

## ピット近似関数を用いた局所解集合探索

肥塚 真由子<sup>†</sup> 渡辺 俊典<sup>†</sup> 古賀 久志<sup>†</sup>

評価コストが高く、かつ連続離散混在変数による必ずしもなめらかでない非線形関数の局所解集合の探索的手法を提案する。取り扱い可能な未知パラメータの個数は限定されるが、関数のタイプに限定されることなく利用できる利点を持ち、局所解の集合を探索する能力を持つため、結果的に大域的最適解を探索する能力が高い点などが特徴である。

### Nonlinear Local Optima Set Extraction by Pit Approximation Functions

MAYUKO KOEZUKA,<sup>†</sup> TOSHINORI WATANABE<sup>†</sup> and HISASHI KOGA<sup>†</sup>

In this paper we propose a search method for local optimum set of a nonlinear cost function with high computation cost, continuous and discrete variables, and non-smoothness. Although the scope of dimensionality is rather modest, it can be applied to a wide class of nonlinear functions. Also, the chance of global optima finding is improved by its set searching property.

#### 1. はじめに

多くの最適化手法では単一解のみを探索するため、大域最適解の発見に失敗した際に、限定的な局所最適解を1つ発見して終了してしまうという問題がある。そこで、大域的最適解を含む局所最適解を網羅的に探す局所最適解集合探索問題 (Local Optima Set Extractor Problem: LOSEP) を導入し、その求解によって、最適解探索能力を改良することを考える。広範囲の関数に対応するために解空間が連続・離散混在変数によって構成されている場合に対応でき、また、シミュレータなどを内部に含み1回の計算に長い時間を要する評価関数の呼び出しが必要な最適化問題にも対応できるアルゴリズムの開発を目指す。

#### 2. 局所最適解集合探索問題 (LOSEP)

評価関数の局所最適解集合の定義と対象とする評価関数の性質を最初に整理する。本論文では最小化のみを扱う。さらに、制約式がある場合にはその充足度に応じた罰金を評価式に加算することで、無制約化した問題を対象とする。

##### 2.1 関数の局所最適解集合

評価関数  $f: [0:1]^n \times (0,1)^m \rightarrow R^1$  が与えられたとする。ここで、 $X^* \in [0:1]^n \times (0,1)^m$  に対して、

ある半径  $\delta_i > 0$  の開球  $B_i(X^*; \delta_i)$  が存在し、 $B_i$  内で  $f(X)$  を最小化するとき、 $X^*$  を局所最適解と定める。 $X^*$  が  $\bar{B}_i(B_i$  の閉包) の境界点のときは局所最適解とはしない。平坦な評価関数などでは無数の局所最適解が存在する。このような場合の局所最適解の集合を求める問題を LOSEP とする。

##### 2.2 評価関数の性質

LOSEP の評価関数について、次の諸特性を想定する。

- (1)  $X$  は連続・離散混在変数ベクトル  
 $X$  が連続・離散混在変数によって構成される場合を考える。  
$$X = (x, w), x \in [0:1]^n, w \in (0,1)^m$$
ここに、連続変数は  $[0:1]$  間の実数、離散変数は  $(0,1)$  の2値整数とする。
- (2)  $f(X)$  の実行時間が大  
 $f(X)$  の1回の呼び出し時間が大きく、呼び出し回数削減が必要と想定する。
- (3)  $f(X)$  は非線形関数  
 $f(X)$  は不連続、あるいは、連続であっても微分不可能である場合も想定する。
- (4) 局所最適解が1個以上  
局所最適解が存在しない、あるいは複数存在する可能性を想定する。無限に存在する場合もあるとする。この場合には、有限個の探索で満足することとする。

<sup>†</sup> 電気通信大学大学院 情報システム学研究所

### 3. LOSEP の求解

#### 3.1 既存手法の適用可能性

最適化手法は、線形計画法や非線形計画法と呼ばれる手法を中心とした多くの手法から成り立っている。<sup>1)</sup> シンプレックス法 (Simplex Method : SM) に代表される線形計画法では、<sup>2)</sup> 評価関数は線形であり、高次元問題を効率的に解くことができる。本研究で対象としている非線形計画法では、非線形の評価関数を扱う。連続微分可能関数については勾配法 (Gradient Method : GM) が利用されるが、評価関数が多峰関数である場合、大域的最適解を発見できず、局所最適解に到達する可能性がある。一方、大域的最適解を探索する非線形計画法として、遺伝的アルゴリズム (Genetic Algorithm : GA) や Price の適応ランダム探索 (Controlled Random Search : CRS) などの確率的探索手法が存在する。<sup>3)</sup> GA は、離散ベクトル点群を、確率的に有望な離散ベクトル点の生成により、最適解へと収束させる手法である。CRS は、解空間に生成したベクトル点群を、確率的に発生させた点との相互比較により、最適解へと収束させる手法である。

#### 3.2 LOSEP への適用可能性

既存手法が LOSEP へ適用可能かどうかを見ていく。2.2 の評価関数の性質から以下の要請が出てくる。

- (1) 連続・離散変数混合評価関数に適用可
- (2) 評価関数  $f(X)$  の呼び出し回数が少ない
- (3) 平坦な関数などの場合でもアルゴリズムが終了すること
- (4) 複数の局所最適解を発見できること

これらの条件の下で SM, GM, GA, CRS が適用可能であるかどうかを表 1 に記した。既存手法が LOSEP に適用困難であることがわかる。

表 1 既存手法の LOSEP への適用可能性

	(1)	(2)	(3)	(4)
SM	×線形連続のみ	○少ない	○終了	×1つ
GM	×連続のみ	○少ない	×終了せず	×1つ
GA	○可能	×多い	×終了せず	×1つ
CRS	×連続のみ	×多い	×終了せず	×1つ

### 4. 提案手法 (LOSE)

上記の LOSEP を対象とすることが可能な新しい局所最適解集合探索法 (Local Optima Set Extractor : LOSE) を提案する。LOSE では、区切られた領域  $S_i$

ごとに評価関数  $f(X)$  の形状を推定することで探索を行なう。各領域  $S_i$  及びそこでの関数形状を同時に推定するために以下に示すピット近似関数を導入する。

#### 4.1 ピット近似関数

各探索領域  $S_i$  とそこでの関数形状を推定するのに用いるピット近似関数について説明する。ピット近似関数は次式のように表される。

$$g_j(X, C_j, h_j, \theta_j) = \sum_{i=0}^n \frac{|x_i - c_i|}{\tan \theta_j} + \sum_{i=0}^m \frac{|w_i - c_{i+n}|}{\tan \theta_j} + h_j$$

$j = 1, \dots, M$  とする。  $M$  はピット近似関数の総数であり探索に先立ってユーザが指定する。ベクトル  $C_j = \{c_0, c_1, \dots, c_{n+m}\}$  はピット近似関数の軸位置、  $h_j$  は頂点の高さを示す。  $\theta_j$  は頂角であるが、本研究では固定値 (ユーザ指定) を用いる。図 1 に示すように  $g_j(X)$  ( $j = 1, \dots, M$ ) は  $X$  の集合上に支配領域  $S_j$  を自然に導入する。すなわち、任意の点  $Y$  が  $S_j$  に属するか否かは、すべての  $k \neq j$  について  $g_j(Y) \leq g_k(Y)$  が成立しているか否かで判定できる。この性質を以下のアルゴリズムの中で活用する。

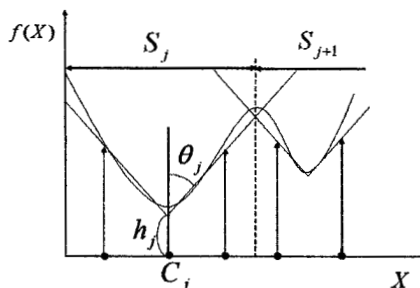


図 1 1次元評価関数でのピット近似関数

#### 4.2 LOSE アルゴリズム

図 2 にアルゴリズムのフローチャートを示す。

- (1) ピット近似関数の生成  
ピット近似関数の軸となる  $C_i$  をランダムに  $M$  個 (ユーザ指定) 生成する。ピット近似関数の頂点の高さ  $h_j$  は、平等に領域分割を行なうために、同一値とする。
- (2) 初期ベクトルの生成  
初期ベクトル  $X_i$  をランダムな  $[0 : 1]$  間の実数をもとに  $N$  個 (ユーザ指定) 生成し、同時にその評価値  $f(X_i)$  を求める。

$$X_i = \{x_0, \dots, x_n, w_0, \dots, w_m\}, f(X_i) \\ x_i, w_j \in [0 : 1]$$

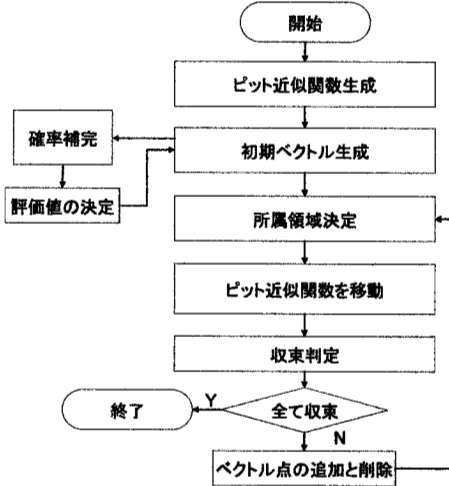


図2 LOSE フローチャート

$w_j$  は,  $w_j \in \{0, 1\}$  であり, 離散値  $0, 1$  のみが許され, 実数  $w_j \in [0 : 1]$  では未定義であり,  $f(X_i)$  の計算は不可能である. 対策として, 次に述べる確率補間法による内挿推定評価値を用いる.

(3) 確率補間法

実数要素  $w_j \in [0 : 1], (j = 0, \dots, m)$  での  $X_i$  における  $f(X_i)$  の値は, 実数  $w_j$  を整数  $w_j^* \in \{0, 1\}$  に置換した時の  $X_i^*$  での  $f(X_i^*)$  の値の  $w_j$  の位置を反映した線形補間値で代用する. 以下に計算法を示す.

$X = (x_0, \dots, x_n, w_0, \dots, w_m)$  とする.

**Step1** 以下を  $i = 1 \rightarrow T$  の間, 繰り返す

- (1) 範囲  $[0, 1]$  にある乱数  $r_i$  を生成
- (2)  $j = 0 \rightarrow m$  の間, 繰り返し,  $w_0, \dots, w_m$  を  $0$  または  $1$  に離散化する
  - a)  $w_j > r_i \Rightarrow w_j^* = 1$
  - b)  $w_j < r_i \Rightarrow w_j^* = 0$
- (3)  $X^* = (x_0, \dots, x_n, w_0^*, \dots, w_m^*)$  における評価値  $f(X^*)$  を求める

**Step2** 以下の式により  $f^i(X^*)$  を平均化し, 擬似的な評価値  $f(X)$  とする

$$f(X) = \frac{1}{T} \sum_{i=1}^T f^i(X^*)$$

(4) ベクトルの所属領域決定

全てのベクトル  $X_i$  に対して, 最小の値を持つビット近似関数  $g_b(X_i)$  を見出し, その支配領域  $S_b$  に  $X_i$  は属するとみなす.

$$g_b(X_i, C_b, h_b, \theta) = \min [g_j(X_i, C_j, h_j, \theta) | j = 1, \dots, M]$$

(5) ビット近似関数の移動

所属が決定されたベクトルを用いて, 各ビット近似関数を移動する. 以下の関数が最小となる軸ベクトル位置  $C_j$ , 頂点  $h_j$  にビット近似関数を移動させる. この計算には  $(C_j, h_j)$  の探索が必要となる.  $h$  は一般的に非線形関数となるが, 新たな  $f(X)$  の計算は不要であることに着目し, 前述の CRS をこの探索に用いる.

$$h(\{X_i, C_j, h_j, \theta\}) = \sum |f(X_i) - g_j(X_i, C_j, h_j, \theta)|$$

(6) 収束判定

局所最適解だけでなく, 不要なビット近似関数の消去, 平坦な関数領域の判定も行なう.

- 局所最適解への収束判定

ビット近似関数  $g_j$  の所持するベクトルの中で最小の評価値を持つベクトルを  $X_b$  とする. このとき, 次式が成立すれば, ビット近似関数の軸ベクトル  $C_i$  を局所最適解とする.

$$|f(C_j) - f(X_b)| < \varepsilon \wedge |C_j - X_b| < \varepsilon$$

- 平坦な領域の判定

ビット近似関数の所持する全てのベクトル  $X_i$  において, 以下の条件を満たすとき, そのビット近似関数の持つ領域は平坦であると判定する.

$$|f(C_j) - f(X_i)| < \varepsilon$$

- 不要ビット近似関数の消去

ベクトルを1つも所持しないビット近似関数  $g_k$  は, 不要と判定する.  $g_k$  は領域分割に貢献しない関数である.

$$k \neq \min [j | g_j(X_i), j = 1, \dots, M]$$

以上3つの収束判定をパスしたビット近似関数は, 収束したとみなし, その後移動させない.

(7) ベクトルの追加と削除

各領域ごとに, 有望なベクトル  $X_{new}$  を追加し, 最悪評価値を持つベクトル  $X_w$  を消去する. これにより, 局所解の方向へのビット近似関数の移動が可能となる.

## 5. 実験

### 5.1 実験計画

提案手法 (LOSE) を既存手法 (GA, CRS) と比較し, その特徴を確認する. 各実験では試行を10回行い, 結果はその平均を示す.

実験 1 既存手法では局所解に陥りやすい、以下の多峰連続関数を用いて、局所解の発見率を検証する。ピット近似関数の個数  $M = 5, 10, 15, 20$ , 初期ベクトル点の個数  $N = M \times (n + m) \times 10$  とする。

$$f_1(X) = \min \left\{ \begin{array}{l} 0.1 + \sum_{i=0}^4 (x_i - 0.1)^2 \\ 5 \cdot \sum_{i=0}^4 (x_i - 0.9)^2 \end{array} \right\}$$

この評価関数における真の局所最適解  $(0.1, \dots, 0.1)$  を A と表記し、真の大域最適解である  $(0.9, \dots, 0.9)$  を B とする。解が A または B との距離  $\varepsilon < 0.3$  のとき局所解を発見したものとす。

実験 2 以下の多峰連続・離散関数に対して、局所解の発見可能性を検証する。LOSE における、次の評価関数の局所解発見率を真の局所最適解からの距離が  $\varepsilon$  以下のとき局所解を発見したとみなして計上する。 $\varepsilon$  を 0.3, 0.4, 0.5 と変動させる。実験 1 と同様に  $M = 5, 10, 15, 20$ ,  $N = M \times (n + m) \times 10$  とする。

$$f_2(X) = \begin{cases} \sum_{i=0}^4 (x_i)^2 & (w_0 = 0) \\ \sum_{i=0}^4 (x_i - 1.0)^2 & (w_0 = 1) \end{cases}$$

この評価関数における局所最適解  $(0.0, \dots, 0)$  を A とおき、 $(1.0, \dots, 1)$  を B とおく。

## 5.2 実験の結果と考察

実験 1 の結果を表 2 に示す。CRS, GA は共に一方の局所解のみを発見しているのに対して、LOSE は高い確率で A, B 両方の局所解を発見した。

表 2 局所解の平均発見率 (実験 1)

		呼び出し回数	A 発見率	B 発見率
CRS		6050	100%	0%
GA		264600	0%	100%
LOSE	M = 5	469	70%	80%
	M = 10	729	70%	80%
	M = 15	1083	80%	100%
	M = 20	1424	100%	80%

実験 2 の結果を表 3 に示す。ピット近似関数の個数 M を増やすことで、局所解の発見率が上昇した。M が多いとき、連続・離散混合変数を持つ多峰関数の局所解近傍を発見していることが示された。

表 3 局所解の平均発見率 (実験 2)

$\varepsilon$	0.3		0.4		0.5	
	A	B	A	B	A	B
M = 5	0%	0%	0%	0%	0%	20%
M = 10	10%	0%	20%	20%	40%	40%
M = 15	10%	10%	30%	10%	40%	10%
M = 20	50%	60%	90%	100%	100%	100%

## 6. 画像認識への応用

画像内の円形オブジェクトを認識する実験を行なった。未知パラメータは  $X = (x, y, r)$  であり、 $(x, y)$  は円の中心、 $r$  は半径である。評価関数値は入力画像とサンプル画像との階調値の差の 2 乗和とする。ピット近似関数の個数  $M = 7$  とし、初期ベクトル点の数  $N = M \times (n + m) \times 50$  とした。下図に結果を示す。認識結果において左下の円は、2 重に認識されている。

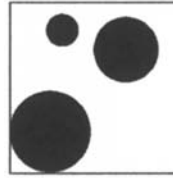


図 3 入力画像

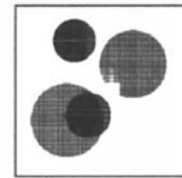


図 4 認識結果 (4 個の円)

## 7. まとめ

ピット近似関数を用いて探索空間の領域分割と評価関数の形状推定とを同時に実現することで、非線形関数の複数の局所最適解を探索する手法 LOSE を提案した。実験によって局所最適解集合の探索能力と、評価関数の呼び出し回数の削減能力を確認できた。確率補間法を用いることで連続・離散混合変数を持つ評価関数への適用も可能とした。今後は、さらに複雑な評価関数への対応が可能となるように改良を行なう。

## 参考文献

- 1) 今野浩, 山下浩: 非線形計画法, 日科技連 (2003).
- 2) 今野浩: 線形計画法, 日科技連 (2003).
- 3) W.L.Price et al.: A controlled random search procedure for global optimisation, The Computer Journal, Vol.20, No.4, pp.367-370 (1977)