

# 多次元空間における近傍構造を利用した 最適化アルゴリズムに関する一検討

広島修道大学商学部

阪井 節子 (Setsuko Sakai)

Faculty of Commercial Sciences, Hiroshima Shudo University

広島市立大学大学院 情報科学研究科 高濱 徹行 (Tetsuyuki Takahama)

Department of Intelligent Systems, Hiroshima City University

## 1 はじめに

進化的アルゴリズム (Evolutionary Algorithm, EA) は、生物進化の過程をモデル化した最適化アルゴリズムの総称であり、遺伝的アルゴリズム (Genetic Algorithm, GA), 進化戦略 (Evolution Strategy, ES), 差分進化 (Differential Evolution, DE)[1, 2] など多くのアルゴリズムが提案されている。EA は、最適化の対象である目的関数の値だけを利用して解を求めることができる直接探索法であり、アルゴリズムの実装が容易であることから、様々な最適化問題を解くために利用されている。

しかし、EA は確率的な多点探索を行うため、比較的局所解に陥りにくいが、関数評価回数が多くなりがちである。近年、最適化問題が大規模化し、目的関数の評価コストが増大してきているため、目的関数の評価回数の削減は大きな課題となってきた。これに対して、近似モデルを用いて関数評価回数の削減をする研究 [3, 4, 5, 6, 7, 8] が行われている。

本研究では、EA を効率化するために、アルゴリズムパラメータの新しい調整方法を提案する。これにより、広い範囲の問題に対して、少ない関数評価回数で精度の高い解の探索を実現することが本研究の目的である。

## 2 近接グラフ

### 2.1 定義

頂点集合  $V$  と辺集合  $E$  からなるグラフ  $G$  を  $G(V, E)$  で表現する。近接グラフ (proximity graph) は、頂点集合  $V$  の近傍構造を表現するグラフであり、最近傍グラフ (Nearest Neighborhood Graph), Gabriel グラフ (Gabriel Graph, GG)[9], 相対近傍グラフ (Relative Neighborhood Graph, RNG)[10],  $\beta$ -skeleton[11] などが提案されている。近接グラフでは、任意の 2 頂点  $v_i, v_j \in V$  が近傍条件を満足するときのみ、 $v_i, v_j$  を結ぶ線分が辺  $(v_i, v_j) \in E$  となる。

Gabriel グラフでは、 $v_i$  と  $v_j$  を結ぶ線分を直径とする超球内に他の頂点が含まれていなければ、 $v_i$  と  $v_j$  が近傍条件を満足する。Gabriel グラフ  $GG(V, E)$  は、以下のように定義できる。

$$(v_i, v_j) \in E \iff HS\left(\frac{v_i + v_j}{2}, \frac{\|v_i - v_j\|}{2}\right) \cap V = \phi \quad (1)$$

ここで、 $HS(v, r)$  は頂点  $v$  を中心とする半径  $r$  の超球内の領域、すなわち

$$HS(v, r) = \{x \mid \|x - v\| < r\} \quad (2)$$

である。この様子を図 1 に示す。超球内に  $v_i, v_j$  の 2 点以外の頂点  $v_k$  が存在しない場合にのみ頂点を接続し、そうでない場合は接続しない。

相対近傍グラフでは、頂点  $v_i$  および  $v_j$  を中心とする半径  $\|v_i - v_j\|$  の 2 つの超球の共通集合内に他の頂点が含まれていなければ近傍条件を満足する。相対近傍グラフは Gabriel グラフの部分グラフである。相対近傍グラフ  $RNG(V, E)$  は以下のように定義できる。

$$(v_i, v_j) \in E \iff HS(v_i, \|v_i - v_j\|) \cap HS(v_j, \|v_i - v_j\|) \cap V = \phi \quad (3)$$

この様子を図 2 に示す。

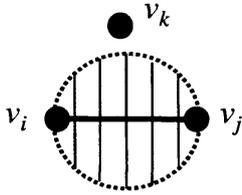


図 1: Gabriel グラフにおける辺の判定

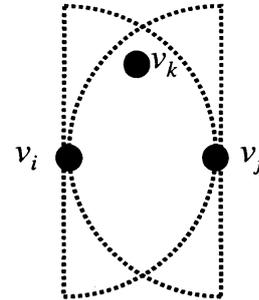
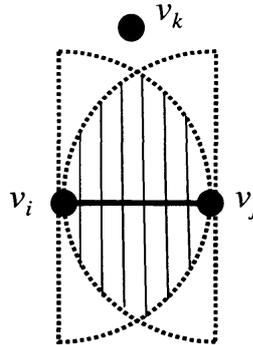
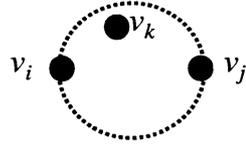


図 2: 相対近傍グラフにおける辺の判定

弓形  $\beta$ -skeleton (lune-based  $\beta$ -skeleton) は、頂点  $v_i, v_j$  およびパラメータ  $\beta$  ( $\beta \geq 1$ ) を用い、点  $(1 - \frac{\beta}{2})v_i + \frac{\beta}{2}v_j$  および点  $\frac{\beta}{2}v_i + (1 - \frac{\beta}{2})v_j$  を中心とする半径  $\frac{\beta}{2}\|v_i - v_j\|$  の 2 つの超球の共通集合内に他の頂点が含まれていなければ近傍条件を満足する。

$$(v_i, v_j) \in E \iff HS \left( \left( \left( 1 - \frac{\beta}{2} \right) v_i + \frac{\beta}{2} v_j, \frac{\beta}{2} \|v_i - v_j\| \right) \cap HS \left( \left( \frac{\beta}{2} v_i + \left( 1 - \frac{\beta}{2} \right) v_j, \frac{\beta}{2} \|v_i - v_j\| \right) \cap V = \phi \quad (4)$$

$\beta = 1$  ならば Gabriel グラフと一致し、 $\beta = 2$  ならば相対近傍グラフと一致する。この様子を図 3 に示す。

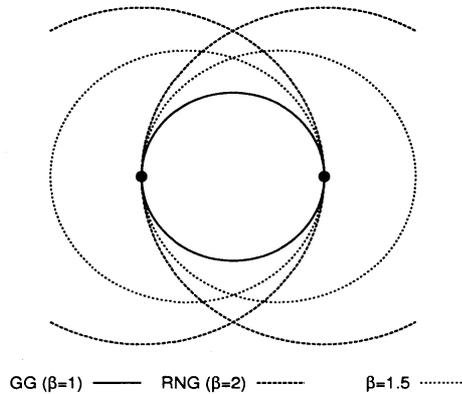


図 3:  $\beta$ -skeleton グラフ ( $\beta = 1.5$ ) における辺の判定

### 3 近傍グラフに基づく最適化アルゴリズム

本研究では、Differential Evolution (DE) と近傍グラフに基づく最適化アルゴリズムである NGDE (Neighborhood Graph Based Differential Evolution) を提案する。

#### 3.1 アルゴリズム

以下にアルゴリズムの概要を示す。

## 1. 初期化

探索領域内に点をランダムに生成し，初期探索点集合を構成する。

## 2. 近傍グラフの構成

点間の距離を求める。近傍関係を求め，近傍グラフを構成する。

## 3. 山谷判定

山と谷を判定するために，各点に対して山度と谷度を付与する。近傍グラフの全ての辺について，その辺を構成する2点のうち，評価値の良い点の谷度を1増加させ，評価値の悪い点の山度を1増加させる。山度が0の点は谷に対応する点，谷度が0の点は山に対応する点とみなすことができる。

## 4. 探索点の生成

山点および山近傍点は大域探索を行うため，大域的なDE操作を行う。このため，ベクトルをランダムに選択し， $F$  および  $CR$  に大きな値に設定する。谷点および谷近傍点は局所探索を行うため，局所的なDE操作を行う。このため，谷点では谷点をベクトルとし，谷点を中心とする探索を行うために， $F$  に小さな値， $CR$  に大きな値を設定する。谷近傍点ではベクトルをランダムに選択し，谷点と同様に  $F$  に小さな値， $CR$  に大きな値を設定する。それ以外の点については，通常のパラメータ値  $F$  と  $CR$  によるDE操作を行う。

## 5. 生存者選択

元の点が探索点より良好でなければ，探索点と置換する。2. に戻る。

## 3.2 擬似コード

以下に，DE/rand/1/exp の擬似コード [3, 7, 8, 12] を参考に，NGDE の擬似コードを示す。

```

NGDE()
{
    P=Generate N individuals {xi} randomly;
    Evaluate xi, i = 1, 2, ..., N;
    for(t=1; t < Tmax; t++) {
        // 距離などの初期化
        for(i=1; i ≤ N; i++) {
            xi.valley=xi.hill=0;
            for(j=i+1; j ≤ N; j++)
                dij=dji=xi と xj 間の距離 d(xi, xj);
        }
        // 近傍グラフの構成
        for(i=1; i ≤ N; i++) {
            for(j=i+1; j ≤ N; j++) {
                for(k=1; k ≤ N; k++) {
                    if(k! = i && k! = j &&
                        dik2 + d(βxj + (1-β)xi, vk)2
                            < βdij2 &&
                        djk2 + d(βxi + (1-β)xj, vk)2
                            < βdij2) break;
                }
            }
            if(k>N) {
                // 山谷判定: 辺 (xi, xj) の生成
                if(f(xi) < f(xj)) {
                    xi.valley++; xj.hill++;
                }
                else if(f(xi) > f(xj)) {
                    xj.valley++; xi.hill++;
                }
            }
        }
        // 全ての個体について山谷判定を行う;
        for(i=1; i ≤ N; i++) {
            switch(xi の種類) {
                case 山点: F'=1; CR'=1; break;
                case 山近傍点: F'=0.9; CR'=0.95; break;
                case 谷近傍点: F'=0.3; CR'=0.95; break;
                case 谷点: p1=i; F'=0.2; CR'=1; break;
                default: F'=F; CR'=CR;
            }
        }
        // 探索点の生成
        (p1, p2, p3)=select randomly in [1, N] \ {i}
            s. t. pj ≠ pk (j, k = 1, 2, 3, j ≠ k);
        xinew=xi ∈ P;
        j=select randomly from [1, n];
        k=1;
    }
}

```

```

do {
   $x_{ij}^{\text{new}} = x_{p1,j} + F'(x_{p2,j} - x_{p3,j});$ 
   $j = (j + 1) \% n;$ 
   $k++;$ 
} while( $k \leq n$  &&  $u(0,1) < CR'$ );
// 生存者選択
if( $f(x^{\text{new}}) \leq f(x_i)$ )  $x_i = x^{\text{new}};$ 
} } }
```

ただし,  $x_i.\text{hill}$ ,  $x_i.\text{vally}$  はそれぞれ点  $x_i$  より良い近傍頂点数, 悪い近傍頂点数である.

### 3.3 山谷判定

本研究では, 4種類の点, 谷点, 山点, 谷近傍点, 山近傍点を定義し, 各点の山谷判定を行う.

**谷** 近傍により悪い点の一つ以上存在し, より良い点が存在しない点. すなわち,  $x_i.\text{valley} > 0$  かつ  $x_i.\text{hill} = 0$  の点.

**山** 近傍により良い点の一つ以上存在し, より悪い点が存在しない点. すなわち,  $x_i.\text{hill} > 0$  かつ  $x_i.\text{valley} = 0$  の点.

**谷近傍** 谷の近傍の点でかつ, 山点, 谷点および山近傍点でない点.

**山近傍** 山の近傍の点でかつ, 山点, 谷点および谷近傍点でない点.

## 4 実験

### 4.1 テスト問題

本実験では, 変数間依存性, 悪スケール性, および大谷構造を有する問題を用いる [3, 4, 5, 7, 8, 13]. 変数間依存性の強い問題として, 稜構造を有する問題を用いる. 悪スケール性のある問題として, 稜構造でかつ変数によりスケールが異なる問題を用いる. 大谷構造とは, 微視的に見れば局所解となる多数の小さな谷が存在するが, 巨視的に見れば大きな谷が一つだけ存在し, その谷が最適解となっている構造であり, Rastrigin 関数とその典型例である.

以下に, 関数とその初期化領域を示す. なお,  $n$  は次元数を表している.

- $f_1$ : Sphere 関数

$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2, \quad -5.12 \leq x_i \leq 5.12 \quad (5)$$

単峰性の関数で, 点  $(0, 0, \dots, 0)$  で最小値 0 をとる.

- $f_2$ : Rosenbrock 関数

$$f(\mathbf{x}) = \sum_{i=2}^n \{100(x_1 - x_i^2)^2 + (x_i - 1)^2\}, \quad -2.048 \leq x_i \leq 2.048 \quad (6)$$

単峰性の稜構造を有する関数で, 点  $(1, 1, \dots, 1)$  で最小値 0 をとる.

- $f_3$ : ill-scaled Rosenbrock 関数

$$f(\mathbf{x}) = \sum_{i=2}^n \{100(x_1 - (ix_i)^2)^2 + (ix_i - 1)^2\}, \quad -2.048/i \leq x_i \leq 2.048/i \quad (7)$$

単峰性の稜構造を有する関数で, 点  $(1, \frac{1}{2}, \dots, \frac{1}{n})$  で最小値 0 をとる.

- $f_4$ : Rastrigin 関数

$$f(x) = 10n + \sum_{i=1}^n \{x_i^2 - 10 \cos(2\pi x_i)\}, \quad -5.12 \leq x_i \leq 5.12 \quad (8)$$

多峰性の大谷構造を有する関数で、点  $(0, 0, \dots, 0)$  で最小値 0 をとる。

図 4.1 に  $n = 2$  のときの関数  $f_1, f_2, f_4$  のグラフを示す。また、表 1 に各関数の特徴を示した [13]。

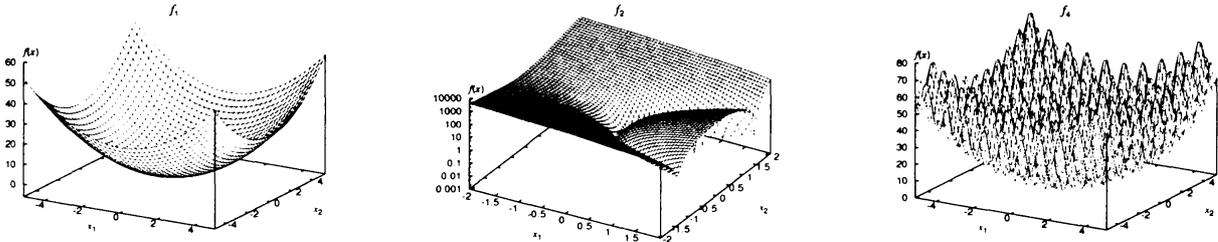


図 4: 関数  $f_1, f_2, f_4$  のグラフ

表 1: テスト問題の特徴

関数	変数間依存性	悪スケール性	大谷構造
$f_1$	なし	なし	なし
$f_2$	強い	なし	なし
$f_3$	強い	あり	なし
$f_4$	なし	なし	あり

## 4.2 実験条件

次元数  $n = 30$  に設定し、 $f_1$  から  $f_4$  の関数を最適化する。DE の設定は、個体数  $N = 50$ ,  $F$  と  $CR$  の組合せを  $(0.7, 0.95)$  と  $(0.5, 0.5)$  の 2 通りとし、各関数について 20 回の試行を行い、結果を考察する。個体数は  $2n$  から  $10n$  程度が良いとされているが、効率性を重視し少し小さい値とした。なお、試行打ち切り条件は、文献 [13] と同様に、(1) 評価値が  $1.0 \times 10^{-7}$  以下に達した、(2) 探索打ち切り評価回数が  $f_1$  および  $f_2$  は  $2n \times 10^5$ ,  $f_3$  は  $5n \times 10^5$ ,  $f_4$  は  $3n \times 10^5$  に達した、のいずれかの条件を満足する場合とした。

なお、NGDE では、Gabriel グラフ ( $\beta = 1$ ) を利用した。

## 4.3 実験結果

実験結果を表 2 に示す。許容精度を満足した回数を “ $< 1e-7$ ”, 実際に関数を評価した回数を  $eval$ , そのうち子ベクトルが良かった割合 (%) を  $succ$ ,  $DE(0.7, 0.95)$  の評価回数との比を  $ratio$ , 評価回数の削減率を  $reduced$  に示した。

パラメータを固定した DE では、 $f_1, f_4$  については、 $(F, CR) = (0.5, 0.5)$  の方が明らかに結果が良い。しかし、 $f_2, f_3$  については、 $(F, CR) = (0.7, 0.95)$  の方が優れており、 $(0.5, 0.5)$  の場合には許容精度の解を見つけることが出来なかった。

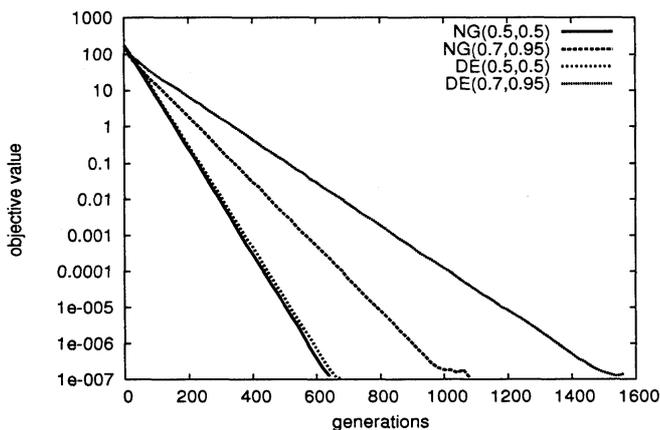
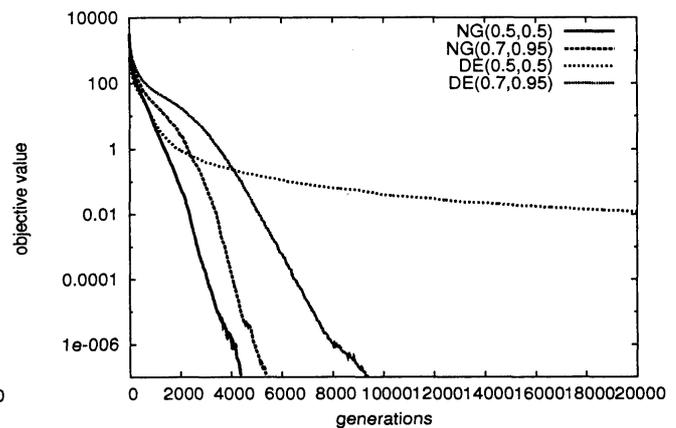
NGDE では、いずれの設定でも全ての問題で許容精度の解を発見できたが、全ての問題で  $(F, CR) = (0.5, 0.5)$  の方がすぐれた結果となっている。これは、山 (近傍) 点, 谷 (近傍) 点については、標準的なパラメータに

表 2: 実験結果

func.	algorithm	$< 1e^{-7}$	eval	succ(%)	ratio	reduced
$f_1$	NG(0.5,0.5)	20	31913.20	38.33	0.42	0.58
	NG(0.7,0.95)	20	50205.85	17.29	0.66	0.34
	DE(0.5,0.5)	20	32958.45	40.36	0.43	0.57
	DE(0.7,0.95)	20	75910.20	14.09	1.00	0.00
$f_2$	NG(0.5,0.5)	20	186525.15	8.93	0.45	0.55
	NG(0.7,0.95)	20	226422.00	7.24	0.55	0.45
	DE(0.5,0.5)	0	—	—	—	—
	DE(0.7,0.95)	20	412663.85	4.19	1.00	0.00
$f_3$	NG(0.5,0.5)	20	188522.55	8.89	0.46	0.54
	NG(0.7,0.95)	20	225498.80	7.26	0.55	0.45
	DE(0.5,0.5)	0	—	—	—	—
	DE(0.7,0.95)	20	413122.60	4.17	1.00	0.00
$f_4$	NG(0.5,0.5)	20	60498.10	24.93	0.22	0.78
	NG(0.7,0.95)	20	243454.45	5.01	0.91	0.09
	DE(0.5,0.5)	20	54366.10	29.37	0.20	0.80
	DE(0.7,0.95)	19	268932.84	5.23	1.00	0.00

よらず,  $F, CR$  を大きくした遠方探索,  $F$  を小さく  $CR$  を大きくした近傍探索を行っているため, その他の場合に中間的な  $F = 0.5, CR = 0.5$  に設定した場合に探索のバランスが取れていたと考えられる。

各問題における最良個体の関数値について, その平均値の変化を図5から図8に示す。横軸は世代数, 縦軸が関数値である。 $f_1$  については, NGDE(0.5,0.5) が最も効率的に探索を行っており, DE(0.5,0.5) がほぼ同程度の効率である。NGDE(0.7,0.95) はやや探索が遅く, DE(0.7,0.95) はかなり遅い。 $f_2, f_3$  は同様の結果となっている。NGDE(0.5,0.5) が最も効率的に探索し, NGDE(0.7,0.95) が比較的効率的である。DE(0.7,0.95) は探索が遅く, DE(0.5,0.5) は最大評価回数までに満足できる解を探索できなかった。 $f_4$  については, DE(0.5,0.5) が最も効率的に探索を行っており, NGDE(0.5,0.5) がほぼ同程度の効率である。NGDE(0.7,0.95) と DE(0.7,0.95) はやや遅く, DE(0.7,0.95) は満足できる解を発見できない場合があった。

図 5:  $f_1$  における関数値の変化図 6:  $f_2$  における関数値の変化

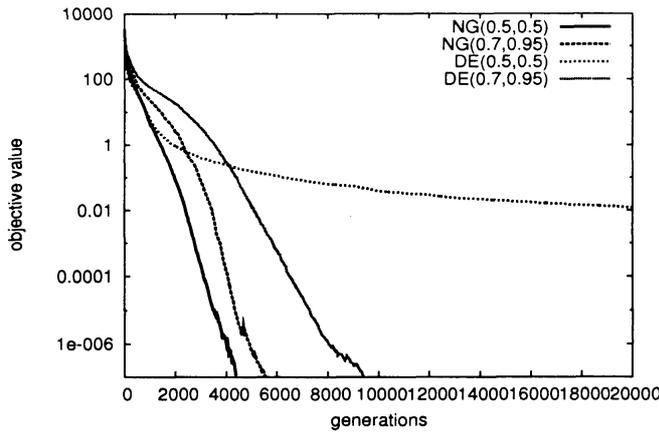


図 7:  $f_3$  における関数値の変化

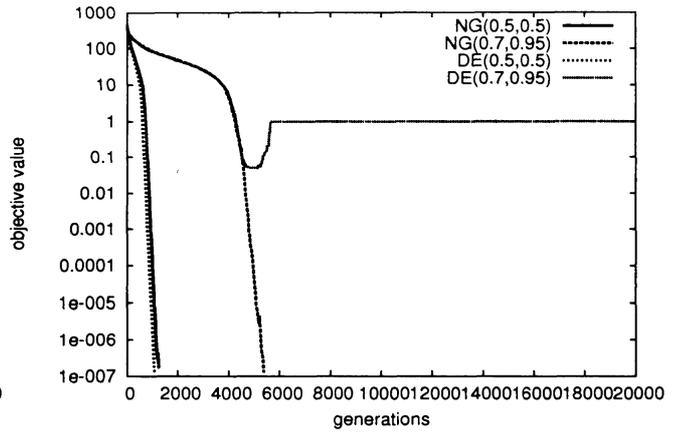


図 8:  $f_4$  における関数値の変化

$f_2, f_3$  については、初期は谷点が1点であるが、次第に増加し、終盤は5個を超えている。谷近傍点は、3個から5個程度の間で増減している。山点は、初期は少数であるが、次第に増加し、中盤以降は7個程度になっている。山近傍点は、急激に25個程度まで増加し、終盤に向かって10数個程度まで減少している。したがって、初期はやや近傍探索を行い、急激に遠方探索に切り替え、終盤に向かって遠方探索と標準的探索のバランスを取っていると考えられる。 $f_2, f_3$  の探索では、一度  $(0, 0, \dots, 0)$  付近に収束し、そこから狭小な谷を通して  $(1, 1, \dots, 1)$  に向かって移動するという探索方法をとる。遠方探索が急激に増加するのは、収束した探索点の多様性を取り戻し、この移動を可能とするためであると考えられる。

$f_4$  については、谷点は1点であり、谷近傍点は、2個から10個程度の間で増減を繰り返す、終盤には25個を超える場合がある。山点・山近傍点は少数であり、通常は1個から2個程度であるが、山近傍点がかかり増加する場合も見受けられる。したがって、序盤は近傍探索と標準的な探索を合わせて行い、遠方探索を少し取り入れ、終盤は近傍探索を行っていると考えられる。

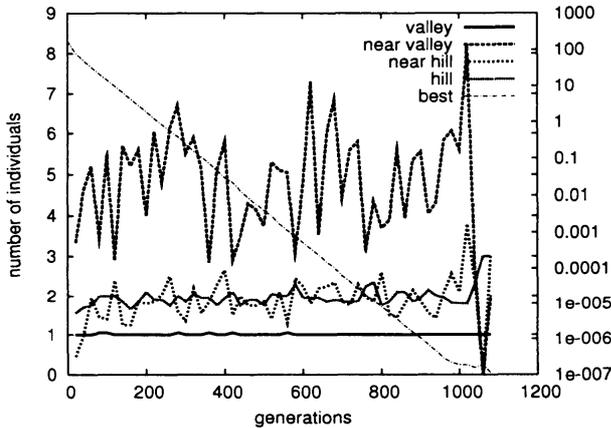


図 9:  $f_1$  における山点判定 ( $F=0.7, CR=0.95$ )

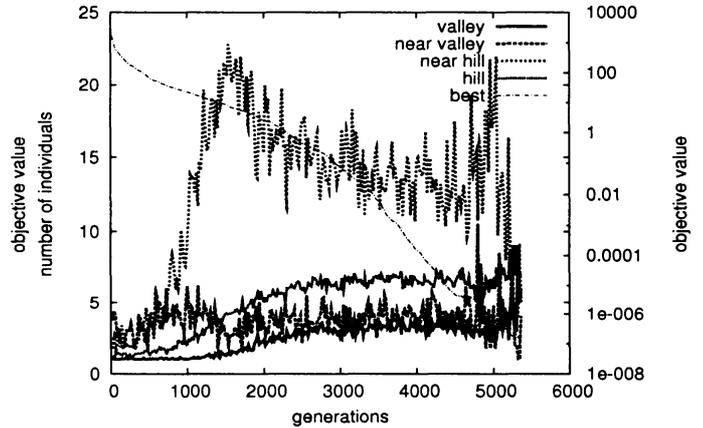


図 10:  $f_2$  における山点判定 ( $F=0.7, CR=0.95$ )

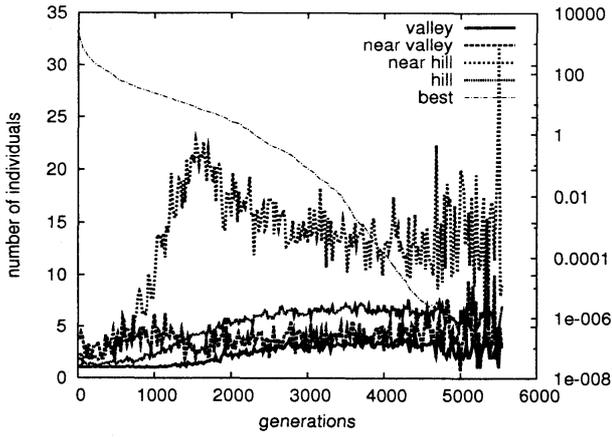


図 11:  $f_3$  における山点判定 ( $F=0.7, CR=0.95$ )

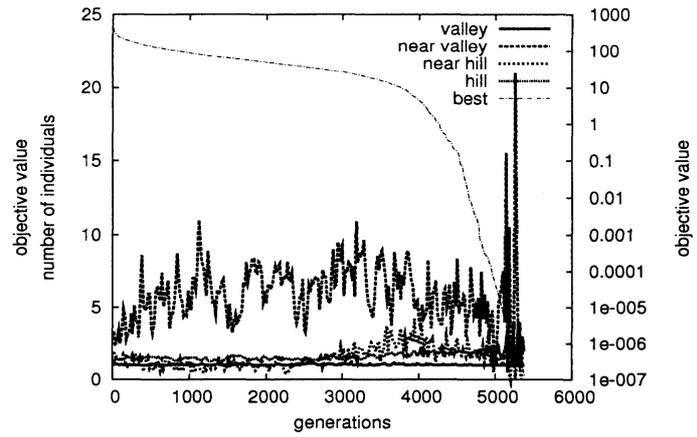


図 12:  $f_4$  における山点判定 ( $F=0.7, CR=0.95$ )

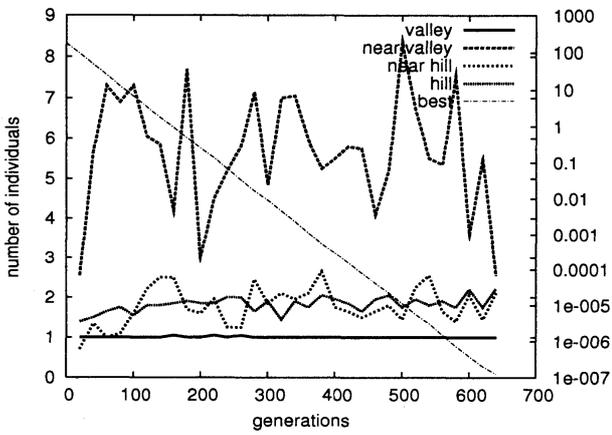


図 13:  $f_1$  における山点判定 ( $F=0.5, CR=0.5$ )

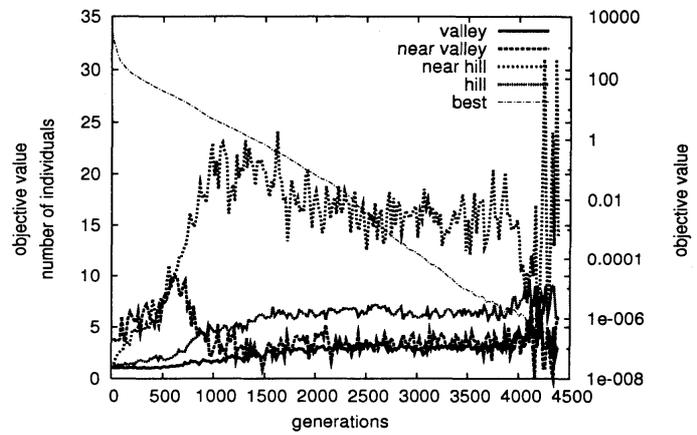


図 14:  $f_2$  における山点判定 ( $F=0.5, CR=0.5$ )

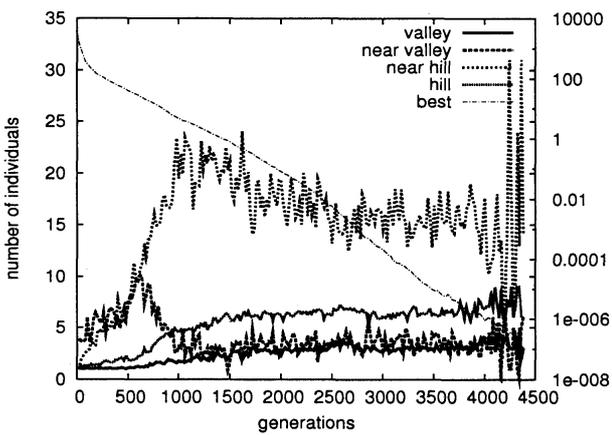


図 15:  $f_3$  における山点判定 ( $F=0.5, CR=0.5$ )

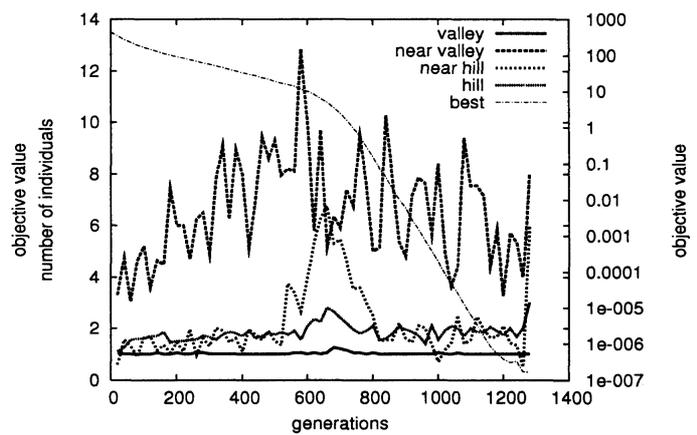


図 16:  $f_4$  における山点判定 ( $F=0.5, CR=0.5$ )

謝辞 この研究の一部は、日本学術振興会科学研究費補助金 基盤研究 (c) (No. 20500138) および広島市立大学特定研究費 (一般研究) 7111 の援助を受けた。

## 参考文献

- [1] Storn, R. and Price, K.: Minimizing the Real Functions of the ICEC'96 Contest by Differential Evolution, in *Proc. of the International Conference on Evolutionary Computation*, pp. 842–844 (1996).
- [2] Storn, R. and Price, K.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, Vol. 11, pp. 341–359 (1997).
- [3] 高濱徹行, 阪井節子, 原章: 低精度の近似モデルを用いた比較推定法による Differential Evolution における関数評価回数の削減, 電子情報通信学会論文誌, Vol. J91-D, No. 5, pp. 1275–1285 (2008).
- [4] Takahama, T. and Sakai, S.: Reducing Function Evaluations in Differential Evolution using Rough Approximation-Based Comparison, in *Proc. of the 2008 IEEE Congress on Evolutionary Computation*, pp. 2307–2314 (2008).
- [5] Takahama, T. and Sakai, S.: A Comparative Study on Kernel Smoothers in Differential Evolution with Estimated Comparison Method for Reducing Function Evaluations, in *Proc. of the 2009 IEEE Congress on Evolutionary Computation*, pp. 1367–1374 (2009).
- [6] 高濱徹行, 阪井節子: 低精度近似モデルを利用した  $\varepsilon$  制約 Differential Evolution による効率的な制約付き最適化, 人工知能学会論文誌, Vol. 24, No. 1, pp. 34–45 (2009).
- [7] 阪井節子, 高濱徹行: 最適化手法における関数評価回数の削減手法 – ポテンシャルモデルに基づく比較推定法の提案 –, 不確実性を含む意思決定の数理とその応用, 数理解析研究所講究録 1548, pp. 61–70 (2007).
- [8] 阪井節子, 高濱徹行: 比較推定法による最適化アルゴリズムの効率性向上, 最適化問題における確率モデルの展開と応用, 数理解析研究所講究録 1559, pp. 22–33 (2007).
- [9] Gabriel, K. R. and Sokal, R. R.: A new statistical approach to geographic variation analysis, *Systematic Zoology*, Vol. 18, pp. 259–270 (1969).
- [10] Toussaint, G. T.: The relative neighborhood graph of a finite planar set, *Pattern Recognition*, Vol. 12, No. 4, pp. 261–268 (1980).
- [11] Kirkpatrick, D. G. and Radke, J. D.: A framework for computational morphology, in Toussaint, G. ed., *Computational geometry*, pp. 217–248, North-Holland (1985).
- [12] Takahama, T. and Sakai, S.: Fast and Stable Constrained Optimization by the  $\varepsilon$  Constrained Differential Evolution, *Pacific Journal of Optimization*, Vol. 5, No. 2, pp. 261–282 (2009).
- [13] 田中雅晴, 土谷千加夫, 佐久間淳, 小野功, 小林重信: Saving MGG: 実数値 GA/MGG における適応度評価回数の削減, 人工知能学会論文誌, Vol. 21, No. 6, pp. 547–555 (2006).