Proc. of The Fourth Int. Symp. on Artificial Life and Robotics (AROB 4th '99)
Oita, Japan, 19-22, January, 1999.

Evolving in Dynamic Environments Through Adaptive Chaotic Mutation

D.P.Thrishantha Nanayakkara†
Keigo Watanabe†
Kyotake Izumi‡
† Dpt. of Graduate School of Science and Engineering, Saga University, Saga 840-8502
‡Dpt. of Mechanical Engineering, Saga University, Saga 840-8502.

Abstract

A new approach to incorporate the features and robustness of the natural evolutionary process to evolutionary computation is proposed. The key feature of this approach is maintaining the evolvability of the population by controlling the mutation process by exploiting the characteristic dynamics of a chaotic neuron. The proposed method has been tested for optimizing in static environments as well as in a changing environment. The results show that it inherits the ability to cope with the change in the optimizing criteria and converges fast while maintaining evolvability.

Keywords:Chaotic Neuron, Evolutionary Programming, Dynamic Environments.

1 Introduction

Evolutionary computation has been given much attention in the recent history of engineering optimization problems. Though evolutionary computation is effective in finding a globally optimum solution to a static optimization criteria, work is yet to be done to enhance it's ability to cope with a dynamic environment, Bäck [1]. In fact, actual engineering problems, are sometimes coupled with phenomena such as emergence and disappearance of constraints, changing optimization criteria, *etc.* Therefore evolutionary computation needs to take a turn towards incorporating the feature of evolvability in dynamic environments.

This paper proposes a new concept that takes a step towards natural evolution. It is important to be noted that the ability of a population to continue surviving in a changing environment comes from the ability to find the global optimum in the current environment, while maintaining enough diversity to face a potential change in the environment. In the proposed method, this phenomenon is simulated by making use of the characteristics of a chaotic neuron proposed by Aihara *et al.* [2].

The new approach of chaotic mutation gives promising results for some benchmark functions. In the sections to come, the meta-evolutionary programming algorithm is taken as the conventional method of evolutionary computation. All comparisons are being made with this algorithm.

2 Meta-EP Algorithm

2.1 The Representation

In evolutionary programming (EP), the individuals are defined based on real valued vectors in the form

$$\boldsymbol{a}_j = (\boldsymbol{x}_j, \boldsymbol{\sigma}_j), \quad j = (1...\mu)$$

to deal with continuous parameter optimization problems. Here $\boldsymbol{x}$ is the object variable vector and the strategy parameters $\boldsymbol{\sigma}_j$ are used for the mutation of individuals as explained later.

2.2 Recombination

Recombination is not found in meta-EP algorithm. In the proposed method arithmetical crossover is adopted with a relatively small probability (0.2). It is basically performed as

$$x'_{S,i} = x_{S,i} + \chi(x_{S,i} - x_{L,i})$$

$$x'_{L,i} = x_{L,i} + \chi(x_{L,i} - x_{S,i})$$

where S and L denote two parent individuals selected at random from the parent population, and $\chi \in [0, 1]$ is a uniform random variable.

2.3 Mechanism of Mutation

Mutation is performed to each individual in the form,

$$x_i^{'} = x_i + \sigma_i N_i(0,1) \quad (1)$$

where x_i is the object variable of a randomly selected individual. σ_i is the standard deviation or so called strategy parameter, and $N_i(0,1)$ is a Gausian random value generator with a normal distribution of zero mean and unity variance.

In the case of meta-EP, the standard deviation is controlled as

$$x'_i = x_i + \sqrt{\nu_i} N_i(0,1). \quad (2)$$

The variance ν_i is a constant and modified as follows.

$$\nu_i^{'} = \nu_i + \sqrt{\nu_i \alpha} N_i(0,1) \quad (3)$$

where the value of α is an exogenous parameter that makes sure that the variance remains positive. In the case where the variance becomes negative or zero, it is set to a small value $\epsilon > 0$. As can be seen in equations (1) , (2) and (3), the drawback in evolutionary programming is that the standard deviation that is gradually being changed is blind to the diversity of the population. As such the method is good in dealing with static environments but lacks the robustness to work in dynamic environments.

2.4 Selection Mechanism

In meta-EP, after creating μ offspring from μ parent individuals by mutating each parent once, tournament selection mechanism is adopted to produce the next generation. In the tournament selection method for each individual $\boldsymbol{a}_j \in P(t) \cup P^{'}(t)$, where $P(t)$ is the parent population and $P^{'}(t)$ is the population of mutated individuals, a random uniform sample of size T is picked up from the population such that $T > 1$. Then a score $w_j \in (0, T)$ is attributed to each $\boldsymbol{a}_j$ such that the score is equal to the number of individuals in T that is less fitter than the individual. Based on this score, the 2μ individuals are ranked and the best μ individuals are selected for the next generation.

3 Adaptive Chaotic Mutation Mechanism

The standard deviation in the proposed mutation mechanism is controlled depending on the instantaneous diversity of the population by exploiting the characteristics of a chaotic neuron proposed by Aihara *et al.* [2], and adopted appropriately for the purpose. An extended version of this chaotic neuron is adopted by Choi *et al.* [3], to avoid local minimal problem in mobile robots. The fundamental idea of avoiding local minimums is almost the same in this method.

Before going into a detailed discussion let us briefly look at the characteristics of the chaotic neuron model used here.

3.1 Characteristics of a Chaotic Neuron

The following equations are used to elaborate the characteristics of the Aihara's chaotic neuron.

$$p(n+1) = f(q(n+1)) \quad (4)$$

$$q(n+1) = kq(n) - \beta g(p(n)) + u(n) \quad (5)$$

$$f(q) = \frac{\phi}{1 + e^{-q/\psi}} - \frac{\phi}{2} \quad (6)$$

where $p(n+1)$ is the output value of the neuron at discrete time $n+1$, which lies between $\phi/2$ and $-\phi/2$, ψ controls the shape of the function $f(q)$ while ϕ determines the magnitude of it. In addition β is a positive parameter, k is the damping factor of refractoriness. $u(n)$ is the input to the chaotic neuron.

The refractory function $g()$ is given by

$$g(p(n)) = \begin{cases} \frac{1}{2\pi\rho^2} e^{-||p(n)||^2/2\rho^2} & \text{if } p(n) - p(n-1) \geq 0 \\ -\frac{1}{2\pi\rho^2} e^{-||p(n)||^2/2\rho^2} & \text{Otherwise.} \end{cases} \quad (7)$$

The value of ρ controls the shape of $g(p(n))$. It can be seen in Fig. 1 that the output undergoes chaotic motions when the input $u(n)$ draws closer to zero. The reason lies in the shape of the refractory function which is sensitive to the sign of the gradient of output.

3.2 Application to Adaptively Mutate Individuals

The relative diversity is defined as

$$\zeta_n = \frac{D(n)}{D(0)} \quad (8)$$

where $D(n)$ is the population diversity [4] of the n^{th} generation, which is given by

$$D(n) = \sqrt{\frac{\sum_{j=1}^{N} \{fit_m(n) - fit_j(n)\}^2}{N}} \quad (9)$$

In which $fit_m(n)$ is the mean fitness of the n^{th} generation, $fit_j(n)$ is the fitness of the j^{th} individual in the

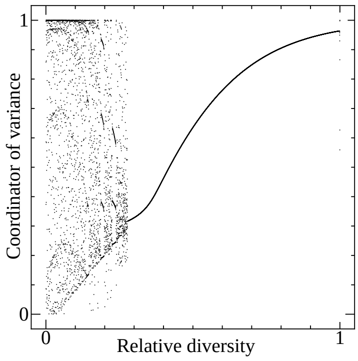

Figure 1: Behavior of the output of the chaotic neuron with the input $u(n)$ used as the bifurcation parameter. The parameter values adopted in this application are, k=0.3, β=0.5, ϕ=2.0, ψ=0.5, ρ=0.2.

n^{th} generation and N denotes the number of individuals. Then

$$a(n) \equiv \zeta_n$$

The relative diversity decreases gradually as the population converges to a global optimum and it gradually decreases the standard deviation due to the characteristics of the chaotic neuron. When the relative diversity is near zero, the standard deviation undergoes chaotic motions regaining diversity and enhancing further evolvability. Therefore the modified version of mutation mechanism proposed in this approach is

$$x_i^{'} = x_i + \sigma_i(\zeta_n) N_i(0,1) \quad (10)$$

where the chaotic standard deviation $\sigma_i(\zeta_n)$ is defined by

$$\sigma_i(\zeta_n) = \theta_i\{(1 + U([0,1)))/2\}\vartheta(\zeta_n) \quad (11)$$

$$\theta_i = cB_i \quad (12)$$

in which the function of chaotic neuron is given by ϑ, B_i is the range of initialization of the object variable x_i, c is set to be 0.125. The value of c can be altered depending on the degree of dynamics in the environment. $U([0,1))$ is the uniform Gausian random number generator between [0,1).

4 Optimization of a Moving Function

For the testing of the proposed method in its ability to optimize in a dynamic environment, the following moving

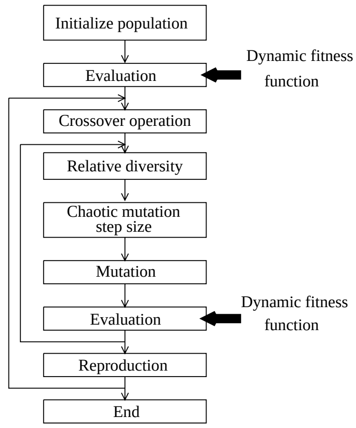

Figure 2: The block diagram showing the proposed adaptive chaotic mutation method

function was considered.

$$H(n,x,y) = \begin{cases} 1 + xe^{(-x^2-y^2)} & \text{if } 0 < n \le 100 \\ 1 + (x+0.2)e^{(-(x+0.2)^2-y^2)} & \text{if } 100 < n \le 200 \\ 1 + (x+0.3)e^{(-(x+0.3)^2-y^2)} & \text{if } 200 < n \le 300 \\ 1 + (x+0.4)e^{(-(x+0.4)^2-y^2)} & \text{if } 300 < n \le 400 \\ 1 + (x+0.5)e^{(-(x+0.5)^2-y^2)} & \text{if } 400 < n \le 450 \end{cases} \quad (13)$$

where n is the number of generations. In this dynamic environment, the test results of the proposed method for tracking the optimum point are shown. A population size of 60, crossover probability of 0.2, tournament size of 10 were used in the proposed method. In the conventional method, same parameters except crossover were used. The population was initialized in $(-2, 2)$. It can be seen

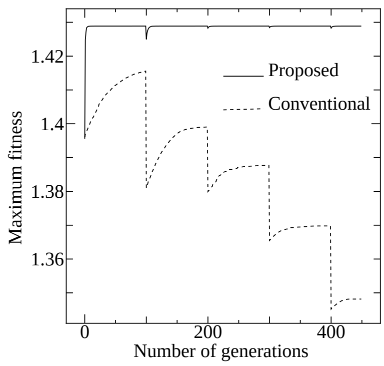

Figure 3: The behavior of the maximum fitness in the proposed method and evolutionary programming method, averaged over 100 trials.

that in the conventional method the ability to track the optimum solution deteriorates when the function moves, compared with the proposed method.

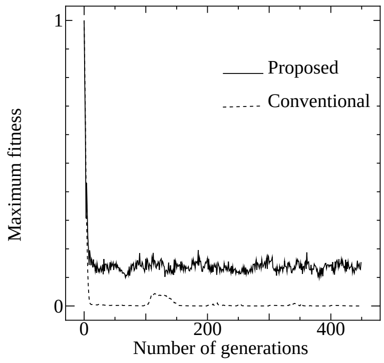

Figure 4: The behavior of the relative population diversity in the proposed method and evolutionary programming method, averaged over 100 trials.

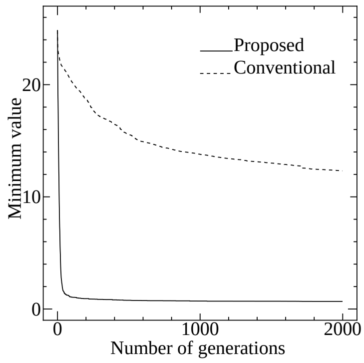

Figure 5: Comparison of the behavior of the best fit value averaged over 100 trials for the quadratic function $f_1(x)$ between the proposed and conventional methods.

5 Optimizing static environments

Two bench mark functions have been used to demonstrate the convergence ability and the reliability of finding the global optimum in a static environment using the proposed method. A population size of 50, crossover probability of 0.2, tournament size of 10 was used in the proposed method. In the conventional method, same parameters except crossover was used. For the function f_1 the population was initialized in $(-30, 30)$ and for f_2 in the range $(-1.28, 1.28)$

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{30} x_i^2 \tag{14}$$

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{30} x_i^4 + U([0,1)) \tag{15}$$

As can be seen in Figs. 5 and 6, in both cases, the proposed method has shown superior performance than the conventional meta-EP method.

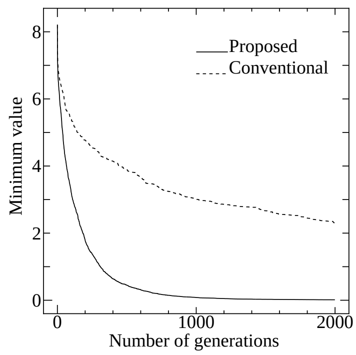

Figure 6: Comparison of the behavior of the best fit value averaged over 100 trials for the noisy quadratic function $f_2(x)$ between the proposed and conventional methods.

6 Summary

A new concept of adaptive mutation with the objective of optimization in dynamic environments has been proposed. The new approach differs from the other methods in considering the diversity of the population as the control input to a chaotic neuron to control the standard deviation of mutation. The process works in observing the diversity and then uses the relative diversity as the input to the chaotic neuron and decides on a suitable standard deviation for mutation.

References

[1] T. Bäck and H. P. Schwefel, "An Overview of Evolutionary Algorithms for Parameter Optimization," *Evolutionary Computation*, vol. 1, No. 1, pp. 1-23,1993.

[2] K. Aihara, T. Takabe and M. Toyoda, "Chaotic Neural Networks," *Physics letters*, A, vol. 144, nos. 6,7, pp. 333-340, 1990.

[3] Changkyu Choi and Ju-Jang Lee, "Dynamical Path-Planning Algorithm of a Mobile Robot: Local Minima Problem and Nonstationary Environments," *Mechatronics*, Vol 6, No. 1, pp. 81-100, 1996.

[4] M.M.A. Hashem, K. Watanabe and K. Izumi, "A Stable-Optimum Gain Tuning Method for Designing Mobile robot Controllers Using an Incest Prevented Evolution Strategy," *Jouranl of Advanced Computational Intelligence*, to be published, 1999.