Evolutionary Programming using the Levy Probability Distribution

Chang-Yong Lee
Dept. of Industrial Information
Kongju National University
Chungnam, 340-800, South Korea
e-mail : clee@knu.kongju.ac.kr

Yoonseon Song
Telecommunication Basic Research Lab.
Electronics & Telecommunications Research Institute
Taejon, 305-600, South Korea

Abstract

In this work, an evolutionary programming, using the mutation operator based on the Levy probability distribution, was studied. The Levy stable probability distribution has an infinite second moment and has been widely studied in conjunction with the fractal structure of nature. Because of the infinite second moment of the Levy probability distribution, the Levy mutation operator is more likely to generate an offspring that is farther away from its parents than the conventional Gaussian mutation operator. As a result, one can search wider area in the variable space with the Levy mutation. The proposed algorithm was applied to the multivariate functional optimization and we showed empirically that in the case of functions having many local optima, the performance of the proposed algorithm was far better than that of the conventional evolutionary programming.

1 INTRODUCTION

There has been much work on the computational methods that mimic biological evolution and natural selections. Among them, genetic algorithms (GAs), evolutionary programming (EP) and evolutionary strategies (ESs) are most prominent. These methods have drawn much attention in conjunction with the parallel computations. Evolutionary programming (EP) (Fogel, 1995) was initially studied for a method of the evolutionary approach for the artificial intelligence (Fogel, 1966). Later, in the middle of 80's, D. Fogel further developed it for the numerical and combinatorial optimizations. EP has been applied successfully for the optimization of the real valued functions (Fogel, 1990)(Foge, 1994) and other practical problems (Sebald, 1994). In contrast to the genetic algorithms that emphasize the crossover operation, the mutation operation is the main operation in the evolutionary programming. In the conventional EP, each parent generates an offspring via the Gaussian mutation and better individuals among parents and offspring are selected as parents of the next generation.

As stated in the ref. (Yao, 1996), one of drawbacks of EP in the optimization of the multivariate functions is a rather slow convergence to the optimized or nearly optimized solutions. To improve the performance of EP and ESs, X. Yao and Y. Liu recently proposed a fast evolutionary programming (FEP) (Yao, 1996) and a fast evolutionary strategies (FESs) (Yao, 1997a) with the mutation operation based on the Cauchy probability distribution. Unlike the Gaussian probability distribution, The Cauchy probability distribution has infinite second moment, as a result, has much longer tail than that of the Gaussian distribution. Due to the long tail in the distribution, one can get mutated offspring quite different from their parents and it is shown that FEP and FESs have some advantage over the conventional EP and ESs.

In this contribution, we generalize FEP and FESs by using mutation operator based on the Levy probability distribution. With the Levy probability distribution, we can extend and generalize the X. Yao and Y. Liu's works since the Cauchy probability distribution is, in a sense, a special case of the Levy probability distribution (The reason will be clear in section II.). The Levy probability distribution differs from the Gaussian probability distribution in that the Levy distribution, like the Cauchy distribution, has infinite second moment. In particular, the Levy distribution exhibits a power law in the tail region that is a characteristic of the fractal structure of nature. Thus, the distribution draws much attention to the physics communities to explain the fractal structure of nature, which were found in areas such as turbulence, polymer, biological systems, and even economics (Mandelbrot, 1982). Moreover, by adjusting the parameter α in the distribution, one can tune the probability density, which in turn yields adjustable variation after the mutation. In order to investigate and compare the performance of the Levy mutation with Gaussian mutation, we fixed the parameters used both in the proposed algorithm and EP.

The rest of the paper is organized as follows. In section 2, we discuss the characteristics of the Levy probability distribution and the algorithm for generating random variables having the Levy probability distribution. Section 3 describes the multi-dimensional function optimization problems using both Gaussian and Levy mutation operators in the EP. In section 4, we analyze the difference between the two operators in terms of the mean square displacement and the distribution of the variables. This is followed by result and discussion of the functional optimization experiments for both mutation operators and the last section is devoted to the conclusion and future studies.

2 LEVY PROBABILITY DISTRIBUTION

For some identically distributed stochastic variables x's, if a stochastic variable z, which is a linear combination of x's, has the same probability distribution as x's, then the stochastic process is called *stable*. A typical example is the central limit theorem: The sum of n independent stochastic variables having finite second moment converges to a Gaussian stochastic process. It should be noted that in order for the central limit theorem to be held, the variance (or the second moment) of the distribution should be finite. It, however, had been shown by Levy (Levy, 1937) that there exists a probability distribution which is stable even the second moment is infinite. This probability distribution is called the Levy probability distribution and has the following form:

$$L_{\alpha,\gamma}(x) = \frac{1}{\pi}\int_0^\infty e^{-\gamma q^\alpha} \cos(qx)\ dq \quad . \tag{1}$$

As can be seen easily from the equation, the distribution is symmetric with respect to $x = 0$ and the distribution has two parameters γ and α. γ is the scaling factor satisfying $\gamma > 0$ and α satisfies $0 < \alpha < 2$. The analytic form of the integral is not known for general α. In particular, for $\alpha = 1$, the integral can be reduced to the analytic form and the distribution is known to the Cauchy distribution. When $\alpha = 2$, the distribution is no longer the Levy distribution and becomes the Gaussian distribution.

The scaling factor γ can be set to $\gamma = 1$ without loss of generality. To see this, let's set $\eta = \gamma^{1/\alpha} q$, then the above distribution becomes

$$\begin{aligned} L_{\alpha,\gamma}(x) &= \frac{1}{\pi}\int_0^\infty e^{-\gamma q^\alpha} \cos(qx)\ dq \\ &= \gamma^{-1/\alpha}\frac{1}{\pi}\int_0^\infty e^{-\eta^\alpha} \cos(\eta(\gamma^{-1/\alpha}x))\ d\eta \\ &= \gamma^{-1/\alpha} L_{\alpha,1}(\gamma^{-1/\alpha}x) \quad . \end{aligned} \tag{2}$$

Thus, by letting $x_s = \gamma^{-1/\alpha}x$, we have

$$L_{\alpha,\gamma}(x) = \gamma^{-1/\alpha} L_{\alpha,1}(x_s) \quad . \tag{3}$$

Therefore, by knowing the distribution when $\gamma = 1$, we can get the distribution of any other γ. In this study, we fix $\gamma = 1$ and denote $L_{\alpha,1} = L_\alpha$.

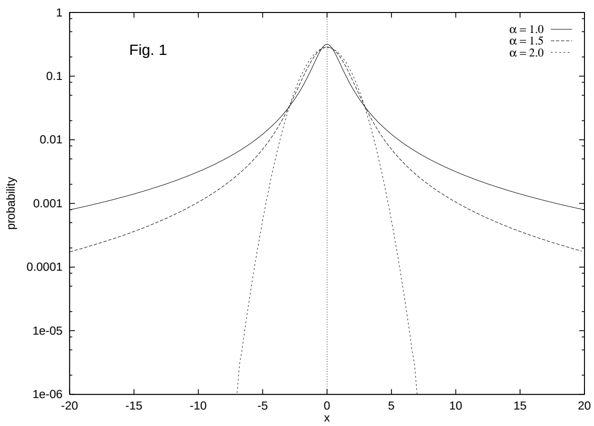


Figure 1: Comparison of the Levy probability distribution of $\alpha = 1.0$ and $\alpha = 1.5$ with the standard Gaussian distribution ($\alpha = 2.0$). The ordinate is log-scale.

The main reason that the Levy distribution has been widely studied in the field of fractal structure is its power law behavior in the large x region. It can be shown (Levy, 1937) that, for large values of x,

$$L_\alpha(x) \propto x^{-(\alpha+1)} \quad . \tag{4}$$

Thus, except for $\alpha = 2$(Gaussian distribution), one can easily see that this distribution has infinite second moment. Figure 1 compares the Levy distribution of $\alpha = 1.0$ and $\alpha = 1.5$ with the standard Gaussian distribution ($\alpha = 2$). The parameter α plays a role of controlling probability distribution. As can be seen in the figure, the distribution exhibits a power law in the large values of x and one can obtain different probability distributions depending on the parameter α. That is, the smaller is the parameter α, the longer the tail.

To implement the Levy distribution to EP we need an algorithm for generating random numbers having the Levy probability distribution, since the analytic form of the distribution (eq.(1)) is not known for general α. Recently, an algorithm for generating random numbers of the Levy probability distribution was introduced (Mantegna, 1994). This algorithm utilizes two random variables from standard Gaussian distribution and perform some transformation using these two variables. In this way, one can generate random variables for $0.3 < \alpha < 1.99$. To be more specific, from the two Gaussian random variables x and y , define the following transformed variable v:

$$v = \frac{x}{|y|^{1/\alpha}} \quad . \tag{5}$$

Then, it was shown (Mantegna, 1994) that the sum of these variables with an appropriate normalization,

$$z_n = \frac{1}{n^{1/\alpha}} \sum_{k=1}^{n} v_k \quad , \tag{6}$$

has a Levy probability distribution being better converges to the Levy probability distribution with larger n.

It, however, has a practical disadvantage. It is necessary to generate many (usually $n = 100$) such random variables to get the desired distribution. To perform simulations, in which random variables from the Levy distribution is involved, a fast and accurate algorithm for generating the random variables is highly desired. To get a more efficient algorithm, the following non-linear transformation was considered.

$$w = \{(k(\alpha) - 1)e^{-|v|/c(\alpha)} + 1\}v \quad , \tag{7}$$

where parameters $k(\alpha)$ and $c(\alpha)$ depend on α and calculated in the ref. (Mantegna, 1994), and v is from eq.(5). In this way, one can generate random variables of the Levy probability distribution for $0.8 < \alpha < 1.95$, even $n = 1$. Figure 2 shows some examples of the distribution of the random variables generated from the above non-linear transformation with $n = 1$.

3 FUNCTIONAL OPTIMIZATION USING THE EVOLUTIONARY PROGRAMMING

In this section, we discuss EP for the functional optimization using Gaussian as well as the Levy mutation operators.

Functional optimization problem can be defined as follows: For a function $f(\vec{x})$,

$$f : S \to R \ , \text{ and } \ \vec{x} \in S \subseteq R \quad , \tag{8}$$

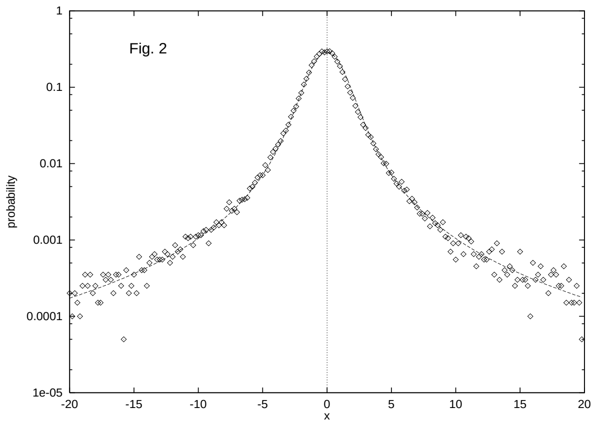


Figure 2: The Levy probability distribution of $\alpha = 1.5$. Solid lines are from the numerical integration of the Levy distribution, eq.(1) and dotted are random numbers generated from eq.(7).

the algorithm aims to find

$$\vec{x}_{min} \ \text{ such that } \ \forall \vec{x} \in S \ , \ \ f(\vec{x}_{min}) \le f(\vec{x}) \quad . \tag{9}$$

For the conventional EP, we adopt the procedure of the algorithm as was used in the ref. (Yao, 1996).

Step 1 : Generate the initial population consisting of μ individuals, each of which can be represented a set of real vectors, $(\vec{x}_i, \vec{\sigma}_i), \ i = 1, 2, \cdots, \mu$. Each $\vec{x}_i$ and $\vec{\sigma}_i$ have m independent components

$$\vec{x}_i = \{x_i(1), x_i(2), \cdots, x_i(m)\} \ , \tag{10}$$
$$\vec{\sigma}_i = \{\sigma_i(1), \sigma_i(2), \cdots, \sigma_i(m)\} \ . \tag{11}$$

Step 2 : For each parent $(\vec{x}_i, \vec{\sigma}_i)$, create an offspring $(\vec{x}'_i, \vec{\sigma}'_i)$ in the following manner. For $j = 1, 2, \cdots, m$,

$$x'_i(j) = x_i(j) + \sigma_i(j) N_j(0,1) \ , \tag{12}$$
$$\sigma'_i(j) = \sigma_i(j) \ exp\{\tau' N(0,1) + \tau N_j(0,1)\} \ , \tag{13}$$

where $N(0, 1)$ stands for a standard Gaussian random variables fixed for each i and $N_j(0, 1)$, a newly generated Gaussian random variable for each component j. The parameter τ and τ' are defined as (Bak, 1993)

$$\tau = \frac{1}{\sqrt{2\sqrt{m}}} \quad \text{and} \quad \tau' = \frac{1}{\sqrt{2m}} \quad . \tag{14}$$

Step 3 : From μ parents $(\vec{x}_i, \vec{\sigma}_i)$ and their μ offspring $(\vec{x}'_i, \vec{\sigma}'_i)$, calculate their fitness values $f_1, f_2, \cdots, f_{\mu+\mu}$.

Step 4 : This step is a selection procedure. First, define and initialize winning functions for each parents and its offspring as $w_i = 0$, for $i = 1, 2, \cdots, \mu + \mu$. For each i, select one fitness function, say $f_j (j \neq i)$ and compare the two fitness functions. If f_i is less than f_j, the winning function for the individual increases by one unit, $w_i = w_i + 1$. Perform this procedure q times for each parent and its offspring.

Step 5 : For the winning function $w_i, i = 1, 2, \cdots, \mu + \mu$, select μ individuals that have the greater winning value to be the parents for the next generation.

Step 6 : Repeat step 2 through step 5 until the required criteria are satisfied.

To see the effect of the Levy mutation, we keep the difference between procedures of the conventional EP and EP with the Levy mutation as minimum as possible. Thus the procedure of EP with the Levy mutation adopted in this work are the same as that of the conventional EP except the eq.(12) of the step 2 in the procedure and is replaced by

$$x_i'(j) = x_i(j) + \sigma_i(j) L_j(\alpha) \quad . \tag{15}$$

Here, $L_j(\alpha)$ is a random number, for each different j, newly generated from the Levy probability distribution of the parameter α. As mentioned before, we fix $\gamma = 1$ and the parameter α is remained unchanged during the procedure. However, to see the effect of the different α, we simulated with different values of α.

4 ANALYSIS OF THE MUTATION OPERATIONS

To compare the effects of the Levy mutation with the conventional Gaussian mutation, we analyze the variables after mutation operations. To do this, let's suppress the index i which represents each individual x_i and denote the superscript t to be an offspring obtained at t^{th} generation. We further assume, for the sake of simplicity, that the adaptive mutation(eq.(13)) is turned off. That is, we regard $\sigma_i(j)$ as some constant β. With the above conventions, the eq.(12) and the eq.(15) can be rewritten in general as

$$x^{(t)}(j) = x^{(t-1)}(j) + \beta X_j^{(t-1)} \quad , \tag{16}$$

where $X_j^{(t-1)}$ is identically distributed random variables from the standard Gaussian or the Levy distributions. Now suppose that j^{th} component of variable x survives after t generation, the component can be written as

$$\begin{aligned} x^{(t)}(j) &= x^{(t-1)}(j) + \beta X_j^{(t-1)} \\ &= x^{(t-2)}(j) + \beta X_j^{(t-2)} + \beta X_j^{(t-1)} \\ &= \cdots \\ &= x^{(0)}(j) + \beta \sum_{k=0}^{t-1} X_j^{(k)} \quad . \end{aligned} \tag{17}$$

For the case of the Gaussian mutation, since $X_j^{(k)}$ are identically distributed random variables from $N(0,1)$, the variation of the variable after t^{th} generation defined as

$$\Delta_j(t) \equiv x^{(t)}(j) - x^{(0)}(j) = \beta \sum_{k=0}^{t-1} X_j^{(k)} \tag{18}$$

also has a Gaussian probability distribution with mean 0 and the variance $\beta^2 t$, *i.e.* $\Delta_j(t) \sim N(0, \beta^2 t)$, without even invoking the central limit theorem.[1] To see the effect of the Gaussian mutation, we calculated the square root of the mean square displacement of $\Delta_j(t) = x^{(t)}(j) - x^{(0)}(j)$ and the result is

$$\sqrt{< \Delta_j^2(t) >} = \beta \sqrt{t} \;\propto\; \sqrt{t} \quad , \tag{19}$$

where, $< \cdots >$ stands for the expectation value. This is similar to the Brownian motion in the statistical mechanics. From this, we know that the average variation after t generation with the Gaussian mutation is proportional to the $\sqrt{t}$. That is, on average, the mutated offspring $x^{(t)}$ after t generation differs from its original parent $x^{(0)}$ by the factor of $\sqrt{t}$. Thus in order to get a large varied (or mutated) offspring, we need to carry out many generations.

Now, let's consider the Levy mutation. In this case, the random variable $X_j^{(k)}$ in the eq.(17) are from the Levy probability distribution and from the fact that the Levy probability distribution is *stable*, their sum also has a Levy probability distribution. Thus the square root of the mean square displacement(or variance) of $\Delta_j(t)$ becomes

$$\sqrt{< \Delta_j^2(t) >} = \infty \quad , \tag{20}$$

from the infinite second moment of the Levy distribution. It should be noted that in the case of the Levy mutation, the mean square displacement does not depend on the generation t. Thus it is possible to get a large variation on the offspring even after just on generation, without having many successive mutation over generations. An advantage of the Levy mutation

[1] The above assertion holds even if $X_j^{(k)}$ are *not* from the Gaussian distribution. As long as these are from same probability distribution having some finite second moment, the central limit theorem could be readily held.

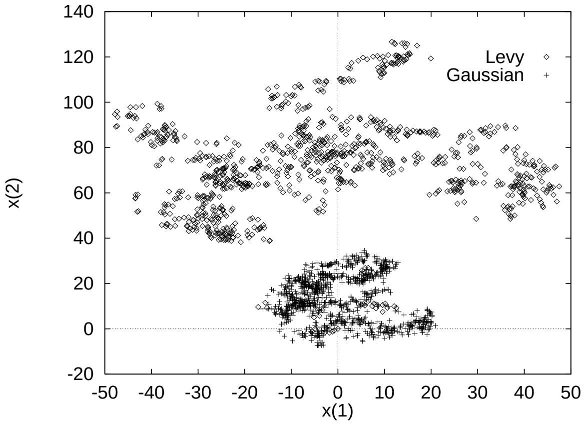


Figure 3: Plot of the two dimensional variables from the Gaussian and the Levy($\alpha = 1.5$) mutations. For both variables, we start from the origin, plot 1000 points, and set $\beta = 1$.

is in its ability to search wider region of the search space due to large variation of the mutated offspring. As a result, it helps to get out of local optima. This is especially true when the local optimum is deep and has wide basin. Once a search algorithm falls into a local optimum, it is hard to get out of the local optimum unless largely varied offspring are generated. The Levy mutation does exactly what is needed. To see this more concretely, we generate offspring having two components using both the Gaussian and the Levy mutations. More specifically, consider the eq.(16) in two dimension as follows.

$$x^{(t)}(1) = x^{(t-1)}(1) + \beta X_1^{(t-1)} \quad , \tag{21}$$

$$x^{(t)}(2) = x^{(t-1)}(2) + \beta X_2^{(t-1)} \quad . \tag{22}$$

Using both mutation operations for X_j^k, we plot the variable $(x^{(t)}(1), x^{(t)}(2)),\ t = 1, 2, \cdots, 1000$ as shown in the fig. 3. As can be seen in the figure, the Levy mutated variables cover wider range than the Gaussian mutated ones.

It is the infinite second moment of the Levy distribution that makes all the difference. We'll see the effect of the Levy mutation in the experiments on the functional optimization in the following section.

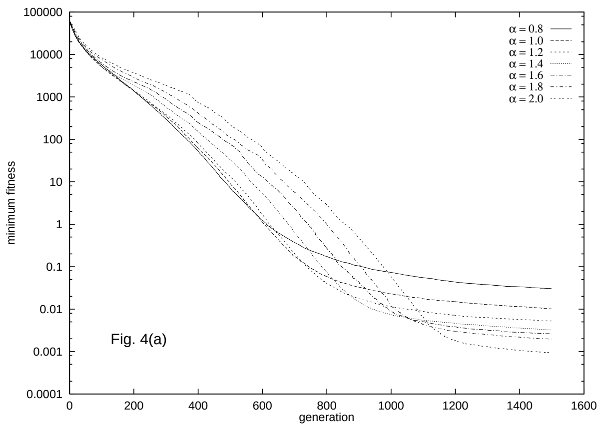
5 EXPERIMENTAL RESULTS AND DISCUSSION

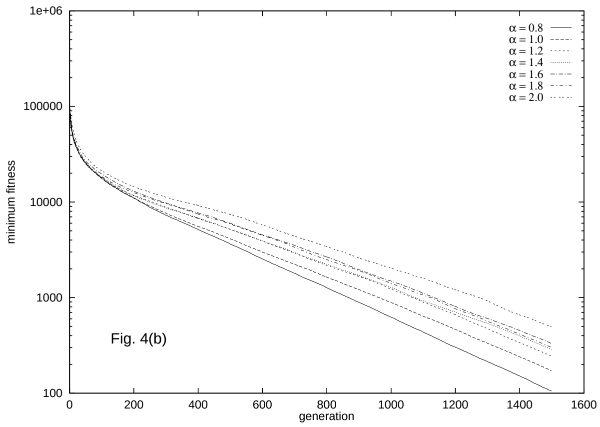
We applied the algorithms in section III to the functional optimization problems to see the difference between these two mutation operators. Table 1 shows the functions and the ranges of the variables that have been used in this study. These are benchmarking functions that were considered in the ref. (Yao, 1996). In particular, we divide the test functions into three classes: functions with no local minima, many local minima, and a few local minima. The following are the parameters that have been used in the simulation:

- number of population: $\mu = 100$
- tournament size: $q = 10$
- initial standard deviation: $\sigma = 3.0$
- number of the generation:
  1500 for $f_1, f_2, \cdots, f_7$ and 20 for f_8, f_9

5.1 UNIMODAL FUNCTIONS

Figure 4 shows the minimum value of the functions from both mutations as a function of the generation averaged over 50 independent trials. As one can see, as far as the convergence rate is concerned, the Levy mutation operator gives faster convergence rate than that of the Gaussian mutation operator. This is because the Levy mutation operator in general generates much longer jump than the Gaussian mutation, thus yields the mutated variables to move farther away from their parents. This was also shown in the case of the Cauchy mutation (Yao, 1997b). The longer jump, however, yields the premature convergence as can be seen in fig. 4 (a). Since the Levy mutation can yield offspring farther away from their parents with higher probability, when the variables are farther away from the global minimum, longer jump is effective. On the other hands, in the vicinity of the global minimum, longer jump is ineffective due to the large variation of the offspring.





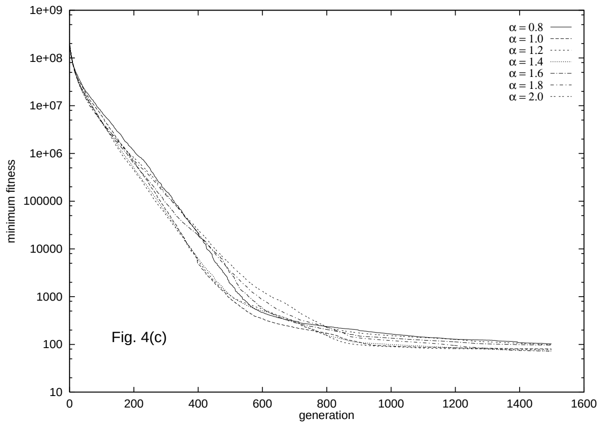
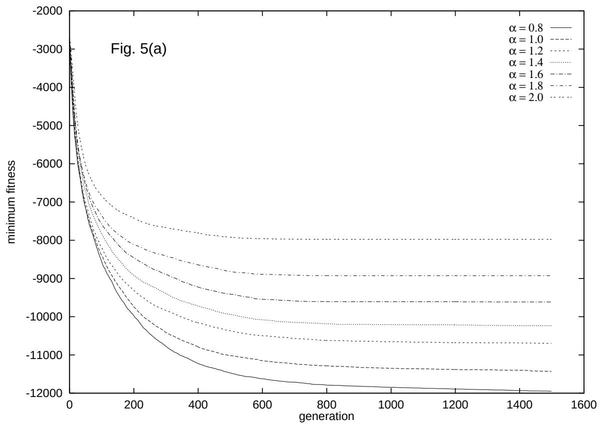
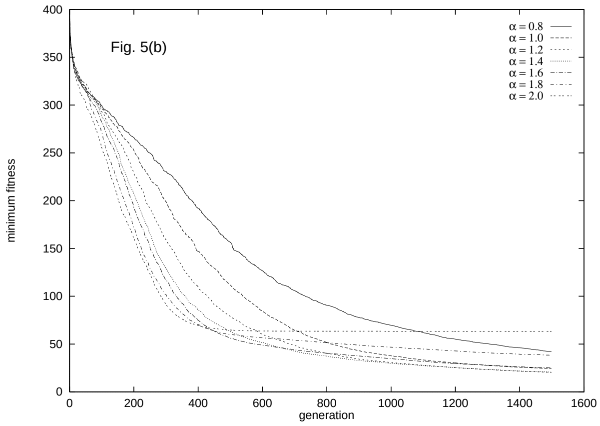


Figure 4: Minimum value of the test functions versus generation using both mutation operations. 50 independent trials were averaged. Figures (a), (b), (c) correspond to the results of test functions f_1, f_2, f_3, respectively.





5.2 FUNCTIONS WITH MANY LOCAL MINIMA

For the functions with many local minima, the Levy mutation yielded better results for both convergence rate and the obtained minimum values as seen in fig. 5. The change of the variables after the Gaussian mutation is so small that it is hard to get out of the local minima especially the local minimum has a wide basin, while the Levy mutation gives higher probability of getting out of the local minima because of the large variation. That is, once the algorithm falls into a local minimum, it is hard to get out of it unless the mutated value is large enough. It is the Levy mutation operation that helps to get out of the local minimum. Furthermore, the Levy mutation can search the variable regions more evenly than the Gaussian mutation. Considering that many practical problems have many local optima in general, it would be better to use the Levy mutation. It should be also noted that the optimal value of for the Levy mutation depends on the function considered and an appropriate value of is needed to be determined.

5.3 FUNCTIONS WITH A FEW LOCAL MINIMA

Figure 6 show the results for the optimization of the functions with a few local minima. In this case, it is hard to tell which mutation operator performs better. This is probably due to the fact that the dimension of the variable is low and it was relatively easy to get to the global minimum.

In summary, the Levy mutation operator yields a fast convergence to the nearly minimum and is beneficial especially for the functions of the many local minima.

6 CONCLUSION

In this study, we utilize the mutation operator based on the Levy stable probability distribution and investigate the advantage and/or disadvantage with the Gaussian mutation. We also analyze the effect of the

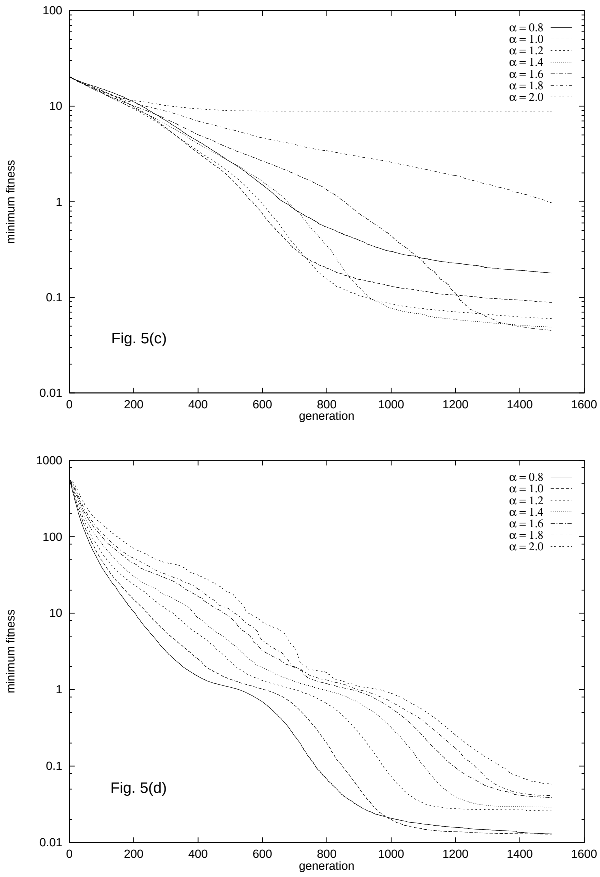


Figure 5: Minimum value of the test functions versus generation. 50 independent trials were averaged for both mutations. Figures (a), (b), (c), (d) correspond to the test functions f_4, f_5, f_6, f_7, respectively.

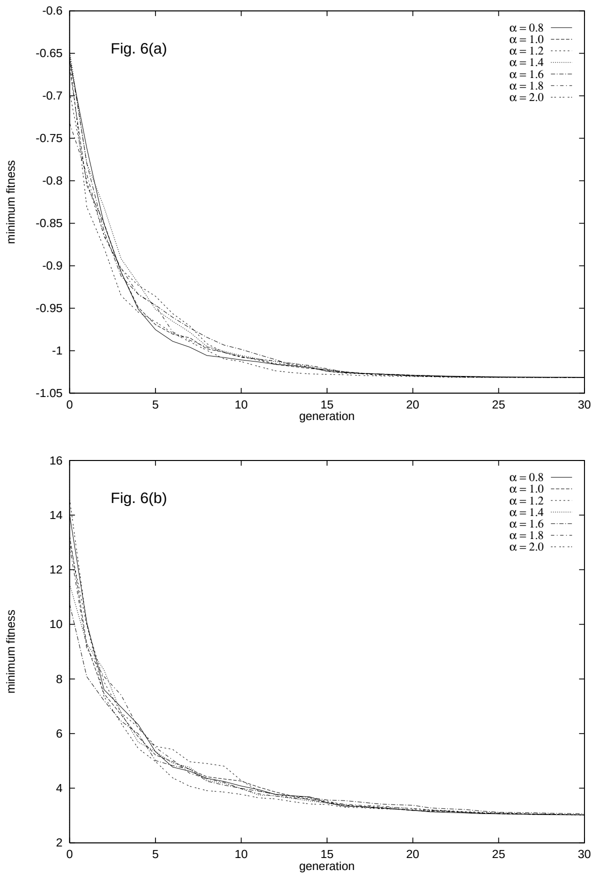


Figure 6: Minimum value of the test functions versus generation for both mutations. 50 independent trials were averaged. Figures (a), (b) correspond to the test functions f_8, f_9, respectively.

two mutation operations and show that the Levy mutation indeed yields a larger variation, which is needed for the better optimization of the complex functions with many local optima.

For the unimodal functions, the Levy mutation yields faster convergence to a minimum but shows the premature convergence. In this class of functions, one can see that the smaller α, the faster the convergence rate and premature convergence. There was not much difference between the two mutation operations for the functions with a few local minima. For the functional optimization with many local minima, however, the Levy mutation has advantage over the Gaussian mutation on both the convergence rate and the obtained minimum values. In particular, the Levy mutation operator can search wider region of the variables due to the long jump (or large variation) of the variables. Considering that many practical optimization problems stem from the complex system having the many local optima, it might be useful to use the Levy mutation than the Gaussian mutation. It is also expected that the above results using Levy mutation can be improved because the parameter chosen are known to be the best for the Gaussian mutation, thus may not be the best for the Levy mutation.

EP with the Levy mutation operator have many aspects of the future studies among which the following three are worth mentioning. Firstly, we have not considered the region of $0 < \alpha \leq 0.7$. This is mainly due to the fact that a fast algorithm of generating the Levy random numbers in the above αregion has not been found. However, in order to see that a further reduction in α would lead to even better result, we need to include this region of the parameter. Secondly, in applying the Levy mutation to EP, we apply the Levy mutation only to the variation of the variables(eq.(12)) not to the self-adaptive deviation(eq.(13)). It is, however, more natural to apply the Levy mutation to the self-adaptive deviation. Finally, even though we fixed the parameter α for each simulation, it is highly desir-

Table 1: List of functions that were tested in this study. N stands for the dimension of the variables and S their ranges.

Test Functions	N	S
$f_1 = \sum_{i=1}^{N} x_i^2$	30	$(-100, 100)^N$
$f_2 = \sum_{i=1}^{N} (\sum_{j=1}^{i} x_j)^2$	30	$(-100, 100)^N$
$f_3 = \sum_{i=1}^{N-1} \{100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\}$	30	$(-30, 30)^N$
$f_4 = -\sum_{i=1}^{N} x_i \sin(\sqrt{x_i})$	30	$(-500, 500)^N$
$f_5 = \sum_{i=1}^{N} \{x_i^2 - 10\cos(2\pi x_i) + 10\}$	30	$(-5.12, 5.12)^N$
$f_6 = -20exp\{-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\} - exp\{\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\} + 20 + e$	30	$(-32, 32)^N$
$f_7 = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \Pi_{i=1}^{N}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$(-600, 600)^N$
$f_8 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$(-5, 5)^N$
$f_9 = \{1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\} \times \{30 + (2x_1 - 3x_2)^2)(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\}$	2	$(-2, 2)^N$

able to make the parameter self-adaptive so that the value of parameter can be also changed adaptively during the procedure. This would bring a more complete adaptive algorithm in the evolutionary computation and should be the future direction of the study.

This study is the first step to generalize the Gaussian mutation in EP. Further studies on this matter should be carried out in the direction of the algorithms itself as well as applications to the other functional optimization and the practical problem.

Acknowledgments

The author thank to Prof. X. Yao for informing his work on the fast evolutionary programming and Dr. N. Saravanan for many useful discussions on the evolutionary programming.

References

T. Bak and H.-P. Schwefel (1993). "An overview of evolutionary algorithms for parameter optimization", Evolutionary Computation Vol. 1, No. 1, pp. 1-23.

D. Fogel (1995). "Evolutionary Computation: Toward a New Philosophy of Machine Intelligence", IEEE Press, NY.

D. Fogel and J. Atmar (1990). "Comparing Genetic Operators with Gaussian Mutation in Simulated Evolutionary Processes Using Linear Systems", Biological Cybernetics, Vol. 63, pp. 111-114.

L. Fogel, A. Owens, and M. Walsh (1966). "Artificial Intelligence Through Simulated Evolution", John Wiley & Sons, New York, NY.

D. Fogel and L. Stayton (1994). "On the Effectiveness of Crossover in Simulated Evolutionary Optimization", BioSystems, Vol. 32:3 pp. 171-182.

P. Levy (1937). "Theorie de l'Addition des Veriables Aleatoires", Gauthier-Villars, Paris, 1937; B. Gnedenko and A. Kolmogorov, "Limit distribution for Sums of Independent Random Variables", Addition-Wesley, Cambridge, MA, 1954.

B. Mandelbrot (1982). "The Fractal Geometry of Nature", Freeman, San Francisco.

R. Mantegna (1994). "Fast, accurate algorithm for numerical simulation of Levy stable stochastic process", Physical Review E. Vol. 49, No. 5, pp. 4677-4683.

A. Sebald and J. Schlenzig (1994). "Minimax Design of Neural Net Controllers for Highly Uncertain Plants", IEEE Trans. Neural Networks, Vol. 5:1, pp. 73-82.

X. Yao and Y. Lin (1996). "Fast evolutionary programming", in Evolutionary Programming V: Proceedings of the 5th Annual Conference on Evolutionary Programming, MIT Press Cambridge, MA.

X. Yao and Y. Lin (1997a) "Fast Evolution Strategies", in Evolutionary Programming VI: Proceedings of the 6th Annual Conference on Evolutionary Programming, pp. 151-161, Springer.

X. Yao, G. Lin and Y. Liu (1997b). "An analysis of evolutionary algorithms based on neighborhood and step size", in the Proceedings of the 6th international conference on evolutionary programming, pp. 297-307.