遺伝的浮動を利用する進化的プログラミング*

大倉 和博[†]・松村 嘉之[‡]・上田 完次[†]

Evolutionary Programming that Utilizes Genetic Drift*

Kazuhiro OHKURA[†], Yoshiyuki MATSUMURA[‡] and Kanji UEDA[†]

Evolutionary Programming (EP) belongs to a class of general optimization algorithms based on the model of natural evolution. EP has also been applied to real-valued function optimization since the early 90's. However, recent research results have proved that EP is not so robust as expected; EP performs very well only when the lower bound of strategy parameters is adjusted to each problem. In order to overcome this difficulty, an extended EP, called Robust EP (REP), is proposed. A major feature of REP is that genetic drift is introduced as another source of changing strategy parameters. Computer simulations are conducted in order to illustrate the robust performance of REP against the lower bound on a set of popular benchmark problems. Some evolutionary characteristics of REP are also clarified by calculating basic statistical values.

1. はじめに

自然進化に基づく進化型計算 (Evolutionary Computation: EC) は、種々の最適化問題に適用できる頑健な手法として知られている。この EC には、進化的プログラミング (Evolutionary Programming: EP), 進化戦略 (Evolution Strategies: ES), 遺伝的アルゴリズム (Genetic Algorithms: GA) の三つの主な流れがあり、個体表現形式・突然変異・組換え・選択等により、それぞれ特徴付けられる。本論文で取り扱う EP は、1960 年代前半、L. J. Fogel が有限状態機械を進化させる枠組みとして提案したものを、その息子 D. B. Fogel が 1990 年代初頭に “meta-EP” として実関数最適化問題向けに定式化しなおしたものである。なお、慣習に習い、以下ではこれも EP と呼ぶ。

EP は “個体群の進化過程” ではなく “種の進化過程” に着目した手法として、参考文献 [1,4] 等で説明されて

いる。これから、EP は GA や ES のように複数個体による組換え操作を用いない。各個体は、実数値表現された目的変数とその突然変異量をだまかに規定する戦略パラメータを組として一列に並べたものとして表現される。再生過程では、その両方が進化する対象となっているため、突然変異量が自然選択を通して適応的に調節される。この振る舞いは自己適応 (self-adaptation) と呼ばれ、EP の重要な特徴であるとされている。

この最適化手法は、多くのベンチマーク問題だけでなく実問題 (たとえば, [5]) でもその優れた探索能力が検証されている。しかし、近年、EP の探索能力にほとんど影響が無いとして重要視されていなかった戦略パラメータに関する下限設定値が、EP の探索能力に大きく影響すること、および、その最適値が強く問題に依存することが Liang らにより示された [9,10]。すなわち、EP の探索能力に関する頑健性が、当初見込まれていたものよりもかなり低いことが明らかになった。

本研究では、この問題から EP を実質的に解放するための手法を構築し、その進化ダイナミクスの特徴を観測することを目的とする。手法の基本的アイデアは、戦略パラメータが探索の比較的早い段階で 0 に近くなり集団が実質的に動かなくなる早期収束の状態に陥るという観察事実に着目し、これを遺伝的浮動の効果を利用して戦略パラメータが特定の値に固着しにくくする機構を導入するところにある。この EP の拡張手法は従来型の EP

* 原稿受付 2000 年 9 月 6 日

[†] 神戸大学 工学部 Faculty of Engineering, Kobe University; Rokkodai, Nada-ku, Kobe city, Hyogo 657-8501, JAPAN

[‡] 神戸大学 大学院 自然科学研究科 Graduate School of Science and Technology, Kobe University; Rokkodai, Nada-ku, Kobe city, Hyogo 657-8501, JAPAN

Key Words: evolutionary programming, real-valued function optimization, strategy parameters, genetic drift.

から下限設定に対する脆弱性を取り除くことになるため、Robust EP (REP) [11] と名付けることとする。また、進化ダイナミクスの特徴を把握するために、いくつかの統計量を計算し、探索挙動を従来手法と比較する。

本稿の構成を以下に示す。まず2.では、実関数最適化問題を定義し、従来のEPについて述べる。3.では、提案手法であるREPを定式化する。4.では、テスト問題を用いて計算機実験を行い、探索性能と進化ダイナミクスを明らかにする。5.では本論文をまとめる。

2. EPによる実関数最適化

2.1 実関数最適化問題

実関数最適化問題とは、 n 次元の実行可能領域 S ($S \subseteq R^n$) を持った有界な実数値目的関数 $f(\mathbf{x})$ において、 $f(\mathbf{x}_0)$ が最小値となる実数値決定変数ベクトル $\mathbf{x}_0$ ($\mathbf{x}_0 \in S$) を見つける問題である。これは、以下のように定式化される：

$$\mathbf{x}_0 = \arg \min_{\mathbf{x} \in S} f(\mathbf{x}) \quad (1)$$

なお、本論文では最小値を求める問題を取り扱うが、この制限は実数最適化問題の解法として一般性を失わない。なぜなら、最大値を求める問題は目的関数に負の記号をかけることによって、最小値を求める問題に置き換えることができるからである。

2.2 EPの計算手続き

EPのアルゴリズムを以下に示す。

- ステップ 1 世代 $g=0$ とし、 μ 個体からなる初期集団をランダムに発生させる。
- ステップ 2 親個体の適応度を目的関数 $f(\mathbf{x}_i)$ をもとに計算する。
- ステップ 3 μ 個体の親から μ 個体の子孫をそれぞれ再生する。
- ステップ 4 子孫の個体の適応度を $f(\mathbf{x}_i)$ をもとに計算する。
- ステップ 5 μ 個体の親と μ 個体の子孫の合計 2μ 個体からランダムに選ばれた q 個体の適応度を比較して各個体の勝ち点を計算する。 2μ 個体の集団から勝ち点の高い μ 個体を次世代の親として選択する。
- ステップ 6 終了条件を満たさない時は $g=g+1$ としステップ 3に戻る。

EPの計算手法自体は、D. B. Fogelによる定式化以来よく知られているが、最適化能力の向上をはかった研究はそれほど多くはない。これは、EPが頑健で十分な探索能力を持つためでなく、どの計算過程に拡張を加えればよいのか絞りがきれなかったためと筆者らは推察する。なお、近年においては、2.2.2に示すように、主として再生過程における突然変異のための確率分布様式に焦点を当てた例がいくつかある。

2.2.1 個体表現

n 次元の実関数最適化問題に対し、個体 $\mathbf{X}_i$ ($i=1,2,\dots,\mu$) を n 次元実数値ベクトル $\mathbf{x}_i$ と n 次元戦略パラメータベクトル $\boldsymbol{\eta}_i$ を用いて、以下のように定義する：

$$\mathbf{X}_i = [\mathbf{x}_i, \boldsymbol{\eta}_i] \quad (2)$$

$${}^t\mathbf{x}_i = (x_i(1), \dots, x_i(j), \dots, x_i(n)) \quad (3)$$

$${}^t\boldsymbol{\eta}_i = (\eta_i(1), \dots, \eta_i(j), \dots, \eta_i(n)) \quad (4)$$

$$(x_i(j) \in R, \eta_i(j) \in R^+)$$

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 $\mathbf{x}_i$ 、 $\boldsymbol{\eta}_i$ の第 j 成分である。

2.2.2 再生

オリジナルのEPでは μ 個の子孫 $\tilde{\mathbf{X}}_i$ を以下の式により、 μ 個の親 $\mathbf{X}_i$ から再生した [12]：

$$\tilde{x}_i(j) = x_i(j) + \eta_i(j)N_{ij}(0,1) \quad (5)$$

$$\tilde{\eta}_i(j) = \eta_i(j) + \alpha\eta_i(j)N_{ij}(0,1) \quad (6)$$

ここで、 $N_{ij}(0,1)$ は平均0、標準偏差1の標準正規分布から i, j ごとに得られる乱数値であり、それぞれ独立な標準正規分布に従う。また、 α は定数である。

近年におけるEPの研究では、後に示す log-normal 法 [12] を用いて η を更新し、その後、対応する実数値を計算するという sigma-first 方式 [6] が標準的計算手続きと認知されている。log-normal 法は、以下のように η を更新するものである：

$$\tilde{\eta}_i(j) = \eta_i(j) \exp(\tau' N(0,1) + \tau N_{ij}(0,1)) \quad (7)$$

ここで、 $N(0,1)$ は平均0、標準偏差1の標準正規分布から得られる各世代で共通な乱数値である。なお、これまでのほとんどの論文（たとえば、[4,14,17]）で言及されていないが、 $\tilde{\eta}_i(j)$ がある小さい正の下限値 ϵ より小さくなるときには、 $\tilde{\eta}_i(j) = \epsilon$ とする。定数 τ および τ' には、標準的な値として以下のように設定される [1]：

$$\tau = \left(\sqrt{2\sqrt{n}} \right)^{-1} \quad (8)$$

$$\tau' = (\sqrt{2n})^{-1} \quad (9)$$

EPの代表的な拡張手法の一つに、(5) 式の Gauss 分布の代りに Cauchy 分布を用いるものがある。これは Yao と Liu [14,15,17] が提唱し、FEP (Fast Evolutionary Programming) と呼んだ。その際、従来の Gauss 分布を用いる手法を CEP (Classical Evolutionary Programming) と呼んで区別した。この FEP は、以下のように定式化され、特に多峰性の問題に対し有効であることが示されている：

$$\tilde{x}_i(j) = x_i(j) + \eta_i(j)\delta_{ij} \quad (10)$$

ここで、 δ_{ij} は以下の Cauchy 分布の確率密度関数 $f(x)$ において $\alpha=1$ 、 $\beta=0$ として、 i, j ごとにそれぞれ独立

に得られる乱数値である：

$$f(x) = \alpha / \pi(\alpha^2 + (x - \beta)^2) \quad (11)$$

この報告に触発され、Gauss 分布と Cauchy 分布の確率密度関数を平均した分布 [2,3] や Levy の確率分布 [8] など他の確率分布に関する議論や、Gauss 分布と Cauchy 分布の両方を同時に用いる手法 [13,16,17] 等、様々な議論が展開されている。

3. Robust EP

CEP では、後の計算機実験で示すように、戦略パラメータが早いうちに 0 に近づき過ぎて集団が過度に収束してしまい、進化の方向が見つからないことが原因で探索がうまくいかない。しかし、FEP のように再生時に導入するランダムな変異量を単に増やしただけでは、この現象を回避することは困難である。さらに、この現象を避けるために設定する戦略パラメータの下限値は問題依存であって単純に定められない。

ここで、戦略パラメータが進化を通して適切に調整されるような新しい機構を導入した EP、すなわち REP を定式化する。これは、各個体の目的関数に対して中立な領域（以下に示すように、まれに活性化される）を用意し、その中の確率的变化が戦略パラメータ進化の主要因となる変異生成機構を備えた EP である。すなわち、REP は、中立域で集団中の遺伝子頻度が確率的に変化する遺伝的浮動 [7] の効果を利用したものであるといえる。なお、提案する冗長な個体表現とそこでの確率的操作を除き、他の計算手続きは CEP, FEP と全く同一である。

3.1 個体表現

n 次元の実関数最適化問題に対し、個体 $\mathbf{X}_i (i=1,2,\dots,\mu)$ を n 次元実数値ベクトル $\mathbf{x}_i$ と n 次元戦略パラメータベクトル $\boldsymbol{\eta}_{i0}$ および、冗長な m 個の n 次元戦略パラメータベクトル $\boldsymbol{\eta}_{il} (l=1,2,\dots,m)$ を用いて、以下のように定義する：

$$\mathbf{X}_i = [\mathbf{x}_i, (\boldsymbol{\eta}_{i0}, \dots, \boldsymbol{\eta}_{ik}, \dots, \boldsymbol{\eta}_{im})] \quad (12)$$

$${}^t\mathbf{x}_i = (x_i(1), \dots, x_i(j), \dots, x_i(n)) \quad (13)$$

$${}^t\boldsymbol{\eta}_{ik} = (\eta_{ik}(1), \dots, \eta_{ik}(j), \dots, \eta_{ik}(n)) \quad (14)$$

$$(x_i(j) \in R, \eta_{ik}(j) \in R^+)$$

ここで $j=1,2,\dots,n$, $k=0,1,\dots,m$ であり、 $x_i(j)$, $\eta_{ik}(j)$ は、それぞれベクトル $\mathbf{x}_i$, $\boldsymbol{\eta}_{ik}$ の第 j 成分である。

3.2 再生

まず、戦略パラメータベクトルの各要素に、以下に定める三つの操作 (g_{dup} , g_{del} , g_{inv}) を確率的に実施する：

$$g_{dup} : \eta'_{i0}(j) = \eta_{i0}(j) \quad (15)$$

$$\eta'_{il}(j) = \eta_{i(l-1)}(j)$$

$$\tilde{\eta}_{ik}(j) = D(\eta'_{ik}(j))$$

$$g_{del} : \eta'_{i(l-1)}(j) = \eta_{il}(j) \quad (16)$$

$$\eta'_{im}(j) = \min \left(\eta_{max}, \sum_{r=1}^m \eta_{ir}(j) \right)$$

$$\tilde{\eta}_{ik}(j) = D(\eta'_{ik}(j))$$

$$g_{inv} : \eta'_{i0}(j) = \eta_{i0}(j) \quad (17)$$

$$\eta'_{i0}(j) = \eta_{i0}(j)$$

$$\tilde{\eta}_{i0}(j) = D(\eta'_{i0}(j))$$

$$\tilde{\eta}_{i0}(j) = D(\eta'_{i0}(j))$$

ここで、 $l_0 \in \{1,2,\dots,m\}$ であり、戦略パラメータの上限 η_{max} は定数であり、 D は以下の (18) 式を表す：

$$\tilde{\eta}_{ik}(j) = \eta'_{ik}(j) \times \exp(\tau' N(0,1) + \tau N_{ijk}(0,1)) \quad (18)$$

ただし、 $N_{ijk}(0,1)$ は i,j,k ごとに得られるそれぞれ独立な標準正規分布に従う乱数値である。

これらは、戦略パラメータの不活性領域を短期記憶域として見れば、 g_{dup} は現在の戦略パラメータを記憶すること、 g_{del} はごく最近の生存に有効であった戦略パラメータを使うこと、 g_{inv} は以前の生存に有効であった戦略パラメータを使い、現在の値を記憶すること、に相当する。なお、(16) 式第 2 式では、不活性域の端に新しく挿入される値¹ を規定している。

子孫の実数値ベクトルは、以下の式により親から再生する：

$$\tilde{x}_i(j) = x_i(j) + \tilde{\eta}_{i0}(j) \delta_{ij} \quad (19)$$

ここで、 δ_{ij} は (10) 式にある δ_{ij} と同じ形式の乱数値である。

$x_i(j)$ は $\eta_{i0}(j)$ のみで定まるから、 $\eta_{il}(j), l=1,\dots,m$ における変化は目的関数に対して中立である。なお、各操作の適用確率 $P(g_*)$ をそれぞれ $P(g_{dup})=1.0$, $P(g_{del})=0.0$, $P(g_{inv})=0.0$ とすると、不活性戦略パラメータが活性化されることがなくなるため、FEP と全く等価になる。

4. 計算機実験

4.1 テスト関数と諸設定

Yao と Liu [14–17] によって整理されたよく使用される 23 個のテスト問題から、Table 1 に示す 5 個の問題を選び、CEP, FEP, REP を用いて実験を行った結果を以下に示す。(なお、他の関数に対しても同様な実験結果を得ている。) これら五つのテスト問題は、すべて 30 次元の探索空間を持つ。 $f_1(x)$, $f_2(x)$ は単峰性の関数、 $f_3(x) \sim f_5(x)$ は多峰性の関数である。これらの関数は原点において最適値 (最小値) 0 をとる。

¹ この値は基本的にどんな値でも可能ではあるが、他の不活性戦略パラメータとそれほどかけ離れた値にならないように設定した。

Table 1 Five test functions

Function	Expression($n=30$)	Range
Sphere Model	$f_1(x) = \sum_{i=1}^n x_i^2$	$-100 \leq x_i \leq 100$
Schwefel's Problem 2.22	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$-10 \leq x_i \leq 10$
Generalized Rastrigin's Function	$f_3(x) = \sum_{i=1}^n \{x_i^2 - 10 \cos(2\pi x_i) + 10\}$	$-5.12 \leq x_i \leq 5.12$
Ackley's Function	$f_4(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) + 20 + e$	$-32 \leq x_i \leq 32$
Generalized Griewank Function	$f_5(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$-600 \leq x_i \leq 600$

Table 2 The numbers of trials that EP does not get trapped in local optima

	f_3			f_4			f_5		
ϵ	CEP	FEP	REP	CEP	FEP	REP	CEP	FEP	REP
10^{-2}	0	0	0	0	0	0	0	0	0
10^{-4}	0	24	50	0	50	50	12	5	21
10^{-6}	0	0	50	0	34	50	12	3	17
10^{-8}	0	0	50	0	2	50	12	3	19
10^{-10}	0	0	50	0	0	50	12	5	18
0	0	0	50	0	0	50	5	1	18

CEP, FEP, REPの各実験において、戦略パラメータの探索範囲を ϵ から η_{max} 、実数パラメータ探索域をTable 1にある探索範囲として、一様にランダムに生成した同一の初期集団を用いた。なお、探索範囲の狭い $f_3(x)$ のみ $\eta_{max}=1.0$ とし、それ以外は $\eta_{max}=3.0$ とした。親個体数 $\mu=100$ 、トーナメントサイズ $q=10$ を用い、最終世代5000として、各EPの実験を50回ずつ行った。なお、REPでは、冗長な戦略パラメータ数 m は5とし、導入した再生時のオペレータの適用確率は $P(g_{dup})=0.6$ 、 $P(g_{del})=0.3$ 、 $P(g_{inv})=0.1$ とした。

はじめに、戦略パラメータの下限值の変化に対する探索性能の違いを検証するために、下限値 ϵ を 10^{-2} 、 10^{-4} 、 10^{-6} 、 10^{-8} 、 10^{-10} 、0と六つの値に変化させて計算機実験を行った。つぎに、最も基本的な f_1 を例に取り上げ、 $\epsilon=0$ の場合を取り上げて計算機実験を行い、集団の進化ダイナミクスの特徴を観測した。

4.2 探索性能

テスト関数 $f_1, \dots, f_5$ のCEP, FEP, REPによる50試行の平均結果をそれぞれFig. 1, Fig. 2, Fig. 3に示す。縦軸には関数値を、横軸には世代をとった。

Fig. 1(a), Fig. 2(a)にあるように、単峰性関数 f_1 のCEPとFEPは、 $\epsilon \leq 10^{-4}$ の場合、200世代付近で早期

収束の傾向がみられ最適解への接近速度が極端に落ちる。一方、Fig. 3(a)から、REPでは、異なる ϵ に対しそれほど探索速度の差異はなく、 ϵ が小さいほど計算世代が進むにつれ精度よく最適解に近づくことがわかる。Fig. 1(b), Fig. 2(b), Fig. 3(b)では、 f_2 における結果を示しているが、 f_1 での結果と同様な傾向が観測されている。

多峰性関数 f_3 の場合では、Fig. 1(c), Fig. 2(c)からわかるように、CEP, FEPともに300世代以前で局所解に陥る。一方、Fig. 3(c)に示されたREPの結果からは、単峰性関数の傾向と同様、 ϵ の違いによる探索速度の差異もほとんどなく、小さい ϵ を用いるほど精度良く大域最適解に近づくことがわかる。また、Fig. 1, Fig. 2, Fig. 3のうちのそれぞれ(d)(e)からわかるように、 f_4 , f_5 でも同様の結果が得られている。

なお、Table 2には、多峰性関数において、局所解に陥らなかった試行回数を比較している。これより、明らかに、CEPとFEPに比べREPは下限設定にほぼ関係なく局所解に陥りにくいことがわかる。

4.3 進化ダイナミクス

最適化ツールとしてECを用いる際、探索点集団中の適切な多様性を保つことは非常に重要である。EPでは、その世代の親集団と子集団の和から次世代集団を生成す

(a) CEP for f_1 (a) FEP for f_1 (a) REP for f_1 (b) CEP for f_2 (b) FEP for f_2 (b) REP for f_2 (c) CEP for f_3 (c) FEP for f_3 (c) REP for f_3 (d) CEP for f_4 (d) FEP for f_4 (d) REP for f_4 (e) CEP for f_5 (e) FEP for f_5 (e) REP for f_5

Fig. 1 The averaged best results of CEP when the lower bounds are 10^{-2} , 10^{-4} , 10^{-6} , 10^{-8} , 10^{-10} and 0

Fig. 2 The averaged best results of FEP when the lower bounds are 10^{-2} , 10^{-4} , 10^{-6} , 10^{-8} , 10^{-10} and 0

Fig. 3 The averaged best results of REP when the lower bounds are 10^{-2} , 10^{-4} , 10^{-6} , 10^{-8} , 10^{-10} and 0

るが、集団の探索様式、すなわち進化的振る舞いを調べたい時には、新しく生成された探索点である子集団に関して観測すればよい。ここでは、最も基本的関数である f_1 を用いて、子集団のいくつかの統計量を観測し、各 EP の進化的振る舞いの特徴の違いを把握する。なお、これらの特徴がより明白に現れるようにするために、 $\epsilon=0$ として以下の実験を行った。結果を Fig. 4~Fig. 9 に示す。

4.3.1 関数値

Fig. 4(a), Fig. 5(a), Fig. 6(a) に子集団中のすべての個体の関数値 $f_1(\mathbf{x})$ を点でプロットしたものを示す。CEP では、子集団中のほとんどの個体がよく似た関数値を取っている。FEP では、そのばらつきは CEP よりは大きい。最適解へ方向が見つけられないことがしばらく続くと、CEP と同じ傾向となる。また、CEP, FEP ともに、最適解への接近の仕方が断続的で、最終世代までに十分な解の精度が得られない。一方、REP では、最適解への接近が継続的に行われ、最終的に 10^{-19} まで小さくなる。また、各世代での関数値の分布にも広がりがあり、様々な関数値をとる子孫を常々生成していることがわかる。

4.3.2 変数値・戦略パラメータ

実数値変数および戦略パラメータの推移をみる。 f_1 は超球関数であるから $x_i(j)$, $\eta_i(j)$ はすべての j に対して対称である。それゆえ、 $x_i(0)$, $\eta_j(0)$ (REP においては $\eta_{i0}(0)$) の結果を観測することとし、それらを Figs. 4~6 の (b) および (c) に示す。

一連の図の (b) からわかるように、CEP, FEP ではかなり早い時期に子集団すべての $x_i(0)$ が 0 近傍に集中するようになり、それ以外の点はその後生成されなくなる。REP においては、計算世代がかなり進んでも、0 から遠い値を取る個体も生成する傾向を示すことがわかる。

同様に、一連の図の (c) から、CEP, FEP では、子集団全体で $\eta_i(0)$ が次第に小さくなっていく。これは、前述の $x_i(0)$ の分布が覆う範囲が次第に狭くなっていくことを裏付けている。一方、REP においては、 $\eta_{i0}(0)$ が世代の経過とともに非常に小さくなるものが多いが、それだけでなく、 10^6 程度の大きさの差をそれぞれ持たせて第 2 グループ、第 3 グループ、... が存在する。おのおのがその戦略パラメータサイズを探索戦略とすることから、集団が複数の戦略を同時に使用していることがわかる。

さらに、一連の図の (d)(e) に示した $x_i(0)$, $\eta_i(0)$ の標準偏差の推移はそれを裏付けている。すなわち、CEP, FEP ともに標準偏差が小幅な振れはあるものの基本的に小さくなっていくが、REP では各世代ごとの振れが次第に大きくなっていく。

なお、ここには示さないが、他の 29 の変数に関しても同様の特徴が現れることを確認している。

4.3.3 Hotelling の T^2

集団内の広がり具合をはかる統計量のひとつに Hotelling の T^2 がある。これは多次元データ集団の中心

から各データ点までの多変量距離を表す尺度であり、この値が大きい個体はデータ群の中心から遠いところに位置し、逆にこの値が小さい個体はデータ群の中心に位置する。それゆえ、大域探索・局所探索のバランスをはかることができる。この Hotelling の T^2 は、以下のように計算される：

$$T^2 = \mu(\mu-1)(\bar{\mathbf{x}}-\boldsymbol{\delta})' \mathbf{S}^{-1}(\bar{\mathbf{x}}-\boldsymbol{\delta}) \quad (20)$$

$$\bar{\mathbf{x}} = \sum_{i=1}^{\mu} \frac{\mathbf{x}_i}{\mu} \quad (21)$$

$$\mathbf{S} = \sum_{i=1}^{\mu} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})' \quad (22)$$

ここで、 μ を個体数とし、 $\mathbf{x}_i$ は平均ベクトル $\boldsymbol{\delta}$ と共分散行列 $\boldsymbol{\sigma}_{\zeta}$ を持つ n 次元正規分布 $N(\boldsymbol{\delta}, \boldsymbol{\sigma}_{\zeta})$ に従うものとする。なお、本実験ではこれは成り立たないが近似的に仮定できるとして用いる。ただし、 $\boldsymbol{\delta}$ は、 $\mathbf{x}_i$ が以下の密度関数 $f(\mathbf{x})$ を持つとき、その期待値として計算される：

$$f(\mathbf{x}) = (2\pi)^{-n/2} |\boldsymbol{\sigma}_{\zeta}|^{-1/2} \times \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\delta})' \boldsymbol{\sigma}_{\zeta}^{-1}(\mathbf{x}-\boldsymbol{\delta})\right] \quad (23)$$

$$E(\mathbf{x}_i) = \boldsymbol{\delta} \quad (24)$$

Figs. 7~9 に、各 EP における子集団の Hotelling の T^2 のスナップショットを示す。各図の横軸には子孫 100 個体を Hotelling の T^2 をもとに降順に並べた。CEP の結果を示した Fig. 7 から、CEP では、Hotelling の T^2 は各スナップショットにおいておよそ 10 から 60 の間で分布していることがわかる。この分布様式は世代が進んでもほとんど変わらない。FEP のそれは Fig. 8 にある。CEP より広い範囲に分散してはいるが、世代を経ても集団の分布様式にはあまり大きな違いが現れない。一方、REP の分布様式を示した Fig. 9 からは、明らかに CEP, FEP のそれらとは違う傾向が見て取れる。REP では、計算初期の分布は FEP に似ているが、次第に 95 以上の大きい値をとるものと 5 以下のごく小さい値を取るものとの割合が増加していく。すなわち、集団の分布において、相対的に広く探索する個体群と局所的に探索する個体群におおまかに分かれることがわかる。

以上より、CEP, FEP では探索初期からその探索戦略を基本的に変えないが、REP ではその世代での実質的探索域に対し広く探索する個体群と中心部分を重点的に探索する個体群に分かれていくことが読み取れる。

4.4 実験結果 (まとめ)

以上より、CEP, FEP, REP の探索特徴を以下のようまとめることができる。

- 単峰性、多峰性関数に関係なく CEP, FEP の探索速度は ϵ の大きさに依存するが、REP の探索速度は ϵ の大きさに依存しない。
- CEP, FEP の戦略パラメータは徐々に小さくなり

Fig. 4 Offspring of CEP for f_1 when $\epsilon = 0$ Fig. 5 Offspring of FEP for f_1 when $\epsilon = 0$ Fig. 6 Offspring of REP for f_1 when $\epsilon = 0$

Fig. 7 The Hotelling's T^2 of CEP for f_1 when $\epsilon = 0$ Fig. 8 The Hotelling's T^2 of FEP for f_1 when $\epsilon = 0$ Fig. 9 The Hotelling's T^2 of REP for f_1 when $\epsilon = 0$

多様性を失うため早期収束の状態に陥りやすくなる。しかし、REPの戦略パラメータは適切なレベルの多様性を維持できるため、進化の方向を見失うことなく最適解に近づくことができる。

- CEP, FEPは、探索戦略を計算世代によって変えないが、REPでは集団を実質的探索域で相対的に広く探索するものと中心部分を重点的に探索するものとに次第に分化していく。

5. おわりに

EPにおける戦略パラメータの下限值に対する脆弱性を取り除くことを目的として、EPの一拡張方式REPを構築した。REPでは、戦略パラメータの進化要因として遺伝的浮動が加えられるように、冗長な戦略パラメータを持つ個体表現と、それらに対するCauchy型突然変異

を含む確率的な順序入れ換え操作を導入した。さらに、標準的なテスト問題を用いた計算機実験を通して、REPによる探索性能の向上、戦略パラメータ下限値に対する頑健性の検証、および基本的な進化的振る舞いの観測を行った。

本論文では、REPで使うパラメータのファイン・チューニングに関して深い考察は行っていない。計算機実験で使用したパラメータ・セットは予備実験結果から設定したものである¹。今後、パラメータ・セットと進化ダイナミクスの解析をさらに進めることが必要であると考えている。また、本論文でテスト問題を用いて検証された探索性能の向上が実応用問題でどの程度有効に働くのか検証したいとも考えている。

謝 辞

本研究の一部は、日本学術振興会未来開拓学術プロジェクト、創発的シンセシスの方法論(96P00702)の研究助成を受けて行われた。

参 考 文 献

- [1] T. Bäck and H. -P. Schwefel: An overview of evolutionary algorithms for parameter optimization; *Evolutionary Computation*, Vol. 1, No. 1, pp. 1–23 (1993)
- [2] K. Chellapilla: Combining mutation operators in evolutionary programming; *IEEE Transactions on Evolutionary Computation*, Vol. 2, No. 3, pp. 91–96 (1998)
- [3] K. Chellapilla and D. B. Fogel: Fitness distribution in evolutionary computation — Analysis of local extrema in the continuous domain; *Proc. of Congress on Evolutionary Computation (CEC'99)*, IEEE Press, pp. 1885–1892 (1999)
- [4] D. B. Fogel: *Evolutionary Computation Toward a New Philosophy of Machine Intelligence*, IEEE Press (1995)
- [5] D. B. Fogel, E. C. Wasson, E. M. Boughton, V. W. Porto and P. J. Angeline: Linear and neural models for classifying breast cancer; *IEEE Transactions on Medical Imaging*, Vol. 17, No. 3, pp. 485–488 (1998)
- [6] D. K. Gehlhaar and D. B. Fogel: Tuning evolutionary programming for conformationally flexible molecular docking; *Proc. of the 5th Annual Conference on Evolutionary Programming*, IEEE Press, pp. 419–429 (1996)
- [7] M. Kimura: *The Neutral Theory of Molecular Evolution*, Cambridge University Press (1983)

¹ 計算機実験で使用したパラメータ値の他に ($P(g_{dup})$, $P(g_{del})$, $P(g_{inv})$) = (0.3, 0.1, 0.03), (0.9, 0.6, 0.3) 等を試しているが、大きな性能の劣化は観測されなかった。 $P(g_{dup}) > P(g_{del}) > P(g_{inv})$ と設定すればおおまかに適切なパラメータが得られることは確認している。

- [8] C. -Y. Lee and Y. Song: Evolutionary programming using the levy probability distribution; *Proc. of Genetic and Evolutionary Computation Conference (GECCO'99)*, Morgan Kaufman, pp. 886–893 (1999)
- [9] K. -H. Liang, X. Yao, Y. Liu, C. Newton and D. Hoffman: An experimental investigation of self-adaptation in evolutionary programming; *Proc. of the 7th Annual Conference on Evolutionary Programming*, Springer-Verlag, pp. 291–300 (1998)
- [10] K. -H. Liang, X. Yao and C. Newton: Dynamic control of adaptive parameters in evolutionary programming; *Proc. of the 2nd Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'98)*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1585, pp. 42–49 (1998)
- [11] Y. Matsumura, K. Ohkura and K. Ueda: Evolutionary programming with non-coding segments for real-valued function optimization; *Proc. of 1999 IEEE International Conference on Systems, Man and Cybernetics (SMC'99)*, Vol. 4, pp. 242–247 (1999)
- [12] N. Saravanan and D. B. Fogel: Learning strategy parameters in evolutionary programming —An empirical study; *Proc. of the 3rd Annual Conference on Evolutionary Programming*, World Scientific Publishing, pp. 269–280 (1994)
- [13] N. Saravanan and D. B. Fogel: Multi-operator evolutionary programming —A preliminary study on function optimization; *Proc. of the 6th Annual Conference on Evolutionary Programming*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1213, pp. 215–221 (1997)
- [14] X. Yao and Y. Liu: Fast evolutionary programming; *Proc. of the 5th Annual Conference on Evolutionary Programming*, MIT Press, pp. 451–460 (1996)
- [15] X. Yao, G. Lin and Y. Liu: An analysis of evolutionary algorithms based on neighborhood and step sizes; *Proc. of the 6th Annual Conference on Evolutionary Programming*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1213, pp. 297–307 (1997)
- [16] X. Yao and Y. Liu: Scaling up evolutionary programming algorithms; *Proc. of the 7th Annual Conference on Evolutionary Programming*, Springer-Verlag, pp. 103–112 (1998)
- [17] X. Yao, Y. Liu and G. Lin: Evolutionary programming made faster; *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, pp. 82–102 (1999)