

이동 에이전트를 이용한 병렬 유전자 알고리즘의 성능연구

조용만¹⁾ 강태원 김미숙
강릉대학교 컴퓨터공학과
(ymcho, twkang, mskim)@kangnung.ac.kr

Parallel Genetic Algorithm Performance Study using Mobile Agent

Yong-Man Cho Tae-Won Kang Mi-Suk Kim
Dept. of Computer Science, Kangnung University

요 약

유전자알고리즘을 병렬로 처리하려는 이유는 수행시간의 향상과 최적해의 향상이다. 하지만 이에 대한 연구와 응용이 적은데 이는 연구 환경이 열악하기 때문이다. 즉, 슈퍼컴퓨터와 같은 고가의 장비가 필요하며, 그것이 보편적으로 우리 주변에 있지 않다는 것이 가장 큰 장애가 되었다. 이를 극복하기 위한 방법은 에이전트라는 소프트웨어를 이용해서 유전자 알고리즘을 병렬로 처리를 하는 것이다. 이 연구에서는 이런 방법으로 유전자 알고리즘을 병렬로 처리를 하여도 수행시간의 향상과 최적해의 향상을 보일 수 있는지를 연구한다.

1. 연구배경

유전자 알고리즘[1]은 문제에 내포된 지식을 사용하지 않으며, 알고리즘이 간단하고 강하며 특히, 다변량 문제에 대한 최적해를 매우 효과적으로 찾을 수 있어 많은 응용분야에서 주목을 받고 있다. 또한, 유전자 알고리즘은 다수의 개체(후보 해)들로 구성된 모집단을 대상으로 동시에 탐색이 이루어지기 때문에 자체에 병렬성(parallelism)이 내재되어 있다. 일반적으로 유전자 알고리즘을 병렬로 처리하려는 이유는 두 가지가 있다[2,3]. 첫 번째는 수행시간의 향상을 꾀하는 경우이고, 두 번째는 최적해의 향상을 꾀하는 경우이다. 이러한 이유로 인하여 병렬 유전자 알고리즘(PGA)에 대한 연구는 유전자 알고리즘의 연구와 출발을 같이하였다. 그러나, 동일한 알고리즘을 병렬로 처리하였을 때의 이득이 명백한 것에 비하여 그에 대한 연구 및 응용이 상대적으로 적은 것이 사실이다. 이렇듯 PGA에 대한 연구 및 응용이 상대적으로 적은 가장 큰 이유는 병렬처리를 위한 환경이 열악했기 때문이다. 즉, 슈퍼컴퓨터와 같은 고가의 장비가 필요하며, 그것이 보편적으로 우리 주위에 있지 않다는 것이

큰 장애가 되었다. 그래서 인터넷상에 분산되어 있는 다수의 일반 컴퓨터들을 이용하는 방법을 생각하게 되었다.

분산 인공지능이라는 분야에서 비롯된 에이전트 개념은 컴퓨터 통신의 보편화와 더불어 현재 가장 중요한 기술의 하나로 인정되어 많은 관련연구가 진행되고 있다. 에이전트란 환경에서 얻은 지식과 자신의 내부 지식을 바탕으로 추론하고, 추론의 결과로 환경에 영향을 미치며 또한, 사용자를 포함한 다른 에이전트와 상호 작용하는 소프트웨어를 말한다. 에이전트는 네트워크의 부하를 줄이고 비동기적이고 자율적으로 실행되며, 동적으로 적용할 수 있고, 플랫폼에 독립적이라는 장점을 갖는 것으로 대단히 많은 연구가 진행되고 있는 최신 기술이다. 이러한 이동 에이전트에서 "이동"이라는 측면을 이용하여 네트워크 상에 분산되어 존재하는 호스트들로 구성된 가상의 병렬분산 컴퓨팅 환경을 구축할 수 있으며, 이 환경은 에이전트가 갖는 자율성, 지능, 및 협동 능력으로 인하여 클러스터링과 다르게 대단히 유연한 것이라는 특징을 갖는다.

이러한 배경아래 이 논문은 이동 에이전트 시스템에 기반한 가상의 병렬분산 컴퓨팅 환경과 거기에서 수행되는

병렬 유전자 알고리즘을 개발하고 그것을 다양한 문제에 적용하여 평가하는 것을 목적으로 한다.

이 논문의 구성은 다음과 같다. 2장에서는 성능분석을 위한 유전자 알고리즘에 대하여, 3장에서는 이동 에이전트의 구성방법에 대하여 설명하고, 4장에서는 실험을 통하여 결과를 분석하고 마지막으로 5장에서 결론을 맺는다.

2. 성능 분석을 위한 유전자 알고리즘

이 논문에서 사용한 유전자 알고리즘은 표현에서 하나의 유전자는 4비트로 이루어진 2진수 벡터이고, 100개의 유전자가 하나의 염색체를 이룬다. 초기 모집단은 임의 생성법을 사용하여 만들었고, 적응도 평가는 4비트로 이루어진 2진수 벡터가 0에서 15까지의 10진수 값을 가지는데 이값들의 범위를 -1에서 1까지 등간격으로 분할하여 대응되는 값들을 더해서 사용한다. 즉, (0 0 0 0)을 가진 유전자는 10진수로 0의 값을 가지면서 적응도 값은 -1이고, (1 1 1 1)을 가진 유전자는 10진수로 15의 값을 가지면서 적응도 값은 1이다. 적응도의 값이 클수록 우수한 유전자로 본다. 우수한 유전자가 선택될 확률은 5%이고, 다음 세대로 전해질 확률은 40%이다.

3. 이동 에이전트의 구성

에이전트는 네트워크 에이전트 또는 순회 에이전트라고 부르며, 프로그램 자체가 네트워크를 돌아다니며 수행되는 것을 말한다.

이 논문에서는 IBM에서 개발한 Aglets[4]을 사용하였다. Aglet은 이름에서 알 수 있듯이 Applet과 Agent를 결합한 것으로, Applet 프로그래머가 쉽게 이동 에이전트 응용을 개발하는데 초점을 맞춘 것이다. Aglet을 이용한 상업용 시스템이 나와있으며, 많은 이동에이전트 연구의 플랫폼으로 사용되고 있다.

이동 에이전트를 이용한 병렬 유전자 알고리즘의 성능 평가를 위한 이동 에이전트의 구성은 마스터 에이전트와 슬레이브 에이전트로 구현하였다. 마스터 에이전트는 여러 개의 슬레이브 에이전트를 분산되어 있는 호스트로 이동시키는 에이전트이고, 슬레이브 에이전트는 각각 독립적으로 유전자 알고리즘을 수행한 뒤 결과를 마스터 에이전트로 보낸다.

이 논문에서는 많은 모집단의 수와 세대수의 계산량을 위해 10개의 슬레이브 에이전트를 사용하여 실험하였다.

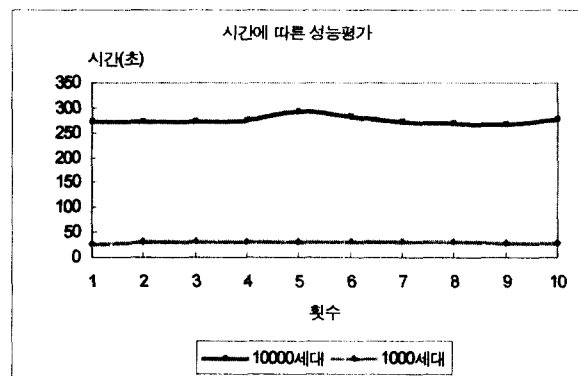
4. 실험결과

첫 번째 실험에서는 많은 계산량을 가진 유전자 알고리즘을 이용하여 모집단의 수가 같은 경우 한 대의 컴퓨터에서 수행되는 시간과 10개의 이동 에이전트가 세대수를 1/10로 하여 수행한 경우의 수행 시간을 비교하여 보았고, 두 번째의 실험에서는 비슷한 수행 시간에서 한 대의 컴퓨터에서 수행되는 유전자 알고리즘의 모집단의 평균 적합도와 10개의 에이전트가 분산된 호스트에서 계산한 모집단들의 평균 적합도를 비교해 보았다.

4.1 수행시간 비교

모집단의 수가 같은 경우, 한 대의 컴퓨터에서 수행한 유전자 알고리즘의 수행속도와 이동 에이전트를 이용해서 10대의 컴퓨터에 나누어서 수행한 경우의 시간차이를 비교하여 보았다.

실험은 한 대의 컴퓨터에서 유전자 알고리즘을 10000세대까지 계산을 하고, 이동 에이전트를 이용해서 10대의 컴퓨터에서 수행하는 유전자 알고리즘을 1000세대까지 계산한 경우의 수행 시간을 10회 반복하여 비교하여 보았다. 실험결과는 다음과 같다 < 그림 1 >.



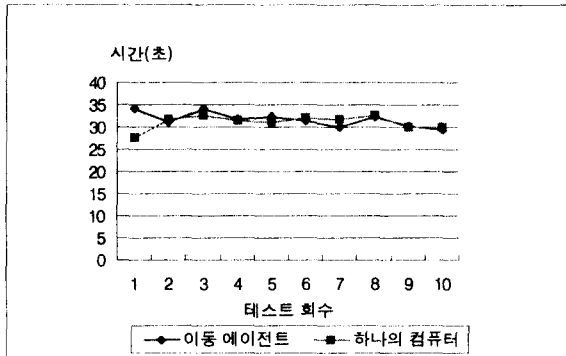
< 그림 1 > 수행시간 비교

실험에서와 같이 10개의 에이전트에 나누어서 수행되는 경우의 수행속도는 한 대의 컴퓨터에서 수행되는 유전자 알고리즘의 수행속도에 비해 약 8.9배의 빠른 결과를 보였다. 같은 문제를 10개의 컴퓨터에 분산해서 병렬로 처리할 경우 일반적으로 10배의 수행시간 향상이 있어야 하지만, 그렇지 않은 것은 상당량의 네트워크 오버헤드와 문제 분할로 인한 추가적인 계산량의 증가로 인한 오버헤드로 생각된다.

4.2 평균 적합도 비교

한 대의 컴퓨터에서 수행되는 유전자 알고리즘을 비슷한 수행 시간에 이동 에이전트를 이용해서 여러대의 컴퓨터에서 수행을 하면 최적해의 향상을 얻을 수 있는지를 실험하여 보았다.

실험은 한 대의 컴퓨터에서 수행한 유전자 알고리즘을 1000세대까지 계산한 경우의 모집단의 평균 적합도값과 이동 에이전트를 이용해서 10대의 컴퓨터에서 수행한 유전자 알고리즘을 같은 수의 세대까지 수행한 모집단의 평균 적합도값을 10회 반복하여 비교하여 보았다. 실험 결과는 다음과 같다 < 그림2, 표1 >.



< 그림2 > 수행 시간 비교

Agent0	99.721	99.792	99.716	99.708	99.761	99.751	99.704	99.700	99.695
Agent1	99.777	99.780	99.759	99.764	99.764	99.744	99.700	99.700	99.695
Agent2	99.777	99.696	99.696	99.697	99.735	99.804	99.705	99.700	99.752
Agent3	99.700	99.784	99.805	99.776	99.687	99.815	99.700	99.700	99.735
Agent4	99.741	99.744	99.752	99.740	99.741	99.787	99.700	99.700	99.721
Agent5	99.696	99.673	99.760	99.776	99.701	99.760	99.705	99.700	99.781
Agent6	99.659	99.852	99.616	99.671	99.631	99.816	99.700	99.700	99.691
Agent7	99.684	99.737	99.760	99.776	99.728	99.772	99.700	99.700	99.691
Agent8	99.686	99.732	99.784	99.784	99.747	99.777	99.777	99.700	99.707
Agent9	99.693	99.733	99.675	99.686	99.785	99.775	99.777	99.691	99.783
Agent10	99.746	99.797	99.622	99.764	99.783	99.783	99.711	99.695	99.706

< 표1 > 모집단의 평균 적합도 비교

<그림2>는 한 대의 컴퓨터와 이동 에이전트를 이용한 10대의 컴퓨터에서 같은 유전자 알고리즘을 수행한 시간을 비교해 보았다. 보여지는 바와 같이 수행 시간의 차이가 없었다.

<표1>에서 가로축은 한 대의 컴퓨터에서 유전자 알고리즘을 수행한 모집단의 평균 적합도이고 세로축은 각각의 이동 에이전트가 수행한 모집단의 평균 적합도이다.

중간의 색칠한 부분은 한 대의 컴퓨터에서 수행된 모집단의 평균 적합도보다 각각의 이동 에이전트에서 수행한 모집단의 평균 적합도가 우수한 경우에 표시를 하였다. 한 대의 컴퓨터에서 수행한 유전자 알고리즘의 평균 적합도보다 이동 에이전트를 이용하여 10대의 컴퓨터에서 수행한 경우의 평균 적합도가 우수한 경우를 보여주고 있는데 이것은 모집단의 크기를 크게했을 때 좀 더 우수한 해를 찾을 수 있다는 것을 보여주고 있다.

5. 결론

이 논문은 한 대의 컴퓨터에서 수행한 유전자 알고리즘 보다는 이동 에이전트를 이용하여 다수의 컴퓨터에서 병렬로 수행했을 때 수행 시간의 향상을 보이는 것을 알 수 있었으며, 일반적으로 이동 에이전트를 이용했을 때 최적해의 향상을 얻을 수 있다는 것도 알 수 있었다. 하지만 최적해의 향상은 모든 경우에 적용되는 것이 아니다.

6. 참고문헌

[1] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
 [2] Erick Cantu-Paz, "Implementing Fast and Flexible Parallel Genetic Algorithms", in *Practical Handbook of Genetic Algorithms*, Vol III, pp. 65-84., 1999.
 [3] S. Baluja, "A Massively Distributed Parallel Genetic Algorithm(mdpGA)", Carnegie Mellon University, 1992.
 [4] D. B.Lange, M. Oshima, "Programming and Deploying java Mobile Agents with Aglets". Addison Wesley, 1998.