Two new mutation operators for enhanced search and optimization in evolutionary programming

Kumar Chellapilla, David Fogel

Kumar Chellapilla, David B. Fogel, "Two new mutation operators for enhanced search and optimization in evolutionary programming," Proc. SPIE 3165, Applications of Soft Computing, (13 October 1997); doi: 10.1117/12.279596

Event: Optical Science, Engineering and Instrumentation '97, 1997, San Diego, CA, United States

Two new mutation operators for enhanced search and optimization in evolutionary programming

Kumar Chellapilla[1] and David Fogel[2]

[1]Dept. of Electrical and Computer Engg., Villanova University, Villanova, PA 19085.
[2]Natural Selection. Inc., 3333 N. Torrey Pines Ct., Suite 200, La Jolla, CA 92037.

ABSTRACT

Evolutionary programming (EP) has been successfully applied to many parameter optimization problems. We propose a mean mutation operator, consisting of a linear combination of Gaussian and Cauchy mutations. Preliminary results indicate that both the adaptive and non-adaptive versions of the mean mutation operator are capable of producing solutions that are as good as, or better than those produced by Gaussian mutations alone. The success of the adaptive operator could be attributed to its ability to self-adapt the shape of the probability density function that generates the mutations during the run.

Keywords: evolutionary programming, mutation operators, global optimization.

1. INTRODUCTION

Roughly 35 years ago, evolutionary programming (EP) was proposed for the evolution of finite state machines in order to solve prediction tasks[1]. Since then a number of modifications, enhancements, and implementations have been proposed and investigated. EP was extended to work on real-valued object variables based on normally distributed mutations[2].

Several approaches to perturb the standard deviations during evolution have been investigated[3,4,5,6]. Self-adaptation allows for adapting the properties of the mutation operators during evolution. Typically, as evolution progresses, and as solutions that are closer to the global optimum are obtained, smaller mutations must be favored over larger ones. But the appropriate step size must be learned on-line. A generalization of this would mean that different phases of evolution could require different degrees and different types of mutation. Most self-adaptation schemes work with the evolution of the variances and covariances of Gaussian mutations[4,5,6]. Recently, Cauchy mutations[7,8] have been proposed and have been shown to perform significantly better than Gaussian mutations on many parameter optimization problems. The idea of evolving the variances of Gaussian mutations has been extended to evolving the scale parameters of Cauchy mutations[7].

This paper proposes two new mutation operators consisting of linear combinations of Gaussian and Cauchy mutations. The efficiency of search and the quality of the solutions obtained using these operators is investigated. Section 2 examines how self-adaptation is used in classical evolutionary programming. Section 3 introduces the new mutation operators and their implementation in EP. In section 4, various experiments on test problems are described. Section 5 presents results, and conclusions are offered in section 6.

2. SELF-ADAPTATION IN CLASSICAL EVOLUTIONARY PROGRAMMING

Evolutionary algorithms (evolutionary programming, evolution strategies and genetic algorithms) address the problem of global optimization (minimization or maximization) in the presence of competing multiple local minima.

1. Further Information -
E-mail: kumarc@ece.vill.edu; WWW: http://www.ece.vill.edu/~kumarc/;
Tel: 610-519-7371; Fax: 610-519-4436.
2. E-mail: dfogel@natural-selection.com; WWW: http://www.natural-selection.com/people/dbf.html;
Tel: 619-455-1560; Fax: 610-455-6449.

A global minimization problem can be formalized as a pair (S,f), where $S \subseteq R^n$ is a bounded set on R^n and $f : S \to R$ is an n-dimensional real-valued function. The problem is to find a point $x_{min} \in S$ such that $f(x_{min})$ is a global minimum on S. More specifically, it is required to find an $x_{min} \in S$ such that

$$\forall x \in S : f(x_{min}) \leq f(x). \tag{1}$$

Here f does not need to be continuous but it must be bounded.

Conventional evolutionary programming (CEP) with self-adaptive mutation is implemented as follows:

1. *Initialization:* Generate the initial population of μ individuals, and set the generation number, k, to 1. Each individual is taken as a pair of real-valued vectors, $(x_i, \sigma_i), \forall i \in \{1, ..., \mu\}$. The x_i's give the i^{th} member's Cartesian coordinates and σ_i's it's self-adaptation parameters.
2. Evaluate the error score for each individual, $(x_i, \sigma_i), \forall i \in \{1, ..., \mu\}$, in terms of the objective function, $f(x_i)$.
3. *Mutation:* Each parent $(x_i, \sigma_i), \forall i \in \{1, ..., \mu\}$, creates a single offspring (x_i', σ_i') by:

$$\sigma_i'(j) = \sigma_i(j)\exp(\tau' N(0, 1) + \tau N_j(0, 1)) \tag{2}$$

$$x_i'(j) = x_i(j) + \sigma_i'(j) N_j(0, 1) \tag{3}$$

   for $j = 1, ..., n$, where $x_i(j)$, $x_i'(j)$, $\sigma_i(j)$, and $\sigma_i'(j)$ denote the j^{th} component of the vectors x_i, x_i', σ_i, and σ_i', respectively. $N(0,1)$ denotes a normally distributed one-dimensional random number with mean zero and standard deviation one. $N_j(0,1)$ indicates that the random number is generated anew for each value of j. The factors τ and τ' are commonly set to 1/sqrt(2sqrt(n)) and 1/(2sqrt(n)), respectively.
4. *Fitness:* Calculate the objective function value $f(x_i)$ of each offspring $(x_i', \sigma_i'), \forall i \in \{1, ..., \mu\}$.
5. Conduct pairwise comparison over the union of parents (x_i, σ_i) and offspring (x_i', σ_i'), $\forall i \in \{1, ..., \mu\}$. For each individual, q opponents are chosen randomly from all the parents and offspring with equal probability. For each comparison, if the individual's error is no greater than the opponent's, it receives a "win."
6. *Selection:* Select the μ individuals out of (x_i, σ_i) and (x_i', σ_i'), $\forall i \in \{1, ..., \mu\}$, that have the most wins to be parents of the next generation.
7. Stop if the halting criterion is satisfied; otherwise, increment the generation number, that is $k = k + 1$, and go to step 3.

In the current simulations, the halting criterion was taken to be a certain maximum number of generations (*kmax*).

A number of self-adaptation strategies have been proposed for use with EP, each with individual merits and demerits. EP implementing any one of these strategies would differ from the CEP given above only in the mutation step. The Gaussian and lognormal self-adaptation strategies are given by:

$$\text{Gaussian self-adaptation: } \sigma_i'(j) = \sigma_i(j) + \sigma_i(j) N_j(0, 1) \tag{4}$$

(any $\sigma'_i(j)$ that go negative are set to a small positive value ε)

$$\text{Lognormal self-adaptation: } \sigma_i'(j) = \sigma_i(j)\exp(\tau' N(0, 1) + \tau N_j(0, 1)) \tag{5}$$

The implementation of CEP described above uses lognormal self-adaptation[4]. The generation of the offspring standard deviations (Eq. 2) followed by updating the offspring's positional vectors using these new standard deviations (Eq. 3) is referred to as the "sigma first" type of self-adaptation. On the other hand, if these two operations are performed in the reverse order, it is referred to as the "sigma last" type of self-adaptation. Sigma first self-adaptation has been shown to be superior to the sigma last strategy irrespective of the selection scheme used on a wide range of test functions and problems in drug design[9]. A Cauchy mutation operator (CMO) has also been proposed[7,8]. The corresponding positional parameter update is accomplished as follows:

$$\text{Cauchy Mutation Operator: } x_i'(j) = x_i(j) + \sigma_i'(j) C_j(0, 1) \tag{6}$$

where $C(0,1)$ denotes a Cauchy random number centered at zero with a scale parameter of 1. EP with the CMO as the primary search operator appears to perform better than CEP for several multi-modal functions with many local minima while being comparable to CEP in performance for several unimodal and multi-modal functions with only a few local minima[7]. This enhancement in the performance of the CMO may be attributed to the greater probability given to occasional long jumps.

3. TWO NEW MUTATION OPERATORS

Two new mutation operators are introduced, namely *mean* and *adaptive mean* mutation operators (MMO and AMMO respectively). Both of these operators consist of a linear combination of Gaussian and Cauchy mutations.

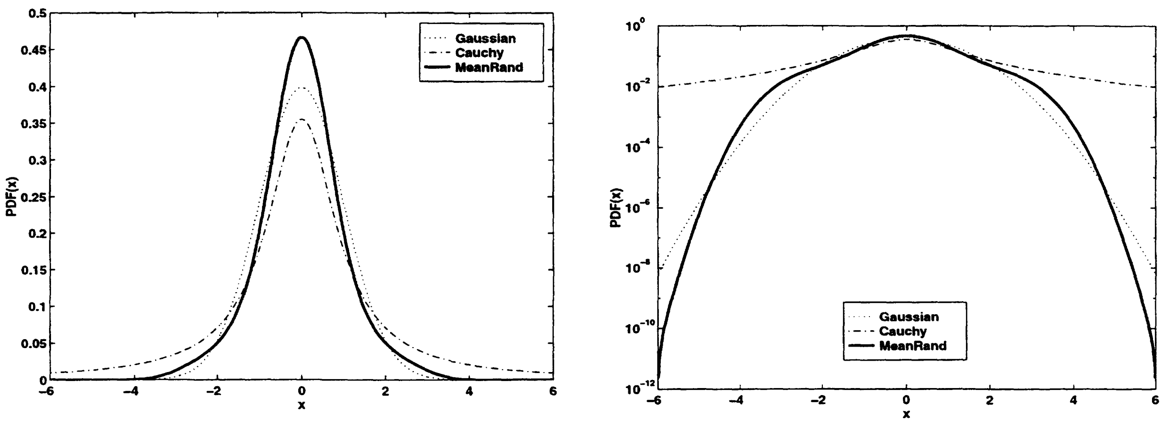
The MMO uses two random variables. The first is a $N(0,1)$ random variable and the second is a $C(0,1)$ random variable. The mean of samples from these two random numbers is scaled by the self-adaptive parameter $\sigma'_i(j)$ and used to perturb the j^{th} component of a parent to obtain the j^{th} component of the offspring. The positional parameter update equation for the MMO is given by:

$$x_i'(j) = x_i(j) + \frac{1}{2}\sigma'_i(j)\{C_j(0,1) + N_j(0,1)\} \tag{7}$$

These mean random numbers follow a probability density function (PDF) which is given by the convolution of the Gaussian and Cauchy PDFs followed by scaling. Mathematically,

$$\begin{aligned} PDF_{mean}(x) &= PDF_{gaussian(0,1)}(2x) * PDF_{cauchy(0,1)}(2x) \\ &= \left\{\frac{1}{\sqrt{\pi}}\exp(-2x^2)\right\} * \left\{\frac{2}{\pi}\frac{1}{1+4x^2}\right\} \end{aligned} \tag{8}$$

where $*$ denotes the convolution operator. Figures 1(a) and 1(b) graph the PDF of the mean random numbers. The Gaussian and Cauchy PDFs are also plotted for comparison. For the sake of analysis, the range of mutations (absolute values) in $[0,\infty]$ is split into five categories, *very small* (0-0.6), *small* (0.6-1.2), *medium* (1.2-2), *large* (2-4.8), and *very large* (>4.8). Among the three distributions there exist trade-offs between the probabilities of generating *very low*, *low*, *medium*, *large*, and *very large* mutations. These are summarized in table 1. In comparison with Gaussian mutations, the mean random mutations generate more *very small* and *large* mutations. Whereas, in comparison with Cauchy mutations, they generate more *very small* and *small* mutations. Thus the MMO generally produces mutations that are larger than Gaussian mutations and smaller than Cauchy mutations.



(a) (b)

Figure 1: A (a) linear, (b) semilog, plot of the probability density function (PDF) of the mean random numbers in comparison with the Gaussian and Cauchy PDFs. Among the three PDFs there exists a trade-off between the probabilities of generating very small (0.0-0.6), small (0.6-1.2), medium (1.2-2.0), large(2.0-4.8), and very large (>4.8) mutations.

During evolution, the shape of the PDF that produces mean mutations is fixed, and the PDF parameters are self-adapted. Self-adaptation of the PDF shape also, along with its parameters would make the self-adaptation scheme more robust to the type of objective function being optimized. In view of this, an *adaptive mean mutation operator* is proposed where positional parameter update equations are given by:

$$
\begin{aligned}
x_i'(j) &= x_i(j) + \sigma'_{i1}(j)C_j(0,1) + \sigma'_{i2}(j)N_j(0,1) \\
&= x_i(j) + \alpha'_i(j)(C_j(0,1) + \beta'_i(j)N_j(0,1))
\end{aligned}
\tag{9}
$$

The AMMO has two sets of self-adaptive parameters $\sigma'_{i1}(j)$ and $\sigma'_{i2}(j)$. The $\sigma'_{i1}(j)$ self-adaptation parameters represent the standard deviations of the Gaussian part and the $\sigma'_{i2}(j)$s represent the scale parameters of the Cauchy part. The mutation step can be rewritten in terms of $\alpha'_i(j)$ and $\beta'_i(j)$ as given in Eq. (9), wherein $\alpha'_i(j)$ plays the role of the overall standard deviation and $\beta'_i(j)$ determines the shape of the PDF. For low values of $\beta'_i(j)$ the PDF resembles that of a Cauchy PDF, whereas for large values of $\beta'_i(j)$ it resembles a Gaussian PDF. Thus with the self-adaptation of the $\alpha'_i(j)$ and $\beta'_i(j)$ parameters it is possible to generate a large number of different PDFs that have a shape between a Gaussian and a Cauchy.

4. EXPERIMENTS

CEP, with the mean mutation operators described in section 3, was tested on a set of five benchmark functions and two time series prediction problems. The five benchmark functions were

$$f_1(\bar{x}) = \sum_{i=1}^{n} x_i^2 \tag{10}$$

$$f_2(\bar{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e \tag{11}$$

$$f_3(\bar{x}) = \sum_{i=1}^{n-1}\left\{100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right\} \tag{12}$$

$$f_4(x) = \sum_{i=1}^{n} x_i^4 + \mathrm{U}([0,1)) \tag{13}$$

where U([0,1)) is a uniform random variable in [0,1).

$$f_5(x) = \sum_{i=1}^{n}\{x_i^2 - 10\cos((2\pi x_i) + 10)\} \tag{14}$$

Function f_1 is the quadratic or sphere function, f_2 is a modified version of the Ackley function[10]. The 20+e term has been added to move the global optimum to the origin. Function f_3 is the extended Rosenbrock function[11], f_4 is the noisy sphere function, and f_5 is the Rastrigin function[12].

The performance of the different operators, namely the Gaussian, the Cauchy, the Mean, and the Adaptive Mean mutation operators, was also tested on two time series prediction problems. The goal was to evolve a reduced parameter bilinear model for a one-step prediction of the Mackey-Glass time series[13] and the Wolf Sunspot data series. The reduced parameter bilinear model is defined in Zhang and Hagan[14] as:

$$A(z^{-1})Z_t = B(z^{-1})e_t + \{C(z^{-1})Z_t\}\{D(z^{-1})e_t\} \tag{15}$$

where $\{Z_t\}$ is the time series to be predicted and $\{e_t\}$ is the sequence of past prediction errors.

$$
\begin{aligned}
A(z^{-1}) &= 1 - a_1 z^{-1} - a_2 z^{-2} - \ldots - a_p z^{-p} \\
B(z^{-1}) &= 1 - b_1 z^{-1} - b_2 z^{-2} - \ldots - b_q z^{-q}
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
C(z^{-1}) &= c_1 z^{-1} + c_2 z^{-2} + \ldots + c_m z^{-m} \\
D(z^{-1}) &= d_1 z^{-1} + d_2 z^{-2} + \ldots + d_k z^{-k}
\end{aligned}
\tag{17}
$$

The identification procedure consists of identifying the model order parameters p, q, m, k and for the determined model order, estimating the coefficients $\{a_j\}$, $\{b_j\}$, $\{c_j\}$, and $\{d_j\}$. The desired optimal model minimizes the minimum description length (MDL) criterion defined in Candy[15] as:

$$(N-\gamma)\log(\sigma_e^2)+\left(\frac{1}{2}\right)(\text{number of independent parameters})\log(N-\gamma) \tag{18}$$

where N is the number of observations of the time-series, $\gamma = \max(p, q, m, k)$ and the mean squared prediction error, σ_e^2, is given by

$$\sigma_e^2 = \left(\frac{1}{N-\gamma}\right)\sum_{t=\gamma+1}^{N}(Z_t-\hat{Z}_t)^2 \tag{19}$$

where $\hat{Z}_t$ is the predicted output at time t obtained using the model with order (p, q, m, k). Using evolutionary programming, the model order is evolved in the following manner: Each organism in the population is a vector which contains the model order parameters (p, q, m, k) followed by the coefficients. Mutations were performed simultaneously on the model order parameters and coefficients. In each generation, the model order parameters were perturbed with a randomly (uniform distribution) selected integer from the set {-1, 0, 1} to give the new model order. The model order and coefficients evolved according to the MDL fitness criterion and the fittest vector after a certain number of generations (after which there is no more improvement in the fitness) contained the desired model order and the coefficients that could be used for time-series prediction. During evolution the model order parameters were limited to a maximum value of 10.

Simulations were conducted with Gaussian, Cauchy, mean, and adaptive mean mutation operators using a population size of 50, and an opponent size $q = 10$. The simulation parameters for the functions f_1-f_5 were chosen as in Saravanan and Fogel[6]. For all the five benchmark functions n was set to 30. For functions f_1 and f_2, all components were initialized uniformly in $[-50,50]^{30}$. For f_3 all components were initialized in $[-30,30]^{30}$. For f_4 all components were initialized in $[-1.28,1.28]^{30}$ and for f_5 all components were initialized in $[-5.12,5.12]^{30}$. For the time series prediction problem the initial model orders were selected randomly from {1, 2, ..., 10} and the model coefficients were initialized in [-0.1,0.1]. For f_1-f_5 the self-adaptive parameters were initialized to 3. For the time series problem all initial self-adaptive parameters were set to 0.1. The termination criterion varied with the function being optimized. The EP runs for f_1 and f_2 were terminated after 3000 generations and for f_3, f_4 and f_5 were terminated after 5000 generations. The time series problems were terminated after 200 generations. A total of 50 independent runs were performed for each problem and each of the four mutation operators. In corresponding runs with different mutation operators the same initial population was used.

5. RESULTS

Figures 2-6 graph the rate of optimization of EP with the four different mutation operators for the five test func-

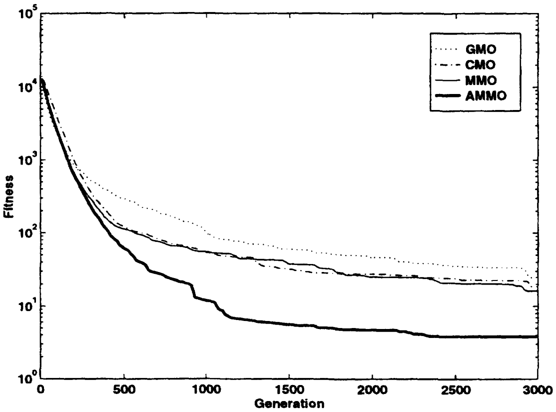


Figure 2: The rate of optimization (averaged over 50 runs) of EP with the four mutation operators for the quadratic function (f_1). In comparison with the other mutation operators, EP with AMMO shows faster convergence. The final mean best solution found by the AMMO is statistically significantly better than that found by EP using the GMO.

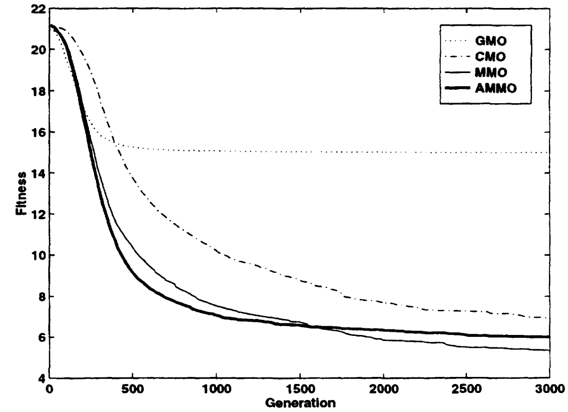


Figure 3: The rate of optimization (averaged over 50 runs) of EP with the four mutation operators for the ackley function (f_2). EP with MMO and EP with AMMO show faster convergence and produces fitter solutions. The solutions found by using MMO and AMMO are statistically significantly better than those found by using the GMO.

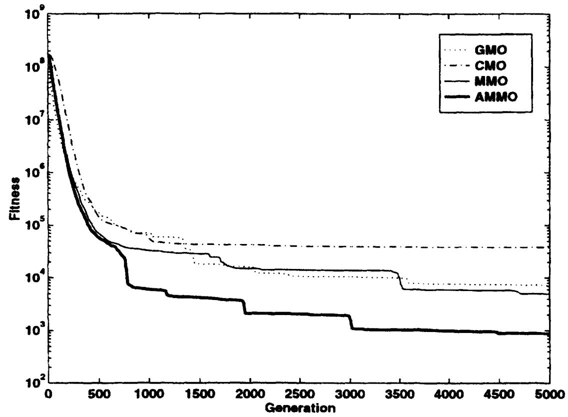


Figure 4: The rate of optimization (averaged over 50 runs) of EP with the four mutation operators for the rosenbrock function (f_3). On average EP with AMMO shows faster convergence and produces fitter solutions. The difference in the quality of the solutions found by EP with AMMO and EP with GMO is statistically significant.

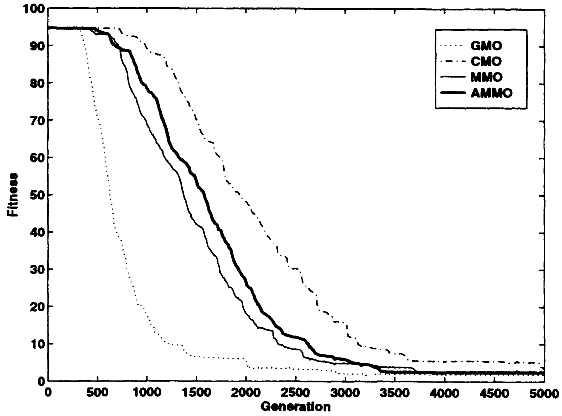


Figure 5: The rate of optimization (averaged over 50 runs) of EP with the four mutation operators for the noisy quadratic function (f_4). In this particular case EP with GMO shows faster convergence and gives the best results. However, all four mutation operators produce solutions of nearly equal fitness. The differences in the quality of the solutions obtained are not statistically significant.

tions f_1-f_5. The mean best scores, mean average scores, and the results of the student's t-test for statistical significance[15], taken over 50 independent runs for f_1-f_5 are given in table 3. On the quadratic function (f_1), EP with the AMMO shows faster convergence than EP with other mutation operators. The final mean best solution found by the AMMO is statistically significantly better than that found by EP using the GMO. In the case of the Ackley function (f_2), both the MMO and the AMMO operators give faster convergence than GMO and the CMO. The results found by MMO and AMMO are statistically significantly better in comparison with those of GMO. On the Rosenbrock function (f_3), AMMO, once again, gives faster convergence than the other three mutation operators. The mean best and mean average solutions obtained using AMMO are statistically significantly better in comparison with the GMO. On

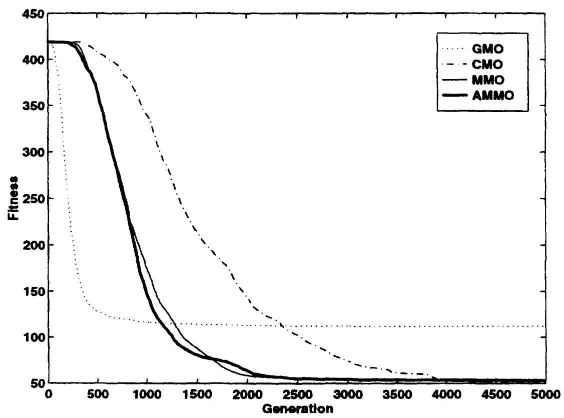
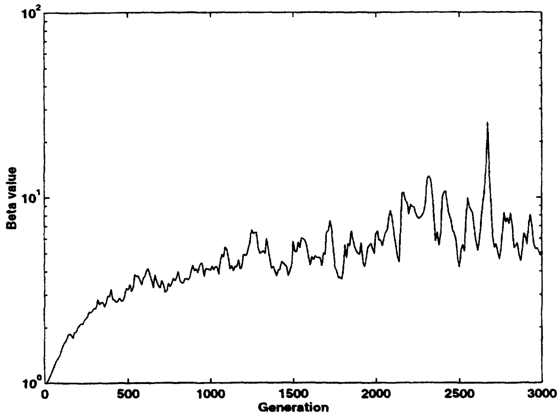
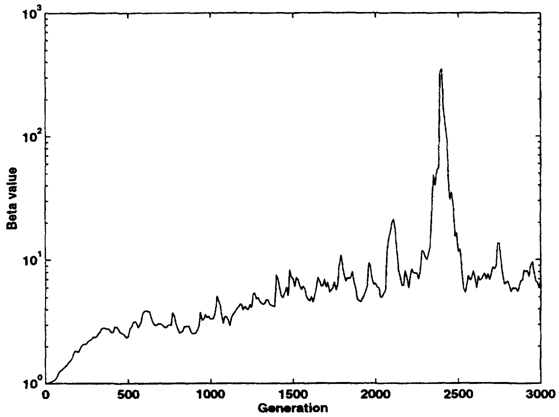


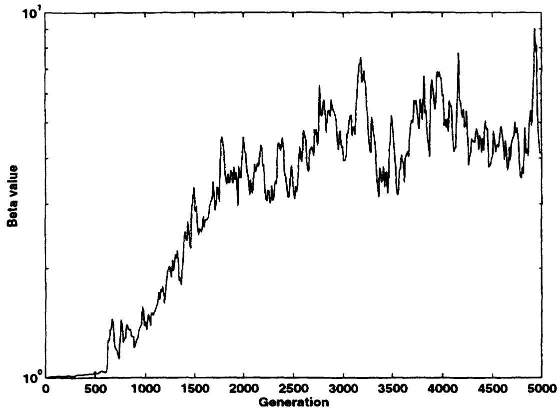
Figure 6: The rate of optimization (averaged over 50 runs) of EP with the four mutation operators for the Rastrigin function (f_5). EP with the GMO converges quickly but stagnates early in the search. In contrast EP with CMO, AMMO, and MMO find better solutions and the MMO and AMMO demonstrate quicker convergence.



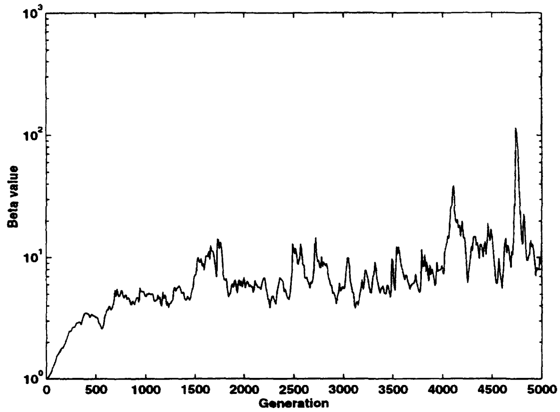
7(a)



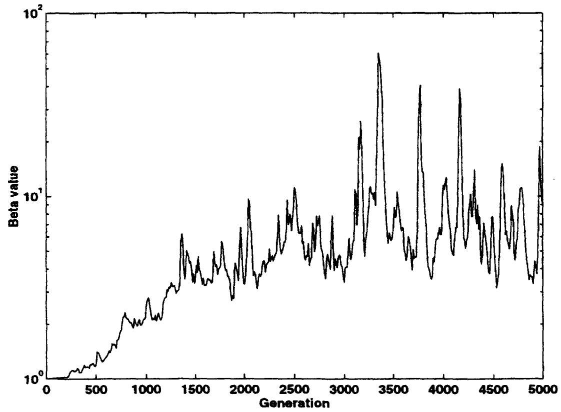
7(b)



7(c)



7(d)



7(e)

Figure 7: The mean trajectory (averaged over 50 runs) of the best member's $\beta'_i(j)$ parameter during the EP with AMMO runs for the (a) Quadratic, (b) Ackley, (c) Rosenbrock, (d) Noisy Quadratic and (e) Rastrigin functions. We observe a general growth in the $\beta'_i(j)$ value in the first 1000 to 2000 generations from 1 to a value between 4 and 10.

the noisy quadratic function (f_4), GMO gives the best convergence and the best solutions. However, the differences in the quality of the solutions obtained using the four operators are not statistically significant. On the Rastrigin function, EP with GMO exhibits faster convergence but stagnates early in the search. The other three mutation operators find better solutions with the MMO and AMMO curves exhibiting faster convergence.

Figures 7(a)-7(e) graph the mean trajectory of the AMMO's $\beta'_i(j)$ parameter (averaged over 50 runs) for the f_1-f_5

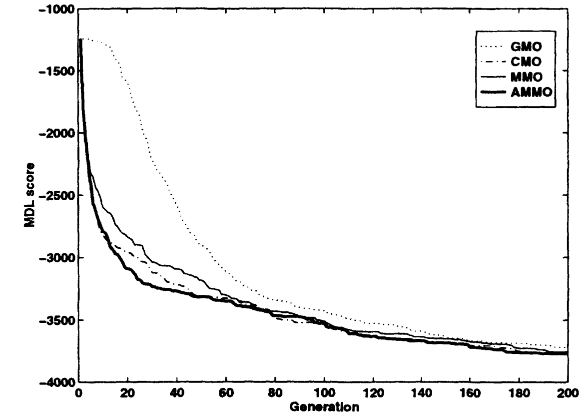


Figure 8: The rate of optimization (averaged over 50 runs) of EP with the four mutation operators for the Mackey-Glass series modeling problem. EP with the AMMO shows significantly faster convergence in comparison with EP using GMO. The differences in quality (MDL score) of the solutions obtained using the different operators at the end of 200 generations are not statistically significant.

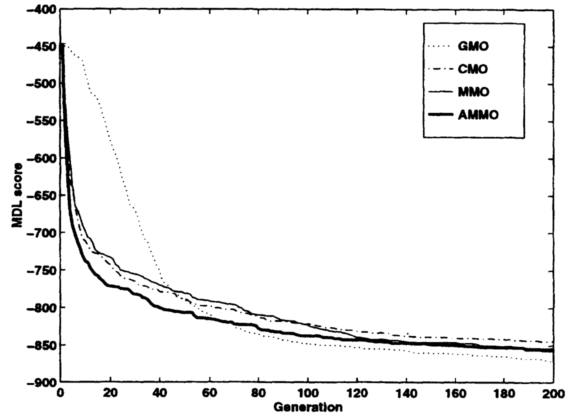


Figure 9: The rate of optimization (averaged over 50 runs) of EP for the four mutation operators on the Wolf sunspot series modeling problem. EP with the AMMO shows significantly faster convergence in comparison with EP using GMO. However the final models evolved by EP using GMO, at the end of 200 generations, are statistically significantly better (lower MDL score) than those produced by the other three mutation operators.

functions respectively. The $\beta'_i(j)$ parameter yields insight as to which form of the PDF produced beneficial mutations during the different phases of evolution. Large values of $\beta'_i(j)$ (>>1) imply a PDF that closely resembles a Gaussian PDF, whereas small values of $\beta'_i(j)$ (<<1) imply a PDF that closely resembles a Cauchy PDF. A value of $\beta'_i(j) = 1$, would be the midpoint, where the PDF is the mean PDF given in Fig. 1. In all the runs the $\beta'_i(j)$ parameters were initialized to 1. From the figures 7(a)-7(e) we observe that in the first 1000-2000 generations there is a gradual increase in the value of $\beta'_i(j)$ from 1 to a value between 4 and 10. After the initial phase of growth, the $\beta'_i(j)$ value oscillates about the final value. Thus, on the five test functions, the AMMO follows a PDF whose form resembles that of the mean PDF at the beginning of the search and gradually changes into a PDF that resembles the Gaussian PDF. In all the runs the value of $\beta'_i(j)$ never went below a value of 1. This could mean that pure Cauchy mutations were found to be less efficient than mean and Gaussian mutations at the beginning of the search and as solutions were obtained that were closer to the global optimum, Gaussian mutations were progressively preferred more and more over Cauchy mutations.

The time series prediction results are given in figures 8 and 9. The mean best MDL scores obtained by the four operators are given in table 2. On the Mackey-Glass series, EP with AMMO shows significantly faster convergence in comparison with GMO. There was no statistically significant difference in the quality of the final solutions obtained at the end of the runs. As might be expected EP with AMMO also shows faster convergence on the sunspot series. However, in this case, EP with GMO produced solutions that were significantly better than those produced by the other three mutation operators.

6. SUMMARY

Two new mutation operators, mean mutation operator (MMO) and adaptive mean mutation operator (AMMO), are proposed to enable EP to vary the probability density function that generates mutations. AMMO and MMO consist of linear combinations of Gaussian and Cauchy mutation operators. These two mutation operators were tested on five well-investigated benchmark functions. In all the test functions considered, MMO and AMMO mutations consistently produced solutions that were as good as or better than those produced by other mutation operators acting alone. In four of the five test functions, the differences in the quality of the solutions produced by the AMMO and those produced by the GMO were statistically significant. On the other hand, in two of the five test functions, the MMO produced solutions that were statistically better than those produced by the GMO. These mutation operators were also tested on the Mackey-Glass and sunspot time series modeling problems. On both of these problems AMMO showed faster convergence than GMO. This success of the AMMO could be attributed to its ability to adapt the shape of the PDF that generates the mutations during the run. These results indicate that self-adapting not only the variances, but also the shape of the PDF, which generates the mutations during search, can significantly enhance the quality of the solutions obtained and reduce the time taken to reach the solution.

REFERENCES

1. Fogel L.J., Owens A.J. and Walsh M.J. (1966). *Artificial Intelligence through Simulated Evolution,* NY: John Wiley

2. Fogel D. B. (1992) *Evolving Artificial Intelligence*. Ph.D. thesis, University of California, San Diego, CA.

3. Fogel D. B, Fogel L. J, Atmar J. W (1991), "Meta-evolutionary programming," in *Proc. of the 25th Asilomar Conf. on Signals, Systems and Computers*, R. R. Chen, Ed. IEEE computer Society, pp. 540-545.

4. Saravanan N, Fogel D.B. (1994), "Learning strategy parameters in evolutionary programming: an empirical study," *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, San Diego, CA, Feb 24-26.

5. Saravanan N, Fogel D.B. (1995), and Nelson K. M., "A comparison of methods for self-adaptation in evolutionary algorithms," *Biosystems* vol. 36, pp. 157-166.

6. Saravanan N, Fogel D.B. (1996), "An empirical comparison of methods for correlated mutations under self-adaptation," *Proc. of the Fifth Annual Conference on Evolutionary Programming* (*EP'96*), San Diego, CA, USA, Feb. 29- Mar. 2, MIT Press. pp. 479-485.

7. Yao X. and Liu Y. (1996), "Fast evolutionary programming," *Proc. of the Fifth Annual Conference on Evolutionary Programming (EP'96)*, San Diego, CA, USA, Feb. 29- Mar. 2, MIT Press. pp. 451-460.

8. Yao X. and Liu Y. (1997), "Fast evolution strategies," *Evolutionary Programming VI: 6th International Conference, EP97,* Indianapolis, Indiana, USA, April 13-16, 1997, pp. 151-161.

9. Gehlhaar D. K. and Fogel D. B (1996), "Tuning Evolutionary Programming for Conformationally Flexible Molecular Docking," *Evolutionary Programming V: Proc. of 5th Annual Conf. on Evolutionary Programming,* L. J. Fogel, P. J. Angeline, T. Bäck (editors), MIT Press, Cambridge, MA.

10. Ackley D.H., (1987) *A connectionist Machine for Genetic Hillclimbing*. Kluwer, Boston, MA. (pp. 13-14).

11. Schwefel H.-P. (1995), *Evolution and optimum seeking.* New York, John Wiley & Sons, Inc.

12. Törn A. and Zilinskas A. (1989) *Global Optimization.*Lecture Notes in Computer Science, vol. 350. Springer-Verlag, Berlin.

13. Mackey M. C and Glass L. (1977), "Oscillations and chaos in physiological control systems," *Science*, 197.

14. Zhang Y. and Hagan M. T. (1994), "A Reduced Parameter Bilinear Time Series Model," *IEEE Trans. Signal Processing,* vol. 42, no. 7, pp. 1867-1870.

15. Candy J. V., *Signal Processing: the model-based approach*, McGraw-Hill Book Company: 1988.

Degree of Mutation	Probability of generating mutations		
	Largest	Medium	Smallest
very small (0.0-0.6)	MMO	GMO	CMO
small (0.6-1.2)	GMO	MMO	CMO
medium (1.2-2.0)	GMO	CMO	MMO
large (2.0-4.8)	CMO	MMO	GMO
very large (>4.8)	CMO	GMO	MMO

Table 1: Probability of generating different degrees of mutations using the Gaussian (GMO), Cauchy (CMO) and Mean (MMO) Mutation Operators.

Series		GMO	CMO	MMO	AMMO
Mackey Glass	Mean	-3717.5	-3778.0	-3756.4	-3768.5
	S.D	445.7	338.9	277.1	307.9
Sunspot	Mean	-872.0	-844.4	-853.9	-856.6
	S.D	30.3	27.4	26.3	27.1

Table 2: The mean best MDL scores for the EP time series modeling runs with different mutation operators: Gaussian (GMO), Cauchy (CMO), Mean (MMO) and Adaptive Mean (AMMO). The S.D rows correspond to the standard deviations. Lower MDL scores imply a better model.

16. Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P. (1992), *Numerical Recipes in C*, New York: Cambridge University Press.

Function	Mutation Operator	Mean Best	S.D	Mean Average	S.D	Test for Significance	
						T-test w.r.t Gaussian	T-test w.r.t Cauchy
Quadratic function (f_1)	GMO	24.8406	54.8097	24.8427	54.8088	-	-
	CMO	18.7292	54.3143	18.7337	54.3169	-	-
	MMO	16.2621	52.7138	16.2631	52.7136	Best: 0.7681 Avg: 0.7683	Best: 0.2324 Avg: 0.2327
	AMMO	3.8129	13.9669	3.8129	13.9670	Best: 2.6381* Avg: 2.6384*	Best: 1.8719 Avg: 1.8724
Ackley function (f_2)	GMO	14.9973	3.3535	14.9973	3.3535	-	-
	CMO	6.9304	6.3437	6.9537	6.3848	-	-
	MMO	5.3377	3.4500	5.3410	3.4485	Best: 15.4544* Avg: 15.4455*	Best: 1.4685 Avg: 1.4792
	AMMO	6.0053	3.4501	6.0054	3.4500	Best: 14.6079* Avg: 14.6079*	Best: 0.8954 Avg: 0.9129
Rosen-brock function (f_3)	GMO	7193.94	13706.63	7215.85	13768.5	-	-
	CMO	37679.44	141421.4	37752.64	141404	-	-
	MMO	4961.64	15041.74	12911.33	57713.8	Best: 0.8302 Avg: -0.7063	Best: 1.6144 Avg: 1.1281
	AMMO	876.95	2651.47	880.31	2650.63	Best: 3.3838* Avg: 3.3837*	Best: 1.8379 Avg: 1.8416
Noisy Quadratic function (f_4)	GMO	1.8761	9.4288	1.8761e0	9.4288e00	-	-
	CMO	3.4503	13.1260	4.6669e9	3.2929e10	-	-
	MMO	2.3367	10.7466	1.5664e4	1.1074e05	Best: -0.2246 Avg: -1.0001	Best: 0.4920 Avg: 1.0022
	AMMO	2.5088	14.1020	4.5507e9	3.2178e10	Best: -0.2612 Avg: -1.0000	Best: 0.3402 Avg: 0.0177
Rastrigin function (f_5)	GMO	111.8413	31.4855	111.8413	31.4855	-	-
	CMO	50.9221	27.6220	50.9229	27.6219	-	-
	MMO	52.2983	19.1424	52.3051	19.1480	Best: 11.7256* Avg: 11.7234*	Best: -0.3150 Avg: -0.3164
	AMMO	53.3251	19.2446	53.3251	19.2446	Best: 12.0254* Avg: 12.0254*	Best: -0.4409 Avg: -0.4408

Table 3: The mean best scores, mean average scores, and the results of the approximate t-test for statistical significance[16] for the EP runs with the different mutation operators: Gaussian (GMO), Cauchy (CMO), Mean (MMO), and Adaptive Mean (AMMO). Positive t-test values indicate better performance. Values with magnitude greater than 1.96 (containing a *) indicate statistically significant results.