

複合指数分布を用いた進化的プログラミング

公文隆・片山謙吾*・成久洋之*

岡山理科大学大学院工学研究科情報工学専攻

*岡山理科大学工学部情報工学科

(2003年11月7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. Forgel²⁾ 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 Song⁵⁾ 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 分布に対応する事から、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 を凌駕する性能が期待されている。しかしながら、EEP の特徴としている乱数分布の分散を制御できるという事は反面、如何に制御すべきかという問題を内包することになる。すなわち、パラメータ λ の値を如何に決定するかが解探索の鍵となっている。

本論文は解の収束度合いに応じたパラメータ λ の変動と解の収束特性につき検討したもので、 λ の値を変更するタイミングとその大きさについて数値実験を実施し、その結果をまとめたものである。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あり、結果的にはこれまで有効であるとされてきた FEP よりも EEP の方が優れた収束特性を示すことができた。

2. 標準進化的プログラミング (CEP)

CEP は、進化的プログラミングにおける基本的なアルゴリズムであり、Gaussian 突然変異を使用するものである。

2.1 個体表現

EP における個体集団 (Population) を構成する、個体 $X_i (i = 1, 2, \dots, \mu)$ は n 次元実数値ベクトル x_i と n 次元戦略パラメータ η_i の対からなり、以下のように定義する。

$$X_i = [x_i, \eta_i], \quad (1)$$

$$x_i = (x_i(1), \dots, x_i(j), \dots, x_i(n)), \quad (2)$$

$$\eta_i = (\eta_i(1), \dots, \eta_i(j), \dots, \eta_i(n)), \quad (3)$$

ここで $j = 1, 2, \dots, n$ 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 成分とする。

2.2 突然変異

各個体 $[x_i, \eta_i]$ について、正規分布に従った突然変異による子孫 $(\acute{x}_i, \acute{\eta}_i)$ は次のように生成される。

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

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

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、 $j = 1, 2, \dots, n$ 。

$N(0, 1)$ は平均 0、標準偏差 1 の正規乱数を、 $N_j(0, 1)$ 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 $[0, 1]$ 区間での一様乱数 x_1, x_2, \dots, x_n より次式のように生成する。

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

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。

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

ただし、 x_j は $0 \leq x_j \leq 1 (j = 1, 2, \dots, 12)$ 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対 $X_i = [x_i, \eta_i]$

Step2. 各親 $[x_i, \eta_i]$ は単一の子孫 $[\acute{x}_i, \acute{\eta}_i]$ を生成。

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

$$\acute{\sigma}_i(j) = \eta_i(j) \exp(\acute{\tau}N(0, 1) + \tau N_j(0, 1)), \quad (9)$$

ただし、 $j = 1, 2, \dots, \mu$ 。

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

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。

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

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。

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

$f_t(x)$ 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) の代わりに次の式 (12) を使用する点が相異している。

$$x_i(j) = x_i(j) + \eta_i(j)C_j(0,1), \quad (12)$$

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

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

となり、これを $C(0,t)$ で表す。

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

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

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

として与えられる。これに対応する分布関数は

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

となる。したがって、平均 $\bar{x} = 0$ 、分散 $\text{var}(x) = \frac{2}{\lambda^2}$ 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、

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

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 (3) 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 tail) 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

対象問題は次ページの表 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 12 個とした。いずれも高次元関数であり、 f_1 から f_6 は単一の局所解を持つ uni-modal 関数であり、 f_7 から f_{12} 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) 個体集団 $\mu = 100$
- (2) トーナメントサイズ $q = 10$
- (3) 世代数 $GEN = 5000$

5.3 実験実施要領

EEP の λ のスイッチングに対する収束効果検討のため λ を様々な値またはスイッチング世代でを試行し、その条件に対して 100 回実行した平均に対する収束特性を従来提案されている中で有効な FEP と比較した。結果の系列の意味は 0.05c とは λ の値が 0.05 を 5000 世代まで持続したものを表しており、1(500)5 は初期の λ の値は 1.0、500 世代で 5.0 に変更したものを表している。

5.4 実験結果

表 2 の $f_1 \sim f_{12}$ 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、すべての EEP が FEP よりも良好な結果を示した。さらに、1.0c よりも 1(1000)5 の方が良い特性を示している。0.1(2000)10 はスイッチングをしてから急激に収束した。 f_2 においても全体的に EEP が FEP よりも優れた結果となった。こちらも 1.0c よりスイッチングを行った方が優れた結果となった。 f_3 についてはスイッチングを比較的早めに行った、1(1000)5 はスイッチングをすることにより悪い結果となった。しかし、1(3500)5 のように遅めにスイッチングすると良好な結果となった。 f_4 は 0.01(2000)1 は FEP よりも劣る結果となったがそれ以外のものは FEP よりも優れた結果となった。特に 0.1(2000)1 はスイッチングを行ってから急激に収束した。 f_5 は全体的に FEP よりも良好な結果を示したが、スイッチングの効果が f_1 等と比べると劣った結果となっている。 f_6 は FEP が全体的に優れた結果となった。FEP が 700 世代ほどでほぼ 0 になるのに対して、EEP は早くも 1400 世代、又は 1700 世代で、特に悪い 0.1(200)10 では 0.06 程度で収束しきってしまいほとんど変化しないものがあつた。また、スイッチングの効果がこの

問題に対してはほとんど表れていない。 f_7 は f_5 同様に優れた結果となっているが、スイッチングの効果はほとんど表れていない。 f_8 は 0.00001c が FEP よりも優れておりほぼ目的解の値になった。 f_9 は 0.05c よりも 0.05(500)0.1 のようにスイッチングを施したほうが優れた結果となり、さらに FEP よりも優れた収束特性を示した。 f_{10} においては 0.1(2000)10 はスイッチングの効果が特に表れた結果となり、FEP よりも優れた結果となった。 f_{11} 、 f_{12} ともに FEP の方がすぐれた結果となっている。しかし、どちらもスイッチングによる影響が少なからず表れている。

表 2. ベンチマーク問題

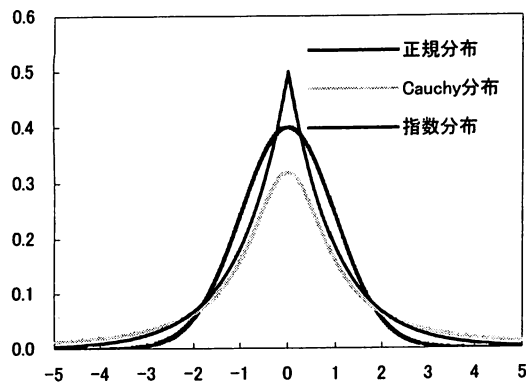
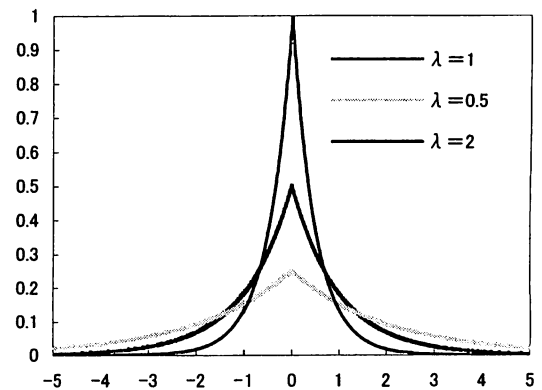
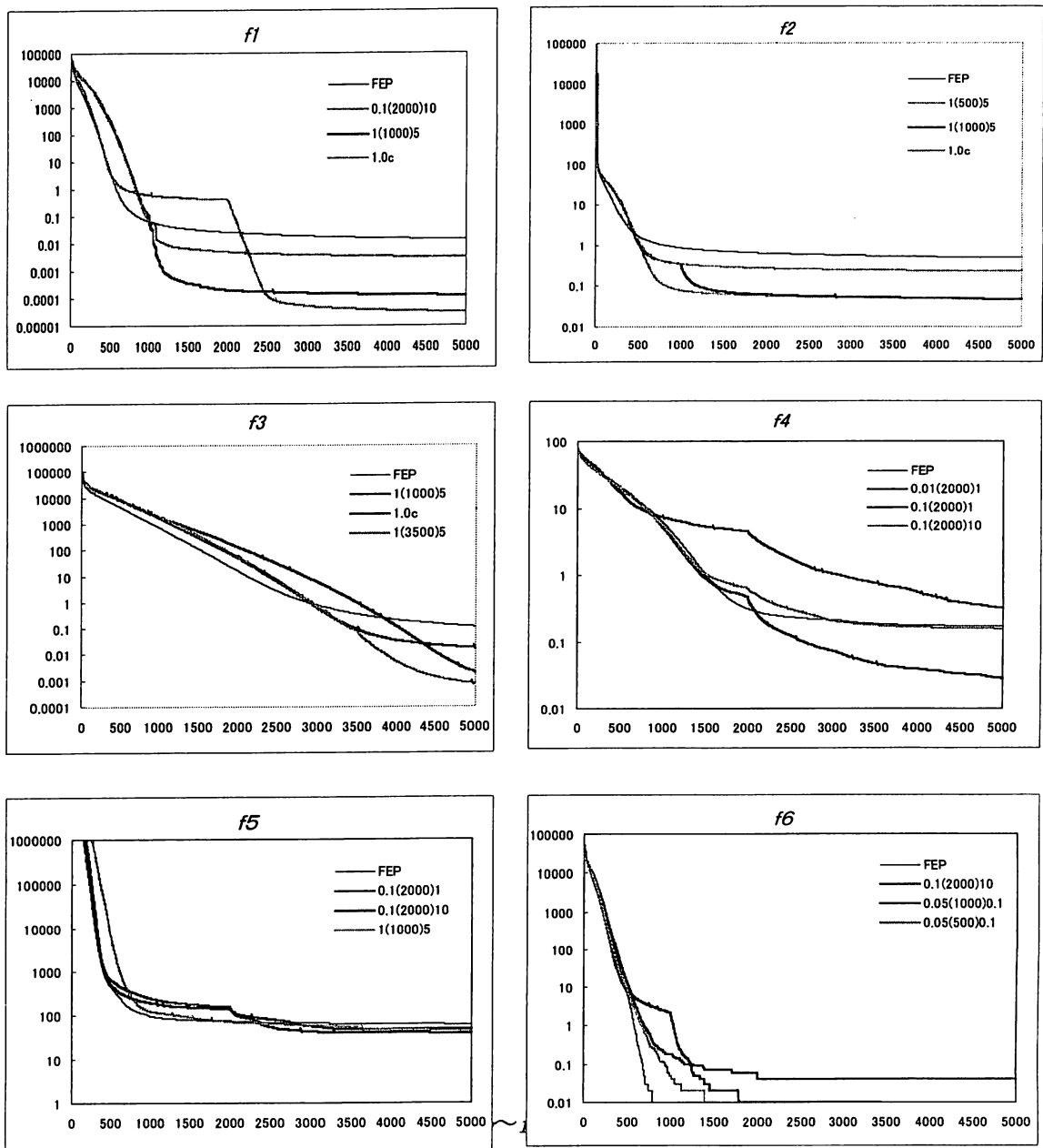
対象関数	s	f_{min}
$f_1(x) = \sum_{i=1}^{30} x_i^2$	$[-100, 100]^{30}$	0
$f_2(x) = \sum_{i=1}^{30} x_i + \prod_{i=1}^{30} x_i $	$[-10, 10]^{30}$	0
$f_3(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^{30}$	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq 30\}$	$[-100, 100]^{30}$	0
$f_5(x) = \sum_{i=1}^{29} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^{30}$	0
$f_6(x) = \sum_{i=1}^{30} ([x_i + 0.5])^2$	$[-100, 100]^{30}$	0
$f_7(x) = \sum_{i=1}^{30} ix_i^4 + \text{random}[0, 1]$	$[-1.28, 1.28]^{30}$	0
$f_8(x) = -\sum_{i=1}^{30} \left(x_i \sin \left(\sqrt{ x_i } \right) \right)$	$[-500, 500]^{30}$	-12569.5
$f_9(x) = \sum_{i=1}^{30} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^{30}$	0
$f_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2} \right) - \exp \left(\frac{1}{30} \sum_{i=1}^{30} \cos 2\pi x_i \right) + 20 + e$	$[-32, 32]^{30}$	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^{30}$	0
$f_{12}(x) = \frac{\pi}{n} \{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1}) + (y_n - 1)^2] \}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4) y_i = 1 + \frac{1}{4}(x_i + 1),$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^{30}$	0

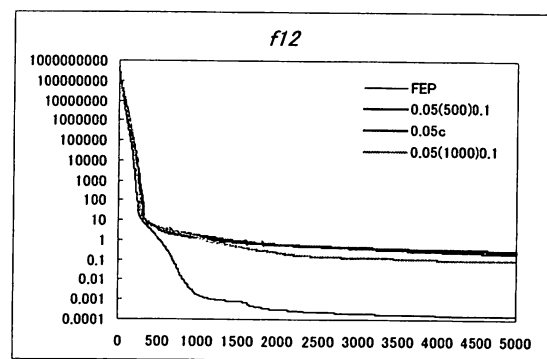
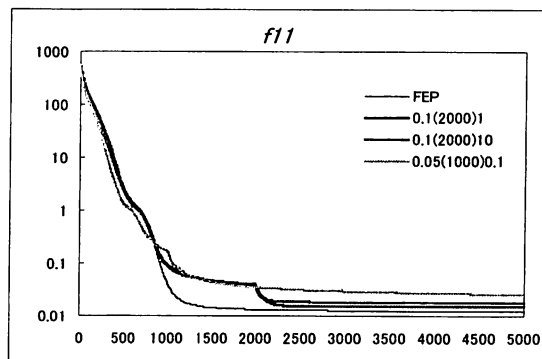
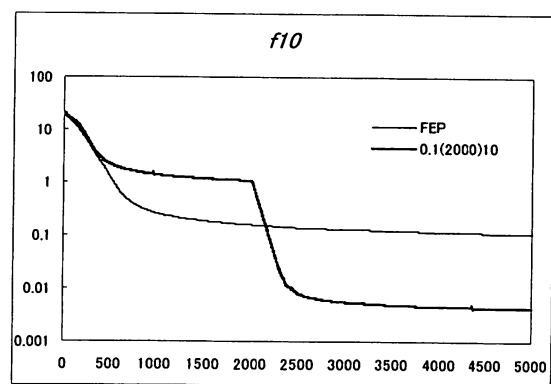
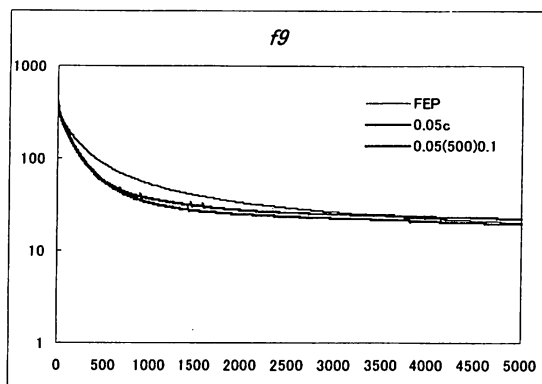
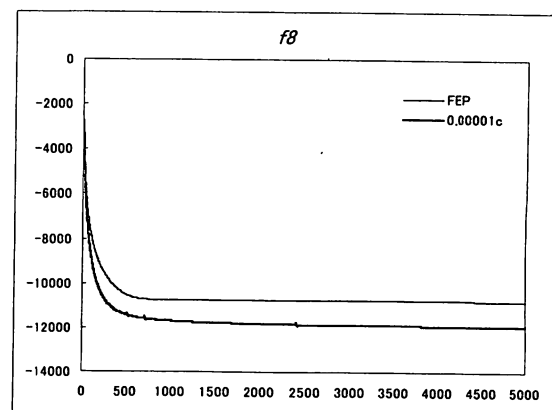
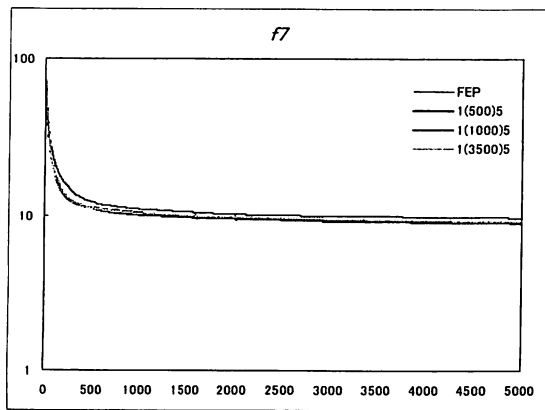
6. まとめ

今回の $f_1 \sim f_{12}$ の結果のうち、9 個の EEP が FEP よりも優れた特性を示したことからも EEP の有効性が表れている。前述した通り EEP は λ の値を自由に設定でき図 2 からわかるように分散を制御でき、進化の過程で λ を変化させる事もできる。この事により問題の特性に合わせた進化が設定できるのでさらなる発展が期待できる。

今後の課題としてはさらに多くの問題にも適用させ λ の最適な値、及び最適なスイッチングの世代又はパラメータ設定を検討する必要がある。

図 1. 乱数分布

図 2. λ の値による指数分布図 3. $f_1 \sim f_6$ の結果



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 Transactions 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, Kengo KATAYAMA* 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 7, 2003)*

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 evolutionary 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 2002, Narihisa et al proposed a new efficient evolutionary programming(EEP) with the mutation based on double exponential distribution which is controllable the variance of its distribution by changing the value of parameter λ . In this paper, we present a changing effect of the distribution parameter λ to convergence characteristics for the given optimization problems. The results of our numerical experiments show that EEP outperforms FEP in almost cases by taking appropriate parameter value according to the landscape of the given function and its convergence characteristics during initial evolving periods.