

진화 하드웨어를 위한 종 적응 진화 방법

Species Adaptive Evolution Method for Realization of Evolvable Hardware

반장봉* · 전호병* · 박창현* · 정구철** · 심귀보*

Chang-Bong Ban, Ho-Byung Chun, Chang-Hyun Park, Goo-Cheol Jeong, and Kwee-Bo Sim

* 중앙대학교 전자전기공학부

** 한국기술교육대학교 정보통신공학과

요 약

종의 분화는 생명체의 다양성을 유지하며, 좀더 환경에 적합한 생명체를 탄생시킨다. 그래서, 자연계의 진화에 모방을 둔 진화 알고리즘은 주어진 환경에 적응하기 위해 다양성을 유지해야 한다. 본 논문에서는 이러한 종의 분화 개념을 도입한다. 개체군의 각 개체들이 돌연변이를 통하여 자손을 생성하고, 그 중 일부가 분화하여 다음 세대의 개체를 이룬다. 각 개체들은 돌연변이에 의해 결정되는 일정한 해밍 공간 내의를 탐색공간으로 하고, 분화를 통하여 유효한 탐색공간을 점차 넓혀 탐색공간 전체에 대한 효율적인 탐색을 수행한다. 돌연변이를 통한 진화 방법으로 진화 하드웨어에 적용할 경우 내부구조의 변경이 적어 빠른 탐색효과를 가질 수 있다. 제안된 알고리즘을 2개의 최적화 문제에 적용하여 그 유용성을 확인한다.

ABSTRACT

Living creatures maintain their variety through species differentiation helps them to have more fitness for nature. So evolutionary algorithm based on biological evolution must maintain variety in order to adapt to nature. In this paper, we use the concept of species differentiation. Each individual of population creates their son using mutation, and next generation consists of them. Each individual explores search space determined by mutation. Useful search space is extended by differentiation, then population explorer whole search space very effectively. If evolvable hardware evolves through mutation, it is useful way to explorer search space because of less varying inner structure. We verify the effectiveness of the proposed method by applying it to two optimization problems.

Key Words : 종 적응 진화, 종의 분화, 탐색공간 분화, 돌연변이, 진화 하드웨어

1. 서 론

생명체는 종의 분화를 통하여 다양성을 유지한다. 자연계에는 다양한 종들이 존재하고, 각각의 종들은 그들에게 주어진 환경에 적응하며 살아간다. 자연계는 환경이 다양하게 변화한다. 생명체가 특정한 환경에 적합한 종만 존재한다면 다양하게 변화하는 환경에서 살아가기 힘들다. 그러나 생명체는 종의 분화를 통해 다양한 환경의 변화에 적응하고, 환경의 변화가 심하더라도 여러 개의 종이 존재함으로써 생명체의 존재를 유지시키나간다. 즉, 다양성은 좀 더 환경에 적합한 생명체가 탄생할 수 있다는 가능성을 제시하며, 환경의 변화에도 큰 적응성

을 갖도록 한다. 그래서 자연계의 진화에 기반을 둔 진화 알고리즘(Evolutionary Algorithms)이 적합한 해를 찾고 다양한 환경의 변화에 적응하려면 다양성을 유지할 필요가 있다. 본 논문에서는 돌연변이(Mutation)와 탐색공간의 분화에 기반을 둔 종 적응 진화방법(Species Adaptive Evolution Method)을 제시한다. 제시된 알고리즘의 진화 알고리즘의 기본 연산자인 돌연변이를 통하여 개체군의 다양성을 유지한다. 또한 각 개체들이 개별적으로 진화하면서 분화하고, 분화된 개체들에 의해 탐색공간을 분화하여 해 공간을 효율적으로 탐색하도록 한다.

재구성 가능한 하드웨어 내에서는 유전자 알고리즘의 개체로 사용되는 구조적 비트열(Bit String)을 돌연변이 시키면, 돌연변이가 이루어진 부분만을 재구성 할 수 있다[1]. 즉, 교차(Crossover)를 이용하여 구조적 비트열을 바꾸는 것 보다 빠른 탐색효과를 갖는다. 또한, 탐색공간의 분화(Differentiation)를 통해 국소해(Local Minimum)에 수렴하지 않고 해 공간 전체를 효율적으로 탐색한다. 제안된 알고리즘을 함수 최적화 문제에 적용하여 유용성을 확인하고, 진화 하드웨어에의 적용 가능성을 제시

접수일자 : 2000년 11월 18일

완료일자 : 2001년 01월 15일

본 연구는 과학기술부의 뇌과학 프로젝트(Braintech21)의 연구비지원으로 수행되었으며 연구비 지원에 감사드립니다.

한다.

본 논문의 구성은 다음과 같다. 2절에서는 유전자 알고리즘과 주요 연산자인 돌연변이 연산자에 대해서 소개한다. 3절에서는 제안된 알고리즘인 종 적응 진화 방법과 종진화 방법에서의 개체의 탐색과 분화에 대해서 소개한다. 4절에서는 제안된 알고리즘을 적용한 시뮬레이션의 결과를 분석하고, 진화 하드웨어의 적용 가능성을 제시한다. 마지막으로 5절에서 결론을 내리고 향후 과제에 대해 논의한다.

2. 유전자 알고리즘과 돌연변이

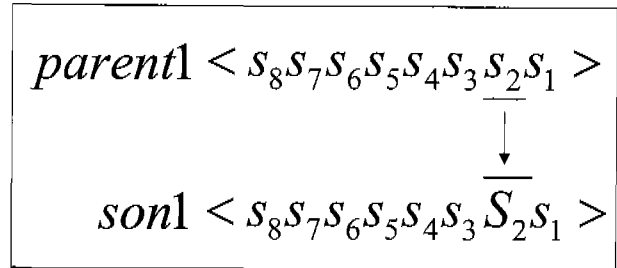
2.1 유전자 알고리즘

자연계의 적자생존의 진화 방식에 기반을 둔 유전자 알고리즘[2-4]은 지능적인 정보처리 시스템의 한가지 방법으로 널리 연구되고 있다. 유전자 알고리즘은 1975년 Holland에 의해 생물의 진화 과정의 계산모델로서 제시되었고[1], 집단기반 최적화 방법에 기반을 둔다. 즉, 기존의 방식인 해 공간내의 특정한 한 지점에서 최적해를 찾아 나가는 방식이 아닌 여러 지점이 집단을 이루어 병렬 적으로 최적해를 찾아내는 방법이다. 유전자 알고리즘은 초기 랜덤하게 생성된 집단에서 환경에 더 적합한 방향으로 각 개체들의 염색체 구조를 변화시켜나간다[5]. 집단내의 개체들이 교차나 돌연변이 등의 연산자에 의해 새로운 개체들을 생성하고, 그들 환경에 적합한 개체들이 선택되어 다음 단계의 집단을 이룬다. 이러한 새로운 하나의 집단 생성과정을 세대라고 한다. 각 세대 내에서, 이전 세대의 적합도가 높은 개체들의 일부분을 이용하여 좀더 적합도가 높은 개체들을 생성해 나간다. 이 유전자 알고리즘은 집단 기반 최적화 방법이기 때문에 다양한 환경에 대해서 강인성을 갖는다. 그러나, 유전자 알고리즘이 여러 가지 문제에 대해서 최적값을 찾는다 하더라도, 유전자 알고리즘의 성능은 시스템 설계자에 의존한다[6].

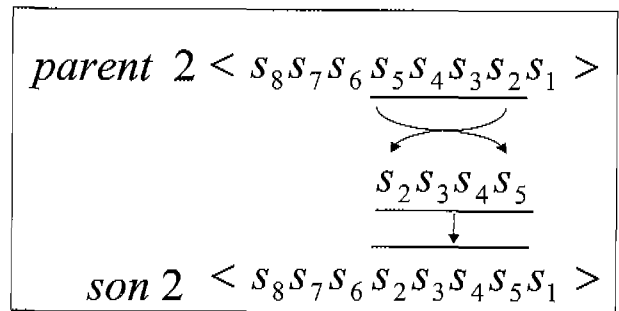
2.2 돌연변이

유전자 알고리즘의 주된 연산자는 교차와 돌연변이다. 교차는 개체군내에서 선택된 임의의 2개의 개체를 부모 개체로 하고, 이 부모 개체의 일부를 교환하여 새로운 개체를 생성한다. 즉, 교차는 부모 개체에게 의존하는 한정된 범위의 자손을 생성하는 연산자로서, 진화 하드웨어(Evolvable Hardware)에 있어서 교차는 하드웨어의 내부구조를 구조적으로 진화시키는데 한계가 있다. 돌연변이는 염색체 상의 어떤 유전자 좌의 값을 다른 대립 유전자로 치환함으로써 자손을 생성하는 연산자로서, 개체군의 다양성을 유지하며 구조적 변화를 일으킨다. 대표적인 돌연변이로 점 돌연변이와 역위가 있다. 점 돌연변이는 개체의 각 유전자 좌의 유전자에 대해서 정해진 돌연변이 확률 p_m 에 의해 다른 대립 유전자로 치환함으로써 달성된다. 예를 들어 그림 1(a)와 같이 비트열 $p_1 = \langle s_n s_{n-1} \dots s_2 s_1 \rangle$ 으로 나타내는 개체에 대해서 두 번째 유전자 좌에 돌연변이가 일어나면 $\langle s_n s_{n-1} \dots s_2 s_1 \rangle$ 인 새로운 개체가 생성된다. 역위는 정해진 확률에 의해 개체 상에서 임의로 선택된 2점간의 부분 문자열의 순서를 반전시키는 조작이다. 예를 들어 그림 1(b)과

같이 개체 $p_2 = \langle s_n s_{n-1} \dots s_2 s_1 \rangle$ 에 대해 2점 a, b ($1 \leq a < b \leq n$)를 임의로 발생시키고, 점 a 에서 점 b 까지의 문자열의 순서를 반전시킨다.



(a) 점 돌연변이
(a) Point mutation



(b) 역위 (a=2, b=5), (b) Inversion (a=2, b=5)

그림 1. 돌연변이 연산자의 연산 과정
Fig. 1. Process of mutation operator

진화 하드웨어의 경우, 비트 스트링을 이용하여 진화 하드웨어의 내부구조를 표현한다[1]. 진화 하드웨어를 진화시킬 경우, 내부구조를 표현하는 비트 스트링은 유전자 알고리즘에서의 하나의 개체로 사용된다. 만약 돌연변이 연산자에 의해 비트 스트링에 돌연변이가 이루어지면, 진화하드웨어의 내부구조의 구조적인 변화를 일으킬 수 있다.

3. 종 적응 진화 방법

3.1 종 적응 진화

본 논문에서는 진화 하드웨어에 적합한 종 적응 진화 방법을 제시한다. 이 방법은 자연계에서 종의 분화를 통하여 생명체의 다양성을 유지하면서 진화하는 방식을 이용한다. 자연계에서는 생물체들이여 가지 종의 형태로 존재하여 다양하게 변화하는 환경에 대해 능동적으로 진화할 수 있다. 종 적응 진화 방법에서는 주 연산자로 돌연변이를 사용하고, 개체군을 이루는 각 개체들이 개별적으로 진화한다. 초기 개체군은 적은 수 개의 개체를 랜덤하게 생성하여 구성한다. 진화 초기에는 N 개의 각 개체에 대해 돌연변이를 수행하여 M 개의 자손을 생성하고 그 중 S 개의 자손을 다음 세대로 넘긴다. S 개의 자손은 돌연변이를 통해 생성된 M 개의 자손중 베스트 개체(s_1)를 먼저 선택하고, 나머지 $M-1$ 개의

자손 중 베스트 개체와의 해밍거리와 적합도 값에 의한 선택도를 식 1과 같이 결정하여 S-1 개를 선택한다.

$$sel(s_n) = \alpha H(s_1, s_n) \cdot fit(s_n) \quad (1)$$

여기서, $H(s_1, s_n)$ 는 자손 베스트와의 해밍거리(Hamming Distance), $fit(s_n)$ 은 s_n 의 적합도(Fitness), α 는 비례 상수를 나타낸다.

다음 세대에서도 각 개체를 개별적으로 돌연변이를 수행하여 M개의 자손을 생성하고 그 중 해밍거리가 크고 베스트인 S개의 자손을 다음 세대로 넘긴다. 이런 방식으로 일정 크기의 개체군이 형성될 때까지 각 개체들을 분화시켜 개체군의 수를 늘린다. 일정 크기의 개체군이 형성되면 각 개체들은 분화를 멈추고 돌연변이에 의해 결정되는 해밍거리 이내의 공간을 탐색한다. 한 세대 내에서 각 개체들과 돌연변이를 수행하여 생성된 개체들의 총합을 제어할 필요가 있다. 여기서는 돌연변이 개체들을 포함해 총 개체군의 수가 일정하게 유지하도록 각 세대에서 개체들이 생성하는 자손의 수를 다음 식 2에 의해 결정한다.

$$M_{i+1} = M_i / S_i \quad (2)$$

여기서, M_{i+1} 은 $i+1$ 세대의 각 개체별 자손 생성 수, M_i 은 i 세대의 개체별 자손 생성 수, S_i 는 i 세대의 각 개체별 분화 수를 나타낸다. 진화 초기의 각 개체는 하나의 종(Species)으로 볼 수 있다. 하나의 개체 즉 종이 돌연변이를 통하여 여러 자손들을 생성하고, 그들 중 적합도와 염색체 구조의 차이를 고려하여 일부의 자손들이 다음 세대에는 새로운 종을 생성하는 것이다. 진화 초기에는 개체들이 분화를 통하여 탐색 영역을 넓히고, 동시에 유용한 탐색공간을 탐색해 나간다. 진화 후기에는 초기 개체들이 분화를 통하여 탐색한 유용한 공간에 대해서 집중적으로 탐색을 실시한다. 다음 그림 2는 종 적응 진화 방법의 개념도이다.

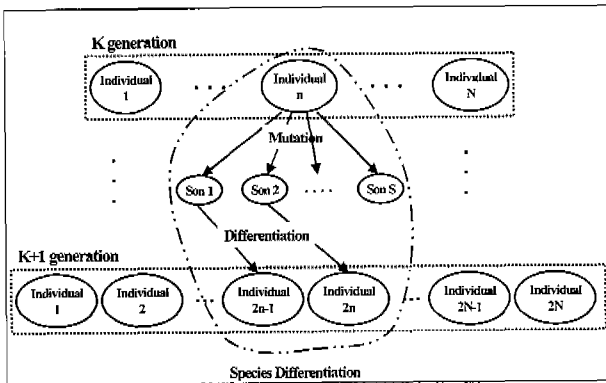


그림 2. 종 적응 진화 방법의 개념도
Fig. 2. Diagram of species adaptive evolution method

3.2 개체의 탐색과 분화

일반적인 유전자 알고리즘의 경우 개체군 전체에 있어

서 적합도 평가에 의해 각 개체가 다음 세대에 살아 남을 것인가 아닌가가 결정되기 때문에, 적합도가 높은 개체들이 급속히 증가하여 개체군 내의 다양성이 크게 감소할 가능성이 있다. 즉, 개체군의 개체들이 해 공간 전체를 탐색하지 못하고 급속히 증가한 개체가 존재하는 일부분만 탐색할 가능성이 크다. 그래서 돌연변이를 이용하여 이러한 문제점을 해결한다. 돌연변이를 통하여 적합도가 큰 특정 개체가 개체군 내에 급속히 확산되는 것을 방지하도록 하고, 해 공간 전체에 대해 두루 탐색하도록 한다. 제안된 알고리즘은 이런 이점을 이용한다. 각 개체들이 돌연변이를 통하여 해밍거리가 일정한 범위 내에 있는 국소적 자손을 생성하고, 각 개체들이 병렬 적으로 생성하도록 한다[7-8]. 즉, 각 개체들이 병렬 적으로 자손을 생성하여 일정한 해밍거리 이내의 해 공간을 탐색하게 된다. 비트열인 경우 해밍거리는 다음 식 3과 같이 나타낸다.

$$H(P_1, P_2) = \sum_{i=1}^N |p_{1i} - p_{2i}| \quad (3)$$

여기서, $H(P_1, P_2)$ 는 P_1 개체와 P_2 개체와의 해밍거리, P_1 과 P_2 는 개체군 내의 개체, P_{1i} 는 P_1 개체의 i 번째 비트, P_{2i} 는 P_2 개체의 i 번째 비트이다.

부모 개체가 돌연변이에 의해 자손을 생성하는 경우, 돌연변이 율에 의해 해밍거리의 범위가 정해진다. 돌연변이에 의한 자손의 적합도가 계속적으로 낮게 유지되면 돌연변이 율을 높임으로서 해밍거리가 먼 공간으로 탐색영역을 넓혀, 탐색 공간을 넓힌다. 전체 탐색 공간을 효율적으로 탐색하기 위해 진화 초기에는 적은 수의 개체군을 생성하고, 이 개체군의 개체들로부터 유용한 개체들로의 분화를 시도한다. 각각의 개체에 대해 돌연변이를 수행하여 해밍거리가 일정범위 이내의 자손들을 생성하고, 그 개체들 중에 해밍거리가 멀고 적합도가 높은 개체를 일정 개수로 분화시킨다. 각 개체들이 개별적으로 해밍거리 근방을 탐색하고, 개체의 분화를 통해 탐색공간을 분화함으로써 탐색공간 전체에 대해 효율적인 탐색을 수행할 수 있다. 다음 그림 3은 비트열의 길이가 10이고 돌연변이에 의한 탐색공간이 해밍거리 2내의일 때 개체탐색과 분화의 개념도를 나타낸다.

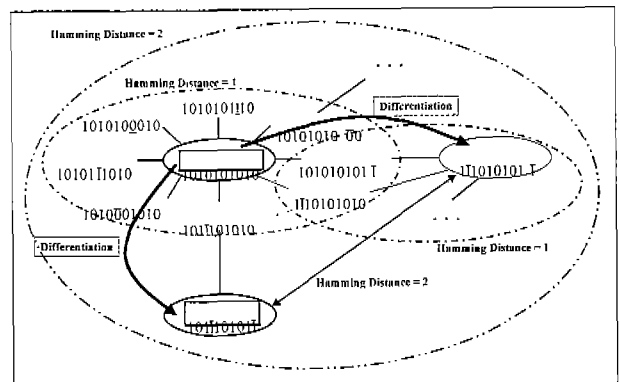


그림 3. 개체의 탐색과 분화
Fig. 3. Individual exploration and differentiation

4. 시뮬레이션과 결과 고찰

4.1 시뮬레이션

본 논문에서 제안된 종 적응 진화방법의 유용성을 확인하기 위해 아래 식 4, 5와 같이 2개의 적합도 평가 함수를 이용하여 유전자 알고리즘과 비교하였다.

$$f_1(x) = x \cdot \sin(10 \cdot \pi \cdot x) + 2.0 \quad (5)$$

$$f_2(x) = \max\{X_1, X_2\} \quad (6)$$

여기서, $X_1 = \sqrt{\frac{x_1^2 + \dots + x_N^2}{N}}$ 이고

$$X_2 = \sqrt{\frac{x_1^2 + (1-x_1)^2 + \dots + (1-x_N)^2}{N+1}}$$

이다.

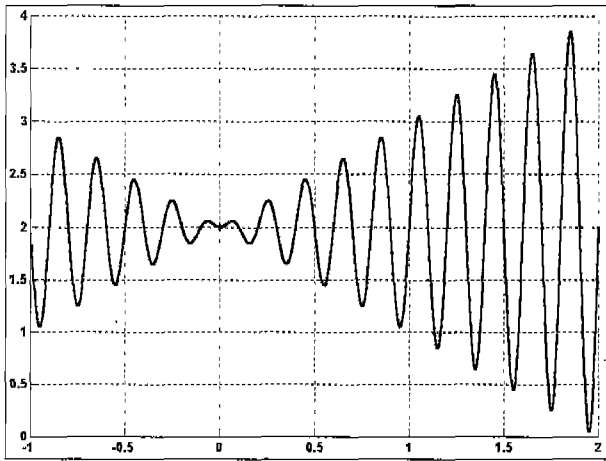


그림 4. 평가함수 f_1 의 그래프
Fig. 4. Graph of the function f_1

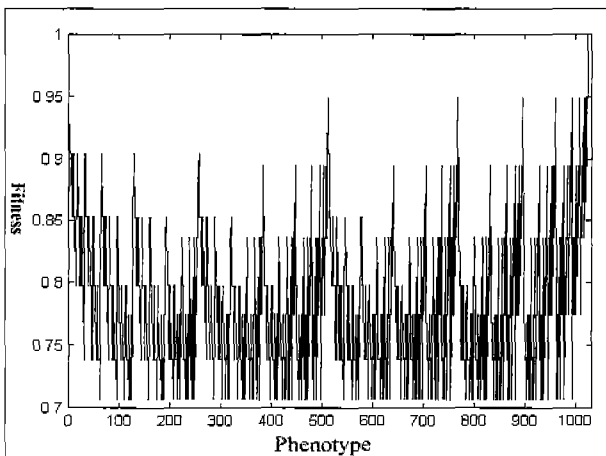


그림 5. 평가함수 f_2 의 그래프
Fig. 5. Graph of the function f_2

평가함수 f_1 은 GA-Easy 문제로 일반적으로 유전자 알고리즘에 의해 최적해를 비교적 쉽게 구할 수 있다[9]. 그림 4는 평가함수 f_1 의 그래프를 나타내고 있다. 이 문제에 있어서 유전자 알고리즘과 종 적응 진화 방법의 파라미터 값은 표 1과 같다. 그림 6은 시뮬레이션 결과를 나타내고 있다. 각 알고리즘을 각각 50회 시뮬레이션을 수행하고 가장 좋은 결과의 베스트 개체를 나타내었다. 유전자 알고리즘의 경우 50회의 시뮬레이션 중 17회는 적합도가 0.9999 인 개체를 베스트로 찾은 반면 종 적응 진화 방법은 50회 모두 적합도가 1.0인 개체를 찾았다. 각 알고리즘이 최적해에 도달하는데 걸린 세대는 10세대 이내로 비슷했지만, 종 적응 진화 알고리즘의 경우 돌연변이만을 수행하였으므로 유전자 알고리즘보다 빠른 탐색효과를 나타냈다.

평가함수 f_2 는 False-Peaks 문제로 국소해가 많아 유전자 알고리즘으로 최적해를 구하기 힘들다[10]. 그림 5는 평가함수 f_2 의 그래프를 나타내고 있다. 각 알고리즘의 파라미터는 표 1과 같이 설정하였다. 그림 7은 각 알고리즘을 50회의 시뮬레이션을 수행하여 가장 좋은 결과의 베스트 개체를 나타낸 것이다. 50회의 시뮬레이션을 통하여 종 적응 진화 방법은 적합도가 1.0인 베스트 개체를 46회 찾았지만, 유전자 알고리즘의 경우에는 25회만 베스트 개체를 찾았고, 나머지 경우 국소해에 수렴했다. 표 2에서는 두 가지 평가함수에 대해 각 알고리즘의 최적해 도달 회수와 최적해 도달까지 걸리는 평균 세대수를 나타내었다.

표 1. 각 알고리즘의 파라미터
Table 1. Parameters of each algorithm

Parameter of Genetic Algorithm	
Length of bit string	30
Population size	80
Probability of crossover	60%
Probability of mutation	1%
Generation	150 generation

(a) 유전자 알고리즘의 파라미터
(a) Parameters of genetic algorithm

Parameter of Species Adaptive Evolution Method	
Length of bit string	30
Initial population size	10
Final population size	40
Number of differentiation	2
Probability of mutation	10%
Generation	150 generation

(b) 종 진화 방법의 파라미터
(b) Parameters of species adaptive evolution method

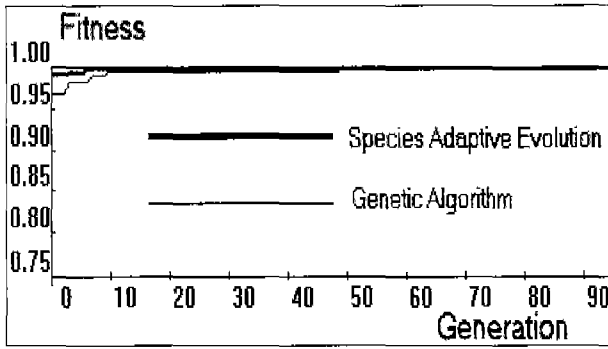


그림 6. GA-East Problem의 결과

Fig. 6. Best fitness of both genetic algorithm and species adaptive evolution in GA-Easy problem.

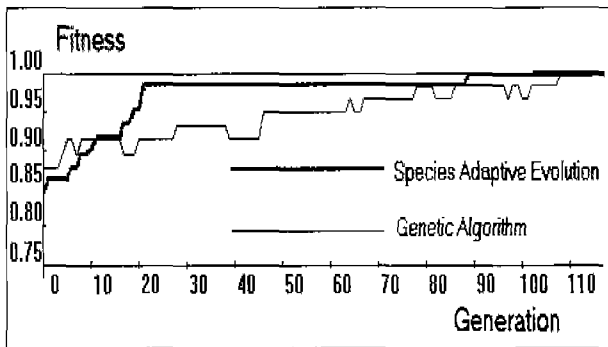


그림 7. False Peaks 문제

Fig. 7. Best fitness of both genetic algorithm and species adaptive evolution in false peaks problem.

표 2. 시뮬레이션 결과
Table 2. Result of simulation

		유전자 알고리즘	종 적응 진화 방법
GA-Easy Problem	최적해 도달 횟수	33회 /50회	50회 /50회
	최적해 도달 세대	10세대 /50회 평균	9세대 /50회 평균
False Peaks Problem	최적해 도달 횟수	25회 /50회	49회 /50회
	최적해 도달 세대	130세대 /50회 평균	110세대 /50회 평균

4.2 결과 고찰

두 가지의 적합도 평가함수를 통하여 제안된 알고리즘이 돌연변이와 탐색공간 분화를 통해 환경에 적합한 염색체를 효율적으로 생성함을 확인할 수 있다. 또한, 탐색공간의 분화를 통해 해 공간 전체를 효율적으로 탐색함으로써 국소해에 쉽게 빠지지 않고 최적해를 찾을 수 있다. 종 적응 진화 방법의 경우 돌연변이 연산자만을 사용하여, 유전자 알고리즘에 의한 진화 방법 보다 진화

시간을 단축할 수 있다. 그리고, 시뮬레이션의 마지막 세대에서 많은 종류의 종이 존재한다. 즉 각 개체들이 개별적으로 진화하여 개체군 내의 모든 개체들이 하나의 최적해 값으로 수렴하지 않고, 개체군내에 최적해 뿐만 아니라 많은 국소해도 존재하고 있음을 보여준다. 이는 다양하게 변하는 환경 하에서 제안된 알고리즘에 의한 진화 방법이 더 유용하다는 것을 보여준다.

제안된 종 적응 진화 방법을 진화 하드웨어에 적용할 경우, 매우 유용한 방법이 된다. 진화 하드웨어를 진화시킬 때 돌연변이 연산자만을 사용할 경우, 돌연변이에 의해 변화된 염색체의 일부분만을 재구성하면 된다. 이 경우 교차에 의해 재구성을 하는 방법보다는 더 빠르게 내부 구조를 재구성할 수 있으므로 한 세대를 진화하는데 걸리는 시간을 줄일 수 있다. 그리고, 유전자 알고리즘에서 개체군의 크기보다 작은 개체군으로 진화함으로써, 좀더 유용한 방법이 될 수 있다. 진화 하드웨어의 경우 개체군 내의 모든 개체들을 각각 진화하드웨어 내에 재구성해야 하므로 적은 수의 개체군을 이용하는 제안된 방법이 더 효율적이다.

5. 결론

본 논문에서는 진화 하드웨어를 구현하기 위해 돌연변이와 탐색공간 분화를 이용한 종 적응 진화 방법을 제안한다. 이 방법은 자연계에서 종 분화를 통하여 다양성을 유지하며 진화하는 방식을 기반으로 한다. 그리고, 제안된 알고리즘은 돌연변이 연산자만을 이용하여 구현하기 쉽다. 그러나, 분화를 통해 개체군의 수가 늘어나므로 진화해 나감에 따라 분화하는 개체 수를 조절하여 일정한 개체군을 유지해야한다. 제안된 알고리즘을 시뮬레이션을 통해 두 가지의 최적해 문제에 적용하여 유전자 알고리즘과 비교하였다. 시뮬레이션에서 제안된 알고리즘이 최적해의 값을 효율적으로 찾아냄을 확인했다. 그리고, 제안된 방법은 돌연변이 연산자만을 이용함으로써 진화 하드웨어 적용할 경우 진화 속도의 향상이 기대된다.

향후 과제로서 일정 세대에서 각 개체의 분화시 돌연변이율을 조절하는 전략이 필요하다. 돌연변이 전략은 각 개체가 진화하여 탐색공간을 설정하는 중요한 역할을 한다. 그리고 효율적인 분화를 위한 방법이 요구된다.

참고 문헌

- [1] A. Thompson, "An evolved Circuit, Intrinsic in Silicon, Entwined with Physics", Evolvable Systems : From Biology to Hardware, No. 1259, 1997.
- [2] John, H. Holland, "Adaptation Natural and Artificial Systems", Ann Arbor, University of Michigan Press, 1975.
- [3] Z.Michalewicz, "Genetic Algorithms + Data Structures Evolution Programs", Third Edition, Springer-Verlag, 1995.
- [4] Melanie Mitchell, "An Introduction to Genetic Algorithm", A Bradford Book, The MIT Press, 1996.
- [5] David E. Goldberg, "Genetic Algorithms in Search,

Optimization, and Machine Learning", Addison-Wesley, 1989.

[6] John, R. Koza, "Genetic evolution and Co-Evolution of Computer Programs", Artificial Life II, Addison Wesley, 1991.

[7] Tamaki, H. and Nishikawa, Y., "A paralleled Genetic Algorithm based on a Neighborhood Model and Its Application to the Job Shop Scheduling", Parallel Problem Solving from Nature 2, Elsevier Science Publishers, 1992.

[8] Gorges Schleuter, M., "ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategy", Proc. 3rd Int. Conf. Genetic Algorithms, Morgan Kaufman, 1989.

[9] Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer Verlag, 1995.

[10] De Jong, K. A. and W. M. Spears, "Using genetic algorithms to solve np-complete problems", Proc. Of the 3rd ICGA, 1989.

저 자 소 개



반 창 봉 (Chang-Bong Ban)
 2000년 중앙대학교 전자전기공학부학사
 2000년~현재 동 대학원 전자전기공학부 석사과정

관심분야 : 진화연산, 진화 하드웨어



전 호 병 (Ho-Byung Chun)
 1996년 중앙대학교 제어계측공학과학사
 1999년~현재 동 대학원 전자전기공학부 석사과정

관심분야 : 진화연산, 공진화 알고리즘



박 창 현 (Chang-Hyun Park)
 2001년 중앙대학교 전자전기공학부학사
 2001년~현재 동 대학원 전자전기공학부 석사과정

관심분야 : 진화연산, 신경회로망 등



정 구 철 (Goo-Cheol Jeong)
 1979년 중앙대학교 전자공학과 학사
 1981년 동 대학 전자공학과 석사
 1988년 동 대학 전파통신과 박사
 1991년~현재 한국기술교육대학교 정보통신공학과 부교수

관심분야 : 퍼지이론, 신경회로망, 유전알고리즘



심 귀 보 (Kwee-Bo Sim)
 1984년 중앙대학교 전자공학과 학사
 1986년 동 대학원 전자공학과 석사
 1990년 The University of Tokyo 전자공학과 박사
 1991년~현재 중앙대학교 전자전기공학부 교수

1997년~현재 한국퍼지 및 지능시스템학회 논문지 편집위원장
 1997년~현재 한국퍼지 및 지능시스템학회 편집이사
 2000년~현재 제어지능화시스템공학회 국문논문지 편집위원
 2001년~현재 대한전기학회 제어 및 시스템 부문회 학술이사

관심분야 : 인공생명, 진화연산, 지능로봇시스템, 뉴로-퍼지 및 소프트 컴퓨팅, 자율분산시스템, 로봇비전, 진화 하드웨어, 인공 면역계 등

Phone : +82-2-820-5319

Fax : +82-2-817-0553

E-mail : kbsim@cau.ac.kr

URL : <http://rics.cie.cau.ac.kr>