

距離に依存せずに多様性を制御する GA による高次元関数最適化

A Genetic Algorithm with Distance Independent Diversity Control for High Dimensional Function Optimization

木村 周平
Shuhei Kimura

理化学研究所 ゲノム科学総合研究センター
RIKEN Genomic Sciences Center
skimura@gsc.riken.go.jp

小長谷 明彦
Akihiro Konagaya

(同上)
konagaya@gsc.riken.go.jp

keywords: genetic algorithm, real-coded GA, function optimization, adaptation

Summary

For genetic algorithms, it is important to maintain the population diversity. Some genetic algorithms have been proposed, which have an ability to control the diversity. But these algorithms use the distance between two individuals to control the diversity. Therefore, these performances become worse on ill-scaled functions. In this paper, we propose a new genetic algorithm, DIDC (a genetic algorithm with Distance Independent Diversity Control), that does not use a distance to control the population diversity. For controlling the diversity, DIDC uses two GAs that have different natures. For realizing different natures, one GA uses a crossover operator as a search operator, and the other GA uses a mutation operator in DIDC. By applying DIDC to several benchmark problems, we show that DIDC has a good performance on high dimensional, multimodal, non-separable and ill-scaled problems. Finally, we show that the control parameter of DIDC has the same effect on the search with the number of generating children n_c .

1. は じ め に

連続な探索空間で関数を最適化する関数最適化問題は、しばしば実問題において現れる重要な問題のひとつである。実問題における関数最適化問題の持つ特徴的な性質として多峰性、変数間依存性、悪スケール性が考えられ、困難とされる実問題にはこれらの性質のいずれかまたはその組み合わせが付随している [小林 01]。従って実問題への適用を考えると、最適化手法はこれらの性質に対応することが望まれる。なお多峰性とは複数の局所最適解が存在する性質のこと、変数間依存性とは目的関数を各変数毎の関数の和として分離して表現することのできない性質のことである。また悪スケール性とは変数毎のスケールが大きく異なる性質のことであり、長さや重さのように異なる単位のパラメータの最適化を行う場合などに現れやすい性質である。

多峰性を持つ困難な実問題への適用において、勾配法のような決定的な探索法による接近は困難である。そのため遺伝的アルゴリズム (GA) のような確率的解法の使用が注目されている。特に遺伝子として実数値ベクトルを使用する実数値 GA は関数最適化問題に適した手法と

して期待されており、様々な交叉が提案されている [Ono 97, Eshelman 93, Wright 91]。

変数間依存性や悪スケール性にも対応した交叉として ENDX [木村 00], SPX [樋口 01], UNDX- m [喜多 00] が提案されている。ところがこれらの交叉を用いた GA による探索では集団が収束しやすく、UNDX [Ono 97] や BLX- α [Eshelman 93] といった変数間依存性と悪スケール性の両方には対応していない交叉と比較すると、多峰性関数においては局所解に陥りやすいという問題がある。また集団の収束しやすい性質のために、長い稜構造を有するような問題では局所解さえ発見できない場合がある。従ってこれらの交叉を用いた GA においては多様性の維持が特に重要となる。

一般に集団サイズを大きくすることで、多様性の維持が図られる。しかしこれは収束を遅らせることによって集団が収束する前に解に到達する確率を高めているに過ぎず、適応度景観が分からない実問題に対して適切な集団サイズを決定することは難しい。同様に MGG [佐藤 97] や島モデル GA [Mühlenbein 91] のような世代交代モデルの使用も収束を遅らせるだけであり、長い稜構造を有するような関数では多様性を維持し続けることは難しい。

他方、積極的に多様性を維持する方法として Fitness Sharing[Goldberg 87] や DDA[Takahashi 99] といった、集団分布を考慮する世代交代モデルの使用が考えられる。これらの世代交代モデルは個体間の距離を考慮することにより、集団の分布を明示的に制御する。これにより状況に応じた適応的な多様性の維持が期待できる。ところがこれらの世代交代モデルは個体間の距離を使用するため、悪スケール性の強い関数最適化問題では性能が悪化する。

そこで本論文では個体間の距離を利用せずに集団の多様性を制御することにより、多峰性や変数間依存性だけでなく悪スケール性を有する関数最適化問題においても良好な性能を示す GA の提案を目指す。その方法として **集団の収束を促す GA による探索と、集団分布を維持・拡大できる GA による探索を状況に応じて使い分ける**ことを考える。これにより集団分布を間接的に制御する。集団の収束を促す GA と集団分布を維持・拡大できる GA を実現するために、それぞれの GA において性質の異なる探索オペレータを使用する。またこれらの探索オペレータには変数間依存性と悪スケール性の影響を受けないオペレータを用いるものとする。以上により多峰性、変数間依存性、悪スケール性を有する関数最適化問題において、従来手法よりも高次元の問題に対しても良好な性能を示す GA として **DIDC** (a genetic algorithm with Distance Independent Diversity Control) を提案する。

以下 2 章では提案手法が使用する遺伝的オペレータの説明を行う。3 章では 2 章で説明した遺伝的オペレータによって作られる二種類の GA の性質が、確かに望まれる性質を満たしていることを実験的に確認する。4 章で新たな GA: DIDC を提案し、その性能を 5 章で確認する。6 章では主に DIDC のパラメータについて考察を行う。7 章では結論と今後の課題について述べる。

2. 分布制御のための遺伝的オペレータ

性質の異なる二種類の GA による探索を実現するための方法として、性質の異なる二種類の探索オペレータの使用を考える。ところで GA には交叉と突然変異という性質の異なる探索オペレータが存在する。そこで本論文では性質の異なる二種類の探索オペレータとして交叉と突然変異を考え、集団の収束を促す GA のために交叉を、集団分布を維持・拡大できる GA のために突然変異を使用するものとする。ただし変数間依存性や悪スケール性に対応した探索を実現するため、遺伝的オペレータにはこれらの性質に対応したものを使用しなければならないことに注意が必要である。

本章ではまず二種類の探索オペレータ (**交叉 ENDX** と **突然変異 NDM**)、およびそれらを適用可能な世代交代モデル (MGG) の説明と提案を行う。さらに 3 章の予備実験において本章で説明した遺伝的オペレータを利用した

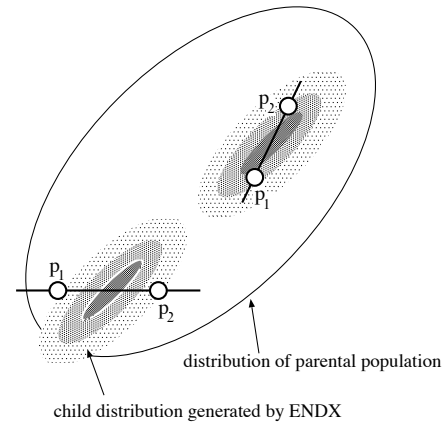


図 1 ENDX による子個体の生成。

GA が確かに必要な性質を満たすことを示す。

2.1 交叉 ENDX

集団の収束を促す GA のための探索オペレータとして交叉 ENDX (Extended Normal Distribution Crossover) [木村 00] を使用する。ENDX は多数の親個体を利用し、主にそのうちの 2 個体の親を結ぶ直線に沿って正規分布に従って子個体を生成する交叉である。両親を結ぶ直線に直交する方向に関して残りの親個体を利用して摂動を加える。ENDX のアルゴリズムを以下に示す：

$$\mathbf{c} = \mathbf{p} + \xi \mathbf{d} + \sum_{i=3}^m \eta_i \mathbf{p}'_i. \quad (1)$$

ただし、

$$\mathbf{p} = (\mathbf{p}_1 + \mathbf{p}_2)/2, \quad (2)$$

$$\mathbf{d} = \mathbf{p}_2 - \mathbf{p}_1, \quad (3)$$

$$\mathbf{p}'_i = \mathbf{p}_i - \frac{1}{m-2} \sum_{j=3}^m \mathbf{p}_j, \quad (4)$$

$$\xi \sim N(0, \alpha^2), \quad (5)$$

$$\eta_i \sim N(0, \beta^2). \quad (6)$$

\mathbf{p}_i は親個体の遺伝子ベクトル、 \mathbf{c} は子個体の遺伝子ベクトルである。また m は使用する親個体の数、 α, β は定数パラメータである。これらの値には ENDX の推奨パラメータを使用する。すなわち $m = n + 2$, $\alpha = 0.434$, $\beta = 0.35/\sqrt{m-3}$ である。ただし n は探索空間の次元数である。

ENDX は両親を結ぶ直線と直交する方向に関して親集団の分布を積極的に活用して子個体を生成する交叉である (図 1)。これにより ENDX は変数間依存性や悪スケール性の影響を受けない探索を実現している。

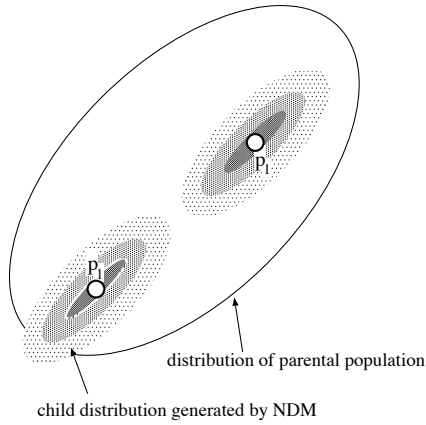


図 2 NDM による子個体の生成.

2.2 突然変異 NDM の提案

変数間依存性や悪スケール性に対応可能な突然変異として、進化戦略 (ES) において相関突然変異 (Correlated Mutation) が提案されている [Bäck 97]. 相関突然変異ではオペレータ自身の制御パラメータも遺伝子として扱い、進化によって自動調整することを必要とする. ところが相関突然変異の制御パラメータの数は非常に多く ($O(n^2)$ 個), 高次元関数での使用は困難である [Hildebrand 99].

そこで本論文では ENDX のアイデアを利用して集団の分布を積極的に活用することで, 変数間依存性や悪スケール性に対応可能な突然変異として **NDM** (Normal Distribution Mutation) を提案する (図 2). NDM は以下に従って子個体を生成する:

$$\mathbf{c} = \mathbf{p}_1 + \sum_{i=2}^m \eta_i \mathbf{p}'_i. \quad (7)$$

ただし,

$$\mathbf{p}'_i = \mathbf{p}_i - \frac{1}{m-1} \sum_{j=2}^m \mathbf{p}_j, \quad (8)$$

$$\eta_i \sim N(0, \gamma^2), \quad (9)$$

である.

ENDX が親個体 $\mathbf{p}_1, \mathbf{p}_2$ を結ぶ直線上に生成された子個体に式 (1) 右辺第 3 項によって摂動を加えているのに対して, NDM は親個体 \mathbf{p}_1 に式 (7) 右辺第 2 項によって摂動を加える. 使用する親個体の数 m および定数パラメータ γ は ENDX のパラメータと同様に, 交叉の設計指針 [喜多 99] および予備実験を利用して $m = n + 2$, $\gamma = 0.35/\sqrt{m-2}$ とした.

NDM は一度に多数の親個体を利用するため, 本来は交叉と呼ぶべきものかもしれない. しかしこのオペレータは本質的にはある 1 個体 (\mathbf{p}_1) に対して摂動を加えるという突然変異の性質を持っているので突然変異と考える. このように複数の親を利用する突然変異として幾つかの手法が提案されている [高橋 01, 染谷 01a].

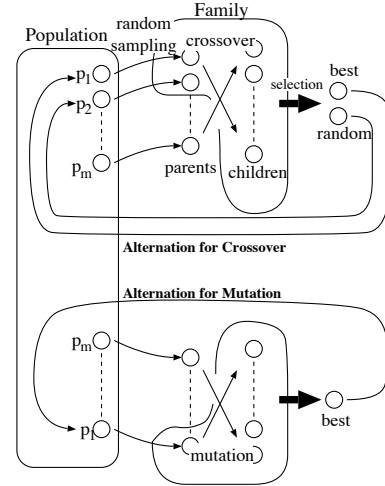


図 3 改良 MGG.

2.3 世代交代モデル MGG

世代交代モデルには MGG (Minimal Generation Gap) [佐藤 97] を使用する. ただし MGG は交叉のための世代交代モデルであるため, 突然変異も適用できるように変更を加える. **改良した MGG のアルゴリズムを以下に示す** (図 3):

- (1) 世代数を 0 に設定. 初期集団として n_p 個体をランダムに生成.
- (2) 集団から親個体として m 個体をランダムに非復元抽出. それぞれの個体を p_i とする ($i = 1, \dots, m$).
- (3) 選択した親個体に交叉または突然変異を適用することで, n_c 個体の子集団を生成する. なお ENDX における親個体 $\mathbf{p}_1, \mathbf{p}_2$ や NDM における \mathbf{p}_1 は残りの親個体とは異なり特別な意味を持つ (主要親個体と呼ぶ) が, 選択された m 個体の中からランダムに選択されるものとする.
- (4) 使用された探索オペレータが**交叉の場合**は, 家族中の最良個体と, 家族中からランダムに選択された個体を主要親個体と交代させる. **突然変異の場合**は家族中の最良個体をその主要親個体と交代させる. ただし家族とは主要親個体とその子個体のことである.
- (5) 終了条件を満たしていれば終了. さもない場合は世代数を 1 増加させてステップ 2 へ.

MGG は個体の選択に評価値のみを利用するため, 関数の変数間依存性や悪スケール性の影響は受けない.

3. 予備実験による GA の性質の確認

本章では探索オペレータとして交叉 ENDX のみを用いた GA と, 突然変異 NDM のみを用いた GA による実験を行う. ただし世代交代モデルには両 GA ともに MGG を使用する. この実験により **ENDX を用いた GA が集団の収束を促す性質を, NDM を用いた GA が状況に応じて集団分布を維持・拡大する性質を, それぞれ有する**

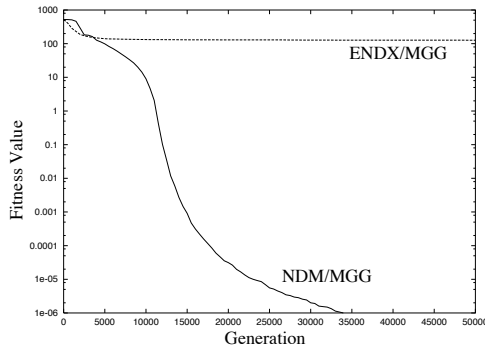


図 4 異なる GA の性能の相違 (offset Sphere 関数).

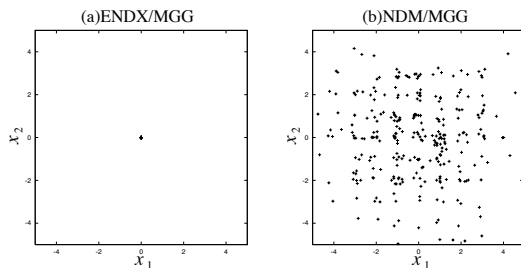


図 5 探索終了時点における親集団の分布 (Rastrigin 関数). (a)ENDX/MGG, (b)NDM/MGG. 図中の + が各個体に相当する.

ことを確認する.

3.1 オフセット Sphere 関数

20 次元 Sphere 関数に対して, 初期集団の分布範囲を $5.11 \leq x_i \leq 5.12 (i = 1, \dots, n)$ として実験を行う. この設定では集団が最適解を囲む領域に分布しておらず, 稜構造を有する関数における GA の挙動を調べる実験に相当する. なお Sphere 関数については表 1 を参照のこと. GA のパラメータは集団サイズ $n_p = 100$, 生成子個体数 $n_c = 50$ とする.

図 4 は各 GA における集団中の最良解の適応度変化の 10 試行平均を示している. ENDX を用いた GA では全ての試行で初期収束が起こるのに対して, NDM を用いた GA では正しく最適解を発見できる.

このように ENDX を用いた GA では初期収束を起こし, 単純な単峰性関数であっても最適解が発見できない場合がある. それに対して NDM を用いた GA では適切に集団分布を拡大することで, 初期収束を起こさなく最適解の発見が可能である.

3.2 Rastrigin 関数

多峰性関数に対する GA の性質を比較するため, 20 次元 Rastrigin 関数を用いて実験を行う (表 1 参照). Rastrigin 関数は大域的最適解のまわりに非常に多くの局所解を持つ多峰性関数である. 最適解は $(x_1, \dots, x_n) = (0, \dots, 0)$ で, その評価値は 0 である. 初期集団の分布範囲を $-5.12 \leq$

$x_i \leq 5.12 (i = 1, \dots, n)$, 集団サイズ $n_p = 300$, 生成子個体数 $n_c = 50$ として実験を行った.

探索打ち切り世代 (5×10^4 世代) までに ENDX/MGG による GA は 10 試行中 7 試行で最適解を発見したのに対して, NDM/MGG では 1 試行も最適解を発見することが出来なかった. ENDX を用いた GA では探索打ち切り世代までに集団が一点に収束する (図 5 左). それに対して NDM を用いた GA では局所的な探索に基づいて各個体の改善が行われるため, 多峰性関数では各個体が異なる局所解の谷に捕われて脱出できず, 集団が収束しない (図 5 右). NDM は集団の分布を利用して子個体の生成範囲を決定するため, 集団が収束しなければ解の洗練化ができず, 探索が停滞する. そのために Rastrigin 関数のような多峰性関数では, NDM を用いた GA の性能は悪くなると考えられる.

以上より NDM を用いた GA では, 多峰性関数において探索不要と思われる領域を適切に切り捨て集団を収束させることが出来ないために探索が停滞することが分かる. それに対して ENDX を用いた GA では重要と思われる領域に徐々に計算リソースを集中させることによって, 効率的な探索が行われる.

4. DIDC の提案

前章の実験により ENDX を用いた GA が集団の収束を促すということ, 逆に NDM を用いた GA を用いることで集団分布を維持・拡大できるということが分かる. 従ってこれらの GA を状況に応じて使い分けることで, 距離情報を利用することなく集団分布を制御することができると考えられる. 以上の考えに基づき, 本章では距離に依存せずに多様性を制御する GA (DIDC, a genetic algorithm with Distance Independent Diversity Control) の提案を行う.

なお DIDC では一種類の世代交代モデル (MGG) に対して二種類の探索オペレータ (ENDX, NDM) を使用することで, 性質の異なる二種類の探索を実現する. このように, 実数値 GA において一種類の世代交代モデルに対して性質の異なる複数の探索オペレータを使い分けるという手法は幾つか提案されている [Ono 99, Sakuma 01, 染谷 01a] が, 提案手法は集団分布を制御するために異なるオペレータを使い分けるという点で従来研究とは異なる.

4.1 設計方針

単純な単峰性関数に対しては NDM を用いた GA だけで最適解が発見できること, ENDX を用いた GA は集団の収束を促すことを鑑み, 以下の方針に従って GA を設計する. これにより変数毎のスケールの影響を受けることなく多様性を適切に維持した探索を実現する:

- 通常は NDM を利用した GA による探索を行う.

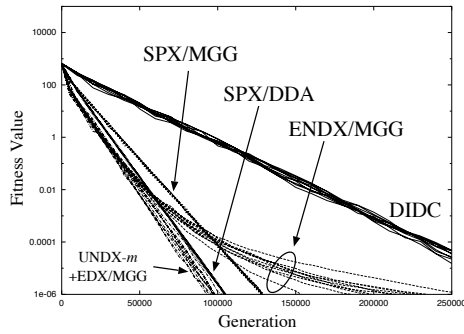


図 6 100 次元 Sphere 関数における探索の様子.

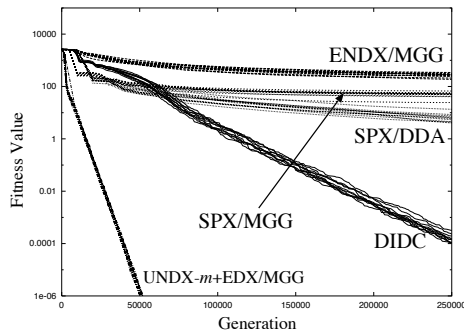


図 7 100 次元 offset Sphere 関数における探索の様子.

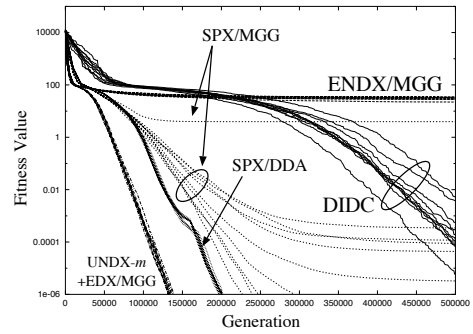


図 8 100 次元 Rosenbrock 関数における探索の様子.

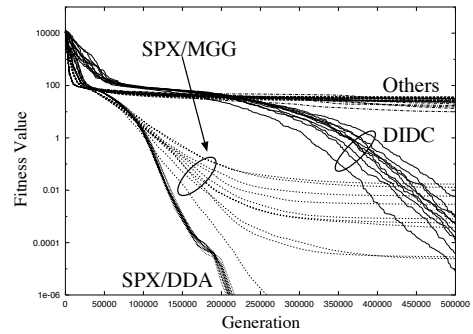


図 9 100 次元 ill-scaled Rosenbrock 関数における探索の様子.

- 探索が停滞したら ENDX を用いた GA による探索を行うことで集団の収束を促す.

探索の停滞は、集団中の最良個体を観察することによって判別する。すなわち一定期間内に集団中の最良個体が改善されなければ、探索が停滞していると判断する。逆に最良個体が僅かでも改善されたら、探索が進行しているものと判断する。

4.2 アルゴリズム

前節の設計方針に従い、DIDC を提案する。DIDC のアルゴリズムは以下の通りである：

- (1) 世代数を 0, 突然変異適用確率 $p_m = 1$ とする.
- (2) 突然変異 (NDM) 適用確率 p_m , 交叉 (ENDX) 適用確率 $1 - p_m$ として MGG による世代交代を一回行う.
- (3) 過去の T_s 世代の間に集団中の最良個体が更新されていなければ $p_m = 0$ とする。集団中の最良個体の評価値が僅かでも改善された場合はすぐに $p_m = 1$ とする. do NDM
- (4) 終了条件を満たしていれば終了。さもなければ世代数を 1 増加させてステップ 2 へ.

DIDC のパラメータ T_s は予備的な実験に基づき $T_s = 3n_p$ とする。ただし n_p は集団サイズである。これは集団中の各個体に対して平均で 3 回 NDM を利用した探索を行っても集団中の最良解が更新されない場合に、探索が停滞していると判断することを意味する。なおこのパラメータに関しては 6 章において考察を行う。

5. 数 値 実 験

本章ではベンチマーク問題を用いて提案手法と代表的な従来手法の性能を比較する。

5.1 対 象 問 題

表 1 に示す関数の最小化を目的として実験を行った。ただし全ての関数において GA の初期集団を表 1 に示される初期化領域に限定して生成し、探索中は探索領域を明示的には考慮しないものとした。実験は 10 次元 ($n = 10$) および 100 次元 ($n = 100$) の関数に対して行った。

Sphere 関数は単純な単峰性関数である。ただし offset Sphere 関数は 3.1 節の実験と同様に初期集団の初期化領域を $5.11 \leq x_i \leq 5.12 (i = 1, \dots, n)$ としている。

Rosenbrock 関数は変数間依存性の強い単峰性関数であり、ill-scaled Rosenbrock 関数は変数毎にスケールを変化させた Rosenbrock 関数である。両者の違いは変数毎のスケールだけであり、その他の性質は全く同じである。

Griewangk 関数は多峰性関数であるが、次元数の増加と共に Sphere 関数に近くなり最適化が容易となるという性質を持つ。Ackley 関数および Rastrigin 関数は多峰性の強い関数である。なお Rastrigin 関数では初期集団の初期化領域を $-100 \leq x_i \leq 100 (i = 1, \dots, n)$ として実験を行った。これは通常の設定よりも広く、難しい問題となっている。これによりサンプリング・バイアスによって探索領域の中央に位置する最適解の発見確率が高くなる影響 [染谷 01b] を緩和できると考えられる。

表 1 実験に使用したベンチマーク関数.

関数		初期化領域	最適解
Sphere 関数	$f_{sp}(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$-5.12 \leq x_i \leq 5.12$	$f_{sp}(0, \dots, 0) = 0$
offset Sphere 関数	$f_{osp}(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$5.11 \leq x_i \leq 5.12$	$f_{osp}(0, \dots, 0) = 0$
Rosenbrock 関数	$f_{ro}(\mathbf{x}) = \sum_{i=1}^n [100(x_1 - x_i^2)^2 + (x_i - 1)^2]$	$-2.048 \leq x_i \leq 2.048$	$f_{ro}(1, \dots, 1) = 0$
ill-scaled Rosenbrock 関数	$f_{iro}(\mathbf{x}) = \sum_{i=2}^n \{100[x_1 - (ix_i)^2]^2 + (ix_i - 1)^2\}$	$-\frac{2.048}{i} \leq x_i \leq \frac{2.048}{i}$	$f_{iro}(\frac{1}{1}, \dots, \frac{1}{n}) = 0$
Griewangk 関数	$f_{gr}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-512 \leq x_i \leq 512$	$f_{gr}(0, \dots, 0) = 0$
Ackley 関数	$f_{ac}(\mathbf{x}) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}\right) - \exp\left(\frac{\sum_{i=1}^n \cos 2\pi x_i}{n}\right)$	$-20 \leq x_i \leq 30$	$f_{ac}(0, \dots, 0) = 0$
Rastrigin 関数	$f_{ra}(\mathbf{x}) = 10n + \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i) $	$-100 \leq x_i \leq 100$	$f_{ra}(0, \dots, 0) = 0$

表 2 ベンチマーク関数に対する各 GA の最適解発見率.

関数	次元数	DIDC	ENDX/MGG	SPX/MGG	SPX/DDA	UNDX-m+EDX/MGG
Sphere 関数	10	10/10	10/10	10/10	10/10	10/10
	100	10/10	10/10	10/10	10/10	10/10
offset Sphere 関数	10	10/10	10/10	10/10	10/10	10/10
	100	10/10	0/10	0/10	0/10	10/10
Rosenbrock 関数	10	10/10	10/10	10/10	10/10	10/10
	100	8/10	0/10	9/10	10/10	10/10
ill-scaled Rosenbrock 関数	10	10/10	10/10	10/10	10/10	10/10
	100	8/10	0/10	6/10	10/10	0/10
Griewangk 関数	10	10/10	10/10	10/10	0/10	10/10
	100	10/10	0/10	10/10	10/10	10/10
Ackley 関数	10	10/10	10/10	10/10	10/10	10/10
	100	10/10	0/10	10/10	10/10	10/10
Rastrigin 関数	10	10/10	8/10	9/10	0/10	10/10
	100	10/10	0/10	0/10	0/10	0/10

5.2 実験計画

集団サイズ n_p は単峰性関数に対しては $n_p = 5n$, 多峰性関数に対しては $n_p = 15n$ とした. ただし n は関数の次元数である. また一世代あたりの生成子個体数 $n_c = 50$ とした. 本論文では以降の実験において特に断りがなければ, 全て上記の設定に基づいて実験を行うものとする.

比較のための GA には ENDX/MGG, SPX [樋口 01] /MGG, SPX/DDA [Takahashi 99], UNDX- m + EDX [Sakuma 01] /MGG を使用した. UNDX- m + EDX/MGG は提案手法と同様に二種類の性質の異なる探索オペレータを使用した GA であり, 稜構造を持つ高次元関数に対して良い性能を示す. UNDX- m + EDX/MGG では文献の推奨パラメータに基づき 10 次元関数では $m = 5$, 100 次元関数では $m = 20$, UNDX- m の生成子個体数 $n_{\text{UNDX-}m} = 100$, EDX の生成子個体数 $n_{\text{EDX}} = 50$, EDX の適用確率 0.5 とした. UNDX- m + EDX/MGG の集団サイズは DIDC と同じとし, ENDX/MGG, SPX/MGG および SPX/DDA については全ての実験において DIDC よりも多い $n_p = 30n$ とした. また ENDX/MGG, SPX/MGG, SPX/DDA の生成子個体数は $n_c = 50$ とした. その他の GA のパラメータは, 全て各文献の推奨パラメータを使用した.

探索の終了は (1) 探索打ち切り世代に達した, (2) 集団が全ての座標軸に関して 10^{-6} 以下の領域に収束した, のどちらかの条件が満たされた場合とした. 探索打ち切り世代は Sphere 関数に対して $2.5 \times 10^3 n$ 世代, Rosenbrock 関数および ill-scaled Rosenbrock 関数に対して $5.0 \times 10^3 n$

世代, 多峰性関数に対して $1.5 \times 10^4 n$ 世代とした. なお UNDX- m + EDX/MGG に関しては関数評価回数が同じになるように正規化している. また Griewangk 関数については最良解の評価値が 10^{-6} 以下になった場合に, それ以外の関数については 10^{-3} 以下になった場合に最適解を発見したと判断した.

5.3 結果

各 GA の各関数に対する最適解発見率を表 2 に示す. また図 6~図 9 には, それぞれ 100 次元の Sphere 関数, offset Sphere 関数, Rosenbrock 関数および ill-scaled Rosenbrock 関数に対する各 GA の集団中の最良解の適応度変化を示す. 10 次元関数では全ての手法が殆どの関数に対して最適解を発見しているのに対して, 100 次元関数では単純な単峰性関数であっても最適解の発見に失敗する場合があることが分かる.

Sphere 関数では DIDC の探索速度が最も遅かった. しかし offset Sphere 関数では ENDX/MGG, SPX/MGG, SPX/DDA は初期収束を起こして最適解を発見できない. これに対して DIDC および UNDX- m + EDX/MGG だけが最適解の発見に成功した.

Rosenbrock 関数においても DIDC の探索速度は他手法と比較すると遅い. 一方 UNDX- m + EDX/MGG は Rosenbrock 関数において最も速く最適解を発見した. しかしながら UNDX- m + EDX/MGG は ill-scaled Rosenbrock 関数では全く最適解を発見できなくなった. これは UNDX- m + EDX/MGG の探索性能が関数の悪スケール

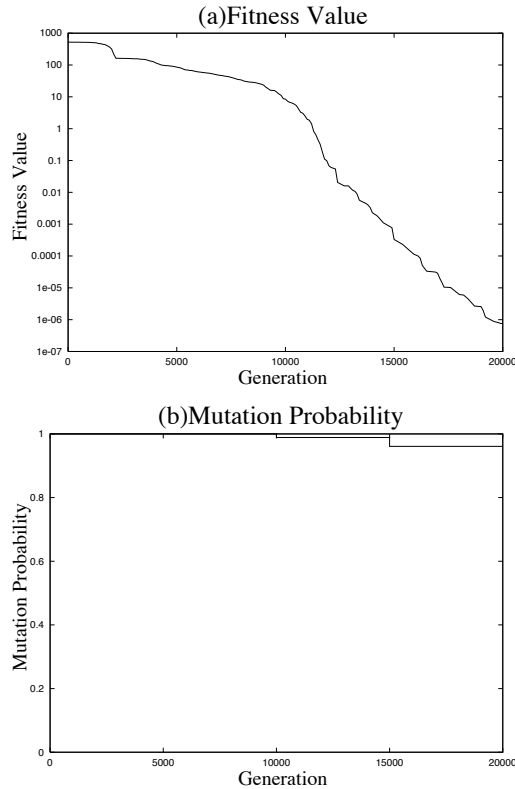


図 10 20 次元 offset Sphere 関数における (a) 適応度変化と (b) 突然変異適応確率の変化。ただし (b) では 5000 世代毎の p_m の平均をプロットしている。

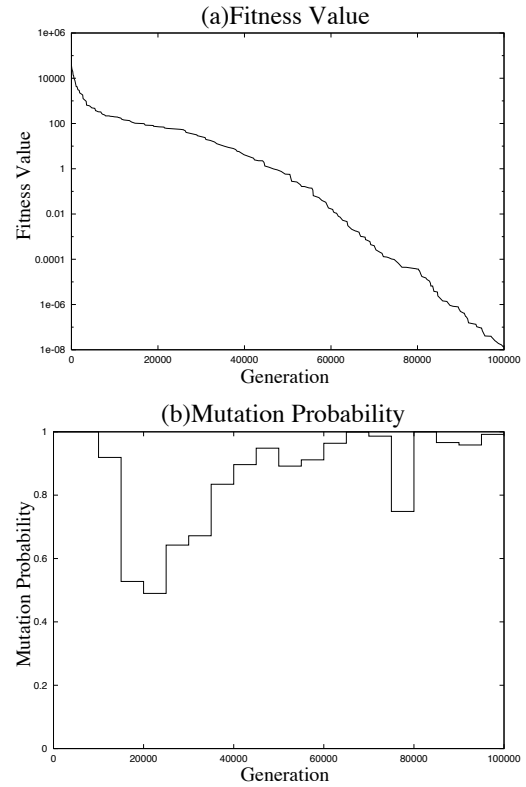


図 11 20 次元 Rastrigin 関数における (a) 適応度変化と (b) 突然変異適応確率の変化。ただし (b) では 5000 世代毎の p_m の平均をプロットしている。

性の影響を受けるためである。これに対して DIDC は悪スケール性の影響を殆ど受けないため、Rosenbrock 関数と同様に最適解の発見に成功した。

多峰性関数において、DIDC は全ての試行で最適解を発見した。特に他の全ての手法が最適解の発見に失敗した 100 次元 Rastrigin 関数においても DIDC は全ての試行で最適解発見に成功した。

以上より DIDC は高次元性、多峰性、変数間依存性、悪スケール性を有する関数最適化問題において従来手法よりも良好な最適解発見性能を持つことが分かる。しかしながらその収束には非常に多くの世代数を必要とすることに注意が必要である。特に比較的単純な関数の最適化では、提案手法は探索速度の点で他手法に劣る場合がある。DIDC を用いた最適化において、必要な関数評価回数を低減することは今後の課題である。

6. 考 察

6.1 探索オペレータ適用確率の適応

DIDC の挙動を確認するために 20 次元 offset Sphere 関数および 20 次元 Rastrigin 関数を用いて、突然変異適用確率 p_m の変化を観察した。

図 10 と図 11 はそれぞれ offset Sphere 関数と Rastrigin 関数における (a) 最良個体の適応度変化と、(b) 突然

変異の適用確率の変化である。なお DIDC では p_m の値は 0 か 1 しかとらない。突然変異の適用確率の変化を見やすくするために、図の突然変異の適用確率は 5000 世代毎の p_m の平均としている。

Offset Sphere 関数では殆ど NDM/MGG のみで探索を行っているのに対して、Rastrigin 関数では探索の途中で突然変異適用確率が大きく変化していることが分かる。Rastrigin 関数では $1.0 \times 10^4 \sim 5.0 \times 10^4$ 世代の探索中盤には多くの個体が異なる局所解に陥り、探索が停滞していると考えられる。そのような場合に DIDC は適応的に交叉 ENDX を用いた探索の適用確率を高めることで集団の収束を促している。

以上のように DIDC は突然変異適用確率を変化させることで関数の性質や探索の状況に合わせて適応的に集団分布を制御することができる。

6.2 パラメータ $T_s = 3n_p$ の妥当性

本節ではパラメータ $T_s = 3n_p$ の妥当性を実験的に示す。実験には 10, 20, 30 次元の offset Sphere 関数と Rastrigin 関数を使用し、 $T_s = \lambda n_p$ として λ を変化させたときの性能を比較した。

表 3 および表 4 は、それぞれ offset Sphere 関数と Rastrigin 関数における最適解発見率および最適解を発見した試行の最適解発見までに要した世代数の平均である。表よ

表 3 パラメータを変えたときの DIDC の性能 (offset Sphere 関数).

λ	10 次元		20 次元		30 次元	
	最適解発見率	平均世代数	最適解発見率	平均世代数	最適解発見率	平均世代数
0.5	100/100	5.24×10^3	100/100	1.49×10^4	100/100	2.70×10^4
1.0	100/100	5.06×10^3	100/100	1.35×10^4	100/100	2.38×10^4
2.0	100/100	5.20×10^3	100/100	1.40×10^4	100/100	2.57×10^4
3.0	100/100	5.18×10^3	100/100	1.43×10^4	100/100	2.62×10^4
4.0	100/100	5.24×10^3	100/100	1.43×10^4	100/100	2.61×10^4
5.0	100/100	5.19×10^3	100/100	1.43×10^4	100/100	2.63×10^4
10.0	100/100	5.23×10^3	100/100	1.43×10^4	100/100	2.66×10^4

表 4 パラメータを変えたときの DIDC の性能 (Rastrigin 関数).

λ	10 次元		20 次元		30 次元	
	最適解発見率	平均世代数	最適解発見率	平均世代数	最適解発見率	平均世代数
0.5	71/100	1.00×10^4	42/100	2.87×10^4	45/100	5.15×10^4
1.0	88/100	1.28×10^4	76/100	3.96×10^4	80/100	7.56×10^4
2.0	93/100	1.69×10^4	92/100	5.58×10^4	99/100	1.16×10^5
3.0	96/100	2.02×10^4	96/100	6.92×10^4	99/100	1.47×10^5
4.0	99/100	2.23×10^4	99/100	8.01×10^4	100/100	1.73×10^5
5.0	98/100	2.51×10^4	99/100	9.07×10^4	100/100	2.01×10^5
10.0	98/100	3.47×10^4	99/100	1.32×10^5	100/100	3.11×10^5

り offset Sphere 関数では最適解発見までに必要な世代数は λ とは無関係にほぼ一定であるのに対して, Rastrigin 関数では λ の値に比例して増加することが分かる. また offset Sphere 関数では全ての試行において最適解を発見している. 他方 Rastrigin 関数では λ が小さいと最適解発見率が低くなること, λ が大きければほぼ全ての試行で最適解を発見できることが分かる.

以上より offset Sphere 関数および Rastrigin 関数における計算コストおよび最適解発見率の観点から, 本論文では $T_s = 3n_p$ とした. T_s の値は問題依存であり, 最適値を決定するためには予備実験が必要であるかもしれない. しかし本論文で用いた他のベンチマーク問題でも $T_s = 3n_p$ で概ね良好な解が得られたことから, 一般に $T_s = 3n_p$ としても良いと考えている.

6.3 DIDC による探索の解析

本節ではパラメータ T_s の観点から, DIDC の挙動に関して解析を行う.

§ 1 多様性の制御

DIDC の探索は主に NDM/MGG を用いて行われる. ところが多峰性関数においては NDM/MGG を用いた探索は集団を収束させることができない. DIDC は T_s 世代連続で集団中の最良個体が更新されなければ NDM/MGG を用いた探索をあきらめ, ENDX/MGG による探索を導入することで集団の収束を促している. 本節ではまず, このような基準で探索の切り換えを行うことの意味について考える.

ある世代において集団中のある特定の一個体 p_A を主要親個体として NDM によって子個体を 1 個体生成することを考える. このとき生成子個体が集団中の最良個体よりも良い評価値を持つ確率を p とする. 簡単のため集団中のその他の個体を主要親個体として NDM を適用しても, 最良解よりも良い評価値を持つ子個体が生成され

る確率は 0 であると仮定する. 一世代に生成する子個体の数は n_c であるから, ある世代において親個体として p_A が選択された場合は確率 $(1-p)^{n_c}$ で, その他の場合は確率 1 で最良解は更新されない. 従ってある世代において最良解が更新されない確率 q は,

$$q = \frac{1}{n_p} \cdot (1-p)^{n_c} + \frac{n_p-1}{n_p} \cdot 1 \quad (10)$$

となる. ただし n_p は集団サイズである. 最良解が更新されるまでは p_A による最良解の更新確率 p が変化しないと仮定すると, T_s 世代連続で集団中の最良解が更新されない確率, すなわち探索が停滞していると判断される確率 r は,

$$r = q^{T_s} \quad (11)$$

$$= \left[\frac{1}{n_p} (1-p)^{n_c} + \frac{n_p-1}{n_p} \right]^{T_s} \quad (12)$$

$$= \left[1 - \frac{1 - (1-p)^{n_c}}{n_p} \right]^{T_s} \quad (13)$$

となる. NDM は親集団の分布を活用して探索を行う探索オペレータである. 従って最良解の更新確率 p が変化しないという仮定は, 多峰性関数において集団の収束が止まり探索が停滞した場合にはほぼ成り立つものと考えられる. 式 (13) より p が大きければ T_s 世代連続で最良解が更新されない確率 r は低くなり, 逆に p が小さければ確率 r は大きくなることが分かる.

以上より DIDC は T_s 世代連続で最良解が更新されない場合には, p が小さすぎると判断していると言える. そのような場合, DIDC は ENDX/MGG の探索により集団の収束を促す. ENDX/MGG による集団の収束は, 探索範囲を狭めることによって p を高める効果があると考えられる. 逆に p が大きい場合は NDM/MGG による探索を行うことで収束を止める. これにより DIDC は p が

一定になるように収束を制御しながら探索を行っていると考えられる。

§2 パラメータ T_s の効果

式 (13) において、最良解の更新確率 p は関数の性質や探索の状況に応じて一意に定まる。このとき r は特定の p に対して、DIDC が NDM による探索をあきらめ ENDX による探索を行う確率を表している。従って DIDC の制御パラメータを変化させた場合でも、同じ p の値に対して r の値が同じになるなら、その GA は同じ挙動を示すと考えられる。

いま集団中のある特定の一個体に対する最良解の更新確率 $p \ll 1$ と仮定すると $1 - (1 - p)^{n_c} \ll 1$ となるから、式 (13) より

$$r \simeq 1 - T_s \frac{1 - (1 - p)^{n_c}}{n_p} \quad (14)$$

$$= 1 - \frac{T_s}{n_p} [1 - (1 - p)^{n_c}] \quad (15)$$

$$\simeq 1 - \frac{T_s}{n_p} [1 - (1 - n_c p)] \quad (16)$$

$$= 1 - \frac{T_s}{n_p} n_c p \quad (17)$$

となる。ただし $|x| \ll 1$ において $(1 - x)^k \simeq 1 - kx$ という近似を用いた。さらにここで $T_s = \lambda n_p$ とすると、式 (17) より確率 r , p , およびパラメータ λ , n_c に関して以下の関係式を得る:

$$r \simeq 1 - p \lambda n_c. \quad (18)$$

式 (18) では $p \ll 1$ を仮定している。これとは逆に p が大きい場合には r が小さくなるため、ENDX/MGG による探索は殆ど行われないと考えられる。従って大きな p に対して、DIDC が ENDX/MGG による探索を行うことの影響は殆ど無視できるだろう。

以上のことから式 (18) より、DIDC においては $\lambda n_c =$ 一定というパラメータ設定では性能が同じになること、すなわちパラメータ λ と生成子個体数 n_c が DIDC の性能に関して同じ効果を持つことが分かる。

6.4 DIDC で使用する探索オペレータ

GA は集団分布を徐々に狭めながら探索を行うのが良いと言われている [Kita 99]。これは GA が解の改善の見込みの無い領域の探索を諦め、徐々に有望な領域の探索に計算資源を集中させることを意味する。この性質は、特に多峰性関数の最適化において重要であると考えられる。

NDM/MGG は主に各個体の近傍を探索する GA である。そのため NDM/MGG は多峰性関数において集団分布を収束させることができず、効率的な探索を行うことができない。他方 ENDX/MGG は集団を強制的に収束させることができる。しかし集団の過度な収束のために、望ましい領域への収束に失敗する場合がある。DIDC は

以上のような性質の異なる二種類の GA を使い分けることで、集団分布を適切に狭めながら探索を行う。

従って上記のような性質を満たす探索手法であれば、どのような手法でも DIDC に適用できると考えられる。例えば ENDX/MGG の代わりに SPX/MGG や UNDX-m/MGG などが利用可能である。また我々は NDM/MGG の代わりに既存のローカルサーチ手法を用いることもできている。これにより DIDC の計算量の削減が期待できるが、実際の適用に関してはさらなる研究の課題である。

7. お わ り に

本論文では性質の異なる二種類の GA を状況に応じて使い分けることによって、距離に依存せずに多様性を制御する GA, DIDC (a genetic algorithm with Distance Independent Diversity Control) を提案した。実験により DIDC が悪スケール性の影響を受けず、また変数間依存性や多峰性を有する高次元関数に対して従来手法よりも局所解に陥りにくいことを示した。しかしながら DIDC は従来手法と比較すると、最適化に多くの関数評価を必要とする。提案手法を用いた最適化において必要な関数評価回数を低減することが今後の課題である。

我々は遺伝子ネットワークのパラメータ推定問題 [Mendes 98] に GA を適用している。遺伝子ネットワークのパラメータ推定問題とは、生体内の化学反応系において実験的に測定不可能な速度定数などを測定可能な物理量を利用して推定する問題であり、非線形性の強い困難な問題である。提案手法を遺伝子ネットワークのパラメータ推定問題に適用したところ、比較的低位次元の問題でもパラメータの推定に失敗する場合があることが分かった。遺伝子ネットワークのパラメータ推定問題において DIDC が探索に失敗する原因は、問題の騙し性にあると考えている。騙し性とは最適解ではない局所解へ GA の集団が収束しやすい性質のこと [Ikeda 00, Sakuma 01, 高橋 01] であり、探索空間に複数の谷が存在し、最適解の存在する谷が小さいあるいは複雑であるなどの理由で起こる。DIDC を拡張することで、騙し性にも対応させることも今後の課題である。

◇ 参 考 文 献 ◇

- [Bäck 97] T.Bäck, U.Hammel and H.-P.Schwefel: Evolutionary Computation: Comments on the History and Current State, IEEE Trans. on Evolutionary Computation, Vol.1, No.1, pp. 3-17 (1997).
- [Eshelman 93] L.J.Eshelman and J.D.Schaffer: Real-Coded Genetic Algorithms and Interval-Schemata, Foundations of Genetic Algorithms 2, pp. 187-202 (1993).
- [Goldberg 87] D.E.Goldberg and J.Richardson: Genetic Algorithms with Sharing for Multimodal Function Optimization, Proc. of 2nd Int. Conf. on Genetic Algorithms, pp. 41-49 (1989).

- [樋口 01] 樋口, 筒井, 山村: 実数値 GA におけるシンプレックス交叉の提案, 人工知能学会論文誌, Vol.16, No.1, pp.147-155 (2001).
- [Hildebrand 99] L.Hildebrand, B.Reusch and M.Fathi: Directed Mutation—A New Selfadaptation for Evolution Strategies, Proc. of Congress on Evolutionary Computation, pp. 1550-1557 (1999).
- [Ikeda 00] K.Ikeda and S.Kobayashi: GA based on the UV-structure Hypothesis and Its Application to JSP, 6th Parallel Problem Solving from Nature, pp.273-282 (2000).
- [木村 00] 木村, 小野, 喜多, 小林: 交叉の設計指針に基づく UNDX の拡張: ENDX の提案と評価, 計測自動制御学会論文集, Vol.36, No.12, pp. 1162-1171 (2000).
- [喜多 99] 喜多, 小野, 小林: 実数値 GA のための正規分布交叉に関する理論的考察, 計測自動制御学会論文集, Vol.35, No.11, pp. 1333-1339 (1999).
- [Kita 99] H.Kita and M.Yamamura: A Functional Specialization Hypothesis for Designing Genetic Algorithms, 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics, pp. 579-584 (1999).
- [喜多 00] 喜多, 小野, 小林: 実数値 GA のための正規分布交叉の多数の親を用いた拡張法の提案, 計測自動制御学会論文集, Vol.36, No.10, pp. 875-883 (2000).
- [小林 01] 小林, 木村, 小野: 生物的適応システムへ進化・学習のアルゴリズムと創発システム論へ, 計測と制御, Vol.40, No.10, pp. 752-757 (2001).
- [Mendes 98] P.Mendes and D.B.Kell: Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation, Bioinformatics, Vol.4, No.10, pp.869-883 (1998).
- [Mühlenbein 91] H.Mühlenbein, M.Schomisch and J.Born: The Parallel Genetic Algorithm as Function Optimizer, Proc. of 4th ICGA, pp.271-278 (1991).
- [Ono 97] I.Ono and S.Kobayashi: A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, Proc. of 7th ICGA, pp.246-253 (1997).
- [Ono 99] I.Ono, H.Kita and S.Kobayashi: A Robust Real-Coded Genetic Algorithm using Unimodal Normal Distribution Crossover Augmented by Uniform Crossover: Effects of Self-Adaptation of Crossover Probabilities, Proc. of ICGA99, pp.496-503 (1999).
- [Sakuma 01] J.Sakuma and S.Kobayashi: Extrapolation-Directed Crossover for Real-coded GA: Overcoming Deceptive Phenomena by Extrapolative Search, Proc. of CEC, pp. 655-662 (2001).
- [佐藤 97] 佐藤, 小野, 小林: 遺伝的アルゴリズムにおける世代交代モデルの提案と評価, 人工知能学会論文誌, Vol.12, No.5, pp. 734-744 (1997).
- [染谷 01a] 染谷, 山村: 探索領域を適応的に調整する GA による関数最適化, 第 28 回 知能システムシンポジウム資料, pp. 165-170 (2001).
- [染谷 01b] 染谷, 山村: 最適解の位置にロバストな実数値 GA を実現する Troidal Search Space Conversion の提案, 人工知能学会論文誌, Vol.16, No.3, pp. 333-343 (2001).
- [Takahashi 99] O.Takahashi, H.Kita and S.Kobayashi: A Distance Dependent Alternation Model on Real-coded Genetic Algorithms, IEEE Int. Conf. on SMC'99, pp. 619-624 (1999).
- [高橋 01] 高橋, 木村, 小林: 交叉的突然変異による適応的近傍探索—騙しのある多峰性関数の最適化—, 人工知能学会論文誌, Vol.16, No.2, pp. 175-184 (2001).
- [Wright 91] A.H.Wright: Genetic Algorithms for Real Parameter Optimization, FOGA, pp. 205-218 (1991).

[担当委員: 山村雅幸]

2002 年 3 月 7 日 受理

著者紹介

木村 周平(正会員)



1998 年京都大学大学院修士課程高分子化学専攻修了, 2001 年東京工業大学大学院博士後期課程知能システム科学専攻修了, 現在理化学研究所ゲノム科学総合研究センター研究員, 博士(工学). 進化計算, バイオインフォマティクスなどの研究に従事. 計測自動制御学会, 日本分子生物学会などの会員.



小長谷 明彦(正会員)

理化学研究所ゲノム科学総合研究センターゲノム情報科学グループプロジェクトディレクター, 東京工業大学大学院客員教授. 博士(工学). 1980 年東京工業大学 理工学研究科情報科学専攻修士課程修了, 同年日本電気株式会社入社, 1996 年 10 月東京工業大学 知能システム科学連携講座 客員助教授, 1997 年 10 月北陸先端科学技術大学院大学 知識科学研究科 教授, 2003 年 4 月より現職. バイオインフォマティクス, 高性能計算機技術, グリッド計算技術に興味をもつ. 人工知能学会分子生物情報研究会(主査), 並列生物情報処理イニシアティブ(会長), 文科省科研ゲノム情報科学特別領域研究ソフトウェア共有化および高速化委員会(委員長). 主要著書「遺伝子とコンピュータ」共立出版(2000). 情報処理学会論文賞(1999).