Performance Improvement of Evolution Strategies using Reinforcement Learning

Kwee-Bo Sim and Ho-Byung Chun

School of Electrical and Electronic Engineering, Chung-Ang University
221, Huksuk-Dong, Dongjak-Ku, Seoul 156-756, Korea

Abstract

In this paper, we propose a new type of evolution strategies combined with reinforcement learning. We use the variances of fitness occurred by mutation to make the reinforcement signals which estimate and control the step length of mutation. With this proposed method, the convergence rate is improved. Also, we use cauchy distributed mutation to increase global convergence faculty. Cauchy distributed mutation is more likely to escape from a local minimum or move away from a plateau. After an outline of the history of evolution strategies, it is explained how evolution strategies can be combined with the reinforcement learning, named reinforcement evolution strategies. The performance of proposed method will be estimated by comparison with conventional evolution strategies on several test problems.

Key Words : Evolution strategy, Reinforcement learning, Cauchy distributed mutation, Reinforcement Evolution Strategies

Ⅰ. Introduction

Evolutionary Algorithms(EAs)[1,2] are the computational models intending to solve complex problems. EAs are based on the collective search process within a population of individuals, each of which represents a search point in the search space. The population is arbitrarily initialized, evolves toward fitter point in the search space by means of operation and selection as the generation goes by. Individuals can generate a new offspring in operation process, and fitter ones can be selected to the next population in selection process. These procedures are iterated until the termination criterion fulfilled.

Four main fields of evolutionary algorithms are genetic algorithms(GAs), evolution strategies(ESs), evolutionary programming(EP), and genetic programming(GP). In particular, ESs are suitable to solve function optimization problems, bcause search points in ESs are represented in n-dimensional real vectors. First efforts toward ESs took place in 1964 to apply hydrodynamic problems in Germany. In 1965, ESs were simulated by Schwefel. Rechenberg developed a convergence theory for (1+1)-ES, and proposed a 1/5-success rule which change the standard deviation of mutations exogenously, in 1973. After a first multimembered ES, $(\mu+1)$-ES, proposed by him, the transition to $(\mu+\lambda)$-ES and (μ,λ)-ES were facilitated by Schwefel, in 1977 and in 1981, respectively[1]. These ESs use the mutation operator as the main operation and the recombination operator as sub operation. One individual of ESs consists of an object variable, a standard deviation, and a rotation angle. These vectors are mutated self-adaptively according to the each updating rule containing modified probability density function. In the mutation process, standard deviations and rotation angles determine the amount of mutation for the objective variables. Although those two vectors, which can be said as step length, are mutated self-adaptively, there are still some problems remained. If these values become too large by their mutation, the objective variables will be mutated too much. Or if they become too low values, the individuals can not make any different offspring from himself. Therefore, in ESs, it is very important to mutate properly for these two vectors, and if we can control these vectors appropriately, the performance of ESs will be improved as it can be expected. But the mutation process of ESs goes on stochastically. It means that an individual of ESs makes its offspring randomly and there is not any directional information in mutation process. Though this makes it difficult to mutate step length appropriately, there have been many efforts to improve the performance of ESs. Schwefel and Rudolph proposed a (μ,k,λ,ρ)-ES which contains the concept of life span[3]. Yao and Liu proposed fast ESs which mutate an individual according to the cauchy distribution not to the gaussian distribution for the purpose of improving global convergence faculty[4].

Now, we propose a new ESs which give some directional information to the mutation process by reinforcement learning mechanism. We named this method as reinforcement evolution strategies. Rein-

Manuscript received March 15, 2001; revised May 1, 2001.
This work was supported by a grant NO. 2000-2-30300-003-3 form the Basic Research Program of the Korean Science & Engineering Foundation (KOSEF). And this paper was supported in part by the Braintech 21 from Ministry of Science and Technology of Korea.

forcement learning[5,6] is a kind of unsupervised learning. The goal of reinforcement learning is that an agent learns the action or the strategies maximizing the reward through trial-and-error interactions with an environment. We can easily match the reinforcement learning to ESs. The reward given to an individual in ESs can be defined as variance of fitness occurred by mutation. It can be used as reinforcement signal to set the step length appropriately according to reinforcement learning. As a result of this, mutation of ESs can be executed to make the variance of fitness larger. Although using reinforcement learning have an effect of improving convergence rate, it can not guarantee improvement of global convergence faculty. For the purpose of acquisition of this, we substitute cauchy distribution for gaussian distribution. Because the mutation according to cauchy distribution can search more widely, population can easily escape from local minimum. Simulation results comparing reinforcement evolution strategies with conventional evolution strategies for several function optimization problems reveal the performance and efficiency of the proposed method.

This paper is organized as follows : Section 2 is a description of conventional evolution strategies. Section 3 introduces reinforcement evolution strategies. Section 4 illustrates simulation results. Finally, conclusions are included in Section 5.

II. Evolution Strategies

2.1 Conceptual Algorithms

Conventional evolution strategies can be divided into $(\mu+\lambda)$-ES and (μ,λ)-ES according to its selection method. ESs recombine and mutate μ parents to generate λ offsprings. Then, $(\mu+\lambda)$-ES select μ individuals to form next population from μ parents and λ offsprings, and (μ,λ)-ES select μ individuals from only λ offsprings. The conceptual algorithm of $(\mu+\lambda)$-ES and (μ,λ)-ES is illustrated in Fig. 1[1,7].

Algorithm **:**

```
$t = 0$;
initialize :  $P(0) = \{a_1(0), \ldots, a_\mu(0)\}$
  where $a_i(0) = (x_i(0), \sigma_i(0), \alpha_i(0))$,
        $x_i(0) \in \mathbb{R}^n, \sigma_i(0) \in \mathbb{R}^n$,
        $\alpha_i(0) \in [0, 2\pi)^{n(n-1)/2}, \forall i \in \{1, \ldots, \lambda\}$;
evaluate :  $P(0)$;
  while termination criterion not fulfilled do
    recombine :  $a_i'(t) = r(P(t)) \ \forall i \in \{1, \ldots, \lambda\}$;
    mutate :  $a_i''(t) = m(a_i'(t))$;
    evaluate :  $P'(t) = \{a_1''(t), \ldots, a_\lambda''(t)\}$;
    select :  $P(t+1) = s(P'(t))$  if  $(\mu, \lambda) -$ ES;
              $P(t+1) = s(P'(t) \cup P(t))$  if  $(\mu+\lambda) -$ ES;
    $t = t+1$;
  end
```

Fig. 1. Conceptual algorithm of evolution strategies

2.2 Mutation

A search point in search space can be represented as a n-dimensional object variable $\vec{x}$, and it include up to n -dimensional standard deviation $\vec{\sigma}$ and rotation angle $\vec{\alpha}$. Each vectors are mutated as follows.

$$\begin{aligned} \sigma_i' &= \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ \alpha_j' &= \alpha_j + \beta \cdot N_j(0,1) \\ x_i' &= x_i + z_i(\vec{0}, \vec{\sigma}', \vec{\alpha}') \end{aligned} \tag{1}$$

where the global factor $\tau' \cdot N(0,1)$ has the same value over all $i \in \{1, \ldots, n\}$, whereas the individual factor $\tau \cdot N_i(0,1)$ is newly generated and applied for every i-th elements. $z(\vec{0}, \vec{\sigma}, \vec{\alpha})$ denotes a realization of a random vector distributed according to the following generalized n-dimensional gaussian distribution having expectation $\vec{0}$, standard deviations $\vec{\sigma}$, and rotation angle $\vec{\alpha}$.

$$p(z) = \sqrt{\frac{\det A}{(2\pi)^n}} \exp\left(-\frac{1}{2} z^T A z\right) \tag{2}$$

where $A^{-1} = c_{ij}$ represents the covariance matrix. $\vec{\sigma}$ is used as n diagonal components of A^{-1}, that is $c_{ii} = \sigma_i^2$, and $\vec{\alpha}$ acts as the other components of A^{-1}. Schwefel suggests to set each parameters in equation (1) as follows.

$$\begin{aligned} \tau &\propto \left(\sqrt{2\sqrt{n}}\right)^{-1} \\ \tau' &\propto \left(\sqrt{2n}\right)^{-1} \\ \beta &\approx 0.0873 \ (5°\ ) \end{aligned} \tag{3}$$

The resulting evolution and adaptation of strategy parameters according to the topological requirements has been termed self-adaptation by Schwefel.

2.3 Recombination and Selection

Recombination of ESs is similar to crossover of GAs. In ESs there are some different recombination mechanisms as follows.

$$x_i' = \begin{cases} x_{S,i} : \\ \qquad \textit{without recombination} \\ x_{S,i} \text{ or } x_{T,i} : \\ \qquad \textit{discrete recombination} \\ x_{S,i} + u \cdot (x_{T,i} - x_{S,i}) : \\ \qquad \textit{intermediate recombination} \\ x_{S_i,i} \text{ or } x_{T_i,i} : \\ \qquad \textit{global, discrete recombination} \\ x_{S_i,i} + u_i \cdot (x_{T_i,i} - x_{S_i,i}) : \\ \qquad \textit{global, intermediate recombination} \end{cases} \tag{4}$$

where indices S and T denote two parent individuals selected at random from population $P(t)$. u_i is uniformly distributed random variables in the range of $[0 \sim 1]$. Discrete recombination is the method selecting randomly one individual from two pre-selected parents and making it as an offspring. Intermediate recombination let an offspring have his object value as intermediate value of two parents. Global recombination select newly two parents for every i-th element. Because variance of object value in recombination is larger than that in mutation, recombination generally have an effect of improving global convergence faculty.

Selection in ESs is completely deterministic, selecting the μ best individuals from λ or $\mu+\lambda$ individuals. Although the $(\mu+\lambda)$ selection is elitist and therefore guarantees a monotonically improving performance, this selection strategy is unable to deal changing environments and jeopardizes the self-adaptation mechanism. Therefore, the (μ,λ) selection is recommended today.

III. Reinforcement Evolution Strategies

3.1 Reinforcement Learning and Evolution Strategies

With reinforcement learning, an agent in dynamic environment can learn the optimal behavior or strategy through trial-and-error interactions with an environment. Reinforcement learning is based on the animals' learning skills and has the concept matching our common-sense ideas, that is, if an action is followed by a satisfactory state of affairs, or an improvement in the state of affairs, then the tendency to produce that action is strengthened or reinforced, otherwise, that tendency is weakened or inhibited. Therefore, an agent learns to perform an appropriate action by receiving evaluative feedback or reward from environment. But, in many cases, the reward is not given to the agent immediately. In that case, the states in a sequence should be evaluated and adjusted according to the final outcome. The problem of evaluate each state individually in such a sequence called the temporal credit assignment problem. There are many methods to deal with this problems such as temporal difference learning[5,6] and Q-learning[5,6].

Reinforcement learning can be matched well with ESs. An agent in reinforcement learning corresponds with an individual in ESs. Similarly, we can find the correspondence between reinforcement learning and ESs as shown in Table 1.

Table 1. Correspondence between reinforcement learning(RL) and evolution strategies(ES)

RL	ES
Agent	Individual
Action	Operation and Selection
Reward	Variance of fitness in an individual
State	object variables

At this point of view, individuals(agent) in ESs change its object variables(state) by means of operation and selection(action). Then the variance of fitness(reward) occurred by operation is used to alter its step length. From this conception, we can realize reinforcement evolution strategies in which individuals learns to maximize the variance of its fitness as time goes by. A detailed feature of reinforcement evolution strategies will be expressed in next section.

3.2. Reinforcement Evolution Strategies

Reinforcement evolution strategies (RES) is a new ESs using reinforcement learning(Fig. 2). In this paper, we defined conventional evolution strategies (CES) as the ESs having no recombination and rotation angles.

Mutation in RES is executed interactively with environments not stochastically in CES. In RES, the rewards calculated by the variance of fitness give the directional information into the mutation process. This can be realized by using similarity shown in section 3.2.

Although there are some correspondence between reinforcement learning and evolution strategies, there are some differences between them. First, in reinforcement learning, there is only one agent learning strategy, but in ESs there is a population consists of many individuals. Also, when we consider variance of fitness as reward, individuals who received many rewards may be more easily alive in selection process and who received many penalties may disappear. Therefore an experience accumulated according to trial-and-error will be included only in the individuals received rewards. And the reward in ESs is acquired immediately, because it can be easily computed within one generation. This difference makes it necessary that we should modify conventional reinforcement learning to apply ESs. Following methods will be used to accomplish that.

Temporal reward $r(t)$ which is given to an individual at every generation is defined as follows

$$r(t)=\begin{cases} +0.5 & \text{if } \Delta f(t) > 0 \\ 0.0 & \text{if } \Delta f(t) = 0 \\ -1.0 & \text{if } \Delta f(t) < 0 \end{cases} \tag{5}$$

where $\Delta f(t)$ is the difference of the fitness of an individual between before and after mutations.

$$\Delta f(t) = f(t) - f(t-1) \tag{6}$$

Now, we define $r_{sum}(t)$ as the summation of temporal rewards for n generations.

$$r_{sum}(t) = \frac{1}{n}\sum_{i=0}^{n-1} r(t-i) \tag{7}$$

This value will be +0.5 if the fitness of mutated individual is better, or -1.0 if it is worse than before for all n generations. Unbalance between reward and penalty reflects the present condition that most individuals consist in population receive the reward rather than the penalty.

In RES, the reward of an individual is used to control the step length of mutation. On the contrary CES evolves and adapts the step length according to self-adaptation mechanisms. There is a directional information for the step length to optimize individual's reward in RES. As the reward becomes larger, the step length becomes larger too, and vice-versa. This can be represented as follows.

$$\sigma_i' = \sigma_i \cdot \exp(r_{sum}(t) \cdot |\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)|) \quad (8)$$

In CES, the standard deviation values become more smaller as generation goes by. However generally they are too small, and make even unfitted individuals not to change any more. This is the reason for worse convergence rate and global search faculty in CES. But in RES as the reward becomes larger, the step length becomes larger too. In most case, the composition of population is the individuals who received more rewards. If the step length becomes larger, then that individual will be dismissed. It means that excessive increase of the step length will be inhibited, therefore, appropriate step length is maintained in RES.

Although this has an effect of improving convergence rate, it can not guarantee improvement of global convergence faculty, because these two purposes are conflicting with each other. For the purpose of acquisition of global convergence faculty, we substitute cauchy distribution for gaussian distribution as follows.

$$x_i' = x_i + \sigma_i' \delta_i \quad (9)$$

where δ_i is cauchy distributed random variable having scale parameter $t=1$. The one-dimensional cauchy density function centered at the origin is defined by :

$$f_t(x) = \frac{1}{\pi}\frac{t}{t^2+x^2}, \quad -\infty < x < \infty \quad (10)$$

Because cauchy distribution is more likely to generate a random number far away from the origin, the probability of escaping from local minimum becomes larger. After all, RES has an effect to improve convergence rate and global convergence faculty by reinforcement learning and cauchy distribution, respectively. This effect is illustrated more deeply by the following simulation results.

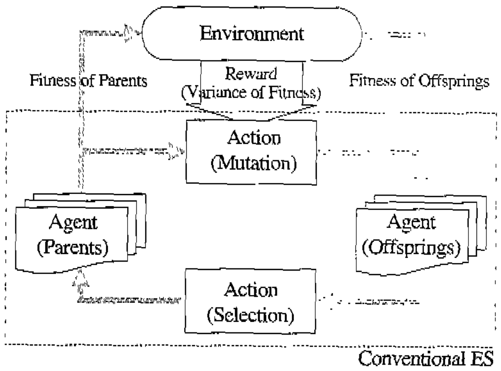

Fig. 2. Reinforcement Evolution Strategies.

IV. Simulation Results

4.1 Unimodal Function

We applied the following unimodal function having no local minima to CES and RES.

$$f(x) = \sum_{i=1}^{n} x_i^2, \quad n=30, \ -100 < x_i < 100, \ f_{\min}=0 \quad (11)$$

The results are shown in Fig. 3 and Fig. 4. They are averaged results for ten trials. (μ, λ) selection is used where μ = 30 and λ = 200. Initial standard deviation set to 3.0. The summation of temporal reward is calculated for last 5 generations as the following equation (12).

$$r_{sum}(t) = \frac{1}{5}\sum_{i=0}^{4} r(t-i) \quad (12)$$

As we can see in Fig 3, convergence rate of RES is better than that of CES. RES uses both reinforcement learning and cauchy distribution. Therefore what brings this result should be analyzed. From Fig 4, by applying only one of these two methods, we can say that using reinforcement learning in RES improve the convergence rate. Reinforcement learning has the effect of accelerating current direction of mutation as it goes better. Therefore, in the search space having no local minimum, mutation having directional information to the fitter search point is more effective than stochastical mutation in CES.

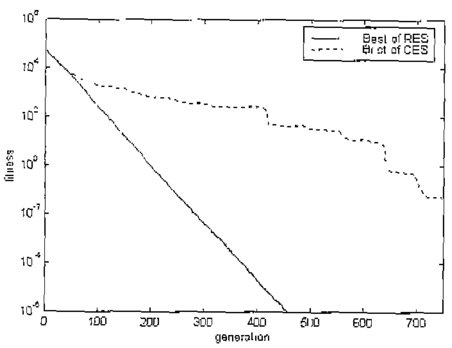

Fig. 3. Best fitness of CES and RES.

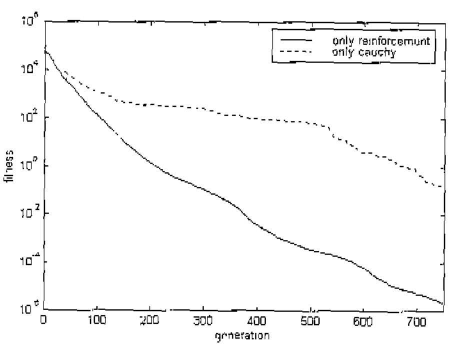

Fig. 4. Best fitness when only one method is used.

4.2 Multimodal Function

The following multimodal function having many local minima is applied to RES and CES.

$$f(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10], \quad n=30, \ -5.12 < x_i < 5.12, \ f_{\min}=0 \quad (13)$$

The results are shown in Fig. 5 and Fig. 6.

Experimental condition is the same with the previous one. In Fig. 4, we can easily see that best fitness of RES is better than that of CES. At the beginning, there is no significant difference in the convergence rate for both ESs. But in CES, once falling into local minima, it is no more better than before contrary to RES. From Fig. 5 we can say using cauchy distribution makes it better searching global minimum. The variance of the cauchy distribution is infinite, and it helps to escape from local minimum. Therefore, in the search space having many local minima, RES using cauchy distribution is more effective than CES using gaussian distribution.

V. Conclusions

In this paper, the reinforcement evolution strategies are introduced after reviewing conventional evolution strategies and reinforcement learning. Reinforcement evolution strategies give some directional information into the mutation process by using the reward from environment, that is the variance of fitness. The step length controled by the reward maintains appropriately and this improves convergence rate. In mutation of object variables, using cauchy distribution makes it possible to find global minimum which is difficult to be found in CES. The effectiveness of the proposed RES is verified by being applied to unimodal and multimodal functions.

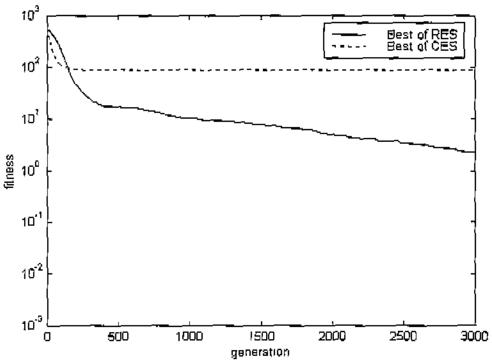

Fig. 4. Best fitness of CES and RES.

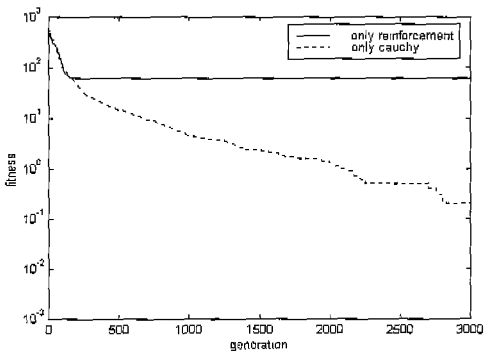

Fig. 5. Best fitness when only one method is used.

References

[1] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *E-volutionary Computation*, vol. 1, no. 1, pp. 1-23, 1993.

[2] D. B. Fogel, Evolutionary Computation, the *IEEE Press*, 1995.

[3] H.-P. Schwefel and G. Rudolph, "Contemporary evolution strategies," *Proc. of European Conference on Alife*, 1995.

[4] X. Yao and Y. Liu, "Fast evolution strategies," *Evolutionary Programming VI: Proc. of the Sixth Annual Conference on Evolutionary Programming*, vol. 1213 of Lecture Notes in Computer Science, pp. 151-161, Springer-Verlag, 1997.

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

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

[7] H.-P. Schwefel, Evolution and Optimum Seeking, A Wiley-Interscience Publication, 1995.

[8] Richard S. Sutton and Andrew G. Barto, Reinforcement Learning, MIT Press, 1998.

[9] Grehard Weiß Sandip Sen (EDs.), Adaption and Learning in Multi-Agent Systems, *Proc. of IJCAI '95 Workshop Montreal*, Canada, Auguest 21, 1995.

Kwee-Bo Sim

Kwee-Bo Sim was born September 20. 1956. He received the B.S. and M.S. degrees in Department of Electronic Engineering from Chung-Ang University, Seoul Korea, in 1984 and 1986 respectively, and Ph. D. degree in Department of Electronic Engineering from the University of Tokyo, japan, in 1990. Since 1991, he has been a faculty member of the School of Electrical and Electronic Engineering at the Chung-Ang University, where he is currently a Professor. His research interests are Artificial Life, Neuro-Fuzzy and Soft Computing, Evolutionary Computation, Learning and Adaptation Algorithm, Autonomous Decentralized System, Intelligent Control and Robot System, and Artificial Immune System etc. He is a member of IEEE, SICE, RSJ, KITE, KIEE, ICASE, and KFIS.

Phone : +82-2-820-5319
Fax : +82-2-817-0553
E-mail : kbsim@cau.ac.kr

Ho-Byung Chun

Ho-Byung Chun was born May 17, 1972. He received the B.S. degree in Department of Control and Instrumentation Engineering from Chung-Ang University, Seoul, Korea, in 1996. He is currently pursuing the M.S. degree at Chung-Ang University. His research interests are Evolutionary Computation, Artificial Life, Artificial Immune System, and Cryptography. He is a member of KITE, KIEE, ICASE, and KFIS.

Phone : +82-2-820-5319
Fax : +82-2-817-0553
E-mail : courtil@jupiter.cie.cau.ac.kr