강화학습을 통한 유전자 알고리즘의 성능개선

Performance Improvement of Genetic Algorithms by Reinforcement Learning

이 상 환*, 전 효 병, 심 귀 보

로보틱스 및 지능정보시스템 연구실
중앙대학교 공과대학 제어계측공학과
Tel : 02) 820-5319, Fax : 02) 817-0553, E-mail : kbsim@cau.ac.kr

Sang-Hwan Lee, Hyo-Byung Jun, and Kwee-Bo Sim

Robotics and Intelligent Information System Laboratory
Dept. of Control and Instrumentation Engineering, Chung-Ang University
221, Huksuk-Dong, Dongjak-Ku, Seoul 156-756, Korea
Tel : +82-2- 820-5319, Fax : +82-2-817-0553
E-mail : kbsim@cau.ac.kr, URL:http://rics.cie.cau.ac.kr

ABSTRACT

Genetic Algorithms (GAs) are stochastic algorithms whose search methods model some natural phenomena. The procedure of GAs may be divided into two sub-procedures : Operation and Selection. Chromosomes can produce new offspring by means of operation, and the fitter chromosomes can produce more offspring than the less fit ones by means of selection. However, operation which is executed randomly and has some limits to its execution can not guarantee to produce fitter chromosomes. Thus, we propose a method which gives a directional information to the genetic operator by reinforcement learning. It can be achived by using neural networks to apply reinforcement learning to the genetic operator. We use the amount of fitness change which can be considered as reinforcement signal to calculate the error terms for the output units. Then the weights are updated using backpropagtion algorithm. The performance improvement of GAs using reinforcement learning can be measured by applying the proposed method to GA-hard problem.

Ⅰ. 서론

유전자 알고리즘(Genetic Algorithms)[1][2]은 생물의 진화과정을 인공적으로 모델링한 알고리즘으로서, 한 세대의 개체군중에서 환경에 대한 적합도가 높은 개체가 높은 확률로 살아남아 교차 및 돌연변이를 거쳐 다음 세대의 개체군을 형성해 가는 과정을 갖는다. 그러나 기존의 연산과정에서는 랜덤하게 수행되는 특성에 의해 적합도가 보다 높은 새로운 개체가 형성될 확률이 불확실해진다. 또한 적합도가 높은 개체가 생성될 수 있다 하더라도 그 가능한 범위가 정해진 연산자의 파라메타값에 의해 제한되어 있으므로 일단 국소해에 빠지면 벗어나기 어려운 단점이 있다. 따라서, 유전자 알고리즘의 연산이 무질서하게 수행되지 않고 적합도가 보다 높은 개체가 생성될 수 있도록 수행될 필요성을 갖게 된다. 이는 곧 진화과정의 경험을 바탕으로 유전자 알고리즘의 연산이 진화를 돕는 방향으로 학습되어지는 개념으로서 바로 강화학습 개념과 상통한다고 볼 수 있다.

강화학습(Reinforcement Learning)[3][4]은 비교사 학습법으로서 시행착오를 통해 환경으로부터의 보상을 최대화할 수 있는 행동 전략을 찾는 학습법이다. 본 논문에서는 이러한 강화학습을 유전자 알고리즘에 적용하여, 연산과정이 보다 높은 적합도를 갖는 개체를 생성할 수 있도록 학습되어지는 방법을 제안한다. 제안된 방법에서는 연산에 의한 개체의 변화정도에 제한을 두지 않으면서 진화를 돕는 방향으로 연산이 수행되어짐으로써 국소해에 빠질 위험을 최소화 할 수 있다.

제안된 방법의 유효성은 GA-hard problem에 적용하여 기존 유전자 알고리즘과의 비교를 통해 검증한다.

II. 유전자 알고리즘과 강화학습

2.1. 유전자 알고리즘

유전자 알고리즘[1][2]은 1975년 John Holland의 연구에 기원한 것으로 진화 알고리즘(Evolutionary Algorithms) 중 하나이다. 이는 개체군(population) 중에서 환경에 대한 적합도(fitness)가 높은 개체가 높은 확률로 살아남아 재생(reproduction)할 수 있게 되며, 이때 교차(crossover) 및 돌연변이(mutation)로서 다음 세대의 개체군을 형성하는 과정을 갖는다. 그러나 유전자 알고리즘의 연산과정은 결정론적인 규칙이 없고 확률적 연산자를 수행하기 때문에 연산과정을 통해 적합도가 보다 높은 개체가 생성되리라는 보장이 없으며, 보다 좋은 개체가 생성된다하더라도 해당 연산자의 파라메타값에 의해 그 범위가 제한되어 탐색에 있어 국소해에 빠질 위험이 존재한다.

이러한 단순 유전자 알고리즘의 단점을 단적으로 보여주는 예로써 GA-hard problems과 deceptive problems등과 같은 문제들이 연구된 바 있다. 즉, 어느 한 국소해에 빠지면 거기서 벗어나기 힘든 유전자 알고리즘의 특성에 의해 최적해와 유사한 해가 최적해와는 거리가 먼 곳에 존재하는 GA-hard problems이나, 최적해가 적합도가 낮은 해들에 의해 둘러싸여 있는 deceptive problems 등과 같은 문제에선 최적해 탐색에 혼란을 빚는 현상이 발생하게 된다. 이러한 현상을 보완하고자 새로운 선택방법인 disruptive selection[5] 및 스키마 공진화 방법[6] 등이 연구되고 있으나 본 논문에서는 강화학습을 통해 유전자 알고리즘의 성능개선을 이루고자 한다.

2.2. 강화학습에 의한 유전자 알고리즘

강화학습[3][4]은 비교사 학습법의 하나로서 환경으로부터의 보상을 최대화할 수 있는 행동전략을 찾는 학습법이다. 따라서 일반적인 강화학습은 변화하는 환경 하에서 동작하는 개체(agent)에 적용되어야 하나 유전자 알고리즘에 적용할 수 있도록 변형이 가능하다. 즉 유전자 알고리즘에서는 하나의 개체가 연산과정을 거치면서 그 개체가 갖고 있던 적합도에 변화가 발생하는데, 이는 환경으로부터의 보상으로 볼 수 있다. 즉, 강화학습의 관점에서 말하자면 현재의 개체가 연산과정을 통해 적합도의 변화를 갖게 되고 이러한 적합도 변화를 환경으로부터의 보상으로 보아 이 보상이 최대가 되도록, 즉, 적합도가 보다 좋아지도록 연산과정을 유도한다는 개념이다. 따라서, 강화학습에 있어서의 상태(states)는 연산과정 이전의 유전자형(genotype) 그 자체로, 보상(reward)은 연산과정 이전의 적합도와 이후의 적합도 사이의 차이로 각각 대응될 수 있다. 여기서, 유전자 알고리즘은 주로 교차와 돌연변이 연산자를 사용하나 본 논문에서는 돌연변이에만 강화학습을 적용하였다. 강화학습을 돌연변이 연산에 적용하기 위하여, 많은 상태에 대해서도 처리가 용이한 신경망을 도입하였다. 따라서, 유전자 알고리즘의 연산중에서 돌연변이 시에만 신경망의 출력값을 이용하여 돌연변이를 수행하며 그 결과인 적합도 변화를 이용해 신경망을 학습하고 다시 유전자 알고리즘을 반복 수행하는 과정을 거치게 된다.

III. 신경망을 이용한 강화학습 알고리즘

신경망의 입력은 각 개체의 유전자형으로, 출력은 각 유전자좌(locus)의 돌연변이 확률로 설정한다. 따라서, 입력뉴런과 출력뉴런의 개수는 스트링의 길이와 일치하게 한다. 또한 출력값의 범위가 확률값이 될 수 있도록 0과 1 사이의 실수치를 갖어야 하므로 신경망의 활성함수로는 다음의 식 (1)과 같은 시그모이드 함수를 사용한다.

$$f(x)=\frac{1}{1+e^{-x}} \tag{1}$$

또한, 입력값은 0또는 1의 이진값이 되므로 0일 때 계산에 영향을 미치지 못하는 것을 방지하기 위하여 각 입력 뉴런에 바이어스값을 일정하게 설정하여야 한다.

신경망을 학습하기 위해선 교사신호가 필요하나 강화학습에서는 직접적인 교사신호가 존재하지 않으므로 보상으로부터 유도하여야 한다. 앞에서 언급한 바와 같이 환경으로부터의 보상이란 적용된 개체의 적합도 변화에 대응된다. 즉 현재의 개체 x_t가 돌연변이를 통해 받는 보상 $r(x_t)$는 식 (2)와 같이 돌연변이 이후의 적합도 $f(x_{t+1})$와 돌연변이 이전의 적합도 $f(x_t)$사이의 차이로 정의된다.

$$r(x_t)=f(x_{t+1})-f(x_t) \tag{2}$$

또, 교사신호의 유도를 위해서, 실제로 해당 유전자좌에서 돌연변이가 일어났는가의 값을 갖는 변수가 필요하다. 이 변수 c_i는 i번째 유전자좌에서 돌연변이가 일어나면 1을, 돌연변이가 일어나지 않으면 0을 갖는다.

위에서 언급한 보상 $r(x_t)$와 변수 c_i를 이용하여 i번째 유전자좌의 원하는 돌연변이 확

률 즉, i번째 출력뉴런의 원하는 출력값 y_i^*은 다음 식 (3)과 같이 정의된다.

$$y_i^* = \begin{cases} y_i + \alpha r(x_l)(c_i - y_i) & \text{if } r(x_l) \geq 0, \\ y_i + \alpha c_i r(x_l)(y_i - \overline{c_i}) & \text{if } r(x_l) < 0 \end{cases} \quad (3)$$

여기서, y_i는 신경망의 실제 출력이고, α는 $\alpha > 0$의 범위를 갖는 상수이다. 이 값이 클수록 신경망의 학습속도가 향상되나 그만큼 국소해에 빠지기 쉽게 된다. 본 논문에서는 $\alpha = 1.0$으로 설정하였다.

i번째 출력뉴런의 원하는 출력값 y_i^*은 적합도 변화와 i번째 유전자좌의 실제 돌연변이 여부에 따라 다음과 같은 4가지 유형으로 분류가 가능하다.

i. 돌연변이를 통해 적합도가 증가하고 i번째 유전자좌가 돌연변이 된 경우 : 현재의 i번째 유전자좌의 돌연변이 확률은 더욱 증가되어야 한다.
ii. 돌연변이를 통해 적합도가 증가하고 i번째 유전자좌가 돌연변이 안된 경우 : 현재의 i번째 유전자좌의 돌연변이 확률은 더욱 감소되어야 한다.
iii. 돌연변이를 통해 적합도가 감소하고 i번째 유전자좌가 돌연변이 된 경우 : 현재의 i번째 유전자좌의 돌연변이 확률은 더욱 감소되어야 한다.
iv. 돌연변이를 통해 적합도가 감소하고 i번째 유전자좌가 돌연변이 안된 경우 : 현재의 i번째 유전자좌의 돌연변이 확률을 변화시키지 않는다.

또한 i번째 유전자좌의 돌연변이 확률을 증가하여야 할 경우 그 확률이 비교적 적은 값이었다면 보다 더 증가시켜야 하고, 반대의 경우에도 마찬가지이다. 이러한 모든 조건은 수식 (3)에 포함되어 있다. 결국 신경망의 i번째 출력뉴런의 오차 추정값은 식 (4)와 같다.

$$e_i = y_i^* - y_i \quad (4)$$

이 출력오차 추정값을 역전파 알고리즘(backpropagation algorithms)[7]에 적용하여 학습을 수행한다.

Ⅳ. GA-hard Problem에의 적용

이와 같은 강화학습에 의한 유전자 알고리즘의 유효성을 검토하기 위해 다음과 같은 GA-hard problem에 적용하였다.

4.1. 문제설정

스트링의 길이를 l이라 할 때 각 개체의 적합도가 다음 식 (5)와 같은 문제를 생각해 보자.

$$fit = \max\left\{\sqrt{\frac{x_1^2 + \cdots + x_l^2}{l}}, \sqrt{\frac{x_1^2 + (1-x_1)^2 + \cdots + (1-x_l)^2}{(l+1)}}\right\} \quad (5)$$

이러한 문제에 대해서 $l = 10$인 경우의 전체 해공간에서의 적합도값을 나타내보면 그림 1과 같다. 이 문제의 최적해는 11…1(적합도 1.0)이지만 해공간의 반대편에 있는 00…0도 비교적 높은 적합도(0.953)이기 때문에 GA의 탐색을 혼란시킨다.

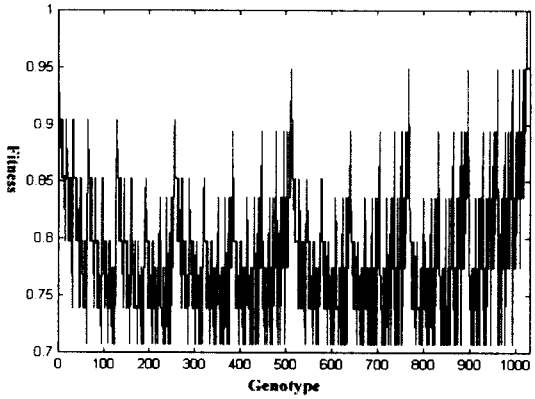

그림 1. 탐색공간
Fig. 1 Fitness Landscape

4.2. 결과 및 분석

우선 신경망의 학습 알고리즘의 유효성을 확인하기 위하여 유전자 알고리즘과는 무관하게 랜덤한 입력에 대하여 학습을 충분히 시킨 후에 전체 해공간에 대한 출력값을 확인하였다. $l = 10$인 경우의 결과를 살펴보면 1024개의 개체 중에서 822개의 개체가 적합도가 높아졌으며, 45개의 개체는 오히려 적합도가 낮아졌다. 나머지 157개는 적합도의 변화가 일어나지 않았다. 이는 랜덤한 입력에 대해 학습된 결과이므로 실제 유전자 알고리즘에 적용될 때에는 그 효과가 나타나지는 않으나 학습 알고리즘이 개체의 적합도가 높아지도록 유도하고 있음을 알 수 있다.

이 문제에 대한 단순 유전자 알고리즘과 강화 유전자 알고리즘과의 시뮬레이션 결과는 표 1과 같다. 이는 스트링의 길이 $l = 20$일 때의 결과로서, 개체군의 크기는 10으로 하였고, 선택압력을 높이기 위해 엘리트를 보존하면서 현 세대의 최대 및 최소 적합도에 따라 [0,1] 구간으로 스케일한 후 선택하는 방법을 사용하였다. 세대수는 10000세대로 정하여 그 안에 최적해를 탐색하지 못한 경우는 최적해 탐색에 실패한 것으로 간주하였다. 강화학습에 의

한 유전자 알고리즘(표 1에서 RGA)의 경우에 사용된 신경망은 입출력 노드가 20개, 중간 노드가 10개인 구조를 가지며 바이어스값으로 0.2, 학습계수 $\eta=0.1$을 사용하였다.

표 1. 10번 수행에 대한 성공회수
Table. 1 The Number of Successful Runs out of Ten Runs.

P_c	P_m	Success	Average Generation
0.00	0.001	4	268.25
	0.01	5	41.80
	0.1	9	3075.11
	0.5	0	-
	RGA	**10**	**965.50**
0.35	0.001	5	273.40
	0.01	4	32.00
	0.1	8	1438.50
	0.5	0	-
	RGA	**10**	**811.60**
0.65	0.001	6	255.67
	0.01	7	40.00
	0.1	10	1508.50
	0.5	0	-
	RGA	**10**	**808.50**
0.95	0.001	7	249.57
	0.01	6	32.83
	0.1	6	1684.83
	0.5	0	-
	RGA	**10**	**735.6**

표 1의 결과를 살펴보면 단순 유전자 알고리즘의 경우 국소해에 빠져 최적해를 찾지 못하는 경우가 많이 발생하고, 교차 및 돌연변이 확률의 변화에 민감한 특성을 보임을 알 수 있다. 특히 돌연변이 확률이 적을수록 국소해에서 벗어나기가 힘들어 최적해 탐색에 실패하기가 쉽다. 반대로 돌연변이 확률이 너무 크면 국소해에 빠지는 위험은 없지만 형질 유전성이 보장되지 않아 진화가 아무리 진행되더라도 최적해 탐색에 실패하게 된다. 즉, 돌연변이 확률이 클수록 랜덤한 탐색에 가깝게 된다. 그러나 강화 유전자 알고리즘에서는 이러한 파라미터들의 영향이 거의 없이 안정적으로 최적해를 탐색할 수 있는데, 이는 돌연변이 과정이 무질서하게 수행되는 것이 아니라 강화학습을 통해 최적해를 탐색할 수 있도록 하는 방향성을 갖게됨으로써 형질 유전성이 유지되면서도 국소해에서 쉽게 벗어날 수 있도록 하는 효과를 갖기 때문이다. 또한 교차확률의 변화에도 비교적 큰 영향없이 최적해를 탐색할 수 있었다. 세대수에 있어서는 최적해 탐색에 성공했을 때의 유전자 알고리즘보다 비교적 많은 세대를 요하는데 이는 신경망의 학습을 위해 소요된 시간으로 볼 수 있다. 다시 말하자면 다소 세대수는 소요되나 국소해에 빠지는 위험없이 안정적으로 최적해를 탐색할 수 있는 방법이라 하겠다.

V. 결론

유전자 알고리즘은 연산과 선택이라는 두 가지 단계를 반복적으로 수행하는 알고리즘이라 볼 수 있다. 그러나 연산과정은 특정한 방향이 없이 랜덤하게 수행되고 파라메타값에 많은 영향을 받게 된다. 따라서, 본 논문에서는 연산과정에서 얻는 경험을 강화신호로 삼아 이를 다시 다음 연산과정에 반영하는 강화학습을 유전자 알고리즘에 도입함으로써 국소해에 빠지기 쉬운 유전자 알고리즘의 단점을 보완하였다. 특히 연산중에서 돌연변이에 강화학습을 도입하여 현재의 개체가 어떤 유전자좌에서 돌연변이가 일어나야 좋은지를 학습해 감으로써 유전자 알고리즘의 성능향상이 이루어짐을 확인하였다.

감사의 글

본 연구는 한국과학재단 특정기초연구비(96-01-02-13-01-3) 지원으로 수행되었으며 지원에 감사를 드립니다.

VI. 참고문헌

[1] M. Mitchell, *An Introduction to Genetic Algorithms*, The MIT Press, 1997.

[2] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1995.

[3] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, 1997.

[4] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research 4*, pp. 237-285, 1996.

[5] T. Kuo and S. Y. Hwang, "A Genetic Algorithm with Disruptive Selection," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, no. 2, pp. 299-307, 1996.

[6] K. B. Sim and H. B. Jun, "Co-Evolutionary Algorithm and Extended Schema Theorem," *J.KSIAM*, vol. 2, no. 1, 1998.

[7] J. A. Freeman and D. M. Skapura, *Neural Networks Algorithms, Applications, and Programming Techniques*, Addison-Wesley Publishing Company, 1991.