

並列遺伝的アルゴリズムによる多峰性関数の最大値探索

山本富士男・荒木智行

情報工学科

Finding the Maximum of Multimodal Functions by a Parallel Genetic Algorithm

Fujio YAMAMOTO, Tomoyuki ARAKI

Abstract

Finding out the maximum of the De Jong's No.5 function (a kind of multimodal function) was performed by a parallel genetic algorithm accompanied with intermittent migration. In a hundred trials of evolutionary computation, we obtained the correct answer 95 times up to 50 generations, and 97 times up to 500 generations. These results prove that this method maintains non-uniformity of individuals in a long period as compared both with parallel evolutionary computation without any migration and serial evolutionary computation treating the whole individuals as a single population.

Key Words: Genetic Algorithm, Parallel Computing, Maximum

1. まえがき

本論文の目的は、多くの極大点を有する関数(多峰性関数)の最大値探索に対する、並列遺伝的アルゴリズムの適用性を明らかにすることにある。

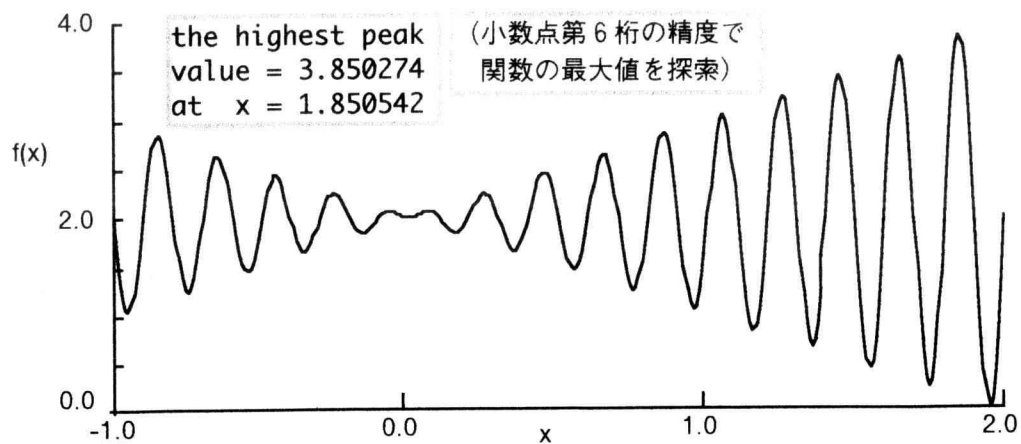
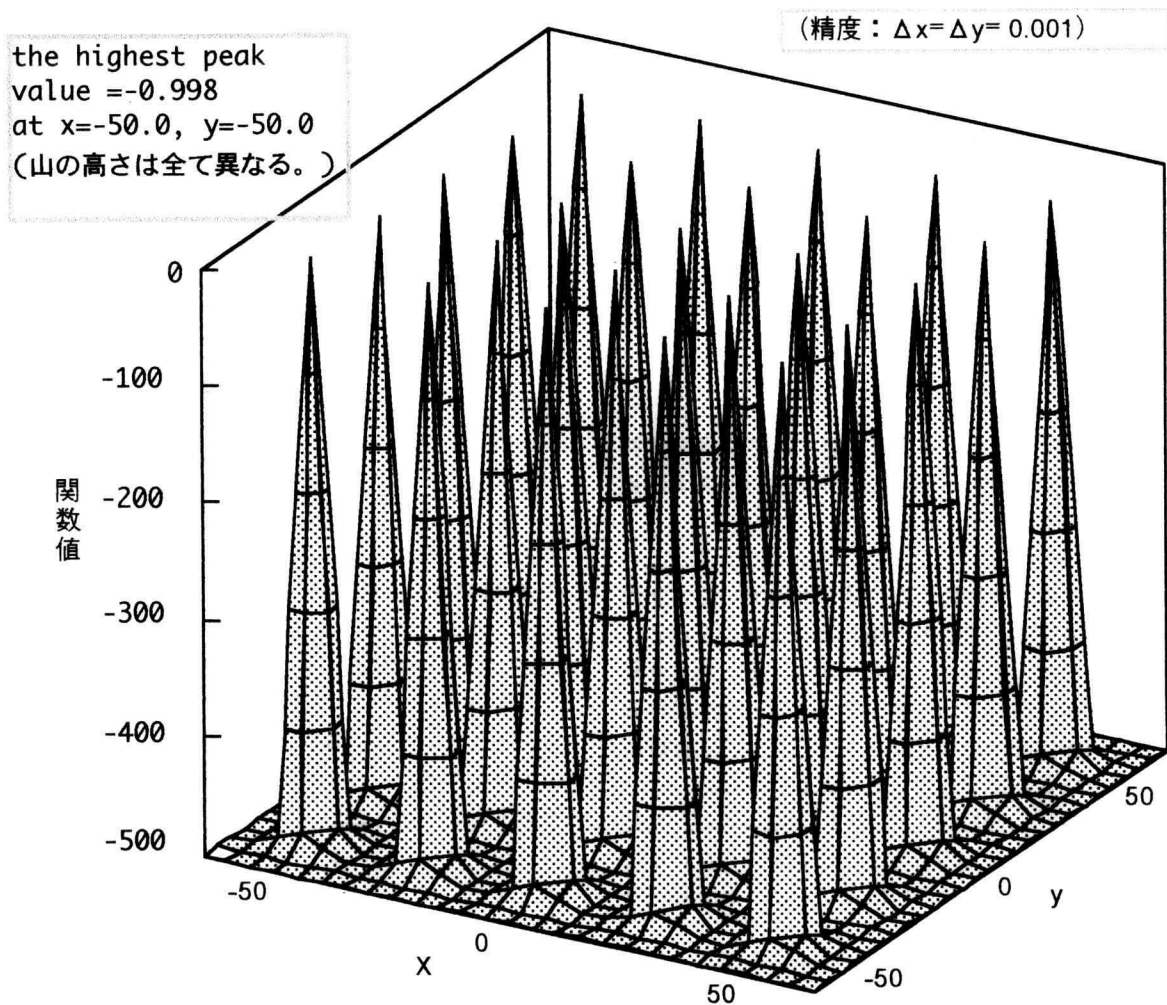
近年、遺伝的アルゴリズムは、工学系や社会系の広い分野で、各種の最適化問題などに適用されている。その多くの場合において、予想外の優れた探索能力を示すことが報告^{1)~13)}されている。しかしながら、その最大の難点は、遺伝操作に係わる個体数、交叉確率、突然変異確率、次世代用個体選択法などのパラメータを、試行錯誤や経験則で決める必要があることである。さらに、これらのパラメータを一応決めたとしても、毎回の試行で(乱数系列が異なる場合)一定の安定した結果が得られるとは限らない。すなわち、乱数系列によっては、局所解へ早期に収束してしまう場合が多い。

局所解から脱する手法^{1), 3), 4)}も色々と提案されている。そのねらいは、集団の進化を促進させると同時に、各個体が急速に均質化することを避け、個体の多様性を長く持続させることにある。これらの手法の多くは、個体の適合度の評価および次世代個体の選択法に関するもので

ある。

一方、コンピュータの急速な発展に伴い、パソコンやワークステーションの幾つかをクラスタ化して並列動作させること、さらには、(超)並列計算機を使える環境も普及しはじめた。そのような環境を使うならば、「個体の多様性を長く持続させる」ことはもっと自然に行うことができると考えられる。つまり、各計算機(または各CPU)に個別の個体群を独立に計算させ、必要に応じて適当な間隔で交流する(移民する)という考え方^{1)~6), 10)}である。これは、実際に多数の生物集団が、それぞれほぼ独自の進化を遂げ、適度に集団間で交流が行われることを模倣するものである。

本論文では、このような考えに基づく遺伝的アルゴリズム(以後、並列遺伝的アルゴリズムと呼ぶ)の有効性を具体的に明らかにする。対象とする例題は、多峰性関数の大域的な最大値を探索する問題^{14), 15)}とした。この種の問題は、通常の遺伝的アルゴリズムでは、上記に述べたように、計算の途中で局所的な解へ早期に収束する可能性が高いので、並列遺伝的アルゴリズムの能力を検証するのに適当であると考えた。

(a) 1変数多峰関数 ($f(x)=x \cdot \sin(10\pi \cdot x)+2.0$)

(b) 2変数多峰関数 (DeJongの第5関数)

図1 最大値探索の対象とした関数の形状

2. 対象とした多峰性関数

まず、どのような問題を解くのか、それをどのように遺伝子表現するのかを説明する。

2.1 多峰性関数の形状

対象とする関数は、図1(a)、(b)に示すものである。このうち、(a)は1変数の波型の関数であり、変数 x の区間 $[-1.0, 2.0]$ で、関数の最大値を小数点6桁の精度で求める。この問題は、文献¹⁾などで紹介されており、幾つかの工夫を施した遺伝的アルゴリズムで比較的容易に正解が得られている。一方、(b)は下記に示す2変数の多峰性関数(DeJongの第5関数)であり、ベンチマークとして使われている。^{1), 4)}

$$f(x, y) = - \left[0.002 + \sum_{j=1}^{25} \frac{1}{j + (x - a_j)^6 (y - b_j)^6} \right]^{-1}$$

$$-65.535 \leq x, y \leq 65.536$$

$$\Delta x = \Delta y = 0.001$$

ここで a_j, b_j は、剣山(極大点)の位置を決める定数である。ここでは図にあるとおり、領域内に25個の剣山があり、その頂点の高さは全て異なる。 $[-50.0, -50.0]$ の点で大域的な最大値-0.998を取る。この最大値を求める。これは、通常の遺伝的アルゴリズムにとってかなり難しい問題とされている。他の剣山を局所的な解として早期に収束しやすく、剣山の谷間をとおり抜けて最高峰に達するのが難しいのである。

そこで、本論文では以後、この(b)の多峰性関数の最大値探索を取り上げることにする。

2.2 解(x,y)の遺伝子表現

文献⁴⁾にあるとおり、上記に示した変数 x の領域範囲と x 標軸方向の刻みから、17ビットの0の列、すなわち、000000000000000000を-65.535に対応付け、1ビット分の増加を、 Δx の増加に対応づける。変数 y についても同様にする。 xy 平面上の点 (x, y) は、この二つのビット列を繋げたものとする。すなわち、解 (x, y) の遺伝子表現は、34ビットの列とすることができる。 $f(x, y)$ を求めるには、このビット列の値を2進から10進に変換する必要がある。

3.1 基本的遺伝操作

今回用いる遺伝操作は、基本的には多くの文献などで定着しているものである。すなわち、(1)集団の個体を初期化した後、(2)交叉、(3)突然変異、(4)適合度評価、(5)次世代個体の選択、からなる遺伝操作を k 世代まで繰り返す。このうちのいくつかについて、以下に若干説明する。

交叉は、最も単純な1点交叉とする。適合度としては、関数値 $f(x, y)$ をそのまま使う。選択については、最良個体は常に次世代に残す、いわゆるエリート選択と、ランキング選択を合わせたような方式とした。すなわち、個体数が n で、交叉の結果 m 個(交叉確率による)の個体が新たにできたとする。 $(m+n)$ 個の個体を適応度が高い順にソートする。次世代には n 個の個体だけを選択して残す。上位から第 s 番目(最上位は $s=0$)の個体を選択される確率を、 $(m+n-s)/(m+n)$ とした。したがって、最下位の個体でも、 $1/(m+n)$ の確率で選択される可能性が残る。また、 n 個になるまでこの選択を繰り返すので、同一個体が複数選択されることもある。

3.2 進化のさせ方の3つの枠組み

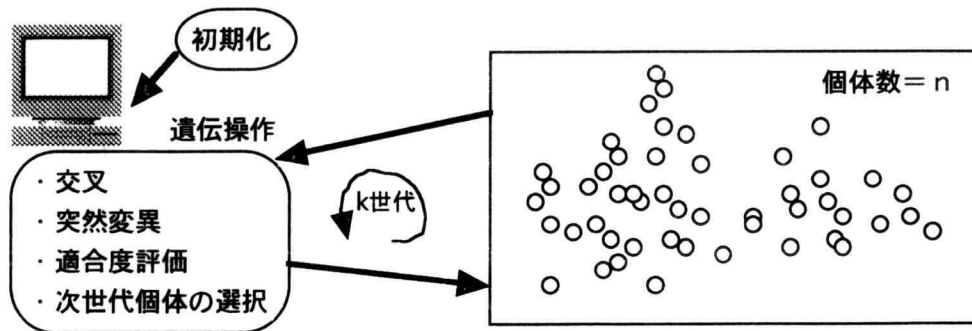
上に述べた遺伝操作を用いて、個体を進化させる枠組みを図2(a)、(b)、(c)に示した。最初の(a)全体一括型は、単一コンピュータ上で全ての個体(個体総数 n)を一つの集団として扱う。

一方、(b)完全分離型と、(c)並列移民型は、並列コンピュータの使用を前提としている。 p 個のプロセッサを使う場合、各プロセッサは、 n/p 個の個体からなる部分集団をそれぞれ進化させる。この場合、個体の初期化も、使用する乱数系列も各プロセッサで異なるように設定する。これにより、各部分集団は独自の進化を遂げることができる。期待した適合度の個体が出現した場合、全ての計算を停止させるため、各集団(各プロセッサ)は、適当な世代間隔毎に、他の集団の最良個体の情報を得られるようにする。

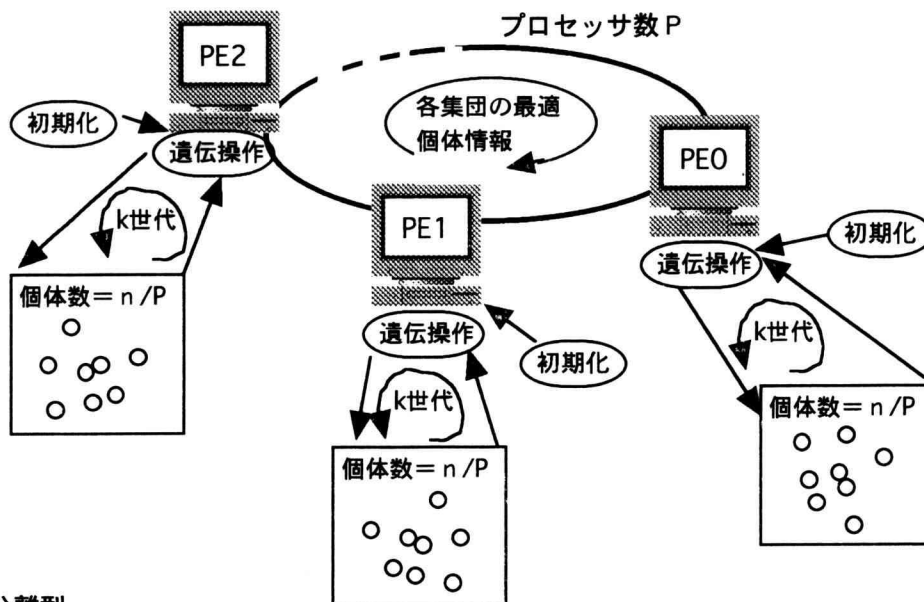
完全分離型では、各部分集団は全く独立に進化するのに対し、並列移民型では、適当な世代間隔毎にいくつかの個体は他の部分集団へ移動する(移民)。これにより、各部分集団がそれぞれの局所解へ早期に収束することを防ぎ、集団全体の多様性を長く持続させることを狙う。移民効果の強さは、移民間隔(世代間隔)と移民率(移動する個体数の割合)で決まるが、わずかな移民でも効果的であるという報告がある。⁵⁾

本論文では、各プロセッサが論理的にリング状に接続されているモデルを使い、移民は左にある部分集団へ向かうとした。

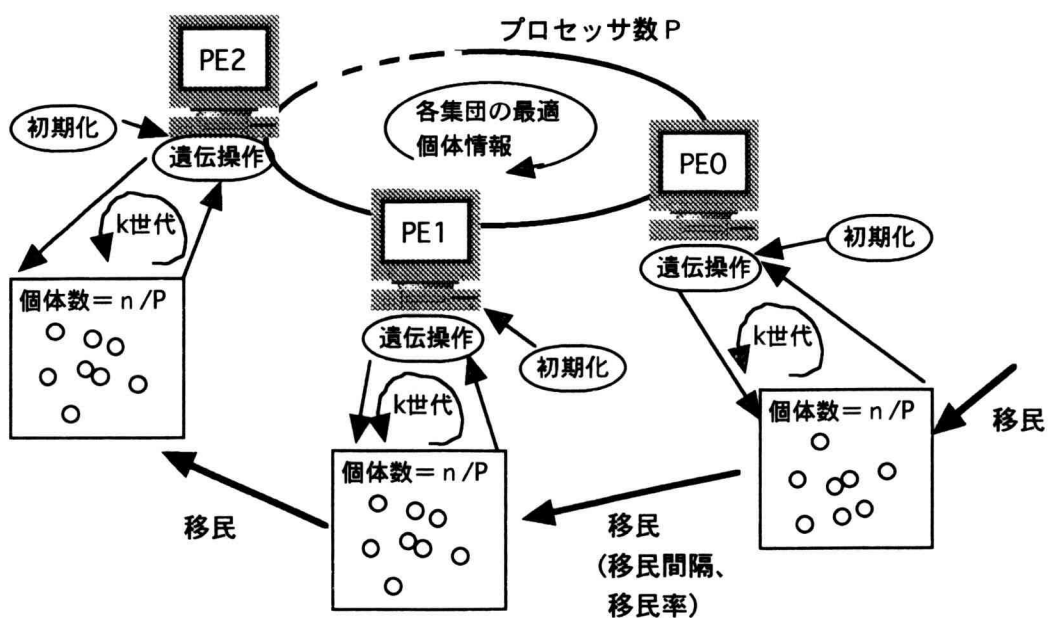
3. 個体の進化のさせ方



(a) 全体一括型



(b) 完全分離型

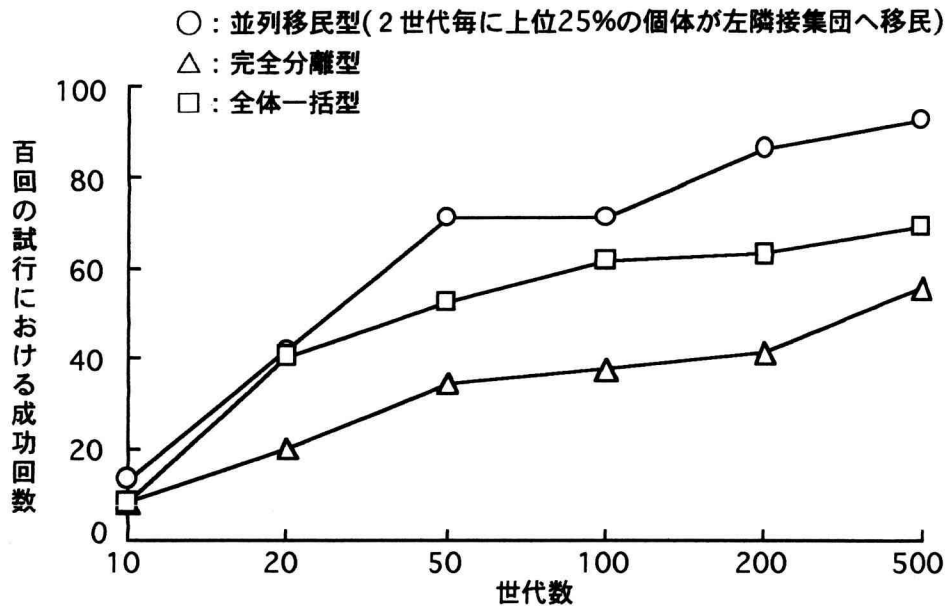


(c) 並列移民型

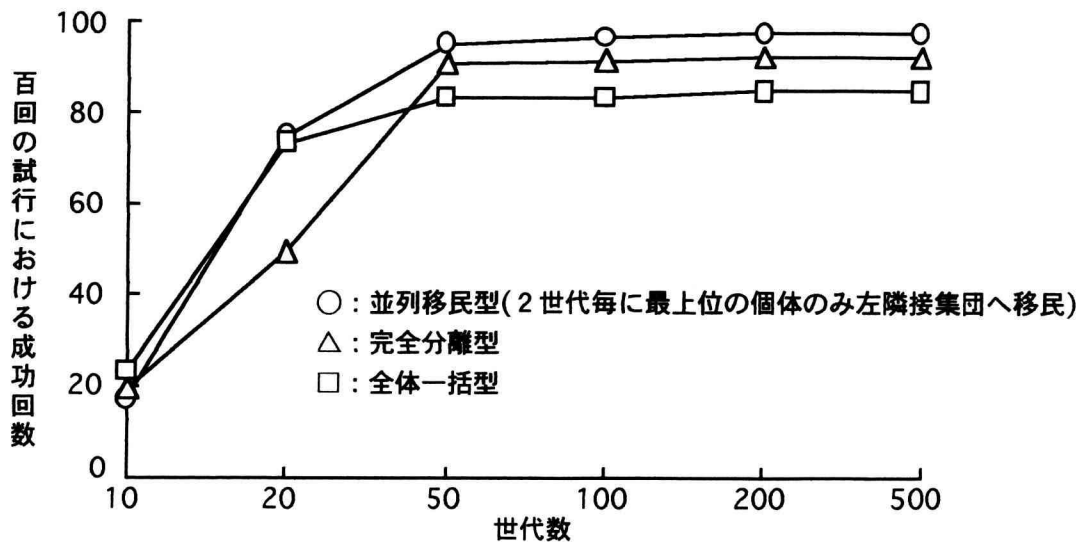
図2 進化のさせ方の3つの枠組み

計算条件：

- (1) 遺伝子の長さ=34ビット、初期化率=0.5、交叉率=0.5
- (2) 集団の大きさ=400、(ただし、並列に進化の場合は、8集団に均等分割。)
- (3) 各世代で、各集団の最適個体の値が、一つでも関数の最大値に達していれば (精度：小数点4桁)、成功とする。



(a) 突然変異率= 0.005の場合



(b) 突然変異率= 0.05の場合

図3 DeJong第5関数の最大値探索成功回数の比較

表1 各枠組みでの500世代までの計算時間(日立製SR2201使用)

諸元 進化の枠組み	1PE当りの 個体数	使用プロセッサ数	計算時間
(a)全体一括型	400	1	51.27秒
(b)完全分離型	50	8	3.88秒
(c)並列移民型	50	8	4.06秒

(計算条件)

1. 遺伝子長=34ビット,
初期化率=0.5,
交叉率=0.5,
突然変異率=0.05
2. (c)では、2世代毎に
最上位の個体のみが、
左隣接集団に移民。

3.3 計算時間

上記に述べた3つの進化の枠組みの計算時間を最初に調べておく。使用計算機は、日立製超並列コンピュータSR2201である。ここでは1~8台のプロセッサを使用した。500世代までの計算を一回行うのに要した処理時間の実測結果を表1に示す。3つの枠組みで、個体に対する遺伝操作の総回数が同一になるよう、プロセッサ当たりの個体数は、(a)全体一括型では400、(b)完全分離型と(c)並列移民型では50とした。

(b)、(c)では8台のプロセッサを使っているが、1台のプロセッサを使う(a)に対し、計算速度は12倍程度になっている。これは、(b)、(c)が扱うプロセッサ当たりの個体数が少なくなったため、遺伝操作の処理に含まれるソートや標準偏差の計算などが線形以上にスピードアップする効果である。また、(c)では移民のためのプロセッサ間通信が起こるため、(b)よりも1.05倍遅くなっているが、この程度の負荷増はほとんど問題にならないと考えられる。

4. 最大値探索性能の実験的比較

4.1 実験方法

進化の3つの枠組みの性能を、第2章で述べた問題で検討する。世代数は最大500まで調べる。遺伝子の長さは2.1節のとおり、34ビットとなる。初期化率(初期個体の遺伝子34ビットのうち1にする割合)=0.5、交叉率=0.5で実験する。プロセッサ当りに割当ててる個体の数は、3.3節で示したとおりである。

個体群を部分集団に分割する2種、(b)完全分離型と(c)並列移民型では、何れかの部分集団の最適個体が正解を与えた場合に正解とする。500世代までの計算を、乱数の種を変えてそれぞれ100回実行し、そのうちの何回正解が得られたかを中心に調べる。これは、第1章に述べたとおり、乱数系列の違いによって、得られる結果にどの程度変動があるかを明かにするためである。

4.2 実験結果

図3(a)、(b)に結果を示す。ケース(a)では、突然変異率=0.005とした。すなわち、突然変異の確率は遺伝子長34に対してはかなり低く、突然変異の効果は低く抑えられる場合である。この場合、並列移民型では、2世代毎に上位25%の個体が左隣接集団に互いに移民するように設定した。3種類の進化の枠組みのうちで、並列移民型がどの世代においても、最も優れた結果を示した。50世代で71%、500世代では92%の割合で正解が得られている。次に成績が良いのは、全体一括型であるが、500世代でも69%の正解率に過ぎない。完全独立型はかなり成績が悪く、各集団毎に早期に局所解へ収束したことを窺われている。

一方、突然変異率をケース(a)より1桁大きくしたケース(b)では、3種類の型とも成績がかなり向上した。しかしながら、50世代以降は何れの型も成績がほとんど向上していない。これは、突然変異は確かに局所解を脱出させる効果があるものの、それだけでは限界があることを示している。図の説明にあるとおり、並列移民型では、ケース(a)の場合より、かなり少ない度合いの移民しか行っていない。しかし、このわずかな移民がやはり効いており、ケース(b)でも並列移民型が最も優れた結果を示した。すなわち、50世代ですでに95%、500世代では97%の正解率に達した。

4.3 「並列移民型」と「完全独立型」の比較

上記に述べたように、図3(b)、すなわち突然変異がある程度起こる場合には、一見すると「並列移民型」と「完全独立型」とあまり差異がないように見える。しかし、解の性質、すなわち、集団の各個体の特質には大きな差が見られることが分かった。すなわち、完全独立型の場合は、解が得られたといっても、ある特定の部分集団のいくつかの個体が正解に達しているだけで、他の集団の個体は別の局所解に収束している場合が多いことが分かった。これに対して、並列移民型の場合で正解が得られた場合は、ほとんど全ての部分集団が正解に達しており、なおかつ、集団毎に多様性を保持しているということである。この事実は、並列移民型の方が、異なる

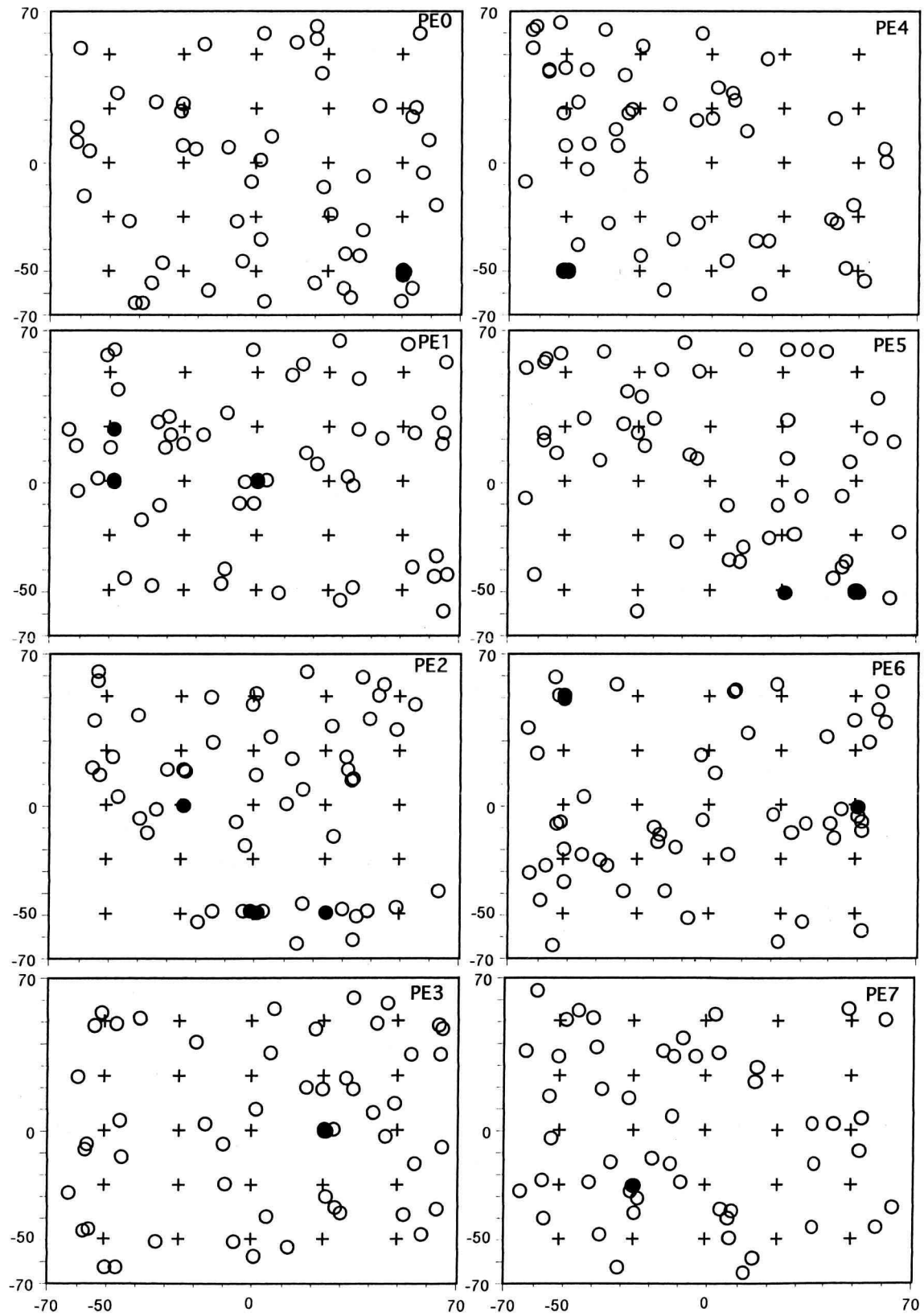


図4 完全分離型の個体進化 (+関数の極大点、○初期個体、●10世代後の個体(重なっている))

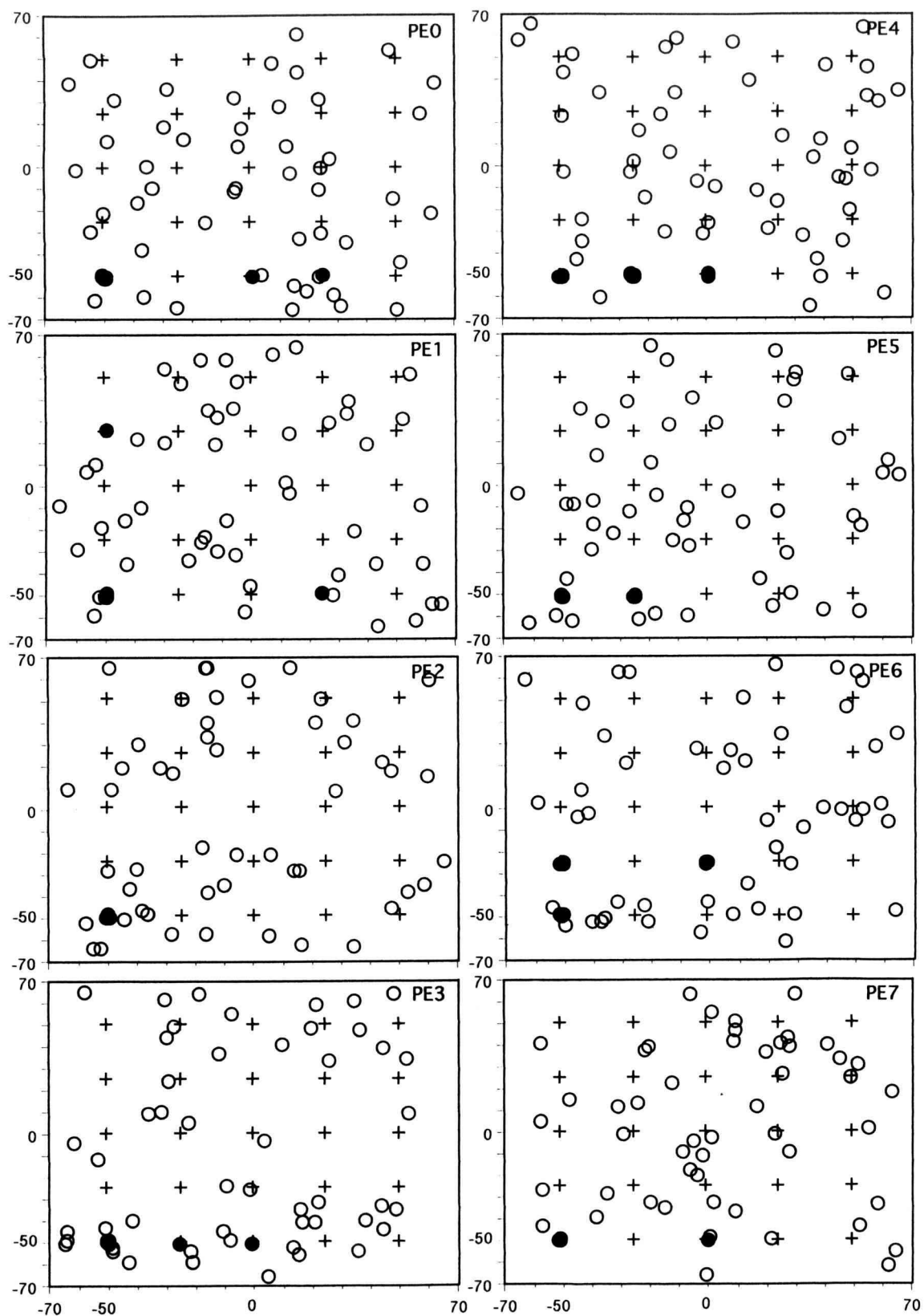


図5 並列移民型の個体進化 (+関数の極大点、○初期個体、●10世代後の個体(重なっている))

乱数系列に対しても安定して正解を与える要因となっていると考えられる。

この事実を図4と図5に例示する。図4は、完全分離型で10世代後に正解が得られた場合である。プロセッサ8台に割り付けられた個体の適合度の分布を示してある。図中で、25個ある+の記号のところは、関数の極大点である。初期個体を示す○印が、10世代目に●印の位置に移動している。多くの個体は、10世代で早くも各部分集団においてそれぞれ1～3個の極大点にほぼ収束してしている。関数の最大点を与える(-50, -50)の点に個体がほぼ収束しているのは、PE4のプロセッサだけである。この後、世代を500代まで進めてもこの分布状況はほとんど変わらなかった。

一方、図5は並列移民型の場合である。図4と同じく10世代で正解が得られた。しかし、図4とは大きな差がある。全ての部分集団においていくつかの個体が正解点(-50, -50)に収束している。しかも、全ての個体が均一化している訳でなく、他の局所解にも適度に収束している。その局所解は図4の場合に比べ良質なものと考えられる。すなわち、PE6の一例を除き、7つの部分集団では、局所解は正解にくらべ、x座標またはy座標の何れか同じになっている。従って、これらは、その後の交叉に参加して正解に近付ける良質な個体であると言える。

5. むすび

通常の遺伝的アルゴリズムでは正解を得るのは難しいとされる、ある多峰性関数(DeJongの第5関数)の最大値探索問題を検討した。個体をいくつかの部分集団に分割し、並列コンピュータの各プロセッサでその部分集団を並列に進化させ、適度にプロセッサ間通信を使って集団間で移民を起こさせる「並列移民型」手法をこの問題に適用した。500世代までの進化計算を、乱数系列を変えて100回試行した結果、50世代までで95回、500世代では97回の正解(少数点以下3桁の精度で)を得ることができた。また、並列に進化させるが各部分集団が完全に孤立している場合や、全個体を一つの集団として一括して処理する場合に比べ、本手法が個体の多様性を持続させるのに効果があることも確認できた。

参考文献

- 1) 坂和正敏、田中雅博：遺伝的アルゴリズム、朝倉書店 (1995)
- 2) 米澤保雄：遺伝的アルゴリズム、森北出版 (1993)
- 3) 北野宏明編：遺伝的アルゴリズム、産業図書 (平成5年)
- 4) 伊庭斉志：遺伝的アルゴリズムの基礎、オーム社 (平成6年)
- 5) 伊庭斉志：遺伝的プログラミング、東京電機大学出版局 (1996)
- 6) E.G. Talbi and P. Bessiere : A Parallel Genetic Algorithm Applied to the Mapping Problem, SIAM NEWS, July (1991)
- 7) 廣田薫：知能工学概論、第6章、昭晃堂 (1996)
- 8) 北村新三、北森俊行：自己組織化の科学、第5章、オーム社 (平成8年)
- 9) 石田良平、村瀬治比古、小山修平：パソコンで学ぶ遺伝的アルゴリズムの基礎と応用 (1997)
- 10) 安居院猛、長尾智晴：ジェネティカルアルゴリズム、昭晃堂 (1993)
- 11) 荻原将文：ニューロ・ファジィ・遺伝的アルゴリズム、産業図書 (1994)
- 12) メラニーミッチェル著、伊庭斉志監訳：遺伝的アルゴリズムの方法、東京電機大学出版局 (1997)
- 13) 竹内勝：遺伝的アルゴリズムと機械学習、コンピュータソフトウェア、Vol.8, No.5, pp. 16-24 (1991)
- 14) 古川知成、矢川元基：多峰性関数最適化のための進化的アルゴリズム、日本機械学会、第71期通常総会講演会講演論文集、March 29-31, pp. 140-142 (1994)
- 15) 福島 誠：遺伝的アルゴリズムによる多次元多峰性関数の大域的最大値の探索、電子情報通信学会和文論文誌 (DII) 情報・システム (情報処理)、No.6, pp. 1169-1171 (1996年)
- 16) Heintz Muhlenbein : Evolution in Time and Space - The Parallel Genetic Algorithm, Foundations of Genetic Algorithms, Morgan Kaufmann Publishers, pp. 316-337 (1991)