의 크기는 얼마로 해야 가장 우수한 해를 구할 수 있을 것인가를 고려해야 한다.

그래서 본 논문에서는 이러한 여러 가지 성능에 영향을 미치는 요소들을 고려하고 수행시간을 줄이면서 성능향상을 가져올 수 있는 분산 유전 알고리즘을 개발하였다.

먼저 후보해의 개체가 변경되어야 할 슬래ιβ 위치를 결정하는 새로운 개체 이동정책을 개발하였다.

그리고 노드간의 이동해야 할 후보해의 크기를 동적으로 변화시킬 수 있는 방법을 개발하였다. 이동해야 할 개체의 크기를 후보해 교환시마다 윈도우 크기가 변하는 방법이다. 가장 작게 개체가 이동할 수 있는 개체의 크기는 1에서 시작하며, 후보해 크기의 10% 범위 크기 안에서 임의로 크기를 생성시켜 이동되어질 개체의 크기를 결정하는 방법을 개발하였다.

또한 후보해 교환 횟수가 수행 시간에 많은 영향을 미치므로, 통신 연산을 고려하여 교환시기를 결정하는 방법을 비교 실험하였다. 먼저 매 세대마다 후보해를 교환하는 알고리즘과 일정한 비율로 세대수를 증가시키면서 후보해들을 교환하는 방법을 개발하고 실험하였다.

본 논문의 구성은 먼저 2장에서 관련 연구를 살펴보고, 3장에서 새롭게 제안한 개체이동 정책과 개체이동 주기를 가지고 수행하는 분산 유전 알고리즘을 기술하고, 4장에서는 후보해들이 이동해야 할 위치를 선정하는 개체 이동 정책과 개체들이 어떠한 주기로 교환되어야 하는가를 결정하는 알고리즘에 대해 논의한다. 그리고 또한 이동해야 할 개체의 크기를 동적으로 조절하는 이동해야 할 개체의 크기를 결정하는 방법을 기술하고, 5장에서는 실험한 내용과 결과를 분석하며, 마지막 6장에서 결론을 내리고 향후 연구과제를 제시한다.

2. 관련 연구

2.1 유전 알고리즘

유전 알고리즘을 특정 문제에 적용시키기 위해서는

개체가 문제의 해결책을 어떻게 표시하도록 할 것인지를 설계해야 하고, 개체의 우수한 정도를 나타내는 적응도를 계산하는 적합성 함수(fitness function)를 어떻게 정의할 것인지, 또 우수한 개체들을 집단에서 어떤 방법으로 선택하고 몇 개로 복제해야 하는지, 교차 연산과 돌연변이 연산을 어떻게 정의할 것인지를 결정해야 한다. 이들 적용 방법이 어느 정도로 문제의 특성을 잘 반영하고 있는가에 따라 알고리즘의 성능이 크게 달라진다. 또한, 유전 알고리즘은 자체에 병렬성이 내재되어 있어서 수행 시간의 향상과 근접해의 향상을 가져오기 위해 병렬로 많이 처리한다.

그리고 분산 인공지능이라는 분야에서 비롯된 에이전트 개념은 컴퓨터 통신의 보편화와 더불어 현재 가장 중요한 기술의 하나로 많은 연구가 진행되고 있다. 에이전트에 대한 정의는 많이 있지만 공통적으로 에이전트는 지속적으로 환경에서 지각된 것과 내부 지식을 바탕으로 추론하며 그 결과 행동하여 환경에 영향을 미치고 또한 사용자를 포함한 다른 에이전트와 의사소통하는 지속적으로 존재하는 소프트웨어 요소라고 할 수 있다. 특히 다중 에이전트 시스템은 상호 작용하는 다수의 에이전트를 이용하여 설계하고 구현된 시스템으로서 문제에 대한 서로 다른 시각과 해결방식을 갖고 있는 다수의 에이전트로 구성된 문제를 표현하기에 적합하다.

3. 분산 유전 알고리즘

분산 환경에서 다중 에이전트들을 이용해서 수행하는 분산 유전 알고리즘의 구성도는 그림 1과 같이 마스터/슬래ιβ(master/slave) 모델로 구성된다. 마스터 에이전트는 후보해를 생성해서 슬래ιβ 에이전트들에게 전송하면 슬래ιβ 에이전트들은 각 자신의 능력에 따라 분산 유전 알고리즘을 수행하여 얻은 결과를 마스터 에이전트에게 전달한다. 마스터 에이전트가 가장 좋은 근사해를 얻기 위해서 각 슬래ιβ 후보들이 반환한 부분 근사해 값들을 관리하는 리스트를 가지고 있다. 부분 후보해들을 유지하는 리스트는 개체 이동이 일어나야 할 슬래ιβ 에이전트들을 선정하기 위해 사용된다. 먼저 슬래ιβ 에이전트들로부터 전송된 결과값들을 비교하면서 동적으로 개체 이동 크기와 개체 이동 정책을 연산하여 얻은 결과를 각 슬래ιβ 에이전트들에게 알려 준다. 그러면 슬래ιβ 에이전트들은 개체 이동크기만큼 개체를 개체 이동 정책 결과에 따라 개체를 이동시킨 후 다시 분산 유전 알고리즘을 수행한다. 이렇게 에이전트간에

새로운 개체 이동 정책이 적용되어 더욱 좋은 근사해를 구할 수 있도록 설계하였다.

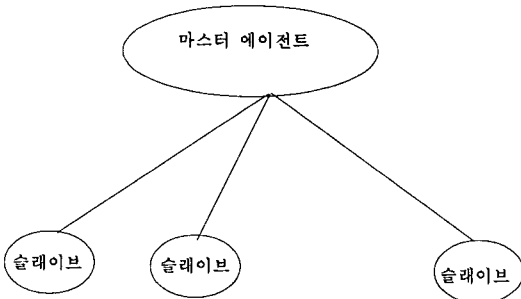


그림 1. 시스템 구조도

4. 에이전트간 개체 이동 정책과 이동 주기

앞에서도 설명했지만 다중 후보해를 가진 분산 유전 알고리즘에서 고려해야 할 중요한 요소가 어떤 에이전트들간에 개체의 이동이 일어나야 하는지를 선택하는 것과 어떤 주기로 개체 이동을 시켜야 성능이 좋은 것인지에 대해 많은 연구들이 수행되고 있다. 본 논문에서는 개체 이동이 일어나야 할 에이전트를 선정할 때, 각 슬레이브에서 전달된 해들을 비교하여 두 그룹으로 나눈 후 에이전트들을 선정하는 개체 이동 정책을 개발하였다. 또한 개체이동이 일어날 때, 이동되어야 할 개체의 크기를 결정하는 방법과, 개체들을 이동시켜야 할 주기를 변경하는 방안을 개발하였다.

4.1 개체 이동 정책

개체 이동이 일어나야 할 슬레이브 에이전트를 선정할 때, 마스터에서 각 슬레이브에서 전달된 근사해들을 비교하여, 근사해가 비슷한 곳을 한 그룹으로 묶고, 나머지를 다른 한 그룹으로 나눈다. 그룹의 크기는 동일하게 유지하며, 개체 이동은 그룹내에서 이루어지는 것이 아니라 그룹간에 이루어진다. 이는 비슷한 해를 발생하는 슬레이브에 새로운 개체를 이동시켜도 비슷한 해를 생성해 근사해의 조기 수렴을 발생시킬 수 있다. 따라서 이러한 조기 수렴 현상을 방지하기 위해서 새롭게 개체 이동정책을 개발하였다. 특히 그룹내에서 개체 이동이 발생할 슬레이브들을 선택할 때에도 바로 전 수행에서 얻은 근사해와 다음에서 수행하여 얻은 근사해의 변화를 비교하여 일정한 임계값 보다 적게 변화가 일어난 슬레이브를 선택한다. 임계값은 임의로 정해지며, 세대가 지날수록 적은 값을 가지도록 한다. 근사해 비교식은 다음과 같다.

$$|(i-1)\text{th 근사값} - (i)\text{th 근사값}| < \text{임계값} \quad \text{식 (1)}$$

4.2 개체 이동 주기

개체 이동 정책에 따라 개체 이동을 할 때, 통신 연산등을 고려하여 매 세대마다 개체를 교환하는 것이 아니라 일정한 세대가 지나고 나면 개체를 이동시키는 방법과, 세대가 지나감에 따라 최적해에 근접하는 해를 생성하므로 이를 보존하기 위해 점차 세대수를 증가시키면서 개체를 교환하는 알고리즘을 실험하였다.

4.3 임의 동적 개체이동 크기

임의 동적 개체이동 크기 방법은 개체 이동이 일어날 때마다 이동해야 할 개체의 크기를 임의로 생성하여 그 크기만큼 교환이 일어나는 알고리즘이다. 여기서 이동해야 할 개체의 크기는 가장 최소의 크기는 1에서부터 최대의 크기는 후보해 크기의 10% 범위 안에서 생성되도록 하였다.

5. 실험 방법

본 논문에서 제시한 내용들은 Java를 사용하여 구현하고 있으며, 실험 환경은 LAN으로 연결되어 있는 분산환경을 사용한다. 슬레이브 에이전트의 수를 점차 증가시키면서 수행하는데, 하나의 슬레이브 에이전트의 후보해 크기는 10으로 하였다. 슬레이브 에이전트간에 이동하는 개체의 윈도우 크기도 변경하면서 실험하였다. 유전 알고리즘에서 사용한 파라미터는 다음과 같다.

표 1 실험에서 사용한 파라미터들

	크기 및 비율
후보해의 크기	가변적(에이전트 당 10)
자손의 세대	가변적
교잡 연산자의 비율	60%
돌연변이 연산자의 비율	10%
역순 연산자의 비율	10%

모든 슬레이브 에이전트에서 사용한 연산의 비율

및 후보해의 크기 그리고 세대수 등 모든 조건은 동일하게 적용하여 수행하였다. 그리고 외판원 문제에 대한 거리의 비용은 외판원 문제를 연구하는 많은 연구자들이 인용하고 있는 TSPLIB[9] 에 있는 값들을 사용하였으며, 일부는 임의로 생성하여 실험하는 중이다. 매 세대마다 개체를 교환하지 않고 통신 연산을 고려하여 10세대를 수행한 후 개체를 교환하도록 실험중이다.

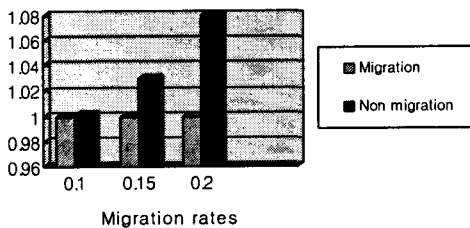


그림 2 변하는 개체 이동 비율에 대한 근사해 비율

6. 결 론

본 논문에서는 외판원 문제를 분산 환경에서 마스터/슬레이브 모델의 다중 에이전트들을 이용해서 우수한 근접해를 구할 수 있는 분산 유전 알고리즘을 개발하였다. 다중 후보해를 이용한 분산 유전 알고리즘을 수행할 때, 고려되어야 할 요소가 에이전트들 간에 이동해야 할 개체의 크기를 얼마로 해야 할 것인가와, 어떤 방법으로 에이전트를 선택해서 개체들을 이동시키느냐이다. 이에 이동 개체 크기를 동적으로 조정해서 사용할 수 있는 윈도우를 설계하였으며, 개체들을 교환할 위치를 결정하는 새로운 개체 이동 정책을 개발하고 실험하였다. 앞으로 연구과제는 다양한 실험을 완성해야 하며, 이동해야 할 개체의 크기 곧 최적의 θ 를 결정하는 좀 더 효율적인 방법을 모색하는 일이다.

[참고문헌]

[1] Erick Cantú-Paz, "Topologies, Migration Rates, and Multi-Population Parallel Genetic Algorithms", *IlligAL Report No. 99007*, January, 1999.
 [2] Fazlollahi, B., Vahidov, R.M., Aliev, R.A., "Multi-agent distributed intelligent system based on fuzzy decision making", *International*

Journal of Intelligent Systems, Vol. 15, No. 9, pp. 849-858, 2000.
 [3] M. Gen, R. Cheng, *Genetic Algorithms and Engineering Design, A Wiley-Interscience Publication*, pp. 118-132, 1997.
 [4] D. E. Goldberg, *Genetic Algorithms : in Search and Optimization, Addison-Wesley*, pp. 1-125, 1989.
 [5] E. Horowitz, S. Sahni, *Computer Algorithms, Computer Science*, pp. 370-421, 1978.
 [6] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs, Springer-verlag*, pp. 209-237, 1995.
 [7] G. Reinelt, *The Traveling Salesman Computational Solutions for TSP Applications, Springer-Verlag*, 1994.
 [8] T.C. Sapountzis, "The Traveling Salesman Problem", *IEEE Computing Futures*, pp. 60-64, Spring, 1991.
 [9] TSPLIB, <http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/ATSP.html>
 [10] M. Wooldrige, *An Introduction to MultiAgent Systems*, John Wiley & Sons, LTD, 2002.