

# 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

## Species-based Differential Evolution with Detecting Landscape Modality and Dynamic Crowding for Multimodal Optimization

有田 紘人 串田 淳一 原 章 高濱 徹行

Hiroto Arita Jun-ichi Kushida Akira Hara Tetsuyuki Takahama

広島市立大学大学院情報科学研究科知能工学専攻

Email: harita @ ints.info.hiroshima-cu.ac.jp, {kushida, ahara, takahama} @ hirosshima-cu.ac.jp

**Abstract**—In multimodal function optimization problems, which have multiple global optimal solutions, Species-based Differential Evolution (SDE) has been proposed in order to find various global optima simultaneously. In this method, the population is divided into multiple subpopulations by using speciation, and each species focuses its search on one optimal solution. In this paper we improve SDE by landscape modality which is detected from a neighborhood graph of the population. At each generation, Hill-Valley detection is performed for the neighborhood graph and the speciation is performed based on it. Furthermore, dynamic crowding where the most similar individual to an offspring is replaced with the offspring is introduced to preserve diversity of population. We confirmed the effectiveness of the proposed method by comparing with conventional method on standard multimodal benchmark functions.

### I. はじめに

複数の最適解をもつ目的関数の最小値または最大値を探索する関数最適化問題において、全ての最適解を発見することは非常に困難である。このような問題を解く方法として、進化的アルゴリズムである Differential Evolution (DE) を拡張し、種分化という概念を用いて空間を分割しながら、部分空間ごとに探索を行う Species-based DE (SDE) [1] が提案されている。また、個体集団の近傍グラフを利用して種分化を行う Graph-based SDE (GDE) [2] がある。

また、多峰性関数を探索するための手法として、通常の DE では親個体を子個体で置き換えるのに対し、子個体に最も近い個体を置き換える Crowding DE (CDE) [3] が提案されている。CDE では crowding を行うことで個体集団が探索過程でサブ集団に分割されていき、多峰性関数を大域的に探索することができる。ただし、crowding は全ての個体に行うと非常に収束速度が遅くなってしまふ。一方、近傍グラフから関数の山と谷 (形状) を判別し、それを用いて DE の制御パラメータを調整することで探索性能を向上させる手法も提案されている [4]。

本研究では複数解を同時に探索する手法として、近傍グラフ上の山谷判定に基づき種分化を行う DE を提

案する。これは、近傍グラフを利用して谷を判定し、谷を中心に種分化を行う方法である。また、crowding によって収束速度が低下するという問題を解決するために、crowding を行う個体を谷に近い個体に制限する動的 crowding を提案する。さらに、crowding を行う際に現在探索している谷とは別の谷を探索することを防ぐために、差分ベクトルを小さくする局所的な差分ベクトルの生成方法を導入する。

提案手法を 5 つのベンチマーク問題に適用し、従来手法との性能の比較を行う。

### II. 従来手法

#### A. Differential Evolution

Differential Evolution (DE) は、実数空間における最適化アルゴリズムである。アルゴリズムの基本的な構造は遺伝的アルゴリズムと似ているが、個体集団の各個体の差分ベクトルを遺伝的操作で利用する点が異なる。DE のアルゴリズムは以下ようになる。

1. 初期集団  $P = \{\mathbf{x}^k | k=1, \dots, NP\}$  を構築
2. 個体の関数値を計算
3. 子個体の生成
  - (a)  $\mathbf{x}^{r_1}, \mathbf{x}^{r_2}, \mathbf{x}^{r_3}$  の選択
  - (b) 変異ベクトル  $\mathbf{v}$  の生成
  - (c) 変異ベクトルと親個体を交叉
4. 置き換え
5. 終了条件を満たしていなければ、3に戻る

初期集団における各個体  $\mathbf{x}^k = (x_1^k, x_2^k, \dots, x_D^k)$ ,  $k=1, \dots, NP$  は、問題の各次元における定義域内でランダムに初期化される。ただし、 $D$  は問題の次元数、 $NP$  は集団中の個体数である。次に、 $D$  次元の情報をもとにして、全  $NP$  個の個体の関数値  $f(\mathbf{x}^k)$  を計算する。

各々の個体  $\mathbf{x}^P$  において、それぞれ一つずつ子個体を生成する。DE における子個体の生成は、親個体  $\mathbf{x}^P$  と式 (1) で生成される変異ベクトル  $\mathbf{v}$  を遺伝子単位で交叉させることで行われる。

$$\mathbf{v} = \mathbf{x}^{r_1} + F(\mathbf{x}^{r_2} - \mathbf{x}^{r_3}) \quad (1)$$

式(1)の $\mathbf{x}^{r_1}$ ,  $\mathbf{x}^{r_2}$ ,  $\mathbf{x}^{r_3}$ は条件 $r_1 \neq r_2 \neq r_3 \neq P$ のもとで選択される。 $\mathbf{x}^{r_1}$ はベクトル、 $\mathbf{x}^{r_2}$ ,  $\mathbf{x}^{r_3}$ は差分ベクトルを生成するのに用いるベクトルである。パラメータ $F$ は、差分ベクトルの伸縮を意味するスケーリングファクターである。

交叉を始める遺伝子座 $i$ をランダムに決め、区間 $[0,1]$ で発生させた乱数が交叉率 $C_r$ より小さくかつすべての遺伝子が交叉済みでなければ、親個体 $\mathbf{x}^P$ をコピーした子個体 $\mathbf{x}^{child}$ の遺伝子を変異ベクトル $\mathbf{v}$ の遺伝子に置換する。交叉の方法としては、exponential crossoverやbinomial crossoverなどが用いられている。本研究ではexponential crossoverを使用する。

DEの置き換えは、生成された子個体 $\mathbf{x}^{child}$ と親個体 $\mathbf{x}^P$ の関数値の比較を行い、子個体の関数値の方が良ければ置換を行う。

### B. Crowding DE

Crowding DE(CDE)[3]は、ニッチング手法の1つであるcrowdingを導入したDEであり、生存選択において通常の親個体を置き換えの対象とするのではなく、子個体の最近傍個体を置き換えの対象とする。crowdingを行うことにより、探索空間を分割しながら大域的な探索を行える。

1. 初期集団 $P=\{\mathbf{x}^k | k=1, \dots, NP\}$ を構築
2. 個体の関数値を計算
3. 子個体の生成
  - (a)  $\mathbf{x}^{r_1}$ ,  $\mathbf{x}^{r_2}$ ,  $\mathbf{x}^{r_3}$ の選択
  - (b) 変異ベクトル $\mathbf{v}$ の生成
  - (c) 変異ベクトルと親個体を交叉
4. 子個体と最近傍個体を決定
5. 置き換え
6. 終了条件を満たしていなければ、3に戻る

1~3については、通常のDEと同じである。特徴的な点は4の部分である。CDEの置き換えは、生成された子個体 $\mathbf{x}^{child}$ と子個体に最もユークリッド距離が近い個体 $\mathbf{x}^{crowd}$ の関数値の比較を行い、子個体の関数値の方が良ければ置換を行う。

### C. 近傍グラフによる山谷判定

1) 近傍グラフの生成: 個体集団 $P=\{\mathbf{x}^k | k=1, \dots, NP\}$ が与えられるとき、近傍グラフの1つであるRelative Neighborhood Graph(RNG)は次のように生成される。

1. 個体集団から2個体 $\mathbf{x}^i$ ,  $\mathbf{x}^j$ を選択する
2.  $\mathbf{x}^i$ ,  $\mathbf{x}^j$ 間の距離 $d_{ij}$ をユークリッド距離で求める
3. 全体から個体 $\mathbf{x}^z$ を選択する
4.  $\mathbf{x}^i$ ,  $\mathbf{x}^z$ 間の距離 $d_{iz}$ 、 $\mathbf{x}^j$ ,  $\mathbf{x}^z$ 間の距離 $d_{jz}$ をユークリッド距離で求める

### 5. RNGの条件の判定

個体 $\mathbf{x}^i$ ,  $\mathbf{x}^j$ を中心とし、距離 $d_{ij}$ を半径とした円の共通部分に他の個体 $\mathbf{x}^z$ が存在するかどうかを式(2)で判定する。

$$d_{ij} < \max\{d_{iz}, d_{jz}\}, \text{ for all } \mathbf{x}^z \in P, \mathbf{x}^z \neq \mathbf{x}^i, \mathbf{x}^j \quad (2)$$

- a) 式(2)を満たさない個体が1つでも存在する場合、個体 $\mathbf{x}^i, \mathbf{x}^j$ は近傍ではない。
- b) 式(2)を満たさない個体が1つも存在しない場合、個体 $\mathbf{x}^i, \mathbf{x}^j$ は近傍である。

### 6. 全ての個体の組み合わせを調べ終われば終了、そうでなければ1に戻る。

2) 山谷判定: 関数景観の山と谷を判定[4]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、その辺を構築する2点のうち、関数値の良い点の谷度を1増加させ、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。

山谷判定を行う際の例を図1に示す。図1の例では、点 $p$ が谷になり、点 $r$ が山となる。

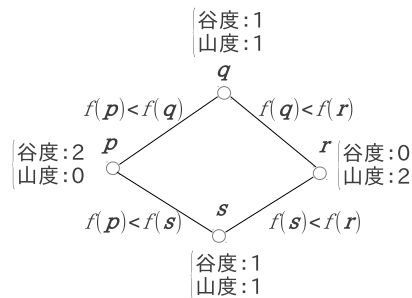


図1. 山谷判定の例

### D. 置き換え防止操作

DEでは、世代を重ねていくことで最適解付近に解は収束していくが、複数の最適解をもつ最適化問題では、進化を続けると個体密度の低い最適解周辺から個体が移動してしまうことがある。

これを防止するために、置き換え防止操作[2]を行う。置き換え防止操作は子個体との置き換えの際に親個体 $\mathbf{x}^i$ から一定の距離 $d_r$ 内に存在する個体数 $N_{\mathbf{x}^i}$ と、生成したトリアルベクトル $U$ から一定の距離 $d_r$ 内に存在する個体数 $N_U$ とを比較する。このとき、 $N_{\mathbf{x}^i} < N_U$ となる場合は子個体との置き換えを行わない。この操作を行うことで、最適解周辺から個体密度の高い位置へ個体が移動するのを防ぐことが可能である。

### III. 提案手法

#### A. 山谷判定による種分化を用いた DE(VGDE)

Hill-Valley detection based GDE(VGDE) のアルゴリズムを以下に示す。

1. ランダムに個体を生成し集団を構築する
  2. 全個体間の距離を求める
  3. 個体間の近傍を求め、相対近傍グラフを生成
  4. 近傍関係をもとに山谷判定を行う
  5. 山谷判定から種分化を行う
  6. 子個体を生成する
    - (a)  $x^{r_1}, x^{r_2}, x^{r_3}$  を選択する
    - (b) 変異ベクトル  $v$  を生成する
    - (c) 変異ベクトルと親個体を交叉する
  7. 子個体との比較および置き換えを行う
  8. 終了条件を満たしていなければ、2に戻る
- さらに、上記5のアルゴリズムを以下に示す。
- a) 山谷判定で得た山度、谷度から谷点を調べる
  - b) 見つかった谷点を種の支配者とする
  - c) 谷点となった個体以外は、自身の1番近くにある谷点の種に所属する

子個体の生成に用いるベースベクトルと差分ベクトルの選択方法を以下のようにする [5]。

- ベースベクトルは親個体が所属する種の支配者を選択する
- 差分ベクトル生成のための個体 ( $x^{r_2}, x^{r_3}$ ) は進化に多様性を持たせるために、全体からランダムに選択する

#### B. 山谷判定による動的 crowding

通常の DE では、子個体を生成した際には親個体をそのまま置き換える。本研究では crowding を行うことで、置き換える個体の変更を行っている。しかし、crowding を用いると集団の多様性は向上するが、親個体を取り残され、探索の速度が大きく低下してしまう。そこで、山谷判定から得られた谷度、山度を用い crowding を行う個体を決定する。

また、通常の DE では差分ベクトル生成のための個体は、進化に多様性を持たせるために集団全体からランダムに選択するが、VGDEにおいて crowding を行っても、差分ベクトルが大きいと図2のように自身の近傍の谷の個体は進化せず別の谷の個体の進化を行ってしまう場合がある。そこで、自身の周りを探索するために、親個体から最も近い個体との距離  $L_{small}$  を半径とする領域を作り、その領域にランダムに個体を2つ生成し、その2つの点で差分ベクトルを取る。 $D$ 次元の球内の個体は次のように作成する。

- (1)  $D$  個の独立な一様乱数  $u(0 \leq u_i \leq 1)$  を作成する ( $i=1,2,\dots,D$ )

- (2) 平均0, 分散1の正規分布をもつ  $D$  個の独立な正規乱数  $z$  を作成する

$$z_{2i-1} = (-2\log_e u_{2i-1})^{1/2} \cos 2\pi u_{2i}$$

$$z_{2i} = (-2\log_e u_{2i-1})^{1/2} \sin 2\pi u_{2i} \quad (i=1,2,\dots)$$

- (3) 作成された  $z$  の大きさ  $L_z$  を求める
- (4)  $z$  を半径  $L_{small}$  の球上に移動させる
 
$$y_i = z_i (L_{small}/L_z) \quad (i=1,2,\dots,D)$$
- (5) 一様乱数  $r(0 \leq r \leq 1)$  を作成する
- (6)  $y$  を球内に移動させる

$$x_i = y_i r^{1/k} \quad (i=1,2,\dots,D)$$

以上の方法を用い、2個体を生成する。

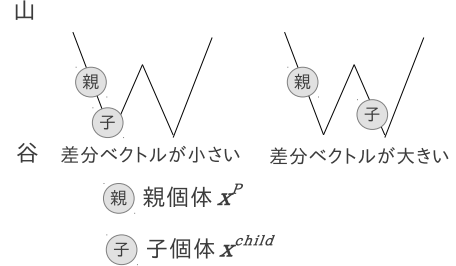


図2. 差分ベクトルの大きさによる置き換えの違い

### IV. 実験

#### A. 使用する関数

従来手法の GDE と提案手法の VGDE を複数の最適解をもつ表 I のテスト関数に適用し、性能比較を行う。

表 I  
テスト関数

関数名	次元数	最適解の数
Shubert 関数 ( $F_1$ )	2	18
Himmelblau 関数 ( $F_2$ )	2	4
Six-Hump Camel Back 関数 ( $F_3$ )	2	2
Branin RCOS 関数 ( $F_4$ )	2	3
3-peaks 関数 ( $F_5$ )	2, 5, 10	3

#### B. 実験設定

1) 使用したパラメータ: 実験で用いたパラメータを表 II に示す。表 II の個体数とは集団中の個体数であり、全ての関数において個体数を 100 とする。次元数は 3-peaks 関数のときのみ 2, 5, 10 の 3 つの次元について実験を行う。それ以外の関数は全て 2 次元での探索となる。最大世代数は 1 回の試行において進化させる世代数であり、最大世代数は 500 とした。また、山谷判定により得られた山度、谷度から交叉率  $C_r$  とスケールリングファクターの調整を行う。表 III のように、谷度 > 山度の個体については局所的な探索を行うために  $F$  を小さくし、 $C_r$  を大きくする。山度 > 谷度の個体については大域的な探索を行うために  $F$  を大きくし、 $C_r$  を小さくする。

#### 2) 評価方法:

- 収束速度 (Convergence Speed)  
各ステップごとに大域最適解 ( $Opt_i$ ) と、その解との距離が 0.1 以内である個体 ( $M_i$ ) の関数値 ( $f(Opt_i)$  と  $f(M_i)$ ) を比較する。全ての大域最

表 II  
実験で用いたパラメータ

パラメータ	内容	値
$NP$	個体数	100
$D$	次元数	2, 5, 10
$G_{max}$	最大世代数	500
$F$	スケールリングファクター	0.5
$C_r$	交叉率	0.9
$d_r$	置き換え半径	$1.0e^{-4}$

表 III  
調整したパラメータ

	$F$	$C_r$
谷	0.3	1.0
谷度>山度	0.3	0.9
谷度=山度	0.5	0.9
谷度<山度	0.8	0.9

適解について関数値の差が  $10^{-5}$  以下となる場合に解が収束しているとし、そのときの評価回数を収束速度とする。

・ 発見個数

終了ステップ時に大域最適解と大域最適解に最も近い個体との距離を求める。個体が最適解から一定の距離  $\delta$  内に存在すれば、解を発見したとみなす。ここでは、 $\delta = 10^{-3}$  とした。

C. 実験結果と考察

RNG による近傍関係から種分化を行った GDE と RNG による山谷判定から種分化を行った VGDE と VGDE に動的 crowding を導入した VGCDE の結果を、次の表 IV に示す。表の収束速度内の括弧内は 50 回の試行の内、収束の条件を満たした場合の回数である。

表 IV  
GDE, VGDE, VGCDE の性能比較

関数	手法	収束速度	発見個数
$F_1$	GDE	35808.00(50)	18.00
	VGDE	24480.00(45)	17.40
	VGCDE	19664.00(50)	18.00
$F_2$	GDE	14132.00(50)	4.00
	VGDE	6852.00(50)	4.00
	VGCDE	4280.00(50)	4.00
$F_3$	GDE	6188.00(50)	2.00
	VGDE	3680.00(50)	2.00
	VGCDE	1928.00(50)	2.00
$F_4$	GDE	8148.00(50)	3.00
	VGDE	4906.00(50)	3.00
	VGCDE	3860.00(50)	3.00
$F_5(D=2)$	GDE	5618.00(50)	3.00
	VGDE	2962.00(50)	3.00
	VGCDE	2004.00(50)	3.00
$F_5(D=5)$	GDE	15938.00(50)	3.00
	VGDE	7222.00(50)	3.00
	VGCDE	6512.00(50)	3.00
$F_5(D=10)$	GDE	35116.00(50)	3.00
	VGDE	16590.70(43)	2.78
	VGCDE	13720.00(50)	3.00

GDE に比べ VGDE は収束速度が全ての関数で速くなっている。VGDE のほうが GDE よりも種の数少なく、一般的に種の数が多い場合のほうが収束速度が速くなる性質があるため、VGDE のほうが収束速度が速くなったと考えられる。発見個数については、 $F_2 \sim F_4$  と  $F_5$  の 2 次元については VGDE は全て発見することができている。しかし、 $F_1$ 、 $F_5$  の 10 次元については全ての最適解を発見することはできなかった。 $F_1$  の最適解はペアとなる最適解が非常に近い位置にあるため、ペアとなる最適解が 1 つの谷に見えてしまい、全ての最適解を発見することができなかったと考えられる。 $F_5$  の高次元については、山谷判定により最適解の位置を谷として見えていなかったため発見することができなかったと考えられる。

VGCDE は、谷度の良い個体に対して crowding を行ったことで、多様性が保たれ、 $F_1 \sim F_5$  の関数においてすべての解を発見することができた。また差分ベクトルを小さくし、パラメータ調整によって谷度の良い個体の周辺を局所的に探索することにより、VGDE と比べさらに収束速度を速めることができたと考えられる。

V. おわりに

山谷判定に基づく種分化と動的 crowding を行う方法を提案し、安定して複数解が探索できることを示した。本研究では、個体の谷度、山度に応じて  $F$  と  $C_r$  の調整を行った。今後の課題として、パラメータ調整を行わずに全ての最適解を発見できるような手法を考察したい。また、さらに高次元の問題にも対応できるように改良する必要がある。

参考文献

- [1] Xiaodong Li, "Efficient differential evolution using speciation for multimodal function optimization", Proc.of GECCO2005, pp.873-880, 2005.
- [2] 木村孝作, 原章, 市村匠, 高濱徹行, "近傍グラフに基づく種分化を導入した Differential Evolution による複数解の探索", 第 26 回フレンジシステムシンポジウム (FSS2010), pp.566-571, 2010.
- [3] Rene Thomsen, "Multimodal Optimization Using Crowding-Based Differential Evolution", 2004 IEEE Congress on Evolutionary Computation, pp.1382-1389 Vol2, 2004.
- [4] 坂井節子, 高濱徹行, "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", 数理解析研究所講究録 第 1682 巻, pp.184-192, 2010.
- [5] 柴坂美祐喜, 原章, 市村匠, 高濱徹行, "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 D, Vol. J92-D, NO.7, pp.1003-1014, 2009.

問い合わせ先

〒 731-3194

広島市安佐南区大塚東三丁目 4 番 1 号

広島市立大学大学院情報科学研究科知能工学専攻  
有田 紘人