

線形変動パラメータによる指数型進化的プログラミング

成久 洋之・谷口 隆裕・津田 倫彰*・片山 謙吾

岡山理科大学工学部情報工学科

* 岡山理科大学大学院工学研究科情報工学専攻

(2004年 9 月28日受付、2004年11月 5 日受理)

1 はじめに

進化的計算 (Evolutionary Computation) [1] は自然の進化と適応の概念を導入した計算システムのことであり、従来の伝統計算手法では解けない大規模で複雑な問題を解くための強靱なしかも効率的計算法と考えられ注目されている。これには進化的戦略 (Evolutionary Strategies;ES)、進化的プログラミング (Evolutionary Programming;EP)、遺伝的アルゴリズム (Genetic Algorithms;GA) と遺伝的プログラミング (Genetic Programming;GP) などがある。

ES は 1965 年に数値最適化手法として Rechenberg と Schwefel により提案されたものである。EP は Fogel により 1965 年有限状態機械の人工知能的アプローチとして提案され、後に組み合わせや数値最適化手法に応用されるに至った。GA は 1975 年に Holland により適応的探索アルゴリズムとして提案されたものである。GP は 1987 年に Koza の LIPS プログラムとして提案されたものであるが、木構造染色体に対する GA の応用とみなすことができる。

これらの計算手法はすべて個体集団 (population) を使用することと個体集団を構成する個体 (individual) の間の情報交換により、解を探索しようとするものである。

本論文は進化的計算の中の EP につき記述するものである。進化的プログラミングは有限個の解集合からなる個体集団に対し突然変異 (mutation) と選択により、より質の高い解をもつ個体集団を発生させようとする集団探索法である。これまでに提案されている EP は主として Gaussian Mutation による CEP と Cauchy Mutation による FEP 手法がある。前者は Gauss 分布乱数を、後者は Cauchy 分布乱数をそれぞれ使用し、関数最適化問題に対しては一般的に FEP の方が有効であるとされている。[2] しかしながら、問題によっては CEP の方がよりすぐれた収束特性を示すこともあり、さらに収束の状況によって結果は異なる。この観点から、Narihisa らは 2002 年に複合指数分布乱数を使用する指数型進化的プログラミング (EEP) を提案した。[3] これは指数分布のパラメータを制御することで分布の分散を変動させ収束の効率化を狙ったものである。これまでの研究では種々のパラメータ値に対する収束特性を検討してきたが [4][5][6][7]、今回はパラメータ値を進化世代 (evolutionary generation) に対応して線形に変動させた EEP の収束特性につき検討したものである。数値実験として、この分野でよく知られたベンチマーク問題に適応した結果、かなり良好な収束特性が得られている。

2 標準進化的プログラミング (CEP)

Bäck と Schwefel による EP のアルゴリズムは次の通り。

Step1: μ 個の個体からなる初期個体集団の生成、 $k = 1$ とする。各個体は実数ベクトルの対 $(x_i, \sigma_i), \forall i \in \{1, 2, \dots, \mu\}$ 、ただし、 x_i は変数ベクトル、 σ_i はガウス分布の標準偏差ベクトルとする。

Step2: 個体集団の各個体 $(x_i, \sigma_i), \forall i \in \{1, 2, \dots, \mu\}$ に対して適合度を計算する。

Step3: 各個体 $(x_i, \sigma_i), i = 1, \dots, \mu$ は単一の子孫 (x'_i, σ'_i) を生成する. $j = 1, 2, \dots, n$ に対して

$$\sigma'_i(j) = \sigma_i(j) \exp\{\tau N(0, 1) + \tau N_j(0, 1)\} \quad (2.1)$$

$$x'_i(j) = x_i(j) + \sigma_i(j) N_j(0, 1) \quad (2.2)$$

ただし、 $x_i(i), x'_i(j), \sigma_i(j), \sigma'_i(j)$ はベクトル $x_i, x'_i, \sigma_i, \sigma'_i$ の j -成分を表わす. $N(0, 1)$ は標準化された 1 次元正規乱数で、その平均は 0、標準偏差は 1 となっていることを示す. $N_j(0, 1)$ は各 j 毎に発生する正規乱数を示し、 $\tau = (\sqrt{2\sqrt{n}})^{-1}, \tau = (\sqrt{2n})^{-1}$ とする.

Step4: 各子孫 $(x'_i, \sigma'_i), \forall i \in \{1, 2, \dots, \mu\}$ の適応度を計算する.

Step5: q トーナメント選択を実施し、 $(x_i, \sigma_i), (x'_i, \sigma'_i)$ の中から μ 個選択し、次世代の親とする.

Step6: 停止条件を満たせば停止、そうでなければ $k = k + 1$ として Step3 へ.

3 Cauchy 乱数を用いた進化的プログラミング (FEP)

CEP では Gauss 分布に従う正規乱数 $N(0, 1)$ を使用した突然変異を考えたのに対して、FEP では Cauchy 分布に従う乱数を使用する. 原点を平均とする 1 次元 Cauchy 分布の密度関数は

$$f(x, t) = \frac{1}{\pi} \frac{t}{x^2 + t^2}, -\infty \leq x \leq \infty, t > 0 \quad (3.1)$$

t はスケールパラメータとする. これに対応する Cauchy 分布関数 $F(x, t)$ は次のようになる.

$$F(x, t) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{t}\right) \quad (3.2)$$

したがって、 $[0, 1]$ 区間の一様乱数を y とすると、

$$x = t \cdot \tan\left\{\pi\left(y - \frac{1}{2}\right)\right\} \quad (3.3)$$

となり、この乱数を $C(0, t)$ で表わす. このスケールパラメータ $t = 1$ とした乱数が $C(0, 1)$ であり、次のようになる.

$$C(0, 1) = \tan\left\{\pi\left(a - \frac{1}{2}\right)\right\}, 0 \leq a \leq 1 \quad (3.4)$$

FEP の処理アルゴリズムは CEP の Step3 における (2.2) 式の代りに

$$x'_i(j) = x_i(j) + \sigma_i(j) C_j(0, 1) \quad (3.5)$$

としたものである.

4 複合指数乱数を用いた進化的プログラミング (EEP)

EEP では Double Exponential 分布に従う乱数 $E(0, \lambda)$ を用いた突然変異を考慮するものである. パラメータ λ の複合指数分布の 1 次元の確率密度関数 $f(x)$ は

$$f(x) = \frac{\lambda}{2} \exp\{-\lambda|x|\}, -\infty \leq x \leq \infty, \lambda > 0 \quad (4.1)$$

として与えられる. したがって、この分布における平均 $\bar{x} = 0$, 分散 $\text{var}(x) = \frac{2}{\lambda^2}$ となる. このことから、この分布の分散は λ が小さければ大きく、 λ が大きければ小さくなり、 λ により制御可能であることを示している. $f(x)$ に対応した分布関数 $F(x)$ は

$$F(x) = \begin{cases} \frac{1}{2} \exp[\lambda x] & x \leq 0 \\ 1 - \frac{1}{2} \exp[-\lambda x] & x > 0 \end{cases} \quad (4.2)$$

として与えられる。したがって、 $[0,1]$ 区間での一様乱数を y とすると、

$$x = \begin{cases} \frac{1}{\lambda} \ln(2y) & y \leq \frac{1}{2} \\ -\frac{1}{\lambda} \ln\{2(1-y)\} & y > \frac{1}{2} \end{cases} \quad (4.3)$$

で発生させる乱数 x は $E(0, \lambda)$ で表わされる。このことから、 $E(0, \lambda) = \frac{1}{\lambda} E(0, 1)$ となり $E(0, 1)$ を発生させることにより $E(0, \lambda)$ を計算できる。EEP では CEP の Step3 における (2.2) 式の代りに

$$x'_i(j) = x_i(j) + \sigma_i(j) E_j(0, \lambda) \quad (4.4)$$

としたものである。

5 EEP における分布パラメータ λ の収束特性に及ぼす効果

EEP ではパラメータ λ の複合指数乱数による突然変異により解を進化させるために、良好な収束特性を得るための有効な λ の値を何に決定するかが重要な研究課題である。定性的には進化の初期段階では分散を大きくして解の多様性を高め、広域探索により解を進化させ、末期段階では局所探索とすることが望ましいわけである。

このような適正な λ の値を決定するために、これまでに実施した実験は次の通り。

(1) $\lambda = 1$ とした場合の EEP の収束検討 [3]

世代数 GEN=1500、 $\mu = 100$ 、 $q = 10$ 、性能 50runs の平均

結果：N(0,1)、C(0,1)、E(0,1) mutation で E(0,1) が *best*

(2) $\lambda = 1, 0.5, 0.05$ とした場合の EEP の収束検討 [5]

GEN=2000、 $\mu = 100$ 、 $q = 10$ 、性能 50runs の平均

結果： $\lambda = 0.05$ が *best*

(3) λ の初期値 $\lambda_1 = 0.05$ とした場合の 1-time switching EEP の検討 [6]

($\lambda_1(g)\lambda_2$ においては、 λ の初期値を g 世代で切換えて λ_2 とすることを示す)

$\lambda : 0.05(500)1$

0.05(1000)1

0.05(1500)1

0.05(2000)1

$\mu = 100$ 、 $q = 10$ 、GEN=1500、性能は 100runs の平均

結果：問題により switching 世代の早いもの、遅いものが望ましい場合があり、いずれも switching 効果が認められた。

その他、

0.1(2000)1, 0.01(2000)1

0.1(2000)5, 0.01(2000)5

0.1(2000)10, 0.01(2000)10

については 2000 世代で $\lambda_2 = 10$ とする方が良好な結果が得られた。なお、問題によって $\lambda = 0.01, 0.0005, 0.00001$ と固定した方が良好な収束特性を示す場合もあった。

(4) 4-times switching EEP の検討 [7]

multi-times switching EEP として、

EEP1 : 0.1(1000)0.5(2000)1.0(3000)5.0(4000)10.0

EEP2 : 0.3(1000)1.0(2000)5.0(3000)10.0(4000)20.0

EEP3 : 0.5(1000)1.0(2000)5.0(3000)10.0(4000)20.0

EEP4 : 1.0(1000)3.0(2000)5.0(3000)10.0(4000)25.0

$\mu = 100$ 、 $q = 10$ 、GEN=5000、性能は 100runs の平均.

結果 : *uni-modal* 関数に対しては EEP が FEP より良好.

multi-modal 関数に対しては 2 個だけ EEP が良い.

残り 4 個は FEP の方が良い.

さらに、4-times switching で switching 効果がよく認められる関数に対して 6-times switching を施したところ精度の高い解が求められた. さらに、*multi-modal* 関数については初期値 λ を 0.1 より小さい値 (*ex* : 0.05) にすると 3~4 個の問題に対して改善が認められ、multi-switching EEP が有効であることが認められた.

6 数値実験

6.1 今回の数値実験の目的

EEP ではもっとも単純な手順として $E(0,1)$ 乱数を使用した突然変異による進化法でも、これまでに提案された中で有効とされている FEP に匹敵するかあるいはそれ以上の収束特性を示し得ることが明らかにされた. しかしながら、CEP や FEP と根本的に異なる点は収束状況に応じて進化の探索中を有効に制御することで収束の効率化を計らうとするものである. これがためには前回までの報告で λ の値を変動させる multi-switching EEP の有効さも示してきた. しかしながら、multi-switching による λ の値を決定する場合、switching point をどこにすべきか、あるいはその大きさはどの程度にすべきか、さらに何回の switching を考慮すべきか等々の問題が生じてくる. 勿論、これまでの実験からはある範囲内のものであれば λ の fixed value のみよりは効率的であることは判明している. 上記状況に鑑み、今回は初期値 λ_1 と最終値 λ_2 を与えることで世代と共に線形に変動する EEP につき検討するものである.

6.2 適用問題

EEP の収束特性を検討するために、今回の実験で対象とした問題はこの分野でよく知られているベンチマーク問題で、表 1 に与えている関数最適化問題である. この中の $f_1 \sim f_6$ は単一の局所解をもつ *uni-modal* 関数であり、 $f_7 \sim f_{11}$ は複数の局所解をもつ *multi-modal* 関数である.

6.3 パラメータ設定

数値実験におけるパラメータは次のように設定した.

- (1) 個体集団の大きさ ; $\mu = 100$
- (2) トーナメントサイズ ; $q = 10$
- (3) 計算世代数 ; GEN=5000
- (4) λ の初期値 ; $\lambda_1 = 0.05, 0.0001$
- (5) λ の最終値 ; $\lambda_2 = 10$
- (6) g 世代で λ ; $\lambda = \frac{(\lambda_2 - \lambda_1)}{GEN} \cdot g + \lambda_1$
- (7) 戦略パラメータの初期値 ; $\sigma_i(j)$; $[0,1]$ 区間の一様乱数

6.4 実験実施要領

本実験は複合指数分布のパラメータ λ を線形に変動させる EEP の収束特性につき検討するものであるが、同様に、CEP および FEP についても 5000 世代までの特性につき記述する。これは EEP との性能比較という目的もさることながら、CEP や FEP についての性能記述が多くの関連文献において 1500 世代か 2000 世代まで位のものであり、5000 世代までのものは殆んどないからである。例えば、FEP の方が CEP よりは一般により特性を示すものとされているが、このことは 1500 世代あるいは 2000 世代までの特性であり、それ以降のものについては殆んど言及されていない。しかし本実験では 2000 世代以降においては CEP の方が良好な結果を齎すことも明確になっている。これらの結果は基本的に Gauss 分布と Cauchy 分布の特徴から類推しうるものである。

本実験での収束特性は各世代毎に各個体の関数値を 100runs の平均値として評価値とし、各世代毎にプロットして収束特性を表わすものとする。ここでは関数最適化問題を取り扱うので関数値が小さい程望ましい状態といえる。進化の過程で、戦略パラメータ $\sigma(j)$ は新しい解を決定する場合の変異巾とに関与するものであるが、 $|\sigma(j)| = 0$ は進化の停止を意味する。この状況を避けるために、 $|\sigma(j)|$ の下限を ϵ とし、 $|\sigma(j)| < \epsilon$ であれば強制的に $|\sigma(j)| = \epsilon$ として進化を続行させる。標準的には $\epsilon = 10^{-2}$ とするが、状況によっては各問題に適した下限値についても検討する。

6.5 実験結果

(1) $\lambda_1 = 0.0001, \lambda_2 = 10, \epsilon = 10^{-2}$

図 1~11 に各 EP の収束特性を示す。

f_1 に対して、1000 世代までは EEP、FEP、CEP の順に良いが、1000 世代以降においては CEP が FEP に逆転している。 f_2 に対していずれも同じような収束傾向を示すが EEP、CEP、FEP の順に良い。 f_3 に対して、全区間で EEP、CEP、FEP の順となっている。 f_4 に対して、EEP は最良であるが、2000 世代で CEP、FEP とが逆転している。 f_5 に対して、EEP は全区間で最良であり、1500 世代で CEP と FEP とが逆転している。 f_6 に対しては全区間で EEP が最良で、EEP、CEP、FEP の順になっている。 f_7 に対して、全区間で FEP が最良で、FEP、EEP、CEP の順になっている。 f_8 に対して、1000 世代までは CEP が最良であるが、それ以降において EEP、CEP、FEP の順となっている。 f_9 に対して、当初 FEP が最良であるが 500 世代以降において EEP、FEP、CEP の順になっている。 f_{10} に対して、当初 FEP が最良であるが 700 世代以降において EEP、CEP、FEP の順になっている。 f_{11} に対して、EEP、FEP、CEP の順になっている。

$f_1 \sim f_{11}$ の問題中、最終的に最良解となっている EP は EEP ; 10, FEP ; 1, CEP ; 0

$f_1 \sim f_6$ の *uni-modal* 関数に関して、EEP ; $\frac{6}{6}$ で全ての問題に対して最良解を得ることができた。CEP と FEP では最終的に CEP の方が良好な解となっているが、初期段階では FEP の方がいずれもよい収束度を示している。 $f_7 \sim f_{11}$ の *multi-modal* 関数に対しては EEP は $\frac{4}{5}$ で最良解を得ているが、CEP と FEP では $\frac{3}{5}$ で FEP の方が良好な収束を示している。以上の事から $\epsilon = 10^{-2}$ とした場合については全面的に EEP が有効な EP 手法であることが認められる。

(2) $\lambda_1 = 0.001, \lambda_2 = 10, \epsilon = 10^{-4}$

5000 世代での最終関数値を表 2 に示す。

この表から、 $f_1 \sim f_6$ の *uni-modal* 関数に対しては、6 問中 5000 世代で最良値を示したものは EEP ; 4, FEP ; 2 となり、EEP の方が良好な収束特性を示している。

$f_7 \sim f_{11}$ は *multi-modal* 関数であるが、これについては 5 問中、5000 世代で最良値を示したものは FEP ; 3, EEP ; 1, CEP ; 1 となり、FEP が良好な収束特性を示す。また f_1 や f_9 に対しては最終値において $\epsilon = 10^{-2}$ とした場合に比し 10^{-4} 程解が改良されている。なお、FEP と CEP については

uni-modal 関数に対して CEP が良いが、*multi-modal* 関数に対して FEP が有効であることを示している。 $f_1 \sim f_{11}$ を全問題に対しては EEP は *uni-modal* 関数については良いが、*multi-modal* 関数に対しては FEP の方が良好な収束特性を示す。

(3) $\lambda_1 = 0.001, \lambda_2 = 10, \epsilon =$ 最適下限

表 3 に 5000 世代での関数値を示す。

この表から、 $f_1 \sim f_6$ の *uni-modal* 関数に対しては最良解は EEP : 5, FEP : 1 となっており、EEP の良好さが認められる。 $f_7 \sim f_{11}$ の *multi-modal* 関数に対して最良解となった数は EEP : 3, FEP : 2 となっている。これらの事から、 $f_1 \sim f_{11}$ の全問題を通して、最終的に良好な解を得たのは EEP : 8, FEP : 3 となり、 ϵ を問題に適合させれば、EEP 手法は EP としてかなり有効なアルゴリズムと期待できる。

7 結論

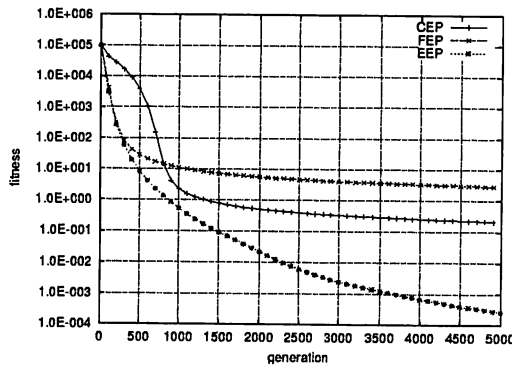
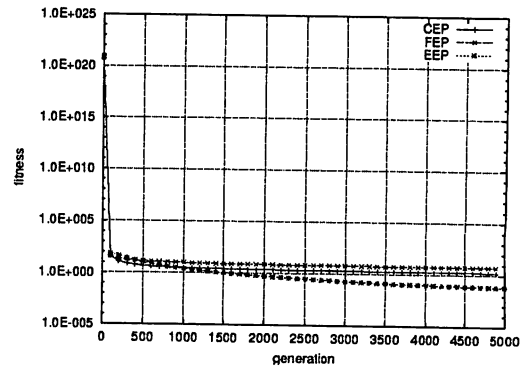
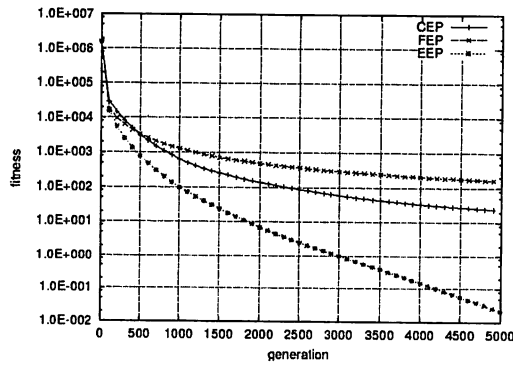
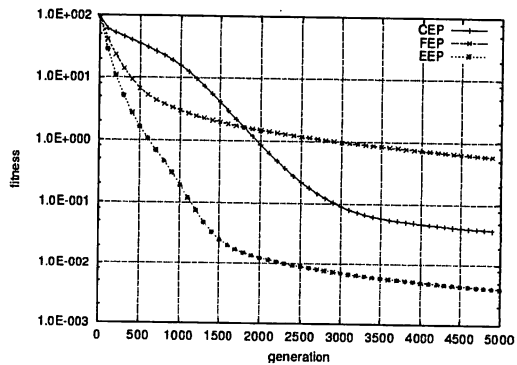
EEP での複合指数分布のパラメータを λ の値を世代と共に線形に変動させることで効率的な EP を実現させることができた。さらに今回の実験では比較のために、CEP、FEP の性能特性についても検討したが、実験結果により *uni-modal* 関数に対しては全て CEP の方が、*multi-modal* 関数に対しては FEP の方が優れた特性を示した。今回使用した EEP のパラメータ設定としては戦略パラメータ σ の下限 ϵ を 10^{-2} としているが、全般的にみて与えられた問題の 8 割以上において最良解を得ていることから EEP が極めて有効な EP 手法であるといえよう。なお、 ϵ としては $10^{-2} \sim 10^{-6}$ 程度 λ_1, λ_2 にしても $\lambda_1 = 0.1 \sim 0.0001, \lambda_2 = 10 \sim 100$ 程度の設定で割合有効な EEP を実現できるものと考えられる。

参考文献

- [1] T.Back and H.-P.Schwefel, "An overview of evolutionary algorithms for parameter optimization", *Evolutionary Computation*, 1(1):1-23, 1993.
- [2] X. Yao and Y. Liu, "Fast Evolutionary Programming," *Proc. of the 5th Annual Conference on Evolutionary Programming*, MIT Press, pp.451-460, 1996.
- [3] K.Kohmoto, H.Narihisa and K.Katayama, "Evolutionary programming using exponential mutation", *Proc. of The 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI2002)*, vol. XI, Computer Science II, pp.405-410, 2002.
- [4] H.Narihisa, K.Kohmoto, and K.Katayama, "Evolutionary Programming with Double Exponential Probability Distribution", *Proc. of the Second IASTED International Conference on Artificial Intelligence And Applications*, September 9-12, Spain, 358-363, 2002.
- [5] K.Kohmoto, H.Narihisa, and K.Katayama, "Performance of Evolutionary Programming Using Exponential Mutation", *Proc. of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL'02)* vol. 2, pp.454-458, 2002.
- [6] H.Narihisa, K.Kohmoto, T.Kumon, and K.Katayama, "Performance of Exponential Evolutionary Programming", *Proc. of IASTED International Conference on Artificial Intelligence and Soft Computing (ASC2003)*, pp.243-248, 2003.
- [7] H.Narihisa, K.Kohmoto, M.Tsuda, and K.Katayama, "CONVERGENCE CHARACTERISTICS OF EXPONENTIAL EVOLUTIONARY PROGRAMMING", *Proc. of the 8th IASTED International Conference on Artificial Intelligence and Soft Computing (ASC2004)*, pp.426-431, 2004.

表 1: ベンチマーク問題

Test function	S	f_{min}
$f_1(x) = \sum_{i=1}^{30} x_i^2$	$[-100, 100]^{30}$	0
$f_2(x) = \sum_{i=1}^{30} x_i + \prod_{i=1}^{30} x_i $	$[-10, 10]^{30}$	0
$f_3(x) = \sum_{i=1}^{30} (\sum_{j=1}^i x_j)^2$	$[-100, 100]^{30}$	0
$f_4(x) = \max x_i , 1 \leq i \leq 30$	$[-100, 100]$	0
$f_5(x) = \sum_{i=1}^{29} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-100, 100]^{30}$	0
$f_6(x) = \sum_{i=1}^{30} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]^{30}$	0
$f_7(x) = -\sum_{i=1}^{30} (x_i \sin(\sqrt{ x_i }))$	$[-500, 500]^{30}$	-12569.5
$f_8(x) = \sum_{i=1}^{30} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^{30}$	0
$f_9(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^{30} \cos 2\pi x_i\right) + 20 + e$	$[-32, 32]^{30}$	0
$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^{30}$	0
$f_{11}(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1),$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i \leq -a \end{cases}$	$[-50, 50]^{30}$	0

図 1: f_1 の収束特性図 2: f_2 の収束特性図 3: f_3 の収束特性図 4: f_4 の収束特性

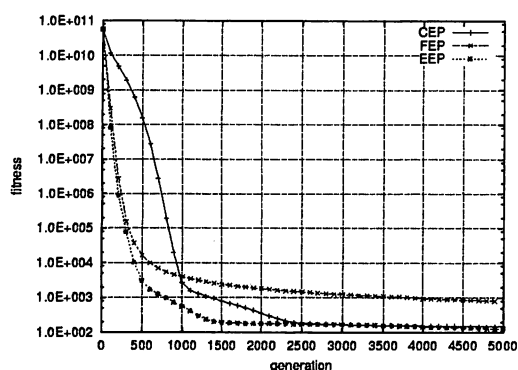
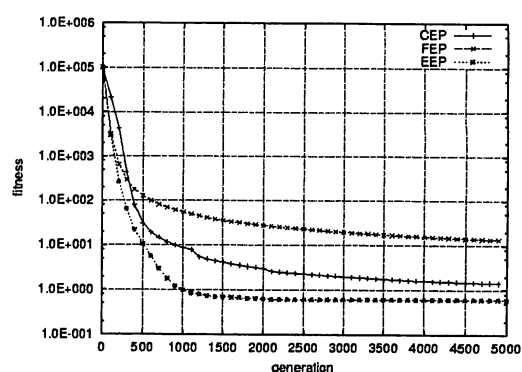
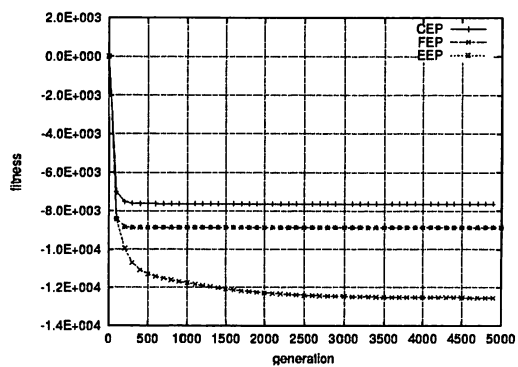
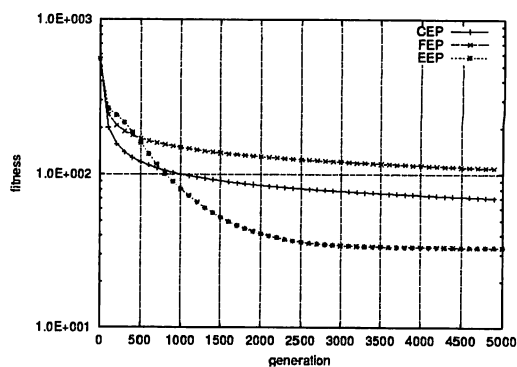
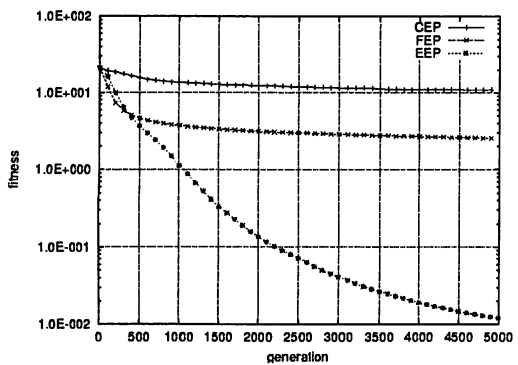
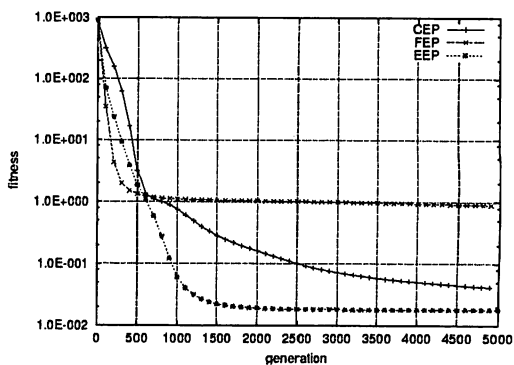
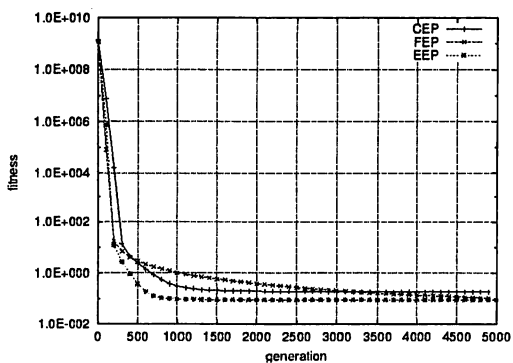
図 5: f_5 の収束特性図 6: f_6 の収束特性図 7: f_7 の収束特性図 8: f_8 の収束特性図 9: f_9 の収束特性図 10: f_{10} の収束特性図 11: f_{11} の収束特性

表 2: 最終世代での適応度 ($\epsilon = 10^{-2}$)

f	CEP	FEP	EEP
f_1	$2.11366e^{-1}$	2.80764	$2.57091e^{-4}$
f_2	1.71999	5.59973	$6.91784e^{-2}$
f_3	$2.30155e$	$1.58996e^2$	$2.14424e^{-2}$
f_4	$3.83286e^{-2}$	$5.76661e^{-1}$	$3.89591e^{-3}$
f_5	$1.49840e^2$	$7.78664e^2$	$1.26704e^2$
f_6	1.42460	1.31743e	$6.10000e^{-1}$
f_7	$-7.62757e^3$	$-1.25424e^4$	$-8.8608e^3$
f_8	$6.96276e$	$1.09076e^2$	3.33348e
f_9	1.08810e	2.53202	$1.19992e^{-2}$
f_{10}	$4.10545e^{-2}$	$8.89725e^{-1}$	$1.81654e^{-2}$
f_{11}	$1.89849e^{-1}$	$1.06290e^{-1}$	$9.33051e^{-2}$

表 3: 最終世代での適応度 ($\epsilon = 10^{-4}$)

f	CEP	FEP	EEP
f_1	$1.40355e^{-4}$	$1.15573e^{-1}$	$3.01536e^{-8}$
f_2	$4.39687e^{-2}$	$9.58763e^{-1}$	$7.35000e^{-4}$
f_3	6.16582	7.66093e	$5.91940e^{-1}$
f_4	2.82547	$1.69291e^{-1}$	$3.95447e^{-1}$
f_5	$1.01883e^2$	$1.65745e^2$	8.27275e
f_6	$5.76040e^1$	$9.71800e^{-1}$	1.23500e
f_7	$-7.63698e^3$	$-1.2561e^4$	$-8.84030e$
f_8	$1.250948e$	$2.76834e$	2.30830e
f_9	2.03914	$3.51060e^{-1}$	$1.23992e^{-4}$
f_{10}	$2.23479e^{-1}$	$8.25179e^{-2}$	$1.80516e^{-1}$
f_{11}	$1.95496e^{-1}$	$4.86307e^{-3}$	$8.72115e^{-2}$

表 4: 最終世代での適応度 ($\epsilon = \text{最適値}$)

f	CEP	FEP	EEP	ϵ
f_1	$1.83253e^{-6}$	$6.35955e^{-4}$	$7.00000e^{-14}$	10^{-8}
f_2	$1.19587e^{-3}$	$5.40637e^{-3}$	$1.13089e^{-7}$	10^{-8}
f_3	$2.30155e$	$1.58996e^2$	$2.14424e^{-2}$	10^{-2}
f_4	$3.83286e^{-2}$	$5.76661e^{-1}$	$3.89591e^{-3}$	10^{-2}
f_5	$1.04217e^2$	$1.2490e^2$	8.21130e	10^{-6}
f_6	$5.76040e^1$	$9.71800e^{-1}$	1.23500e	10^{-4}
f_7	$-7.53590e^3$	$-1.2532e^4$	$-8.71936e^3$	10^{-8}
f_8	$3.34505e$	$2.08208e^{-1}$	1.83968e	10^{-8}
f_9	2.03914	$3.51060e^{-1}$	$1.23992e^{-4}$	10^{-4}
f_{10}	$4.10545e^{-2}$	$8.89725e^{-1}$	$1.81654e^{-2}$	10^{-2}
f_{11}	$1.89849e^{-1}$	$1.06290e^{-1}$	$9.33051e^{-2}$	10^{-2}

Performance of Exponential Evolutionary Programming with Linear Varying Parameter

Hiroyuki Narihisa, Takahiro Taniguchi, Michiaki Tsuda* and Kengo Katayama

*Department of Information and Computer Engineering,
Faculty of Engineering,*

** Graduate School of Engineering.
Okayama University of Science*

1-1 Ridai-cho, Okayama, 700-0005, Japan

(Received September 28, 2004; accepted November 5, 2004)

The term evolutionary computing refers to the study of the foundation and applications of certain heuristic techniques and based on the principles of natural evolution; thus the aim of designing evolutionary algorithms is to mimic some of the processes taking place in natural evolution. These algorithms are classified into three main categories, depending more on historical development than on major functional techniques. In fact, their biological basis is essentially the same.

These algorithms contain evolutionary programming (EP), evolutionary strategies (ES), genetic algorithms (GA) and genetic programming (GP).

An evolutionary algorithm (EA) is an iterative and stochastic process that operates on a set of individuals (called population). Each individual represents a potential solution to the problem being solved. Initially, the population is randomly generated. Every individual in the population is assigned, by means of a fitness function, a measure of its goodness with respect to the problem under consideration, which guides the search.

EP was originally proposed by Fogel as an approach to artificial intelligence in 1966. Since late 1980's, EP was also applied to various combinatorial and numerical optimization problems. Standard EP uses Gaussian mutation which is based on Gaussian random number, and referred to CEP. In order to improve EP convergence, various mutations have been proposed. Among them, FEP algorithm which was based on Cauchy mutation was considered to be the most efficient algorithm.

In 2002, we proposed a new efficient EP with double exponential mutation as referred to EEP.

In this paper, we present a new EEP with linear varying parameter of double exponential distribution. The experimental results show that EEP is superior to CEP and FEP on applying to function optimization problems.