

# 種分化のための $\beta$ 緩和相対近傍グラフの提案

## A Study on $\beta$ -Relaxed Relative Neighborhood Graph for Speciation

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)

Graduate School of Information Sciences, Hiroshima City University

### 概要

多峰性最適化は、多数の最適解を持つ多峰性問題において、一度に全ての最適解を探索するという非常に難しい最適化である。このための方法として、近接グラフによる種分化が提案されている。しかし、グラフとして Gabriel グラフ (GG) を用いており、高次元の問題では少数の最適解しか発見できないという問題がある。本研究では、比較的高速に生成できるグラフであり、パラメータ  $\beta$  によって  $GG(\beta=1)$  と相対近傍グラフ ( $\beta=2$ ) の中間的な近傍グラフを生成できる  $\beta$  緩和相対近傍グラフ ( $\beta$  RNG) を提案する。2次元から20次元の5-peaks問題において、様々な  $\beta$  値に基づいて探索点を種分化し、差分進化により最適解を探索した。その結果、低次元の問題では小さい  $\beta$  で高速に解を探索し、高次元の問題では大きい  $\beta$  となるべく多くの解を探索することが有効であることが分かった。

## 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。

工業設計などにおいて、最適解を1つだけ求めるのではなく、準最適解も含むなるべく多くの最適解を求め、その中から別の観点で採用する解を決めたい場合がある。このような問題は多峰性最適化 (multimodal optimization) 問題と呼ばれ、多数の最適解・準最適解を求める必要がある困難な問題である。しかし、多峰性最適化問題を EA で解こうとすると、一般的には探索が進むにつれて多様性が減少し、ある1つの解の近傍に収束する。単純に、探索を繰り返すことにより多数の解を求めるという方法もあるが、同じ解を繰り返し探索する可能性があり、必ずしも多数の解を求めることにはつながらない。このため、1回の試行で多数の最適解を求めるための研究が活発に行われている。

多数の解を同時に求めるためには、探索点集団が幾つかの部分集団 (subpopulation) に分かれ、各部分集団が探索領域を分担し、多様性を保持しつつ探索を進める手法が必要となる。このような手法は棲み分け (niching) あるいは種分化 (speciation) と呼ばれている [3, 4]。代表的な手法としては、ある範囲で適合度を分け合う sharing, 部分集団中で上位個体以外の適合度を削除し部分集団のサイズを制限する clearing, 生存者選択の際に最も類似した探索点と置換することにより多様性を維持する crowding, 縄張り半径により種を構成する方法 [5, 6], クラスタリングを利用して部分集団を構成する方法 [7], 近接グラフ (proximity graph) を利用して種を構成する方法 [8, 9] などがある。

本研究では、近接グラフを利用する種分化について検討する。種分化に用いる近接グラフとして、Gabriel グラフ (Gabriel Graph, GG), 相対近傍グラフ (Relative Neighborhood Graph, RNG),  $\beta$  スケルトン ( $\beta$  skelton) などが用いられている。GG は収束が速いが、特に高次元の問題では探索初期に完全グラフに近い

状態となり、種に分かれにくいという問題がある。RNG は種に分かれやすいが収束が遅くなるという問題がある。グラフによる種分化では主に GG が使用されているが、高次元の問題ではあまり多くの最適解を発見できないという問題点があった。本研究では、GG と RNG の中間であり、グラフ生成の計算量が小さい  $\beta$  緩和 RNG ( $\beta$ -relaxed RNG,  $\beta$ RNG) を提案する。 $\beta$ RNG は  $\beta$  を指定することにより RNG の条件を緩和して GG に近いグラフを生成できる。さらに、 $\beta$  を指定せずに、枝に  $\beta$  の重みを付けた重み付き  $\beta$ RNG を生成することも可能である。

$\beta$ RNG をグラフに基づく種分化に適用し、ベンチマーク関数を最適化することにより、その性質を調べる。

## 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 [10], 相対近傍グラフ [11],  $\beta$  skeleton [12] 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、頂点  $v$  を中心とする半径  $r$  の超球内の領域を  $HS(v, r)$  で表現すれば、以下のように定義できる。

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

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

この様子を Fig.1 に示す。任意の頂点  $v_k$  が超球内に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。

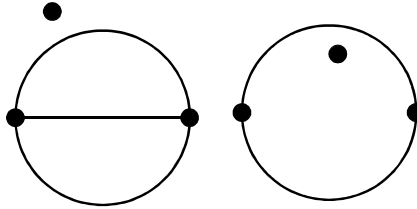


Fig. 1: Gabriel グラフにおける辺の判定

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である弓形領域 (lune) 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 Fig.2 に示す。

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。

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

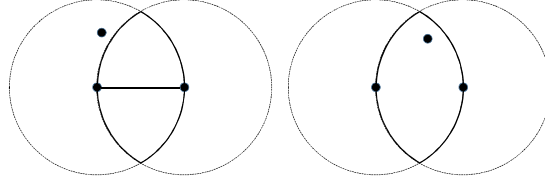


Fig. 2: 相対近傍グラフにおける辺の判定

$\beta = 1$  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 Fig.3 に示す.  $\beta$  スケルトンでは,  $v_i, v_j$  間を  $\frac{\beta}{2} : 1 - \frac{\beta}{2}$  に分割した内分点を中心とする超球と逆の比で分割した内分点を中心とする超球の共通集合を対象とする.

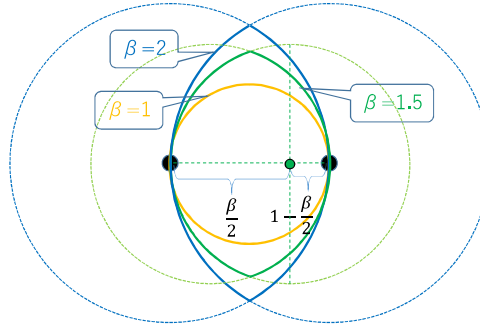


Fig. 3:  $\beta$  skeleton グラフ ( $\beta = 1, 1.5, 2$ ) における辺の判定

## 2.2 近接グラフの生成と計算量

2つの点を  $v_i, v_j$ , それ以外の任意の点を  $v_k$  とすると, 各近接グラフにおける辺の判定は以下のようになる. ただし,  $d(u, v) = \|u - v\|$  は距離関数である.

### 1. ガブリエルグラフ

$d(v_i, v_k)^2 + d(v_j, v_k)^2 < d(v_i, v_j)^2$  を満足する点  $v_k$  が存在しなければ,  $v_i$  と  $v_j$  を辺で接続する.

### 2. 相対近傍グラフ

$d(v_i, v_k) < d(v_i, v_j)$  かつ  $d(v_j, v_k) < d(v_i, v_j)$  を満足する点  $v_k$  が存在しなければ,  $v_i$  と  $v_j$  を辺で接続する.

### 3. $\beta$ skeleton

$d(v_i, v_k)^2 + d(\beta v_j + (1 - \beta)v_i, v_k)^2 < \beta d(v_i, v_j)^2$  かつ  $d(v_j, v_k)^2 + d(\beta v_i + (1 - \beta)v_j, v_k)^2 < \beta d(v_i, v_j)^2$  を満足する  $v_k$  が存在しなければ,  $v_i$  と  $v_j$  を辺で接続する.

$D$  次元空間における頂点数  $N$  のグラフに対する単純なアルゴリズムの計算量を考える. 条件判定の回数は, 任意の2点の対  $(N(N - 1)/2)$  に対して2点を除く全ての点を判定するため, どのグラフでも  $N(N - 1)(N - 2)/2$  回である. 次元数を考慮すると, 計算量は  $O(DN^3)$  となる. しかし, ガブリエルグラフと相対近傍グラフでは,  $d(v_i, v_j)$  (の2乗) のみを使用するため, それらの値を  $O(DN^2)$  の計算量で行列として求めておき, その行列を用いることにより1回の条件判定を  $O(1)$  で行えるため,  $O(DN^2 + N^3)$  となり, 実用上は  $\beta$  スケルトンよりも高速に計算可能である. なお, 任意の次元数  $D$  のグラフについて, RNG に対しては RNG のスーパーグラフを求めて条件に合わない辺を削除するという方法で  $O(N^2)$  以下のアルゴリズム [13, 14] が提案されている. GG に対してはヒューリスティックによる高速化 [15] が提案されているが  $O(DN^3)$  未満のアルゴリズムは知られていない.

### 3 重み付き $\beta$ 緩和相対近傍グラフ

#### 3.1 $\beta$ 緩和

新しいパラメータ付き近接グラフである  $\beta$  緩和 RNG ( $\beta$ RNG) を提案する.  $\beta$ RNG の近傍条件は, RNG の弓形領域とパラメータ  $\beta$  で指定される超球の共通集合内に他の頂点が含まれないことである. 任意の 2 点を  $v_i, v_j$  とし, 2 点の midpoint を中心とする超球上の点  $v_k$  が以下の式を満足するように超球を構成する.

$$\beta = \frac{\|v_i - v_k\|^2 + \|v_j - v_k\|^2}{\|v_i - v_j\|^2}, \quad 1 \leq \beta \leq 2 \quad (5)$$

これを变形すると以下のようにになる.

$$\begin{aligned} \|v_i - v_k\|^2 + \|v_j - v_k\|^2 &= \beta \|v_i - v_j\|^2 \\ 2\left\|\frac{v_i + v_j}{2} - v_k\right\|^2 + \frac{1}{2}\|v_i - v_j\|^2 &= \beta \|v_i - v_j\|^2 \\ \|v_k - \frac{v_i + v_j}{2}\|^2 &= \left(\frac{\sqrt{2\beta - 1}}{2}\|v_i - v_j\|\right)^2 \end{aligned} \quad (6)$$

したがって,  $\beta$ RNG( $V, E$ ) は以下のように定義できる.

$$(v_i, v_j) \in E \iff HS(v_i, \|v_i - v_j\|) \cap HS(v_j, \|v_i - v_j\|) \cap HS\left(\frac{v_i + v_j}{2}, \frac{\sqrt{2\beta - 1}}{2}\|v_i - v_j\|\right) \cap V = \phi \quad (7)$$

この様子を Fig.4 に示す. 斜線部が  $\beta$ RNG( $\beta=1.5$ ) の条件領域である.

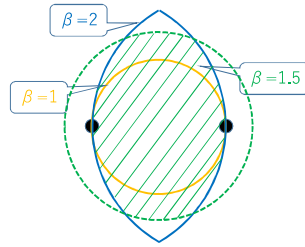


Fig. 4:  $\beta$ RNG における辺の判定

$\beta$ RNG は GG の部分グラフであり, RNG のスーパーグラフである.  $\beta$ スケルトンと同様に,  $\beta=1$  の場合が GG,  $\beta=2$  の場合が RNG である.

$\beta$ RNG における辺の判定は, 以下のようにになる.

- $d(v_i, v_k) < d(v_i, v_j)$  かつ  $d(v_j, v_k) < d(v_i, v_j)$  かつ  $\frac{\|v_i - v_k\|^2 + \|v_j - v_k\|^2}{\|v_i - v_j\|^2} < \beta$  を満足する  $v_k$  が存在しないならば,  $v_i$  と  $v_j$  を辺で接続する.

$\beta$ RNG の計算量は,  $d(v_i, v_j)$  を行列として求めておけば, 条件判定を  $O(1)$  で行えるため, GG や RNG と同様に  $O(DN^2 + N^3)$  である.

Fig.5 にアルゴリズムを示す. 辺  $(v_i, v_j)$  が存在するときは  $e_{ij} = 1$ , そうでないときは  $e_{ij} = 0$  となる.

#### 3.2 重み付きグラフ

重み付きグラフは, 辺に重みやコストが付いたグラフである.  $\beta$ RNG では, 辺に  $\beta$  の重みが付いた重み付き  $\beta$ RNG を生成することが可能である. RNG の弓形領域中に入った頂点  $v_k$  の中で最小の  $\beta$  値が重みとなる.

任意の 2 点  $v_i, v_j$  について, 以下のように辺  $(v_i, v_j)$  の重み  $w_{ij}$  を決める.

```

βRNG() {
  for(i=1; i≤N; i++)
    for(j=i+1; j≤N; j++) dij=dji=d(vi, vj)2;
  for(i=1; i≤N; i++)
    for(j=i+1; j≤N; j++) {
      for(k=1; k≤N; k++)
        if(k!=i && k!=j && dik<dij && djk<dij) {
          if(dik2 + djk2<βdij2) break;
        }
      if(k>N) eij=1; else eij=0;
      eji=eij;
    }
}

```

Fig. 5: βRNG のアルゴリズム

1.  $w_{ij} = 2$  に初期化する
2.  $d(v_i, v_k) < d(v_i, v_j)$  かつ  $d(v_j, v_k) < d(v_i, v_j)$  を満たす  $v_k$  に対して、以下を計算する.

$$w_{ij} = \min_{k \notin \{i, j\}} \left\{ \frac{\|v_i - v_k\|^2 + \|v_j - v_k\|^2}{\|v_i - v_j\|^2} \right\} \quad (8)$$

3.  $w_{ij} \geq 1$  ならば  $v_i$  と  $v_j$  を辺で接続する. そうでなければ  $w_{ij} = 0$  とし接続しない.

Fig.6 にアルゴリズムを示す.

```

weighted βRNG() {
  for(i=1; i≤N; i++)
    for(j=i+1; j≤N; j++) dij=dji=d(vi, vj)2;
  for(i=1; i≤N; i++)
    for(j=i+1; j≤N; j++) {
      wij=2;
      for(k=1; k≤N; k++)
        if(k!=i && k!=j && dik<dij && djk<dij) {
          β=(dik2 + djk2)/dij2;
          if(β<wij) {
            wij=β;
            if(wij<1) { wij=0; break; }
          }
        }
      wji=wij;
    }
}

```

Fig. 6: 重み付き βRNG のアルゴリズム

## 4 種分化

種分化 (speciation) は、生物学的には新しい種が誕生するための進化プロセスであるが、EA においては、多峰性関数において峰を捉え、複数の解を同時に探索するために用いられてきた [6].

種分化の方法としては、種半径 (species radius) を用いる方法、k-means クラスタリングを用いる方法、k 近傍法を用いる方法、近接グラフを用いる方法などが提案されている [16]. 以下では、種分化の対象となる探索点集合  $P$  を  $P = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  とし、各探索点が属す種の代表 (seed) を  $\mathbf{x}_{seed}(\mathbf{x}_i)$  と表記する.

## 4.1 種半径による種分化

種半径による種分化では、近傍構造は種半径  $R$  で定義される。種の代表と代表を中心とする半径  $R$  の超球の内部 (超球上を含む) に存在する点により 1 つの種を構成する。点集合と各点の関数値が与えられたとき、本方法による種形成のアルゴリズムを以下に示す [17, 18, 19]。各点は 1 つの種にのみ所属することになる。

1. 探索点集合を関数値の良い順にソートする。
2. 探索点集合の最良点  $\mathbf{x}_b$  を新しい種の代表とする ( $seed(\mathbf{x}_b) = b$ )。代表から半径  $R$  以内の全ての点  $\mathbf{x}_i$  をその種に属させる ( $seed(\mathbf{x}_i) = b$ )。
3. 新しい種に所属した探索点を探索点集合から削除する。
4. 探索点集合が空になるまで 2. へ戻る。

適切な種半径は問題および探索過程により変化するため、この方法では種半径の選択が困難であるという問題点がある。また、種の形状が超球でない場合に対応しにくいという問題点もある。

## 4.2 クラスタリングによる種分化

k-means クラスタリングによる種分化 [20] では、近傍構造はクラスタ数  $k$  によって定義される。1 つのクラスタが 1 つの種に対応し、クラスタ内の最良の点が種の代表となる。各点は 1 つの種にのみ所属することになる。本方法のアルゴリズムを以下に示す。

1. k-means 法により、探索点集合  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  を  $k$  個のクラスタ  $C = \{C_1, C_2, \dots, C_k\}$  に分割する。
  - (a) 初期化. 各探索点  $\mathbf{x}_i$  について、所属するクラスタ  $C_j$  をランダムに選択する。
  - (b) 更新ステップ. クラスタの重心を新しいクラスタ中心とする。

$$\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i \quad (9)$$

- (c) 割当ステップ. 各点を最も近いクラスタに割り当てる。

$$C_j = \{\mathbf{x}_i | \arg \min_{1 \leq h \leq k} \|\mathbf{x}_i - \boldsymbol{\mu}_h\| = \boldsymbol{\mu}_j\} \quad (10)$$

- (d) 割当に変化がなくなるまで、(b) へ戻る。

2. 各クラスタにおける最良点を種の代表とする。

## 4.3 近傍法による種分化

k 近傍法による種分化では、近傍構造は隣接点の数  $k$  によって定義される。各点と  $k$  個の最近傍点によって種が構成される。各点に対して種が構成されるため、各点は複数の種に属することがある。本方法のアルゴリズムを以下に示す。

1. 全ての探索点間の距離を求める。
2. 各探索点  $\mathbf{x}_i, i = 1, 2, \dots, N$  に対して、
  - (a) 距離が近い上位  $k$  個の近傍点を求める。
  - (b)  $\mathbf{x}_i$  と近傍点により種を構成し、各種における最良点を種の代表とする。

## 4.4 近接グラフによる種分化

近接グラフによる種分化では、近傍構造は辺(接続)の有無によって定義される。各探索点とその点に接続された探索点によって種が構成される。各探索点に対して種が構成されるため、探索点は複数の種に属することがある。本方法のアルゴリズムを以下に示す。

1. 近接グラフを作成する。
2. 各探索点  $\mathbf{x}_i, i = 1, 2, \dots, N$  に対して,
  - (a) 辺の集合  $E$  を参照し、接続されている点を求める。
  - (b)  $\mathbf{x}_i$  と接続されている点により種を構成し、種における最良点を種の代表とする。

$$seed(\mathbf{x}_i) = \arg \min_{\{l|i=l \text{ or } (v_i, v_l) \in E\}} f(\mathbf{x}_l) \quad (11)$$

## 5 近接グラフに基づく多峰性最適化

本研究では、近接グラフに基づく種分化型 DE (Speciation-based DE using  $\beta$ RNG, SDE-G/ $\beta$ RNG) を提案する。

### 5.1 最適化問題

上下制限約付きの最適化問題 (P) は、以下のように定義できる。

$$\begin{aligned} \text{(P) minimize } & f(\mathbf{x}) \\ \text{subject to } & l_i \leq x_i \leq u_i, i = 1, \dots, D \end{aligned} \quad (12)$$

ここで、 $\mathbf{x} = (x_1, \dots, x_D)$  は  $D$  次元決定変数ベクトル、 $f(\mathbf{x})$  は目的関数である。 $l_i, u_i$  はそれぞれ、 $D$  個の決定変数  $x_i$  の下限値、上限値である。以下では上下制限約のみを満足する領域である探索空間を  $S$  で表現する。

### 5.2 差分進化

差分進化 (DE) は EA の一つであり、Storn and Price[2] によって提案された。DE は確率的な直接探索法であり、解集団を用いた多点探索を行う。DE は非線形問題、微分不可能な問題、非凸問題、多峰性問題などの様々な最適化問題に適用されてきており、これらの問題に対して高速で頑健なアルゴリズムであることが示されてきている。

DE には幾つかの形式が提案されており、DE/best/1/bin や DE/rand/1/exp などがよく知られている。これらは、DE/base/num/cross という記法で表現される。“base” は基本ベクトルとなる親の選択方法を指定する。例えば、DE/rand は基本ベクトルのための親を集団からランダムに選択し、DE/best は集団の最良個体を選択する。“num” は基本ベクトルを変異させるための差分ベクトルの個数を指定する。“cross” は子を生成するために使用する交叉方法を指定する。例えば、DE/base/num/bin は一定の確率で遺伝子を交換する交叉 (binomial crossover) を用い、DE/base/num/exp は、指数関数的に減少する確率で遺伝子を交換する交叉 (exponential crossover) を用いる。

DE では、探索空間中にランダムに初期個体を生成し、初期集団を構成する。各個体は決定ベクトルに対応し、 $D$  個の決定変数を遺伝子として持つ。各世代において、全ての個体を親として選択する。各親に対して、次のような処理が行われる。突然変異のために、選択された親を除く個体集団から互いに異なる  $1 + 2 \text{ num}$  個の個体を選択する。最初の個体が基本ベクトルとなり、残りの個体対が差分ベクトルとなる。差分ベクトルは  $F$  (scaling factor) が乗算され基本ベクトルに加えられる。その結果得られたベクトルを変

異ベクトルといい、親と交叉する、すなわち、 $CR$ (crossover factor)により指定された確率で親の遺伝子を変異ベクトルの要素で置換することにより、子のベクトル (trial vector) が生成される。最後に、生存者選択として、子が親よりも良ければ、親を子で置換する。

本研究では、差分ベクトル数を1 ( $num = 1$ )としたDE/species-best/1/binを用いる。species-best戦略は、次節で説明するが、ベースベクトルとして個体が所属する種の代表を選択する戦略である。

### 5.3 SDE-G/ $\beta$ RNGのアルゴリズム

以下にアルゴリズムの概要を示す。

- (0) 設定. 本研究では重みのない $\beta$ RNGを用いるために $\beta$ を指定する。DEのために、個体数 $N$ 、スケールリングファクター $F$ 、交叉率 $CR$ を指定する。複数解を記憶するためにアーカイブを用いるので、アーカイブサイズ $N_A$ を指定する。
- (1) 初期化.  $N$ 個の初期個体 $\mathbf{x}_i$ を初期探索点として生成し、初期集団 $P = \{\mathbf{x}_i, i = 1, 2, \dots, N\}$ を構成する。全ての個体を評価する。アーカイブ $A$ を初期化するために $A = P$ とする。
- (2) 終了判定. 終了条件を満足すれば、アルゴリズムは終了する。終了条件としては、最大の繰り返し回数や関数評価回数を用いることが多い。
- (3) 種分化.  $\beta$ RNGを生成する。グラフに基づく種分化を行い、各個体に対する代表 $\mathbf{x}_{seed}(\mathbf{x}_i)$ を決定する。
- (4) 突然変異と交叉. 全ての個体 $\mathbf{x}_i$ に対して、下記の突然変異と交叉操作を行う。
- (4a) 突然変異. 突然変異には、species-best戦略 [18, 16]を用いる。各個体 $\mathbf{x}_i$ に対して、2個体 $\mathbf{x}_{r_2}, \mathbf{x}_{r_3}$ を $\mathbf{x}_i$ および互いに重複しないようにランダムに選択する。変異ベクトル $\mathbf{m}_i$ を基本ベクトルである種の代表および差分ベクトル $\mathbf{x}_{r_2} - \mathbf{x}_{r_3}$ から以下のように生成する。

$$\mathbf{m}_i = \mathbf{x}_{seed}(\mathbf{x}_i) + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (13)$$

- (4b) 交叉. ベクトル $\mathbf{m}_i$ を親 $\mathbf{x}_i$ と交叉し、子ベクトル $\mathbf{x}'_i$ を生成する。交叉点 $j$ を全ての次元 $[1, D]$ からランダムに選択する。子ベクトル $\mathbf{x}'_i$ の $j$ 番目の要素を $\mathbf{m}_i$ の $j$ 番目の要素から継承する。それ以外の次元は、交叉率 $CR$ の確率で $\mathbf{m}_i$ の要素から、 $1 - CR$ の確率で $\mathbf{x}_i$ から継承する。実際の処理では、(4a)と(4b)は一まとまりの処理で実現される。
- (5) 生存者選択. 子ベクトルを評価する。子ベクトル $\mathbf{x}'_i$ が親ベクトルよりも良ければ子ベクトルを生存者とし、アーカイブ $A$ に保存する。ただし、 $A$ の要素数が $N_A$ 未満ならば単に追加するが、 $N_A$ 以上ならば、最も近い要素と比較し、その要素よりも良ければ、置換する。そうでなければ親ベクトルを生存者とする。
- (6)  $P$ を生存者で置換し、(2)に戻る。

Fig.7にSDE-G/ $\beta$ RNGの擬似コードを示す。

## 6 実験

### 6.1 テスト関数

テスト関数として、5つの峰(谷)を持つ5-peaks関数を用いる。この関数は、本研究のために設計した、2次元以上の任意の次元数で定義される関数であり、以下のように定義される。

$$f(\mathbf{x}) = -\sum_{i=1}^5 \exp\left(-\frac{\sum_{j=1}^D (x_j - a_{ij})^2}{\sigma^2}\right), \quad (14)$$

$$a_{1j} = -1, a_{2j} = 0, a_{3j} = 1, a_{4j} = (-1)^j, a_{5j} = (-1)^{(j-1)}, \sigma = 0.3$$



```

SDE-G/ $\beta$ RNG/species-best/1/exp()
{
  P=N individuals  $\{x_i\}$  generated randomly in  $S$ ;
  A=P;
  for(t=1; termination condition is false; t++) {
    E=Edges of  $\beta$ RNG created from P;
    Obtain  $\{seed(x_i)\}$  by Speciation using E;
    for(i=1; i  $\leq$  N; i++) {
       $r_1=seed(x_i)$ ;
       $x_{r_2}$ =Randomly selected from  $P(r_2 \notin \{i, r_1\})$ ;
       $x_{r_3}$ =Randomly selected from  $P(r_3 \notin \{i, r_1, r_2\})$ ;
      k=select randomly from  $[1, D]$ ;
      for(j=1; j  $\leq$  D; j++) {
        if(j==k ||  $u(0, 1) < CR$ )
           $x'_{i,j}=x_{r_1,j}+F(x_{r_2,j} - x_{r_3,j})$ ;
        else
           $x'_{i,j}=x_{i,j}$ ;
      }
      if( $f(x'_i) < f(x_i)$ ) {
         $z_i=x'_i$ ;
        if( $|A| < N_A$ )  $A=A \cup \{x'_i\}$ ;
        else {
           $x_a=\arg \min_{x_a \in A} \|x'_i - x_a\|$ ;
          if( $f(x'_i) < f(x_a)$ )  $x_a$  is replaced by  $x'_i$ ;
        }
      }
      else
         $z_i=x_i$ ;
    }
    P= $\{z_i\}$ ;
  }
}

```

Fig. 7: SDE-G/ $\beta$ RNG の擬似コード。ただし、 $u(0, 1)$  は区間  $[0, 1]$  の一様乱数である。

最適解は、 $(1, 1, \dots, 1)$ ,  $(0, 0, \dots, 0)$ ,  $(-1, -1, \dots, -1)$ ,  $(1, -1, \dots, (-1)^{(D+1)})$ ,  $(-1, 1, \dots, (-1)^D)$  の 5 個、最適値は  $-1$  である。Fig.8 に  $D = 2$  における関数のグラフを示す。

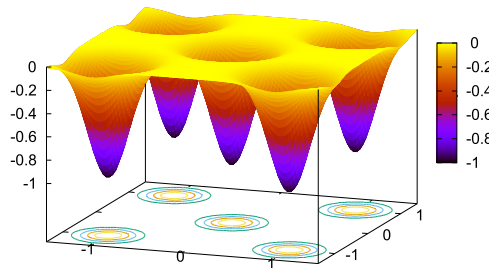


Fig. 8: 5-peaks のグラフ

## 6.2 実験条件

5-peaks 問題において  $D \in \{2, 3, 5, 10, 20\}$  の次元数の問題を最適化する。SDE-G のパラメータ設定は、個体数  $N = 50 + 10D$ 、 $F$  と  $CR$  については、 $F \in \{0.4, 0.5, \dots, 0.8, 0.9\}$ 、 $CR \in \{0, 0.2, \dots, 0.8, 1\}$  から個体毎にランダムに選択した。アーカイブサイズは  $N_A = 3N$  とした。

1 試行あたりの最大関数評価回数を  $20000D$  とし, 25 回試行の平均値を調べる. アークオブ中の個体の関数値と大域最適値との誤差が  $10^{-5}$  以下で,  $10^{-2}$  以上距離の離れた異なる解を, 発見した大域最適解とする. 実験結果を評価するために, 多峰性最適化で良く使用される以下のような指標を用いる [21].

- ピーク率 (peak ratio, PR) は, 全ての試行において発見した大域最適解の割合である. ただし, 大域最適値との誤差が許容誤差以下の解が見つければ, 大域最適解を見つけたと判断する. 全試行で全ての最適解を発見できれば 1 となる.

$$PR = \frac{\sum_{i=1}^{NR} NPF_i}{NKP * NR} \quad (15)$$

$NR$  は試行回数,  $NPF_i$  は第  $i$  試行で見つけた大域最適解の数,  $NKP$  は既知の大域最適解の数である.

- 成功率 (success rate, SR) は, 全ての大域最適解を発見した成功した試行の割合である.

$$SR = \frac{NSR}{NR} \quad (16)$$

$NSR$  は成功した試行の数である.

- 収束速度 (convergence speed, CS) は, 全ての大域最適解を発見するまでに要した関数評価回数の平均である.

$$CS = \frac{\sum_{i=1}^{NR} FE_i}{NR} \quad (17)$$

$FE_i$  は第  $i$  試行で全ての大域最適解を発見するまでに要した関数評価回数である. 全てを発見できなかった場合は, 最大関数評価回数とする.

$\beta$  RNG のパラメータは,  $\beta \in \{1, 1.25, 1.5, 1.75, 2\}$  に設定して結果を比較する.

### 6.3 実験結果

実験結果を Table 1 に示す. 各次元数の問題に対して,  $\beta$  RNG の  $\beta$  を変化させた結果について, SR, PR, CS の平均値を示した. なお, CS については標準偏差も示した. 結果を比較する基準として様々なものが考えられる. ここでは, SR が最大となる結果, SR が同じならば PR が最大となる結果, RP が同じならば CS が最小となる結果という基準で比較し, 最良の結果を太字で示した.

$2D, 3D, 5D$  については,  $2D$  の  $\beta = 1.25$  と  $5D$  の  $\beta = 1$  を除き, 全ての試行で全ての大域最適解を探索できている.  $2D$  の  $\beta = 1.25$  と  $5D$  の  $\beta = 1$  の場合は 1 試行で 1 個の大域最適解を見つけられなかったという結果になっており, 性能に大きな差は無い. 収束速度については, 成功した試行のみを考慮すれば  $\beta = 1$  の場合が最も速く,  $\beta$  が大きくなるにしたがって遅くなる傾向がある.  $\beta$  が小さくなると, 多くの辺が生成されるため, 多くの個体の中で最良となる個体を中心に種を形成することになる. このため, 種のメンバー数が大きくなり, 種の数が少なくなる. 少ない種で探索を進めるため, 大域最適解に速く収束したと考えられる.

$10D, 20D$  については, いずれの場合も全ての大域最適解を探索できていない試行がある. SR と PR については,  $\beta$  が大きい方が大きくなる傾向があり,  $\beta = 2$  の場合が最も大きい. 一般に次元数が大きくなると, 探索空間が広大になるため, 個体数を多少大きくしても探索点が疎になり,  $\beta$  が小さいと完全グラフに近い状態となる. このような状況では, 種の数が大域最適解の数よりも少なくなり, 少数の最適解しか見つけられなくなってしまう.

Fig.9 に各次元毎の種の数推移を示す. 種数は, 種における最良の個体が自分自身となる数, すなわち  $seed(x_i) = x_i$  となる個体数である. 全ての次元数において,  $\beta$  が小さいほど種の数も小さいことが分かる. 最も種の数小さい  $\beta = 1$  の場合に注目すると,  $2D, 3D$  では種数がほぼ 5 以上となっているが,  $5D$  では初期に種数が小さく次第に 5 に近づいている.  $10D, 20D$  では, 常に種数が 5 未満となっており, 全ての最適解を見つけるのが非常に困難であることが分かる.

Table 1: 実験結果

		$\beta = 1$	$\beta = 1.25$	$\beta = 1.5$	$\beta = 1.75$	$\beta = 2$
2D	SR	1	0.96	1	1	1
	PR	1	0.992	1	1	1
	CS	<b>5146 ± 396</b>	6828 ± 6786	5510 ± 462	5732 ± 436	5650 ± 426
3D	SR	1	1	1	1	1
	PR	1	1	1	1	1
	CS	<b>9226 ± 525</b>	9984 ± 567	10307 ± 566	10746 ± 771	10954 ± 760
5D	SR	0.96	1	1	1	1
	PR	0.992	1	1	1	1
	CS	22628 ± 15811	<b>22048 ± 1153</b>	23688 ± 1114	24732 ± 1133	25624 ± 1259
10D	SR	0.04	0.76	0.88	0.92	<b>0.96</b>
	PR	0.72	0.952	0.976	0.968	<b>0.992</b>
	CS	194226 ± 28287	97362 ± 57710	90708 ± 40429	91906 ± 32013	<b>89084 ± 22853</b>
20D	SR	0	0.04	0	0.16	<b>0.2</b>
	PR	0.24	0.352	0.504	0.608	<b>0.648</b>
	CS	400000 ± 0	391520 ± 41543	400000 ± 0	373230 ± 63390	<b>379050 ± 46044</b>

## 7 おわりに

本研究では、多峰性問題を解くために、グラフによる種分化に着目した。従来のグラフによる種分化では主に GG を用いており、高次元の問題では少数の最適解しか発見できないという問題点があった。これに対して、GG と RNG の中間的な近傍グラフを比較的高速に生成できる  $\beta$ RNG を提案した。2次元から20次元の 5-peaks 問題を様々な  $\beta$  値で最適化することにより、低次元の問題では小さい  $\beta$  で高速に解を探索し、高次元の問題では大きい  $\beta$  でなるべく多くの解を探索することが必要であることが分かった。ただし、最適解の数がより大きな問題ではより大きな  $\beta$  を用いる必要があると考えられる。

今後は、より多様な多峰性最適化問題について、提案手法を適用する予定である。最適解の数は事前に知ることはできないため、初期は大きな  $\beta$  を用い、次第に  $\beta$  を減少させることにより、なるべく多くの解をより高速に求める方法について検討する予定である。また、重み付きの辺を生成できる重み付き  $\beta$ RNG の利用方法についても検討したいと考えている。

謝辞 本研究は、JSPS 科研費 JP17K00311, JP19K04916 の助成を受けている。

## 参考文献

- [1] R. Storn and K. Price: “Minimizing the real functions of the ICEC’96 contest by differential evolution”, Proc. of the International Conference on Evolutionary Computation, pp. 842–844 (1996).
- [2] R. Storn and K. Price: “Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces”, Journal of Global Optimization, **11**, pp. 341–359 (1997).
- [3] B. Sareni and L. Krahenbuhl: “Fitness sharing and niching methods revisited”, IEEE Transactions on Evolutionary Computation, **2**, 3, pp. 97–106 (1998).
- [4] J.-P. Li, X. Li and A. Wood: “Species based evolutionary algorithms for multimodal optimization: A brief review”, IEEE Congress on Evolutionary Computation, pp. 1–8 (2010).
- [5] X. Li: “Efficient differential evolution using speciation for multimodal function optimization”, Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp. 873–880 (2005).
- [6] 柴坂, 原, 市村, 高濱: “種分化を導入した Differential Evolution による複数解をもつ多峰性関数の最適化”, 電子情報通信学会論文誌, **J92-D**, 7, pp. 1003–1014 (2009).
- [7] W. Gao, G. G. Yen and S. Liu: “A cluster-based differential evolution with self-adaptive strategy for multimodal optimization”, IEEE transactions on cybernetics, **44**, 8, pp. 1314–1327 (2014).
- [8] T. Takahama and S. Sakai: “Differential evolution with graph-based speciation by competitive hebbian rules”, Proc. of the Sixth International Conference on Genetic and Evolutionary Computing (ICGEC2012), pp. 445–448 (2012).

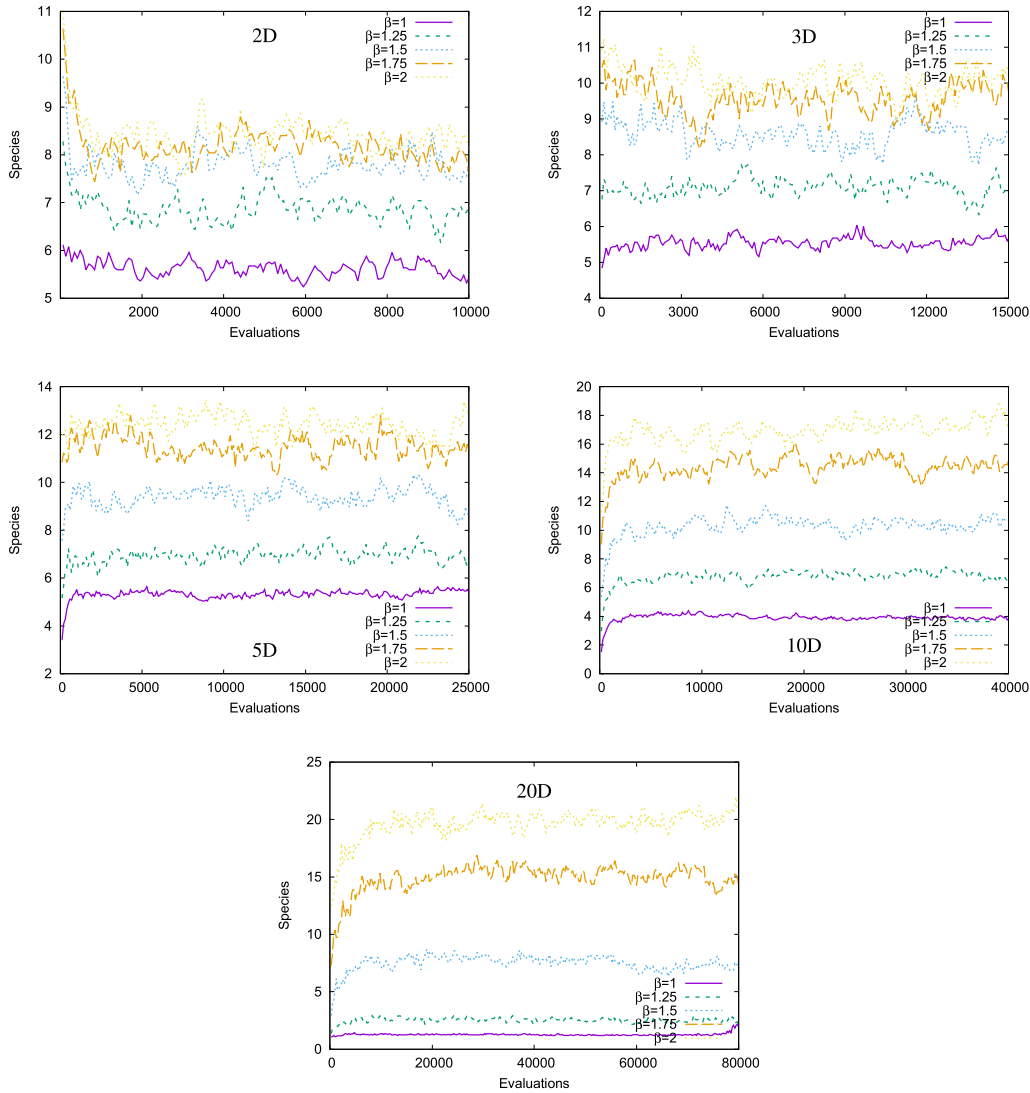


Fig. 9: 種の数推移

- [9] 有田, 串田, 原, 高濱: “近傍グラフ上の山谷判定に基づく種分化と動的 crowding を用いた Differential Evolution による複数解探索”, 2014 IEEE SMC Hiroshima Chapter 若手研究会講演論文集, pp. 93–96 (2014).
- [10] K. R. Gabriel and R. R. Sokal: “A new statistical approach to geographic variation analysis”, *Systematic Zoology*, **18**, pp. 259–270 (1969).
- [11] G. T. Toussaint: “The relative neighborhood graph of a finite planar set”, *Pattern Recognition*, **12**, 4, pp. 261–268 (1980).
- [12] D. G. Kirkpatrick and J. D. Radke: “A framework for computational morphology”, *Computational Geometry* (Ed. by G. Toussaint), North-Holland, pp. 217–248 (1985).
- [13] K. J. Supowit: “The relative neighborhood graph, with an application to minimum spanning trees”, *Journal of the ACM (JACM)*, **30**, 3, pp. 428–448 (1983).
- [14] T.-H. Su and R.-C. Chang: “On constructing the relative neighborhood graphs in euclidean-dimensional spaces”, *Computing*, **46**, 2, pp. 121–130 (1991).
- [15] B. K. Bhattacharya, R. S. Poulsen and G. T. Toussaint: “Application of proximity graphs to editing nearest neighbor decision rule”, *International Symposium on Information Theory*, Santa Monica (1981).
- [16] S. Sakai and T. Takahama: “A comparative study on neighborhood structures for speciation in species-based differential evolution”, *Social Systems Solutions Applied by Economic Sciences and Mathematical Solutions* (Eds. by M. Kitahara and C. Czerkawski), Kyushu University Press, Fukuoka, pp. 111–135 (2012).
- [17] X. Li: “Efficient differential evolution using speciation for multimodal function optimization”, *Proc. of the 2005 Conference on Genetic and Evolutionary Computation*, pp. 873–880 (2005).

- [18] M. Shibusaka, A. Hara, T. Ichimura and T. Takahama: “Species-based differential evolution with switching search strategies for multimodal function optimization”, Proc. of the 2007 IEEE Congress on Evolutionary Computation, pp. 1183 –1190 (2007).
- [19] B. Qu and P. Suganthan: “Modified species-based differential evolution with self-adaptive radius for multimodal optimization”, Proc. of the 2010 International Conference on Computational Problem-Solving, pp. 326 –331 (2010).
- [20] A. Passaro and A. Starita: “Particle swarm optimization for multimodal functions: A clustering approach”, Journal of Artificial Evolution and Applications, **2008**, pp. 1–15 (2008).
- [21] X. Li, A. Engelbrecht and M. G. Epitropakis: “Benchmark functions for CEC’2013 special session and competition on niching methods for multimodal function optimization”, RMIT University, Evolutionary Computation and Machine Learning Group, Australia, Tech. Rep (2013).