岡山理科大学紀要 第38号 A pp. 113-122(2002)

複合指数分布を用いた進化的プログラミング

公文隆・河本敬子*・成久洋之**

岡山理科大学大学院工学研究科情報工学専攻

*岡山理科大学大学院システム工学専攻

**岡山理科大学 工学部 情報工学科

(2002年11月1日　受理)

1　はじめに

複雑で難解な問題を解く場合、解空間が小さいものであれば、古典的なしらみつぶしの方法で最適解を求めることが出来る。しかしながら大きな解空間に対しては特殊な人工知能的な技法が有望視されている。この人工知能的な技法として、最近、進化の原理を模倣した計算法として進化的アルゴリズムが注目されている。この進化的アルゴリズムの中には遺伝的アルゴリズム (Genetic Algorithm,GA) や進化的プログラミング (Evolutionary Programming,EP) などがある。EP は人工知能に対する最初のアプローチとして、L.J. Forgel により提案され、後に、D.B. Forgel[2] によって数値的ならびに組み合わせ最適化手法として開発されたものである。以降、EP は実数値関数の最適化 [4] やその他の現実問題の解法等に多く適用されるに至っている。GA が交叉演算を強張するのと対照的に EP では突然変異演算が主要なものであり、標準的な EP では各個体は Gaussian 突然変異により子孫を生成し、親と子孫の個体群の中から良質の個体が選択されて次世代の親となる。この Gaussian 分布を使用した突然変異の方法を CEP と呼ぶがこの CEP の欠点は収束が遅いということである。この収束特性を改善するために Yao ら [7] は 1996 年に Cauchy 分布を使用した突然変異により高速な EP 手法を提案し、これを FEP と呼んでいる。Cauchy 分布は Gaussian 分布と異なり無限大の二次モーメントを持つため Gaussian 分布よりも長い尾を持っている。このため、突然変異の結果として生起する子孫は親と全く異なった解を発生することになり、このことが CEP より FEP の方が有効であるとされている。その後 CEP の方が適用範囲によっては FEP より良い特性を持つことが報告され、CEP と FEP の双方をある割合で使用した混合方式等も提案されているが、基本的には Gaussian 分布か Cauchy 分布によるものが EP の主流といえる。これ以外の分布を使用したものとして Lee と Song[5] は 1999 年に Levy 分布を用いた EP を提案しているが、この分布はパラメータの取り扱いが複雑で乱数発生における安定性に欠ける面があり、この分布のあるパラメータ値が Gaussian 分布と Cauchy 分布に対応するものであり、結論的には両分布の混合処理法とみなすことが出来る。

Narihisa ら [9] は 2002 年に複合指数分布による突然変異を用いた EP を提案し、これを EEP と呼んでいる。これは複合指数分布に従う乱数を使用するもので Gaussian 分布より長い尾を持ち、しかも平均値を中心とした領域での分布確率も大きく取れるような制御可能な分布で FEP よりもその収束確率も大きく取れるようなものとして注目されている。しかしながら、EEP におけるパラメータ λ の決定法については今のところ経験則とする以外に決定する方法は存在しない。

本論文は EEP におけるパラメータ λ の収束特性に与える効果につき検討したもので CEP や FEP と合わせてよく知られたベンチマーク問題に適用した結果につきまとめたものである。この実験結果からは適応対象問題によって収束特性は異なるがパラメータ λ の有効な値のとり方で少なくとも従来の CEP や FEP よりはその有効性を期待し得るものであると考える。

2 標準進化的プログラミング (CEP)

Forgel[1] や Back ら [3] は適応的突然変異を用いた CEP はそれを使用しない場合よりも通常良い収束特性を示すことを報告しているので、本論文では適応的突然変異を持つ CEP につき検討する。

2.1 個体表現

EP における個体集団 (Population) を構成する個体として、個体 $X_i(i=1,2,\cdots,\mu)$ を n 次元実数値ベクトルと n 次元戦略パラメータ η_i を用いて、以下のように定義する。

$$X_i = [x_i, \eta_i] \tag{1}$$

$$x_i = (x_i(1), \cdots, x_i(j), \cdots, x_i(n)) \tag{2}$$

$$\eta_i = (\eta_i(1), \cdots, \eta_i(j), \cdots, \eta_i(n)) \tag{3}$$

ここで $j=1,2,\cdots,n$ であり、$x_i(j), \eta_i(j)$ はそれぞれベクトル x_i, η_i の第 j 成分である。

2.2 CEP による突然変異について

各個体 (x_i, η_i) について、正規分布に従った突然変異による子孫 $(\acute{x_i}, \acute{\eta_i})$ の生成式は

$$\acute{x_i}(j) = x_i(j) + \eta_i(j) N_j(0,1) \tag{4}$$

$$\acute{\eta_i}(j) = \eta_i(j) \exp(\acute{\tau} N(0,1) + \tau N_j(0,1)) \tag{5}$$

ここで、$x_i(j), \acute{x_i}(j), \eta_i(j), \acute{\eta_i}(j)$ は、それぞれ ベクトルの第 j 成分を表す。
ただし、$j=1,2,\cdots,n$。
$N(0,1)$ は平均 0、標準偏差 1 の正規乱数を、 は各 j ごとに発生させた正規乱数を示す。
また $\tau = (\sqrt{2\sqrt{n}})^{-1}, \acute{\tau} = (\sqrt{2n})^{-1}$ とする。
標準正規乱数 $N(0,1)$ の発生方法は、一般に n 個の一様乱数 $x_1, x_2, \cdots, x_n$ に対して

$$N(0,1) = \frac{\frac{1}{n}\sum_{j=1}^{n} x_j - \frac{1}{2}}{\sqrt{\frac{1}{12n}}} \tag{6}$$

この式を計算すれば、n が大きい時、$N(0,1)$ は中心極限定理より標準偏差に従う確率変数とみなすことができる。n が大きいほど正規分布への近似度はよくなるが、必要な一様乱数の個数が多くなり、その発生時間が長くなる。だが $n=12$ の場合、近似度はきわめて高くなり平方根の計算をしないですむ。本研究では、$n=12$ とし、以下のように簡略して標準正規乱数を発生させた。

$$N(0,1) = \sum_{j=1}^{12} x_j - 6.0 \tag{7}$$

ただし、x_j は $0 \leq x_j \leq 1 (j=1,2,\cdots,12)$ の一様乱数とする。

2.3 CEP の処理アルゴリズム

Step1. μ 個の個体からなる初期集団の生成。各個体は n 次元の実数ベクトルは $X_i = [x_i, \eta_i]$

Step2. 各親 (x_i, η_i) は単一の子孫 $(\acute{x_i}, \acute{\eta_i})$ を生成。

$$\acute{x_i}(j) = x_i(j) + \eta_i(j)N_j(0,1) \tag{8}$$

$$\acute{\sigma_i}(j) = \eta_i(j)exp(\acute{\tau}N(0,1) + \tau N_j(0,1)) \tag{9}$$

ただし、$j = 1, 2, \cdots, \mu$。

Step3. 全ての個体についての適応度関数値を計算。$f(x_i), f(\acute{x_i}), i = 1, 2, \cdots, \mu$

Step4. 全ての親と子孫から q 個選ぶ。

Step5. 適応度と比較し、各 2μ 個の個体が q 回のうちの各々に対して悪くなければ 1 回につき 1 の勝ち点を得る。

Step6. 2μ 個の個体につき勝ち点の多い順に μ 個選択。

Step7. 停止条件を満足するまで Step2 から Step6 を繰り返す。

3 高速進化的プログラミング (FEP)

EP における Cauchy 突然変異オペレータの効果を検討するために突然変異オペレータを除いて CEP の処理手順はそのままに使用することにする。平均を原点とする 1 次元 Cauchy 密度関数は次のように与えられる。

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

ただし、$t > 0$ はスケールパラメータとする。これに対応する分布関数は次のとおり。

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} arctan\left(\frac{x}{t}\right) \tag{11}$$

$f_t(x)$ の形は Gaussian 密度関数と似ているが x 軸に近づくのが極めて遅くその期待値は存在しない。結果として、Cauchy 分布の分散は無限大となる。FEP の処理アルゴリズムは大部分が CEP のそれと同じであるが式 (8) の変わりに次の式 (12) を使用する点が相異している。

$$\acute{x_i}(j) = x_i(j) + \eta_i(j)C_j(0,1) \tag{12}$$

ただし、$C_j(0,1)$ は中心が $x = 0$ でスケールパラメータ t が $t = 1$ を持つ Cauchy 乱数とする。Cauchy 乱数の発生については [0,1] 区間の一様乱数を y とすると (11) 式に対応して

$$x = t \cdot tan\left\{\pi\left(y - \frac{1}{2}\right)\right\} \tag{13}$$

となり、これを C(0,t) で表す。

$$C(0,1) = \left\{tan\left(\pi\left(y - \frac{1}{2}\right)\right)\right\} \tag{14}$$

4 複合指数分布を用いた進化的プログラミング (EEP)

EEP においても FEP の場合と同様に、CEP との違いをを最小限に止めた処理手順とし、乱数分布による収束特性の違いを検討する。パラメータ λ における 1 次元の複合指数分布の確率密度関数 $f(x)$ は

$$f(x) = \frac{\lambda}{2} exp\{-\lambda|x|\}, -\infty \leq x \leq \infty, \lambda > 0 \tag{15}$$

として与えられる。これに対応する分布関数は

$$F(x) = \begin{cases} \frac{1}{2} exp[\lambda x] & x \leq 0 \\ 1 - \frac{1}{2} exp[-\lambda x] & x > 0 \end{cases} \tag{16}$$

となる。したがって、平均 $\bar{x} = 0$、分散 $var(x) = \frac{2}{\lambda^2}$ となる。この分布は明らかに原点に対して対称であり、その分散はパラメータ λ によって制御し得るものである。この乱数の発生方法としては [0,1] 区間の一様乱数 y に対して、

$$x = \begin{cases} \frac{1}{\lambda} ln(2y) & y \leq \frac{1}{2} \\ -\frac{1}{\lambda} ln\{2(1-y)\} & y > \frac{1}{2} \end{cases} \tag{17}$$

として与えられる。この乱数を $E(0,\lambda)$ と記し、これは平均が 0 で、パラメータ λ からなる乱数を意味する。このことから $E(0,\lambda) = \frac{1}{\lambda} E(0,1)$ となる。

複合指数乱数 $E(0,\lambda)$ は $E(0,1)$ を求めることにより簡単に計算される。EP において使用する乱数は、$N(0,1), C(0,1)$ および $E(0,\lambda)$ であるから、これらの分布の様子を図 1 に、λ を変動させたときの様子を図 2 に示す。Yao ら [8] によると $C(0,1)$ と $N(0,1)$ の分布特性により次のことを指摘している。

・FEP では $C(0,1)$ の分布における尾の長い (long flat tails) ことが親とかなり異なった子孫を発生しうる。
・この事実が局所解に落ち込むことによる収束の悪化を防ぎうる。
・一方、$C(0,1)$ では中心付近の丘の高さが低いことにより局所解近傍における解探索のための処理時間を少なくし結果として最適解近傍における良質な解の探索力 (fine-tuning ability) を弱体化している。

図 1 からもわかるように $E(0,1)$ は $C(0,1)$ と $N(0,1)$ の中間的な分布をしており、$E(0,\lambda)$ は図 2 をみると λ を小さくすれば分布における尾の拡がりを長くできるし、λ を大きくすれば中心付近の分布の高さを大きくすることもできる。すなわち、$E(0,\lambda) = \frac{1}{\lambda} E(0,1)$ の関係で分散を制御できる特徴を持っている。$E(0,\lambda), N(0,1)$ および $C(0,1)$ における分布状態をまとめたものが表 1 である。分布の拡がりについては $|x| > 5$ における確率は $C(0,1)$ で 12.52、$E(0,0.3)$ で 22.32 と大きくできるが $|x| > 10$ では $C(0,1)$ が 6.30 であるのに対し、$E(0,0.3)$ では 4.98 で $C(0,1)$ に比べて劣るが全体としては FEP 並みの探索能力を期待し得る。一方、局所近傍解の探索能力については $|x| \leq 1$ において、$N(0,1)$ は 68.26 に対して、$E(0,2)$ は 86.46、$E(0,5)$ については 99.32 となって EEP の探索能力の向上を期待しうる分布となっている。

表 1. 分布特性 (%)

	$\|x\| \leq 3$	$\|x\| \leq 2$	$\|x\| \leq 1$	$\|x\| > 3$	$\|x\| > 5$	$\|x\| > 10$
$N(0,1)$	99.70	95.44	68.26	0.30	0.00	0.00
$C(0,1)$	79.54	70.05	50.00	20.46	12.52	6.30
$E(0,1)$	95.02	86.46	63.21	4.98	0.68	0.00
$E(0,0.5)$	77.68	63.21	39.34	22.32	8.20	0.67
$E(0,0.3)$	59.34	45.12	25.92	40.66	22.32	4.98
$E(0,2)$	99.75	98.16	86.46	0.25	0.00	0.00
$E(0,5)$	99.99	99.99	99.32	0.01	0.00	0.00

5 数値実験

5.1 実験対象問題

対象問題としては表2に与えるようにこの分野でよく知られているベンチマーク問題10個とした。いずれも高次元関数であり、f_1 から f_5 は単一の局所解を持つ uni-modal 関数であり、f_6 から f_{10} は複数の局所解を持つ multi-modal 関数となっている。

5.2 パラメータ設定

実験におけるパラメータは以下のように設定した。

(1) 個体集段　$\mu = 100$
(2) トーナメントサイズ　$q = 10$
(3)EEP のパラメータ λ　EEP1$\cdots\lambda = 1.0$
EEP2$\cdots\lambda = 0.5$
EEP3$\cdots\lambda = 0.1$

表 2. ベンチマーク問題

対象関数	s	f_{min}
$f_1(x) = \sum_{i=1}^{30} x_i^2$	$[-100, 100]^{30}$	0
$f_2(x) = \sum_{i=1}^{30} \lvert x_i \rvert + \prod_{i=1}^{30}$	$[-10, 10]^{30}$	0
$f_3(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^{30}$	0
$f_4(x) = max_i\{\lvert x_i \rvert, 1 \leq i \leq 30\}$	$[-100, 100]^{30}$	0
$f_5(x) = \sum_{i=1}^{29} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	$[-30, 30]^{30}$	0
$f_6(x) = \sum_{i=1}^{30} ([x_i + 0.5])^2$	$[-100, 100]^{30}$	0
$f_7(x) = -\sum_{i=1}^{30} \left(x_i sin\left(\sqrt{\lvert x_i \rvert} \right) \right)$	$[-500, 500]^{30}$	-12569.5
$f_8(x) = \sum_{i=1}^{30} [x_i^2 - 10cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^{30}$	0
$f_9(x) = -20exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2} \right) - exp\left(\frac{1}{30}\sum_{i=1}^{30} cos2\pi x_i \right) + 20 + e$	$[-32, 32]^{30}$	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

5.3 実験実施要領

EEP の λ に対する収束効果検討のため λ の3個の値に対して50回実行した平均に対する収束特性を1500世代まで計算し、従来提案されている CEP,FEP と比較した。ただし、f_3 と f_4 については世代数を5000とした。

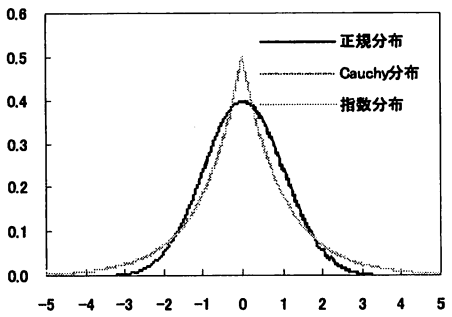


図1. 乱数分布

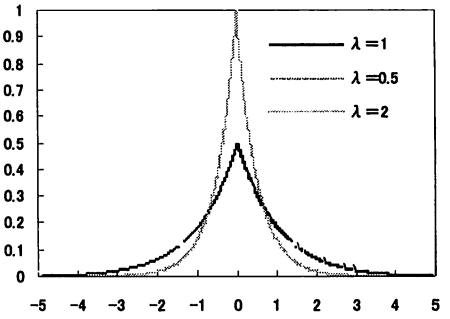
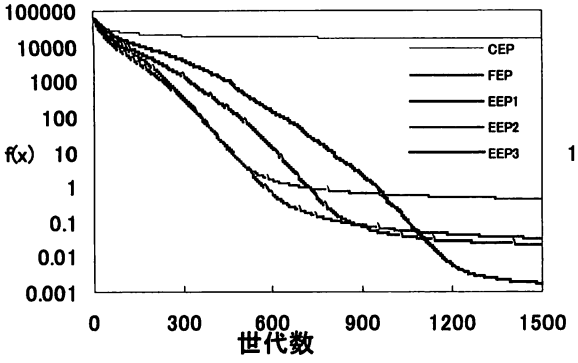
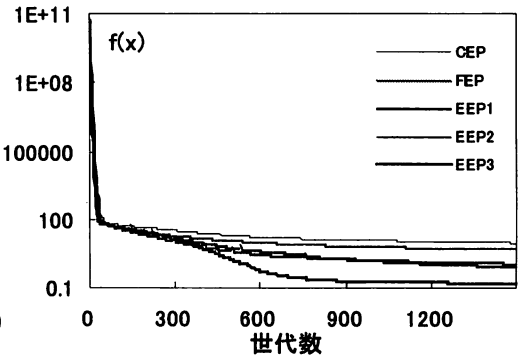


図2. λの値による乱数分布

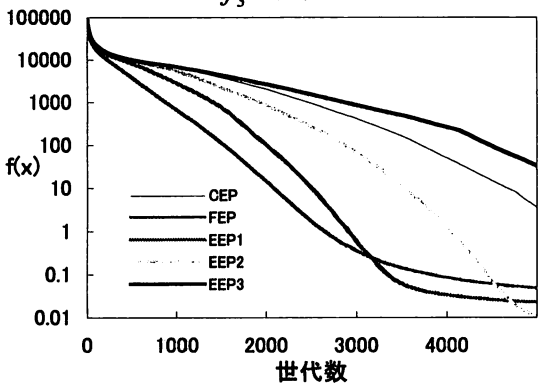
f_1の結果



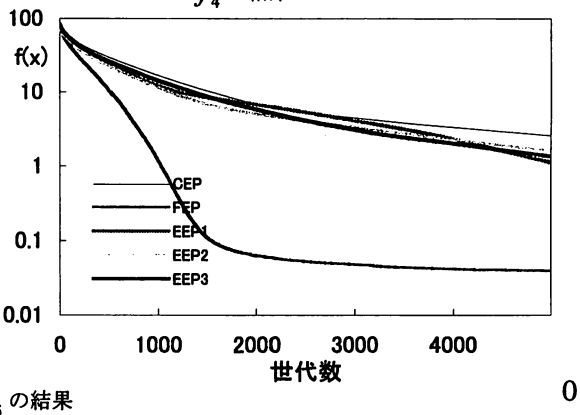
f_2の結果



f_3の結果



f_4の結果



f_5の結果

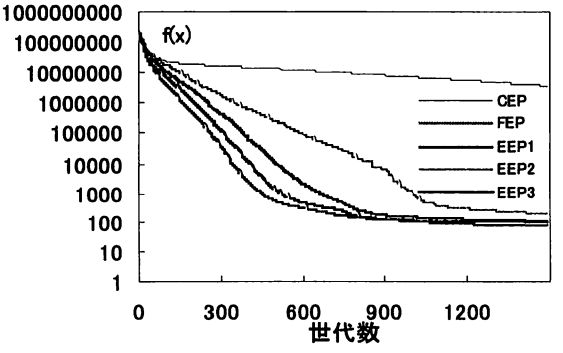


図3. uni-modal関数の収束特性

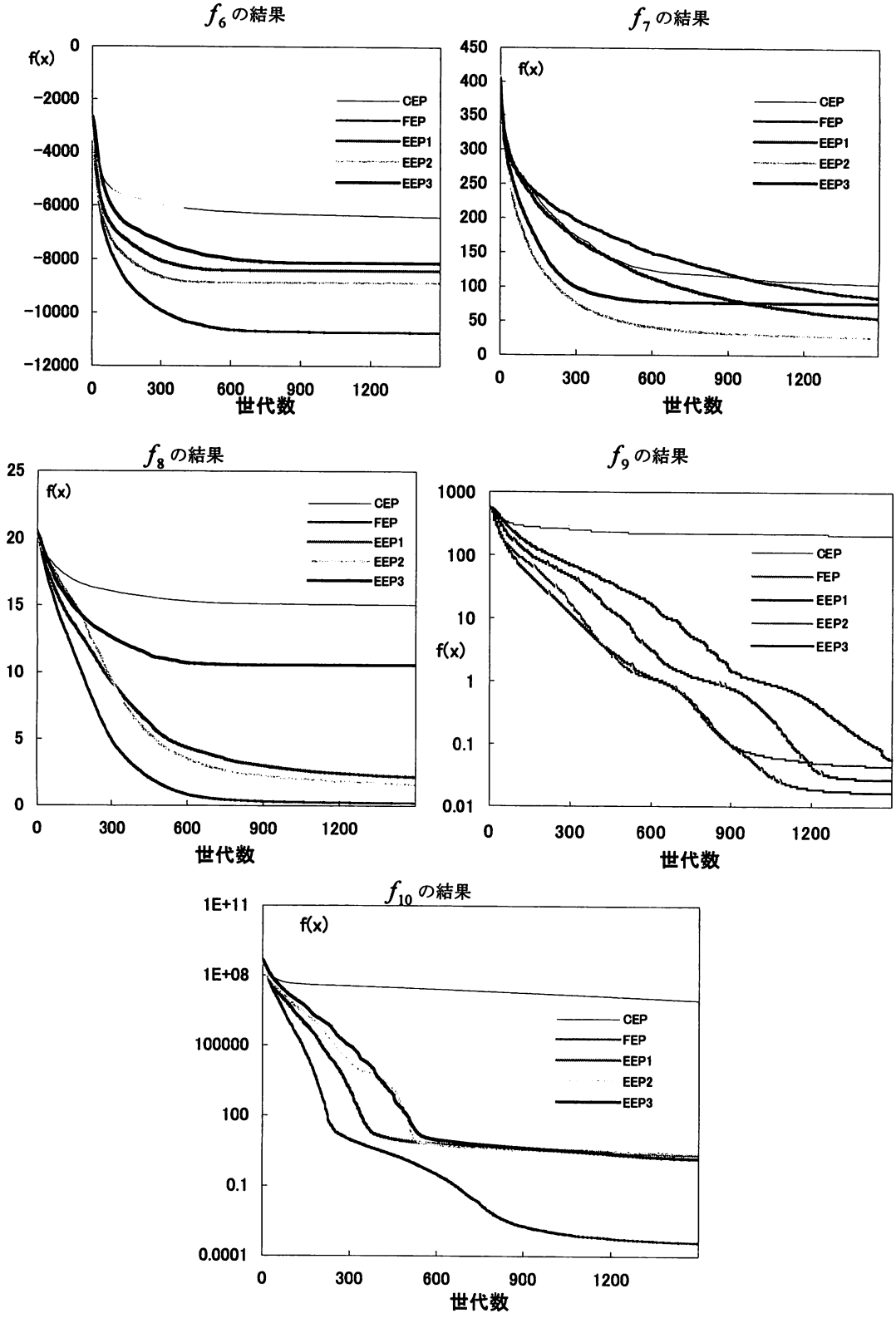


図４. multi-modal 関数の収束特性

5.4 実験結果

表2の $f_1 \sim f_{10}$ は問題によって特徴も異なるので収束特性も変わってくる。収束特性を図3,4に示す。f_1 においてはEEP3が1200世代を超えたあたりからFEPよりも優れた収束特性を示しておりEEP1もEEP3のように急激な収束状況は見られないが、徐々にFEPよりも良い特性を示している。EEP2に関してはCEPより優れた特性を示しているがFEPほどではない。f_2 に関してはEEP3が300世代を超えたあたりからFEPよりも優れた収束特性を示しておりEEP1もEEP3のように急激な収束状況は見られないが、徐々にFEPよりも良い特性を示している。EEP2に関してはCEPより優れた特性を示しているがFEPほどではない。f_3 についてはEEP1が、3000世代を超えたあたりから、EEP2は4000世代を超えたあたりからFEPよりも優れた特性を示しているが、f_1, f_2 において効果のあったEEP3が5000世代までではCEPよりも劣る結果となっている。f_4, f_6 ともにFEPが優れた結果を示しており、EEP1,EEP2,EEP3はともにCEPよりも優れてはいるもののFEPほどの収束特性を示していない。f_5 は、900世代を超えたところでEEP1,EEP3がFEPと遜色なく収束しているのに対してEEP2はCEPよりも劣ってしまっている。f_7 は、EEP2,EEP1,EEP3の順に収束しており、FEPよりもEEPが全体的に優れた結果となった。f_8, f_{10} は、FEP,EEP2,EEP1,EEP3の順で収束しておりEEPよりFEPの方が優れた結果となっている。f_9 は、600世代までにおいてEEP2はFEPよりも収束状況が良い。しかしその後900世代を超えたところでFEPよりも劣る結果となった。

6 まとめ

乱数分布の形状から云えることは図1、図2からも判断できるように、分散の大きい方が最適解の探索能力は高いと思われる。この観点からすれば、$C(0,1), E(0,0.1), E(0,0.5), E(0,1), N(0,1)$ 順に収束特性が決定されることが期待されるが実験結果からは $E(0,0.5)$ が $E(0,1)$ より劣っていることが判明した。特にuni-modal関数に対しては $E(0,0.5)$ を除いてこの理論的な面が裏付けされたと思われる。$C(0,1)$ を使用するFEPは分散が大きい乱数の効果により1000世代くらいまでの収束状況は大体良好な特性を持っている。multi-modal関数に対しては、ほとんどのケースでFEPの方が優れた収束特性を示しているがEEPの方もCEPよりは良い特性を持っていることから λ の値を $\lambda = 0.01$ あるいは $\lambda = 0.001$ 位のオーダーで、multi-modal関数については世代数を2000から5000に増加させた場合の収束特性についても検討する必要があると思われる。数値計算の観点からすれば、実際に現れる収束特性は個体 (individual) を構成する実数値ベクトル η の初期値の設定要領や自己適応応用 (self-adaptability)[6] さらには η の下限値の設定法などが影響しているものと考えられる。したがって、今回の実験ではuni-modal関数に対しては $\lambda = 0.1$ がmulti-modal関数に対しては $\lambda = 1.0$ が有効な収束特性を示し、全般を通じてFEPが良いが対象問題10個のうち、半分の5個についてはEEPの方が良好な結果を示していることから、EEP手法がEPの中では今後の発展が期待しうるものといえる。

今後の課題としては初期段階としての のとり方、最適解近傍における有効な λ のとり方につき検討する必要があると思われる。

参考文献

[1] L.J.Forgel,A.J.Owens, and M.J.Walsh, "Artificial Intelligence Through Simulated Evolution," John Wiley Sons, New York, NY, 1966.

[2] D.B.Forgel, "Applying evolutionary programming to selected traveling salesman problems," Cybernetics and Systems, 24:27-36, 1993.

[3] T.Back and H.-P.Schwefel, "An overview of evolutionary algorithms for parameter optimization," Evolutionary Computation, 1(1):1-23, 1993.

[4] K.Chellapilla, "Combining Mutation Operators in Evolutionary Programming," IEEE Transactions on Evolutionary Computation, 2(3), 91-96, 1998

[5] C.-Y.Lee and Y.song, "Evolutionary Programming using the Levy probability Distribution," Proc. of Genetic and Evolutionary Computation Conference (GECCO'99), Morgan Kaufman, 886-893, 1999.

[6] K.-H.Liang, X.Yao, Y.Liu, C.Newton and D.Hoffman, "An Experimental Investigation of Selfadaptation in Evolutionary Programming," Proc. of the 7 th Annual Conference on Evolutionary Programming, Springer-Verlag, 291-300, 1998.

[7] X.Yao, Y.Liu, "Fast Evolutionary Programming," Proc. of the 5 th Annual Conference on Evolutionary Programming, MIT Press, 451-460, 1996.

[8] X.Yao, Y. Liu, G. Lin, "Evolutionary Programming Made Faster," IEEE Transcations on Evolutionary Computation, 3(2),82-102, 1999.

[9] H.Narihisa, K.Kohmoto, 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.

Evolutionary Programming Using Exponential Mutation

Takashi KUMON, Keiko KOHMOTO and Hiroyuki NARIHISA*

Graduate School of Engineering,
**Department of Information and Computer Engineering,*
Faculty of Engineering,
Okayama University of Science,
1 - 1 Ridai-cho, Okayama, 700-0005, Japan.
(Received November 1, 2002)

During the last two decades there has been a growing interest in algorithms which are based on the principle of evolution(survival of the fittest). These related techniques are called evolutionary algorithms(EA) or evolutionary computation methods(EC).

The best known algorithms in this class include genetic algorithms and evolutonary programming. In general, solving a complex system can be perceived as a search through a space of potential solutions. For small spaces, classical exhaustive methods usually suffice. However, special artificial intelligence techniques must be employed for large spaces. Among of these evolutionary algorithms, evolutionary programming is originally developed by L.J.Forgel. Later, D.B.Forgel proposed EP algorithms for numerical and combinatorial optimization problems. In order to improve the classical EP(CEP), Yao et al. proposed fast evolutionary programming(FEP) using Cauchy mutation.

In this paper, we propose a new approach, in evolutionary programming(EEP) using the mutation based on double exponential distribution and present the empirical analysis of its convergence performance.