

[9-67] DC(Deterministic Crowding) 와 Struggle 유전 알고리즘을 위한 전용 프로세서(Genetic Algorithm Processor) 설계

이 은 경*, 김 동 순**, 정 덕 진*

*인하대학교 정보통신공학과, **전자부품연구원

*only4simple@naver.com **dskim@keti.re.kr, *djchung@inha.ac.kr

Design of exclusive processor for DC(Deterministic Crowding Method) and Struggle genetic algorithm

Eun-Kyung Lee*, Dong-Sun Kim**, Duck-Jin Chung*

Dept. Information and Communication Engineering, Inha University,
Korea Electronics Technology Institute

요약

유전 알고리즘(Genetic Algorithm)은 주어진 환경에 능동적으로 적응하여 기존의 계산 알고리즘으로는 풀기 어려웠던 최적화 문제들을 좋은 성능으로 해결한다. 이런 유전 알고리즘은 그 특성상 반복적인 진화과정을 거치게 되므로 소프트웨어로 처리 하는데 많은 연산시간을 필요로 한다. 그렇기 때문에 유전 알고리즘의 연산을 빠르게 처리해 줄 하드웨어가 요구된다. 이 논문에서는 일반적 유전 알고리즘이 가질 수 있는 Genetic drift 와 지역 해 문제 등의 한계를 해결하는 중분화 알고리즘, 특히 DC(Deterministic Crowding) 와 Struggle GA 를 하드웨어로 구현하였으며, 이 두 유전 알고리즘은 선택신호에 의해 개별 동작하도록 설계되었다. 실험결과 GAP(Genetic Algorithm Processor)는 FPGA 구현 시 소프트웨어보다 DC 의 경우 약 145 배, Struggle 의 경우 약 300 배 빠른 연산시간을 보였다.

1. 서론

유전 알고리즘(Genetic Algorithm)은 생물의 유전과 진화 기법(mechanism)을 공학적으로 모형화(modeling)함으로써 생물이 갖는 환경에 대한 적응능력을 문제해결 방법에 적용한 알고리즘 이다. 유전 알고리즘은 풀고자 하는 문제의 해 집합을 정해진 형태의 자료구조로 표현한 후, 이들을 변형함으로써 점차적으로 해에 가까운 값들을 얻어내는 방법으로 진행된다.[1] 최근 유전 알고리즘은 공학의 여러 가지 응용 분야 중 탐색, 최적화 및 기계학습 등 복잡한 제약성을 가진 대규모의 문제 해결에 폭넓게 사용 되고 있으며, NP-완전 문제 등 기존의 계산 알고리즘으로 풀기 어려운 최적화 문제들에 좋은 성능을 보여주고 있다. 또한 유전 알고리즘은 구조가 단순하고 방법이 일반적이어서 응용범위가 넓은 특징을 가지고 있다.

유전 알고리즘은 그 특성상 반복적인 진화과정과 적응과정을 거치게 되는데 이러한 과정을 프로그램(상위레벨)으로 처리하면 많은 연산시간을 필요로 하게 된다. 따라서 유전 알고리즘의 연산을 빠르게 처리해 줄 별도의 하드웨어, 즉 GAP (Genetic Algorithm Processor)가 요구된다.

본 논문에서는 사용 환경에 대해 유연한 특성을 갖는 EHW(Evolvable hardware)에 적용이 적합한 중분화 알고리즘(Speciation Algorithm)을 하드웨어로 구현 한다. 특히, 중분화 알고리즘을 기본으로 하는 DC(Deterministic Crowding Method)와 Struggle 유전 알고리즘을 최적의 하드웨어로 설계하는 것을 목적으로 한다. DC와 Struggle 유전 알고리즘은 Simple 유전 알고리즘과 같은 일반적 유전 알고리즘이 하나의 최적 해만을 찾아내는 것과는 달리 2개 이상의 다중 해를

찾아내는 특징을 가지며, 일반적 유전자 알고리즘이 가질 수 있는 Genetic drift와 지역해 문제 등의 한계를 해결한다. [2] 기존의 유전자 알고리즘은 하나의 해만을 집중적으로 탐색하는데, 알고리즘의 특성상 우수한 염색체가 나타나면 그 염색체 중심의 해 영역으로 탐색 범위가 축소된다. 따라서 해 영역 전체를 동시에 고려하지 못한 채 하나의 해만을 얻게 된다. 하지만 Multi-object 기반의 중분화 알고리즘은 여러 종의 좋은 해를 찾는 것을 목적으로 하기 때문에, 이미 발견된 우수한 염색체 모두를 중요시한다. 따라서 좋은 유전자들이 발견되면 그것들을 유지하면서 그 값들을 중심으로 탐색교차에 의해 새로운 영역을 탐색해 나간다. 이러한 특성 때문에 진화의 후반까지 탐색 범위가 넓게 유지된다. 이러한 중분화 알고리즘의 특성 때문에 지역해 및 genetic drift 문제를 해결 할 수 있는 것이다.

DC와 Struggle 유전자 알고리즘은 수렴속도와 다양성(diversity)의 측면에서 서로 다른 장단점을 갖는데, 이러한 특성을 문제에 따라 선택적으로 적용할 수 있도록 선택신호에 의해 개별 동작하는 구조로 설계되었다. 이러한 설계 방법은 제안된 하드웨어의 문제해결 적용 범위를 넓힌다.

본 구조는 유전자 알고리즘의 성능 검증에 널리 사용되는 최적화 문제인 Dejong function(2)에 적용되었으며 FPGA로 구현되어 성공적으로 동작함을 확인하였다.

II. DC & Struggle Method 유전자 알고리즘

2.1 Deterministic Crowding Method

DC는 부모집단의 모든 개체들을 임의로 짝 짓는 것으로부터 시작 한다. 들쭉 짝지어진 개체들은