

マルチスイッチング制御による 指数型進化的プログラミングの効率化

津田 倫彰・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

(2004年9月28日受付、2004年11月5日受理)

はじめに

進化的アルゴリズム (Evolutionary Algorithm : EA) は生物の進化を模倣したアルゴリズムであり, 数値最適化²⁾, 組み合わせ最適化などに使用されている. EA のひとつである進化的プログラミング (Evolutionary Programming : EP) [L.J.Forgel] は, 進化対象 (問題最適化対象) を突然変異という確率的に解を改善させる手法である. 一般的な突然変異の方法, つまり解の発生方法には Gaussian 分布に基づいた古典進化的プログラミング (Classical Evolutionary Programming : CEP)[B.D.Forgel]¹⁾, Cauchy 分布に基づいた高速進化的プログラミング (Fast Evolutionary Programming : FEP)[Yao et al]⁴⁾, そして, Narihisa らが提案した複合指数分布に基づいた指数型進化的プログラミング (Exponential Evolutionary Programming : EEP)⁵⁾ がある. 解を進化させるとき CEP や FEP は分布を固定して突然変異を起こす事に対して, EEP は進化の途中でも分散を変更できるという特徴があり, 進化する解の探索領域を調節できる. これによって CEP や FEP よりも良い解が期待できる. しかし, EEP には分散を変更するタイミングや大きさを誤れば良い解を得難いという問題がある.

本研究では進化の途中で分散を変更させるスイッチング EEP の解の収束特性を示す. その収束特性を 4 点スイッチング EEP (進化の途中で分散を 4 度変更させる EEP) と CEP, FEP を表 1 のベンチマーク問題に適用した結果から比較・検討を行い, スwitchング EEP の解探索特性についてまとめる.

1. 進化的プログラミング (Evolutionary Programming:EP)

EP は生物の進化や自然淘汰のしくみを模倣した進化的計算 (Evolutionary Computation : EC) の一分野である. EP は人工知能へのアプローチとして開発されたが, 最近では多くの数値や組み合わせ最適化問題などに適用されている.

1.1 EP の流れ

EP の基本的なアルゴリズムである古典進化的プログラミング (Classical Evolutionary Programming : CEP) を用いて EP を説明する.

Step1. 初期集団の生成

μ 個からなる初期集団を生成する. EP の個体 $X_i (i = 1, 2, \dots, \mu)$ は, 解を構成する n 次元の実数値ベクトル x_i と x_i の変化幅を決定する n 次元の戦略パラメータ η_i の対からなる. 個体 X_i は以下のように定義する.

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

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

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

$j = 1, 2, \dots, n$ であり, $x_i(j), \eta_i(j)$ はベクトル x_i, η_i の第 j 成分である.

Step2. 子孫の生成

EP の個体 (x_i, η_i) は正規分布に従った突然変異のみによって子孫 (x'_i, η'_i) を生成する。生成式を以下に示す。

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

$$\begin{aligned} \eta'_i(j) &= \eta_i(j) \exp(\tau' N(0, 1) + \tau N_j(0, 1)) \\ \tau &= (\sqrt{2\sqrt{n}})^{-1}, \tau' = (\sqrt{2n})^{-1} \end{aligned} \quad (5)$$

ここで $j = 1, 2, \dots, n$ とし $x_i(j), x'_i(j), \eta_i(j), \eta'_i(j)$ は個体 X_i の第 j 成分を表す。 $N(0, 1), N_j(0, 1)$ は平均 0, 標準偏差 1 の正規乱数であり, $N(0, 1)$ は各個体ごとに, $N_j(0, 1)$ は個体 X_i の j ごとに正規乱数を発生する。

Step3. 適応度の計算

すべての個体について適応度計算を行う。親子の適応度はそれぞれ $f(x_i), f(x'_i), i = 1, 2, \dots, \mu$ とする。

Step4. 淘汰 (q トーナメント選択)

q トーナメント選択とは, すべての親と子から無作為に q 個選び出された適応度を元に親と子の 2μ 個に点をつけ, 得点の高い μ 個を次世代の親とする選択法である。各 2μ 個の個体に与えられる得点は, 取り出された q 個の個体とそれぞれ比較し, より最適値に近い場合に 1 点ずつが加点される。

Step5. 終了条件を満たすまで **Step2.** から **Step4.** まで繰り返す。

終了条件とは最適な適応度をとる個体が存在する, または予定の進化回数に達することとする。

また, 正規乱数 $N(0, 1)$ の発生方法は, n 個の一樣乱数 a_1, a_2, \dots, a_n に対して

$$N(0, 1) = \frac{\frac{1}{n} \sum_{i=1}^n a_i - \frac{1}{2}}{\sqrt{\frac{1}{12n}}}, \quad 0_i \leq 1 (i = 1, 2, \dots, n) \quad (6)$$

(6) 式から, n が大きいとき $N(0, 1)$ 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良くなるが, 必要な一樣乱数の個数が多くなり乱数の発生時間が長くなる。そこで $n = 12$ とした時, (6) 式が (7) 式のように簡略化されるとともに, 近似度は極めて高くなる。よって本研究では近似度が高く, 計算時間を短縮できる (7) 式の正規乱数に従う。

$$N(0, 1) = \sum_{i=1}^{12} a_i - 6.0, \quad 0 \leq a_i \leq 1 (i = 1, 2, \dots, 12) \quad (7)$$

1.2 高速進化的プログラミング (Fast Evolutionary Programming : FEP)

FEP は, 正規分布に従う CEP の進化が遅いという欠点を改善した Cauchy 分布の乱数によって子孫を生成する EP である。平均を原点とする 1 次元 Cauchy 密度関数 $f_t(x)$ は以下に示す。

$$f_t(x) = \frac{1}{\pi} \cdot \frac{t}{x^2 + t^2}, \quad -\infty \leq x \leq \infty, t > 0 \quad (8)$$

t はスケールパラメータとする。また, Cauchy 分布関数 $F(x)$ は以下に示す。

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \cdot \arctan\left(\frac{x}{t}\right) \quad (9)$$

Cauchy 乱数を $C(x, t)$ とするとき, FEP で使用される Cauchy 乱数は中心 $x = 0$, スケールパラメータ $t = 1$ の $C(0, 1)$ とする。Cauchy 乱数 $C(0, 1)$ は Cauchy 分布関数 $F(x)$ と $[0, 1]$ 区間の一樣乱数 a に対応させて

$$C(0, t) = t \cdot \tan\left\{\pi \left(a - \frac{1}{2}\right)\right\} \quad (10)$$

となり, $C(0, 1)$ は次式となる.

$$C(0, 1) = \tan\left\{\pi\left(a - \frac{1}{2}\right)\right\}, 0 \leq a \leq 1 \quad (11)$$

FEP のアルゴリズムは CEP での子孫生成方法を (4) 式から (12) 式に変更したものとする. $C_j(0, 1)$ は $N_j(0, 1)$ と同様に個体 X_i の j ごとに乱数を発生する.

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

1.3 指数型進化的プログラミング (Exponential Evolutionary Programming : EEP)

EEP で使用されるパラメータ λ における複合指数分布の確率密度関数 $f(x)$ を以下に示す.

$$f(x) = \frac{\lambda}{2} \exp(-\lambda|x|), -\infty \leq x \leq \infty, \lambda > 0 \quad (13)$$

また, 同分布関数 $F(x)$ は以下に示す.

$$F(x) = \int_{-\infty}^x f(t)dt = \frac{\lambda}{2} \int_{-\infty}^x \exp[-\lambda|t|]dt \quad (14)$$

$$= \begin{cases} \frac{1}{2} \exp[\lambda x] & x \leq 0 \\ 1 - \frac{1}{2} \exp[-\lambda x] & x > 0 \end{cases} \quad (15)$$

(16) 式で表される分布は $F(x) = 0$ に対して対称であり, 分散はパラメータ λ によって決定される (制御できる) ことがわかる. そして, 平均が 0, パラメータ λ の複合指数乱数 $E(0, \lambda)$ は $[0, 1]$ 区間の一様乱数 a に対して次式で与えられる.

$$E(0, \lambda) = \begin{cases} \frac{1}{\lambda} \ln(2a) & 0 \leq a \leq 0.5 \\ -\frac{1}{\lambda} \ln\{2(1-a)\} & 0.5 < a \leq 1.0 \end{cases} \quad (16)$$

EEP のアルゴリズムは FEP のそれと同様, CEP での子孫生成方法を (4) 式から (17) 式に変更したものとする. $E_j(0, \lambda)$ は $N_j(0, 1)$ と同様に個体 X_i の j ごとに乱数を発生する.

$$x'_i(j) = x_i(j) + \eta_i(j)E_j(0, \lambda) \quad (17)$$

1.4 各 EP の特徴

各 EP で使用される乱数の分布を図 1, EEP で使用される乱数のパラメータ λ の値を変化させたときの分布を図 2 に示す. それぞれの図の横軸は $-5 \sim 5$ の乱数を, 縦軸はその乱数の分布を意味する. $N(0, 1), C(0, 1)$ に関しては次のようなことが言われている³⁾.

- ・ $C(0, 1)$ は尾の長い分布のため, FEP は親とはかなり異なった子を生成しうる. そのため, 局所解に陥ることのない進化が期待できる.
- ・ しかし, $C(0, 1)$ の分布は $N(0, 1)$ に比べ背が低く尾の長い分布特性のため, 局所的な解探索のための時間が少ない. そのため最適解付近では良質な解の探索能力は CEP に劣るといえる.

つまり, FEP は分散が無限大のため, 生成する子の種類は無限大となり局所解を避けることができる EP である. それと同時に最適解までも避ける可能性を持った EP とも言える. そして CEP はその逆と言える. $E(0, 1)$ は図 1 が示すように尾の長さや背の高さが中間的な特性を示しているため, パラメータ $\lambda = 1$ の EEP の解探索特性は FEP と CEP の中間といえる.

EEP の特徴である分散制御可能にするということは, EP の解探索特性力の制御を可能にするということになる. 図 2 のように λ の値を大きくすれば尾が短く背の高い $N(0, 1)$ のような分布特性を持ち, CEP のような局所解探索が可能となり, λ の値を小さくすれば尾が長く背の低い $C(0, 1)$ のような分布特性を持ち, FEP のようなグローバルな探索が可能となる. また, 分散の制御は解の進化の途中でも可能である.

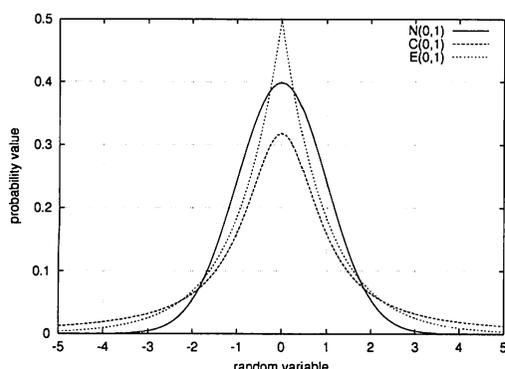
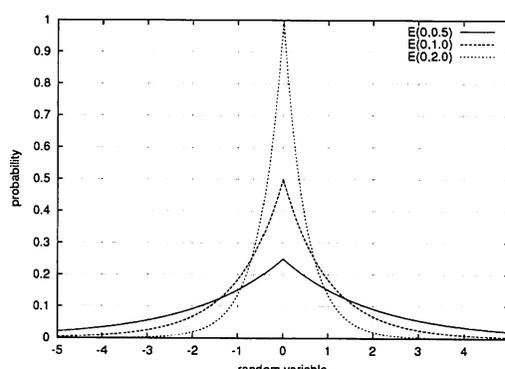


図1 各 EP の乱数分布

図2 λ の値による複合指数乱数分布

2. 実験

本研究では、解が進化している途中で複合指数乱数のパラメータ λ の値を変化させるスイッチングという効果について検討する。

2.1 対象問題

対象問題として表 1 にあるベンチマークの関数最小化を行う。 f_1 から f_6 は単一の局所解をもつユニモーダル関数、 f_7 から f_{12} は複数の局所解をもつマルチモーダル関数となっている。ベンチマーク問題に CEP, FEP, スwitching EEP を適用し、スイッチングの効果について検討する。パラメータは以下ようになる。また、EP のアルゴリズムは偶発的な解生成を行うアルゴリズムのため、各 EP をベンチマーク問題 1 つに対して 100 回適用し、得られた結果を平均化して収束特性を検討する。

- ・ 個体数 μ : 30
- ・ トーナメントサイズ q : 10
- ・ 進化回数 5000
- ・ スwitching EEP : $EEP1$: 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

$EEP1$ は進化開始時の λ の値を 0.1 とし、1000 世代目になると λ の値を 0.5 に変え、そして 2000 世代目になると、 λ の値を 1.0 にするという意味である。つまり、各スswitching EEP は初期探索から 4 段階に分けて局所的な探索へと切り替えている。

3. 実験結果

実験結果を図 3 から図 14 に、各 EP の最終世代での適応度を表 2 に示す。 f_1 の実験結果 (図 3) からスswitching EEP の効果が明確である。進化の初期では、FEP の方が優れているが、最初のスswitching である 1000 世代以降はスswitching ごとに解の最小化の速度が進んでいる。 f_2, f_3 (図 4, 図 5) においては f_1 のような顕著な収束特性を示してはいないが、スswitching 効果により良い結果となっている。しかし、 f_4 (図 6) においてはどの EEP にもスswitching 効果が見られなかった。 f_5 (図 7) に関してはスswitching 効果は見られないが、5000 世代到達時においても進化が進む様子を見せている。 f_6 (図 8) では CEP, FEP, 全 EEP が最適解へと収束した。そのときの進化の早さと収束の順番は FEP, $EEP1$, $EEP3$, $EEP2$, $EEP4$, CEP という順番になり、ほぼ分散が大きかった順に (EEP に関しては初期 λ が小さかった順に) 最適解へと達した。表 2 のようにスswitching EEP は一部のユニモーダル関数を除いて有効であるということがわかる。特に $EEP3, EEP4$ のように初期 (最終) λ を大きくした EEP は、局地探索向けの EP であるということがいえる。

f_7 (図 9) においては全 EEP が良い収束特性を示している。表 1 に示すように f_7 は他のベンチマークと比べて解の探索幅が狭い問題のため、初期 λ が大きい $EEP3, EEP4$ が効果的良い結果を示している。 f_8 (図 10) においては、EEP の最初のスswitching の前にすべての EEP が局地解に収束している。また f_9 (図 11) に関しては、進化の初期段階では他の EP と比べ $EEP1$ の進化が遅れているが、 $EEP1$ のみスswitching 効

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
$f_4(x) = \sum_{i=1}^{29} m_i a_i x_i , 1 \leq i \leq 30$	[-100, 100]	0
$f_5(x) = \sum_{i=1}^{29} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30] ³⁰	0
$f_6(x) = \sum_{i=1}^{30} [(x_i + 0.5)]^2$	[-100, 100] ³⁰	0
$f_7(x) = \sum_{i=1}^{30} ix_i^4 + random_i[0, 1)$	[-1.28, 1.28] ³⁰	0
$f_8(x) = -\sum_{i=1}^{30} (x_i \sin(\sqrt{ x_i }))$	[-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
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^{30} \cos 2\pi x_i^2\right) + 20 + e$	[-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
$f_{12}(x) = \frac{\pi}{60} \{10 \sin^2(\pi y_i) + \sum_{i=1}^{29} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_{30} - 1)^2\}$ $+ \sum_{i=1}^{30} u(x_i, 10, 100, 4)$	[-50, 50] ³⁰	0

果が現れ、全 EEP の中でよい結果を示した。 f_{10} (図 12) では f_8 の時と同様に最初のスイッチングの前で全 EEP の進化が停止している。この場合でも f_8 と同様に EEP1 が全 EEP の中では最良の結果を示している。 f_{11}, f_{12} (図 13, 図 14) では、それぞれの EEP1 が初期の進化数百世代までは FEP と同等の進化を示しているが、 f_{11} は最初のスイッチング後、 f_{12} に関しては f_8, f_9, f_{10} の実験結果のように最初のスイッチング前に、進化が急速に遅くなるか、または停止してしまっている。マルチモーダル関数に対してはユニモーダルのような良好な結果は得られなかった。それは図 9 のスイッチング後に進化の速度が一時的に停止してしまつたこと、図 10 のようにスイッチング前に進化が停止してしまつたことから進化の初期では実験で使用した λ よりも小さな値にすれば良い結果が期待できる。そのことは EEP1 を見てわかるように初期 λ が小さいほど FEP には及ばないものの EEP1 の中では良い結果をもたらし易いということからもそのような期待ができる。

4. 考察

f_8 の実行結果 (図 10) のように最初のスイッチング前に解が収束したことで、 EEP1 のように初期 λ が小さいものが殆どのベンチマーク問題の進化初期での探索素特性が高かつたことから、 λ の値を EEP1 よりも小さくした実験を行う。パラメータ設定は EEP1 の最初の λ を 0.1 から 0.01, 0.001, 0.0001, 0.00001 と小さくした EEP1 をそれぞれ、 EEP5, EEP6, EEP7, EEP8 とする (図 15)。その結果、 EEP5 から EEP8 は EEP1 から EEP4, FEP よりも優れた結果を示しており、また EEP7, EEP8 は最適解へと収束した。 EEP1 から EEP4 に見られなかつたスイッチング効果が EEP6, EEP7 に現れ、 EEP6 については探索素特性を向上させ、 EEP7 はその効果により最適解へと収束した。また、 f_8 に対して良い特性を示した EEP8 を他の関数に

適用させた結果 (表 3), 全てのマルチモーダル関数に対して EEP1 から EEP4 よりも優れた結果が得られたが, ユニモーダル関数に対してはその逆の結果となった。

実験を通して, ほとんどの問題に対して初期 λ が小さかった EEP1 の急激な進化が見られたこと, 図 9 のように解が効率良く進化しているときに λ を大きく変えて解探索特性が落ちたこと, ユニモーダル関数のように進化が落ち着く最適値周辺で λ を大きく変えることによって解探索特性力が上がった事ことから, 解が効率良く進化するであろう進化初期から中期あたりでは λ の値の変化の小さい方が効果的な解探索が可能になるという期待ができる。また, EEP5 から EEP8 の実験結果からユニモーダルに関しては最初の λ を 0.1 から以上に, マルチモーダルに関しては 0.001 以下のパラメータの設定が好ましい。

今回の実験では複数のスイッチングを行ったが, 今後は全進化の過程でのスイッチングを検討する。

表 2 最終世代での適応度

f	CEP	FEP	EEP1	EEP2	EEP3	EEP4
f_1	$1.925049e^{-5}$	$1.473935e^{-4}$	$4.440836e^{-7}$	$1.110075e^{-7}$	$1.104651e^{-7}$	$7.139321e^{-8}$
f_2	$1.887970e^{-2}$	$5.093474e^{-2}$	$2.867935e^{-3}$	$1.428136e^{-3}$	$1.435613e^{-3}$	$1.162933e^{-3}$
f_3	$7.224361e^{-3}$	$3.588663e^{-2}$	$1.861515e^{-4}$	$4.330550e^{-5}$	$4.723577e^{-5}$	$3.337712e^{-5}$
f_4	$4.789981e^{-1}$	$6.228893e^{-3}$	$1.248648e^{-1}$	<u>$4.557398e^{-2}$</u>	$4.711148e^{-2}$	2.919063
f_5	$4.104509e^{+1}$	$4.485276e^{+1}$	$3.636511e^{+1}$	$3.545317e^{+1}$	$3.396918e^{+1}$	$3.572557e^{+1}$
f_6	$0.000000_{(1556)}$	$0.000000_{(753)}$	$0.000000_{(962)}$	$0.000000_{(1113)}$	$0.000000_{(1025)}$	$0.000000_{(1582)}$
f_7	$4.160619e^{-2}$	$8.301105e^{-2}$	$3.650455e^{-2}$	$2.924329e^{-2}$	$2.783769e^{-2}$	$2.806214e^{-2}$
f_8	$-7.278017e^{+3}$	$-1.108609e^{+4}$	<u>$-7.817093e^{+3}$</u>	$-7.380442e^{+3}$	$-7.362007e^{+3}$	$-7.336524e^{+3}$
f_9	$9.877292e^{+1}$	$3.615645e^{-2}$	<u>$5.617537e^{+1}$</u>	$6.313997e^{+1}$	$7.403472e^{+1}$	$8.608359e^{+1}$
f_{10}	$1.754789e^{+1}$	$9.158581e^{-3}$	<u>6.147755</u>	6.403375	6.343575	$1.029754e^{+1}$
f_{11}	$2.648814e^{-2}$	$1.937632e^{-2}$	$3.624663e^{-2}$	$4.104509e^{-2}$	<u>$2.970047e^{-2}$</u>	$1.154182e^{+1}$
f_{12}	1.690723	$1.447893e^{-6}$	$5.483711e^{-1}$	$5.448380e^{-1}$	<u>$3.844782e^{-1}$</u>	$7.109251e^{-1}$

各 EP の中で最良だったものを太字, 各 EEP で最良だったものを 下線 で表す。また, f_6 の () 内に示した数字は最適値に達した世代。

参考文献

- 1) B.D.Forgel, "Applying evolutionary programming to selected traveling salesman problems", *Cybernetics and Systems*, 22:27-36, 1993.
- 2) T.Back and H.-P.Schwefel, "An overview of evolutionary algorithms for parameter optimization", *Evolutionary Computation*, 1(1):1-23, 1993.
- 3) X. Yao, Y. Liu and G. Lin, "Evolutionary Programming Made Faster", *IEEE Transactions on Evolutionary Computation*, vol.3, no.2, pp.82-102, 1999.
- 4) X.Yao, Y.Liu, "Fast Evolutionary Programming", *Proc. of the 5 th Annual Conference on Evolutionary Programming*, MIT Press, 451-460, 1996.
- 5) 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.
- 6) H.Narihisa, K.Kohmoto, M.Tsuda, 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.

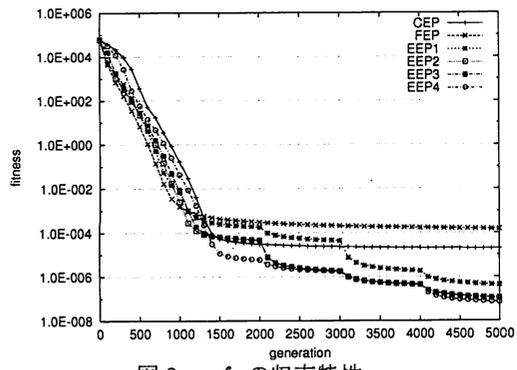


図3 f_1 の収束特性

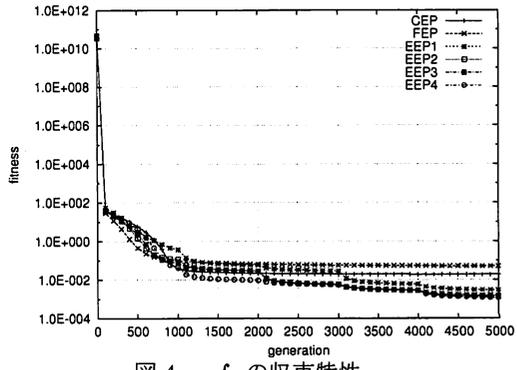


図4 f_2 の収束特性

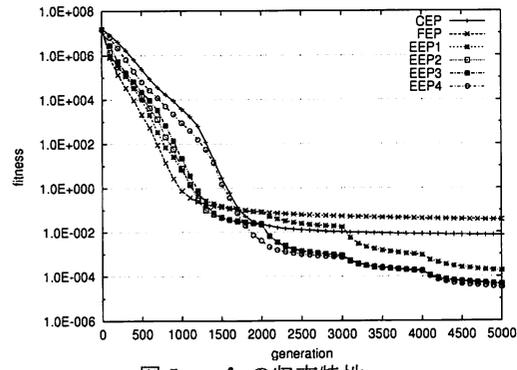


図5 f_3 の収束特性

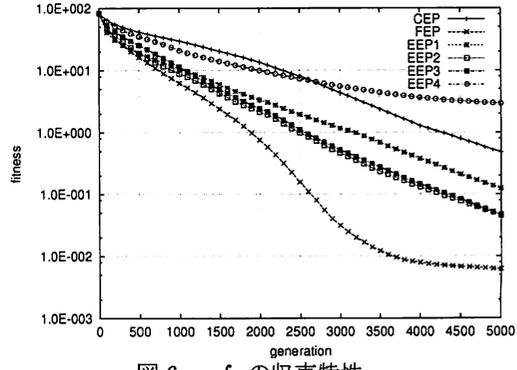


図6 f_4 の収束特性

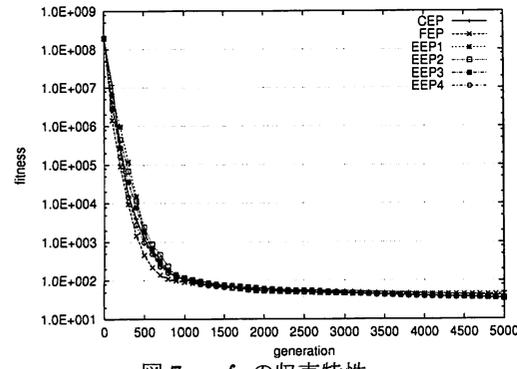


図7 f_5 の収束特性

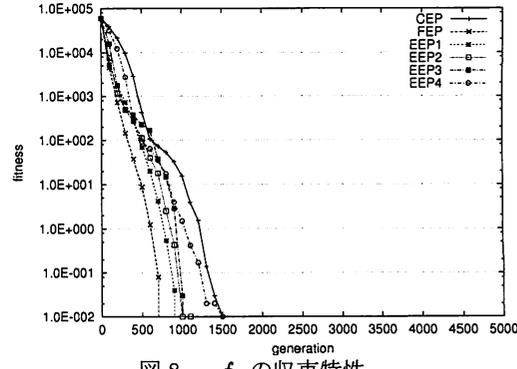


図8 f_6 の収束特性

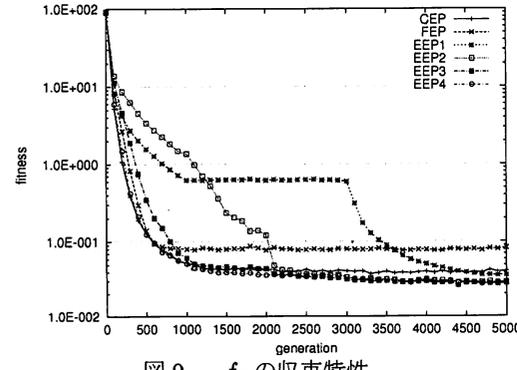


図9 f_7 の収束特性

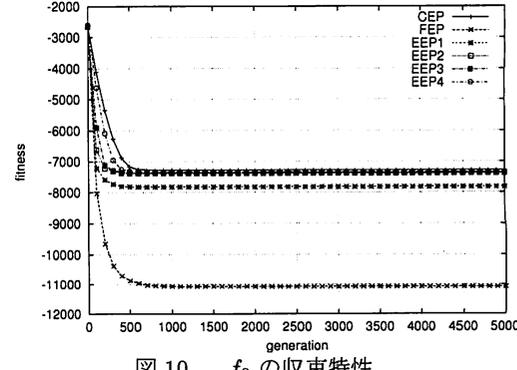


図10 f_8 の収束特性

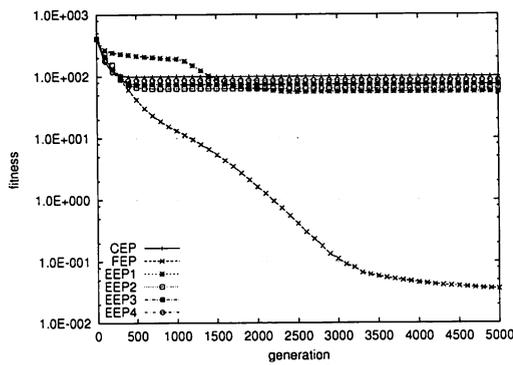


図 11 f_9 の収束特性

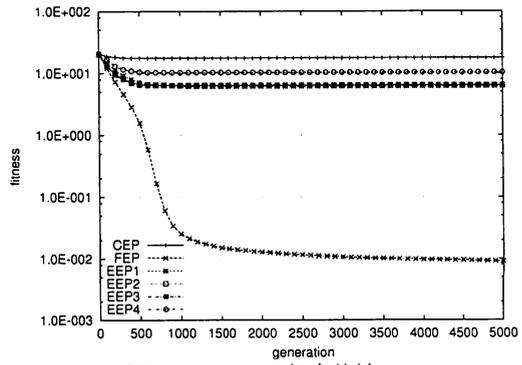


図 12 f_{10} の収束特性

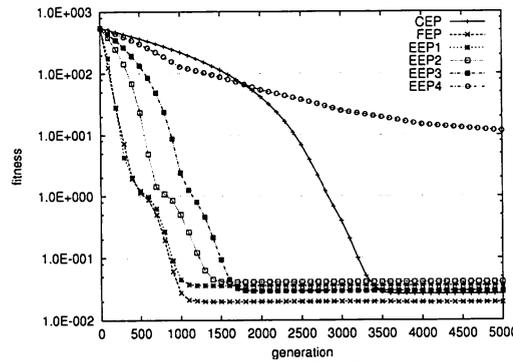


図 13 f_{11} の収束特性

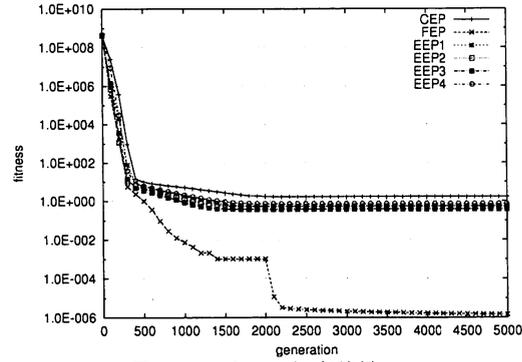


図 14 f_{12} の収束特性

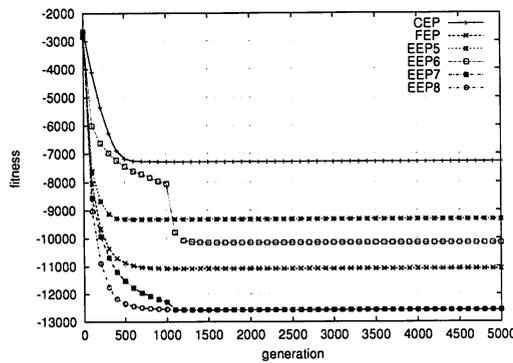


図 15 初期 λ を小さくした f_8 の収束特性

表 3 EEP8 での最終世代での適応度

f_1	f_2	f_3	f_4	f_5	f_6
$4.373470e^{-7}$	$2.847478e^{-3}$	$1.697496e^{-4}$	$8.482937e^{-2}$	$4.374170e^{+1}$	$0.000000_{(1778)}$
f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
$3.056084e^{-2}$	$-1.256949e^{+4}$	4.487354	$5.631469e^{-1}$	$1.251620e^{-2}$	$7.256623e^{-3}$

f_6 の () 内に示した数字は最適値に達した世代。

Efficiency of Exponential Evolutionary Programming with Multi-Switching Parameter

Michiaki TSUDA, 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 September 28, 2004; accepted November 5, 2004)

In this paper, we present the convergence characteristics of Exponential Evolutionary Programming (EEP) which uses a mutation based on double exponential probability distribution with positive parameter value. Hitherto, Evolutionary Programming (EP) mainly used Gaussian mutation or Cauchy mutation in order to improve some given solutions on adopting to numerical optimization problems. Due to the characteristics of these probability distribution, Cauchy mutation is superior to Gaussian mutation in convergence performance at early stage, but Gaussian mutation is desirable at mean-optimal stage.

At this point of view, Narihisa et al proposed exponential mutation which was based on double exponential distribution as EEP. The double exponential distribution can be considered as a middle distribution between Gaussian and Cauchy distribution. Moreover, double exponential distribution is controllable its variance with distribution parameter. Concerning the distribution of random number which is used in mutation of Evolutionary Programming, it is desirable that the variance of the distribution should be large in first stage of evolution and should be small in mid to final stage of evolution applying on optimization problems.

In order to realize such distribution of random number, we propose EEP with multi switching of parameter value of the distribution from initial value to next value at some arbitrary generation so as to decrease the variance of the distribution. Experimental results show that these switching EEP can improve convergence performance of EEP and switching effect can be recognized.