

分類型と近似型のハイブリッドサロゲートに基づく粒子群最適化法

宮原 悠司[†] 園田 拓海[†] 西原 慧[†] 中田 雅也[†]

横浜国立大学[†]

1 はじめに

実社会の複雑な最適化問題において現実的な計算時間で解く最適化手法として、これまで様々なメタヒューリスティクスが提案されている。その一つに、解を粒子に見立て、群内の粒子の相互作用により最適化を行う粒子群最適化法 (Particle Swarm Optimization: PSO)¹⁾ が存在する。PSOやその改良手法は、多くの実問題においてその有効性が示されている^{2, 3)}。

しかしながら、航空機の翼形状設計問題⁴⁾のように解評価にCFDシミュレーションを用いる場合など、実問題には1回の解評価に数時間、数日といった長い時間を要する高計算コストな問題が存在する。この問題領域において通常のメタヒューリスティクスを適用し、探索において数万回の解評価回数を消費することは現実的でない。そこで、アルゴリズムの設定を自動調整する自己適応型進化アルゴリズム⁵⁾や、機械学習によって構築されるサロゲートと呼ばれるモデルを利用して解探索を効率的に進めるサロゲート型進化アルゴリズム (SAEA)⁶⁾ が有力とされている⁷⁾。このSAEAには、大きく分けて、Direct Fitness Replacement (DFR)とIndirect Fitness Replacement (IFR)と呼ばれる2種類のアプローチが存在する⁸⁾。

DFRは、評価値と決定変数の組から目的関数の近似関数モデルを構築し、モデルが出力する近似的な評価値を用いて解探索を進める方法である。近似関数モデル上で良好とされる解を実際の目的関数で解評価することにより、効率化を図ることができる。本稿では、これを近似型サロゲートと呼ぶ。したがって、DFRアプローチの探索性能は近似関数モデルの精度に大きく

依存する傾向がある。すなわち、低精度な近似関数モデルでは誤った領域に解探索を進めるリスクがある。また、高計算コストな最適化問題においては、近似関数モデルの高精度化に十分な数のデータを用意することができず、その近似関数モデルの精度を十分に高めることが困難となる。特にこの傾向は、高次元かつ複雑な目的関数の形状をもつ問題で顕著になる。

一方で、IFRは目的関数を直接近似せず解の有望性を決定することで、解探索を進める方法である。IFRの1つに、優良解の存在領域を学習することで、新たな解が優良解か否かの分類を行う分類型サロゲートを用いた手法がある。本稿ではIFRとして分類型サロゲートに着目する。この手法は、複雑な目的関数空間を持つ多目的最適化問題において性能向上に成功しており、探索効率が改善できることが示されている。しかしながら、分類型サロゲートは優良解の存在領域のみを学習するため、DFRのように近似関数モデル上での仮想的な解探索や解同士の比較ができない。

分類型サロゲートには、突然変異によって解が不連続に移動(生成)される非力学的なメタヒューリスティクスを用いる場合が多い。この挙動は学習した優良解領域に解を生成する点で親和性が高い。一方で、力学系に従い解を連続的に移動させるPSOにおいては、近似型サロゲートと親和性が高いことから、近年では盛んに研究されている^{9, 10)}。しかしながら、高次元かつ目的関数の形状が複雑な問題に対しては、前述した近似関数モデルの信頼性が低下する問題から、最適化性能が低下する可能性がある。また、分類型サロゲートに基づくPSOについては、制約付き最適化問題において実行可能領域をSupport Vector Machine (SVM)¹¹⁾で分類する手法¹²⁾が研究されている。しかし、著者らの調査の限りでは、優良解を特定する目的で分類型サロゲートを用いた研究はない。

そこで本稿では、近似型サロゲートに基づく

Hybrid surrogate-assisted particle swarm optimization based on approximation and classification models

[†] Yushi Miyahara(miyahara-yushi-nz@ynu.jp)

[†] Takumi Sonoda(sonoda-takumi-gp@ynu.jp)

[†] Kei Nishihara(nishihara-kei-jv@ynu.jp)

[†] Masaya Nakata(nakata-masaya-tb@ynu.ac.jp)

Yokohama National University (†)

PSOに分類型サロゲートを協調的に組み込むことで、高次元かつ複雑な最適化問題をより効率的に解くためのPSOアルゴリズムを提案する。近似型・分類型サロゲートをハイブリッド化する理由は、サロゲートの精度低下リスクを緩和するためである。精度が高い近似関数モデルが構築できれば、解同士の比較によって特定の優良解を詳細に絞り込める。この精度が低い場合、分類型サロゲートによって、大まかな優良解を推定する。このとき、未知入力に対するサロゲートの精度が検証できないため、近似・分類型サロゲートのどちらを使うのが問題になる。この問題を解決するために、本稿ではPSOのフレームワークを活かす点が特徴である。具体的には、近似型サロゲートより求めた解を $gbest$ に設定し、粒子ごとの $pbest$ を分類型サロゲートより推測する。 $gbest$ と $pbest$ の独立性を活かせば、近似・分類型サロゲートのハイブリッド化による相乗効果に加え、サロゲートの精度低下リスクも管理できる。本稿の貢献は、両サロゲートの欠点を補いあうハイブリッド化がPSOアルゴリズムで実現できることを示し、(優良解を特定する)分類型サロゲートとPSOの融合方法を提示することにある。なお、本稿では近似・分類型サロゲートにそれぞれ動径基底関数補間法(RBF補間)¹³⁾とSVMを用いる。

本稿の構成は次の通りである。第2章で提案手法の要素技術を紹介し、第3章で提案手法のメカニズムの説明を行う。第4章では、IEEE CEC2013 real-parameter single-objective benchmark function suite¹⁴⁾を用いて、提案手法を通常のPSOと近似型サロゲートに基づくPSOであるOUPS¹⁵⁾と比較する。そして、問題次元数が50, 100といった高次元かつ目的関数が複雑な形状である問題において提案手法が性能を改善することを示す。なお、紙面の制限上、OUPSの概要は第4章で簡単に述べる。第5章では提案手法の考察を行い、第6章で結論を述べる。なお、以下では「優良解が存在すると判断された領域」を単に「優良領域」という。

2 要素技術

本章は、PSOに加え提案手法のサロゲートとして用いるRBF補間とSVMを説明する。

2.1 粒子群最適化法 (PSO)

本稿では、単一目的実数値連続最小化問題を対象とする。同問題では、決定変数を

$\mathbf{x} \in \mathbb{R}^D$ とする目的関数 $f(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$ に対し、 $\mathbf{x}^* = \arg \min_{\mathbf{x}} f(\mathbf{x})$ で与えられる D 次元の大域的最適解 $\mathbf{x}^* = [x_1, x_2, \dots, x_D]$ 、あるいはその近似解を求めることが目的となる。

粒子群最適化法 (Particle Swarm Optimization: PSO)¹⁾では、 N 個の粒子 $\mathbf{x}_i \in \mathcal{P}$ ($i = 1, 2, \dots, N$)は解ベクトル $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$ として表現される。サイズ N の粒子群 \mathcal{P} に属する粒子は速度ベクトル \mathbf{v}_i と位置ベクトル \mathbf{x}_i の情報をもち、これらを更新しながら問題を探索する。ここで、サイズ N の群 \mathcal{V} を \mathbf{v}_i ($i = 1, 2, \dots, N$)の集合とする。具体的には、ある世代 t において、式(1)で表される \mathbf{v}_i^t を用いて、式(2)に示すように \mathbf{x}_i^t を更新し、これを $t+1$ 世代目における粒子の位置 \mathbf{x}_i^{t+1} とする。

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1r_1(\mathbf{pbest}_i^t - \mathbf{x}_i^t) + c_2r_2(\mathbf{gbest}^t - \mathbf{x}_i^t), \quad (1)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}. \quad (2)$$

ここで、 $r_1, r_2 \in [0, 1]$ は一様分布からランダムに選ばれる乱数であり、PSOはハイパーパラメータとして慣性係数 $w \in \mathbb{R}$ と加速度係数 $c_1, c_2 \in \mathbb{R}$ を持つ。なお、 \mathbf{pbest}_i^t は i 番目の粒子 \mathbf{x}_i^t が過去に探索した点のうち評価値が最良となる点を、 \mathbf{gbest}^t は群 \mathcal{P} の全ての粒子の過去の探索点のうち評価値が最良となる点をそれぞれ表す。

近年では、constriction factorを導入したPSOの改良手法¹⁶⁾が広く用いられている^{9, 10)}。同手法では、先の式(1)は次に示す式(3)のように書き換えられる。

$$\mathbf{v}_i^{t+1} = \chi(\mathbf{v}_i^t + c_1r_1(\mathbf{pbest}_i^t - \mathbf{x}_i^t) + c_2r_2(\mathbf{gbest}^t - \mathbf{x}_i^t)), \quad (3)$$

$$\chi = \left| \frac{2k}{2 - \phi - \sqrt{\phi^2 - 4\phi}} \right|. \quad (4)$$

上式で新たに導入されるハイパーパラメータは ϕ と k であり、 $\phi = c_1 + c_2$ である。一般に、 $\phi > 4$ を満たすように c_1 と c_2 が設定される。本稿ではこの設定を用いる。

2.2 動径基底関数補間 (RBF補間)

動径基底関数 (Radial Basis Function: RBF)とは、その値が原点からの距離のみに依存する実数値関数のことを指す。RBF補間では、このRBFを用いることで、任意の次元数において散布する点列をノンパラメトリックに補間する。した

がって、複数の粒子とそれらの評価値から非線形な目的関数の近似が可能であり、提案手法ではこれを近似型サロゲートとして組み込む。

D 次元の点 $\mathbf{x} \in \mathbb{R}^D$ と関数 $f(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$ を考える。 n 組の点列 $\{\mathbf{x}_1, f(\mathbf{x}_1)\}, \{\mathbf{x}_2, f(\mathbf{x}_2)\}, \dots, \{\mathbf{x}_n, f(\mathbf{x}_n)\}$ が与えられたとき、RBFによる補間は次のように表現される。

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^n \lambda_i \Phi_i(\|\mathbf{x} - \mathbf{x}_i\|). \quad (5)$$

ここで、 $\Phi_i(\cdot)$ はRBFを、 $\|\cdot\|$ はユークリッド距離を示す。本稿ではRBFを $\Phi_i(r) = r^3$ とする。 λ_i はRBFの重み係数 $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_n]^T$ の i 番目の要素であり、関数が点列を通る条件 $\mathbf{F} = \Phi \boldsymbol{\lambda}$ から導かれる式(6)を解くことによって求められる。

$$\boldsymbol{\lambda} = \Phi^{-1} \mathbf{F}. \quad (6)$$

なお、 $\mathbf{F} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n)]^T$ 、 Φ は $\Phi_{ij} = \Phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ を要素とする $n \times n$ 対称行列である。

2.3 Support Vector Machine (SVM)

SVMはクラス分類を行う学習器の一つである。具体的には、クラスを分類する境界線に最も近いデータ群(サポートベクトル)と、境界線とのマージンの総和を最大化するような境界線を学習する。SVMを線形分離不可能な問題に適用する方法として、カーネル関数を用いた非線形変換によりデータを高次元特徴空間に写像し、その空間上で線形に分離を行う方法がある。提案手法では線形分離不可能なデータを扱うため、カーネル関数として次で表されるRBFカーネルを用いる。

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|) \quad (7)$$

ここで、 $\gamma \in \mathbb{R}$ は境界線の複雑さを決定するハイパーパラメータであり、 $\gamma > 0$ である。 γ が大きい値をとるほど教師データに適合したモデルが構築されるが、汎化性能は低下するため過学習を引き起こしやすくなる。そこで、過学習を防ぐために、学習誤差値を導出する式に正則化項を設けることがあり、その係数を正則化係数 C とする。この C は、モデルの誤分類に対する許容度を意味する。

3 提案手法

本章では、提案手法の全体の流れを説明したあと、具体的なメカニズムについて述べる。

3.1 全体の流れ

始めに、初期個体群の多様性を高めるためにLatin Hypercube Sampling (LHS)¹⁷⁾によって、次元数 $D+1$ 個の初期解を生成する。そして、これらの初期解を実際の目的関数を用いて評価し、上位 N 個を初期個体群 \mathcal{P} に属する粒子 $\mathbf{x}_i \in \mathcal{P}$ として設定する。 $D+1 < N$ であるときは、初期個体群の個数が N になるまで決定変数の定義域 $[lb, ub]^D$ から一様分布によって解を生成し、初期個体群に追加する。また、速度ベクトル \mathbf{v}_i は $[v_{min}, v_{max}]^D$ で定義される範囲から一様分布によってランダムに初期化する。提案手法では、決定変数の定義域 $[lb, ub]^D$ に対し $v_{min} = -(ub - lb)/2, v_{max} = (ub - lb)/2$ とする。なお、探索過程において、粒子の位置ベクトルが定義域外になった場合や、速度ベクトルが $[v_{min}, v_{max}]^D$ を超えた場合には、定義域を超えた次元の決定変数をその端点に設定する。

初期化に続くプロセスでは、分類型と近似型の2つのサロゲートを用いて探索終了条件まで解更新を行う。まず、優良領域を判定するために、初期解からデータセットを作成しSVMで学習を行う。続いて、同じく初期解から構成されるデータセットによりRBF補間を用いて近似関数モデルを構築する。これらのモデルは、各世代における解更新プロセスの最初に、過去の探索履歴からなるデータセットを用いて更新される。次に、 \mathbf{gbest} を改良するために、RBF補間による近似関数モデル上で \mathbf{gbest} の近傍を探索する。この探索によって発見された解 \mathbf{gbest}' を実際の目的関数で評価し、評価値が改善された場合のみ、これを新たな \mathbf{gbest} とする。続いて、粒子 \mathbf{x}_i の優良領域方向への探索を促進するために、式(3)における \mathbf{pbest}_i を優良領域内の解と置き換える。具体的には、 \mathbf{pbest}_i をSVMに入力し、優良解か否かを判定する。優良解と判断されなかった場合のみ、 \mathbf{pbest}_i を既に発見された解のうち優良領域内に存在する解と置き換える。以上の操作後、式(3), (2)にしたがって、粒子 \mathbf{x}_i の速度ベクトル \mathbf{v}_i と位置ベクトル \mathbf{x}_i を更新する。

3.2 メカニズム

ここでは、提案手法におけるサロゲートの定義と利用法について説明する。最初に、本研究の特徴である、SVMによる分類型サロゲートの構築方法を述べたのちに、これを用いた \mathbf{pbest}_i

の更新メカニズムを示す。次に、RBF補間による近似型サロゲートを用いた $gbest$ 更新メカニズムを説明する。

3.2.1 SVMによる分類型サロゲートの構築

まず、分類型サロゲートのデータセットを構築するために、世代 t まで評価した決定変数値(粒子) \mathbf{x} とその評価値 $f(\mathbf{x})$ を全て記憶するアーカイブ A を設ける。そして、計算時間の削減も視野に入れ、少ないデータ数で効率的に学習を行うために、アーカイブ A 内の(評価値が)上位 M 個の粒子を用いてデータセット D を作成する。次に、SVMによって分類型サロゲートを構築する。ここで、 $M \in \mathbb{N}$ は新たに導入するハイパーパラメータであり、 $|D| = M$ である。提案手法で用いるSVMは、2値クラス $\{0, 1\}$ に分類し、1は優良、0は非優良である。具体的には、 \mathbf{x}_k ($k = 1, \dots, M$)のラベル付けは、 $pbest_i^t$ の評価値 $f(pbest_i^t)$ の中央値をしきい値 θ^t として、式(8)より決定する。

$$L_k^t = \begin{cases} 1 & \text{if } f(\mathbf{x}_k) < \theta^t, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

ただし、

$$\theta^t = \text{Median}(\{f(pbest_i^t) | \forall i = 1, 2, \dots, N\}) \quad (9)$$

である。

上式のように設定することで、およそ半数の粒子に対して $pbest_i^t$ の置き換えの発生が期待され、これにより優良解の存在領域へ誘導する圧力を高めることが可能になる。例えば、優良解のしきい値 θ^t を $\theta^t > \max_i f(pbest_i^t)$ と設定した場合、 $pbest_i^t$ のほとんどが優良解と判断され、置き換えが発生する頻度が小さくなるだけでなく、優良領域の信頼性が低下する。一方、しきい値を小さすぎる値に設定した場合、データセット内に優良解とラベル付けされる粒子が存在しない可能性が生じる。以上の理由から、本稿では解の置き換えをバランスよく発生させるため、提案手法ではSVMのラベル付けを式(8)のように行う。

3.2.2 分類型サロゲートに基づく $pbest_i$ の置換

前項で述べたように、 $pbest_i^t$ がSVMによって非優良解(ラベル:0)と分類された場合、これをデータセット D 内の優良解(ラベル:1)を用いて優良領域内の解と置き換える。一連のアルゴリズムを、疑似関数 $\text{Replace-pbest}(pbest_i^t, D)$ としてAlgorithm 1に示す。まず、 D 内の優良解集合 S に

Algorithm 1 $\text{Replace-pbest}(pbest_i^t, D)$

```

1:  $S = \{\mathbf{y} | \mathbf{y} \in D, \text{SVM}(\mathbf{y}) == 1\}$ 
2:  $\mathbf{x}_i^* = \arg \min_{\mathbf{y} \in S} \|\mathbf{y} - pbest_i^t\|$ 
3: for  $l = 1$  to  $T$  do
4:    $\mathbf{z} = \sigma(\mathcal{N}_1(0, 1), \mathcal{N}_2(0, 1), \dots, \mathcal{N}_D(0, 1))$ 
5:    $\mathbf{x}'_i = \mathbf{x}_i^* + \mathbf{z}$ 
6:    $\text{predict} = \text{SVM}(\mathbf{x}'_i)$ 
7:   if  $\text{predict} == 1$  then
8:     if  $\|\mathbf{x}'_i - pbest_i^t\| < \|\mathbf{x}_i^* - pbest_i^t\|$  then
9:        $\mathbf{x}_i^* = \mathbf{x}'_i$ 
10:   $pbest_i^t = \mathbf{x}_i^*$ 

```

属する \mathbf{y} のうち、 $pbest_i^t$ とのユークリッド距離 $\|\mathbf{y} - pbest_i^t\|$ が最も小さい粒子を初期候補粒子 \mathbf{x}_i^* とする。この初期候補粒子に対してGaussian mutation¹⁸⁾を行い摂動 \mathbf{z} を与えることで新たな解 \mathbf{x}'_i を生成する。なお、Algorithm 1のGaussian mutationにおいて、 $\mathcal{N}_j(0, 1)$, ($j = 1, \dots, D$)は D 個の独立な標準正規分布からサンプリングされる乱数であり、 σ は分散を表す。次に、 \mathbf{x}'_i をSVMに入力した結果、優良解と分類され、かつ $pbest_i^t$ とのユークリッド距離が小さくなれば($\|\mathbf{x}'_i - pbest_i^t\| < \|\mathbf{x}_i^* - pbest_i^t\|$)、 \mathbf{x}_i^* を現在の \mathbf{x}'_i と置き換える。ここで $pbest_i^t$ とのユークリッド距離が比較的小さい粒子のみを採用する理由は、置き換え先の粒子が $pbest_i^t$ と大きく離れている場合を避けることで、粒子の速度ベクトルの大きさが急激に増加することを防ぎ、過剰な速度増加による収束速度の低下を防ぐためである。提案手法では、Gaussian mutationからユークリッド距離比較までの一連の操作を、ハイパーパラメータ T 回に渡って繰り返す。最後に、 T 回時点での \mathbf{x}_i^* で $pbest_i^t$ を置き換えて本項のプロセスを終了する。

3.2.3 近似型サロゲートに基づく $gbest$ の置換

提案手法では、過去の全ての探索点の情報、すなわちアーカイブ A をデータセットとし、RBF補間による目的関数の近似関数モデルを構築する。ここでの目的は、膨大な解評価回数の追加なしに、問題と探索状況により相応しい粒子に $gbest$ を更新することである。具体的には、RBF補間より近似関数モデルを構築したあと、同モデル内で $gbest$ の近傍を探索し、得られた最適解を $gbest'$ とする。なお、決定変数の定義域 $[lb, ub]^D$ に対し、 $\xi = 0.1(ub - lb)$ のもと、探索範囲は $(gbest \pm (\xi/2)^D) \cap [lb, ub]^D$ とする。高次元かつ複雑な問題では近似関数モデルの精度が低下することを考えると、近似関数モデル上で探

Algorithm 2 提案手法

```
1:  $t = 0$ 
2: Initialize  $\mathcal{P} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 
3: Initialize  $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$ 
4: Evaluate  $\forall \mathbf{x}_i \in \mathcal{P}$ 
5:  $\mathcal{A} \leftarrow \mathcal{P}$ 
6: while termination criteria are not met do
7:    $t = t + 1$ 
8:    $\mathcal{D} \leftarrow \text{GenerateDataset}(\mathcal{A})$ 
9:    $\text{SVM} \leftarrow \text{GenerateSVM}(\mathcal{D})$ 
10:   $\text{RBF} \leftarrow \text{GenerateRBF}(\mathcal{A})$ 
11:   $\mathbf{gbest}^t \leftarrow \text{LocalSearch}(\text{RBF}, \mathbf{gbest}^t)$ 
12:  for  $i = 1$  to  $N$  do
13:    if  $\text{SVM}(\mathbf{pbest}_i^t) == 0$  then
14:       $\mathbf{pbest}_i^t \leftarrow \text{Replace-pbest}(\mathbf{pbest}_i^t, \mathcal{D})$ 
15:       $\mathbf{v}_i^{t+1} \leftarrow \text{Eq.}(3)$ 
16:       $\mathbf{x}_i^{t+1} \leftarrow \text{Eq.}(2)$ 
17:      Evaluate  $\mathbf{x}_i^{t+1}$ 
18:       $\mathbf{pbest}_i^{t+1} \leftarrow \text{Update-pbest}(\mathbf{pbest}_i^t, \mathbf{x}_i^{t+1})$ 
19:      Add  $\mathbf{x}_i^{t+1}$  to  $\mathcal{A}$ 
20:   $\mathbf{gbest}^{t+1} \leftarrow \text{Update}(\mathbf{gbest}^t, \mathcal{P})$ 
```

探索空間全体における探索をしても、真の評価値と大きく異なった解を導出する結果となる可能性が高い。そこで提案手法は、実際の解評価で得られた最適解 \mathbf{gbest} の近傍を探索することにより、探索空間を狭めつつ効率的に \mathbf{gbest} の改善する効果を狙う。最後に、実際の目的関数で \mathbf{gbest}' の解評価を行い、評価値を改善した場合は式(10)のように、現在の \mathbf{gbest} を \mathbf{gbest}' に置き換える。一連の \mathbf{gbest} の置換に関する操作で行われる実際の解評価回数は世代につき1回であるため、膨大な解評価回数を消費することは無い。また、近似関数モデル上で発見された最適解は現在の最良解の評価値を改善する可能性が高いため、探索性能が向上することが期待できる。この操作を導入した効果については第5章で考察を行う。

$$\mathbf{gbest} = \begin{cases} \mathbf{gbest}' & \text{if } f(\mathbf{gbest}') < f(\mathbf{gbest}), \\ \mathbf{gbest} & \text{otherwise.} \end{cases} \quad (10)$$

本章のまとめとして、Algorithm 2に提案手法の疑似コードを示す。なお、近似型サロゲートにおける \mathbf{gbest} の近傍探索は、疑似関数 $\text{LocalSearch}(\text{RBF}, \mathbf{gbest}^t)$ と表す。

4 実験

本章では、ベンチマーク問題を用いて提案法、通常のPSOとOUPSの性能を比較する。

4.1 実験設定

4.1.1 ベンチマーク問題

IEEE CEC2013 real-parameter single-objective benchmark function suite (CEC 2013)¹⁴⁾で定義される28個の制約なし単一目的実数値連続最適化問題ベンチマーク関数F1, ..., F28を用いる。これらの関数は、単峰性関数F1 ~ F5、多峰性関数F6 ~ F20およびF1 ~ F20から選択した関数を合成した関数F21 ~ F28に分類される。問題次元数は $D = \{50, 100\}$ とする。また、コンペティションの規定に倣い、 \mathbf{x} の定義域は $\mathbf{x} \in [-100, 100]^D$ とする¹⁴⁾。

4.1.2 比較手法と評価方法

近似型サロゲートを用いるOUPSでは、一つの粒子 \mathbf{x}_i^t に対し r 個の速度ベクトル $\mathbf{v}_{i,1}^t, \mathbf{v}_{i,2}^t, \dots, \mathbf{v}_{i,r}^t$ を仮生成し、これらによって得られる位置ベクトル $\mathbf{x}_{i,1}^t, \mathbf{x}_{i,2}^t, \dots, \mathbf{x}_{i,r}^t$ を仮更新する。次に、RBF補間によって得られる近似関数モデルでこれらの仮位置ベクトルを評価し、最良の評価値を持つものを次世代の粒子 \mathbf{x}_i^{t+1} とする。

PSOにおけるハイパーパラメータ設定は $N = 20, c_1 = c_2 = 2.05, k = 0.729$ であり、PSOをベースとする提案手法でも関連するハイパーパラメータは同じ値を採用する。提案手法のその他のハイパーパラメータは、 $M = 5N, \gamma = 1/D, C = D, T = 2000, \sigma = 0.01$ である。なお、OUPSにおける各ハイパーパラメータは原著¹⁵⁾に従うが、公平を期すため $N = 20$ とし、近似関数モデルの基底関数も提案手法と同じく $\Phi_i(r) = r^3$ とした。

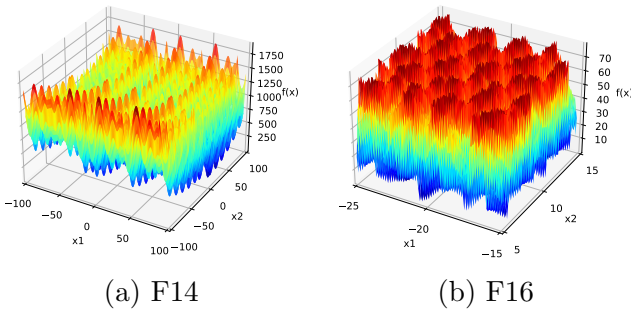
サロゲートを用いた手法の実験で広く用いられる設定を参考に、最大解評価回数を1,000とする^{10, 19)}。実験結果は、異なる乱数シードを用いた独立した10試行で得られた性能(最良解の中央値)で比較する。また、性能比較は解評価回数500, 1,000回で行うことで、収束速度も併せて確認する。加えて、統計的有意差を確認するために有意水準0.05のもとでFriedman検定を行う。Friedman検定が有意差を検出した場合は、事後検定としてWilcoxonの符号順位検定を行った後、Holm法で調整する。

4.2 実験結果

Table 1に $D = 50, 100$ 、評価回数500, 1,000回における最良解の中央値を示す。最大解評価回数(1,000回)時点においては、 $D = 50, 100$ をまとめて、56個中24個の実験ケースで提案手法が最も

Table 1 PSO, OUPSおよび提案手法の性能比較 (10試行の中央値)

次元数 評価回数	$D = 50$						$D = 100$					
	500回			1,000回			500回			1,000回		
	PSO	OUPS	proposal	PSO	OUPS	proposal	PSO	OUPS	proposal	PSO	OUPS	proposal
F1	6.34E+04	9.72E-01	8.58E+00	5.90E+04	3.12E-02	1.47E-06	2.05E+05	1.98E+03	8.59E+03	1.95E+05	8.15E-01	2.09E-01
F2	6.26E+08	2.09E+08	2.69E+08	5.37E+08	7.05E+07	1.09E+08	4.36E+09	1.79E+09	1.96E+09	3.87E+09	4.68E+08	4.91E+08
F3	2.02E+12	3.83E+11	4.41E+11	9.81E+11	1.66E+11	1.92E+11	2.01E+20	8.65E+18	1.71E+18	8.99E+19	5.78E+16	4.60E+16
F4	2.32E+05	1.80E+05	2.91E+05	2.07E+05	1.89E+05	2.85E+05	4.52E+05	3.98E+05	4.67E+05	4.40E+05	3.56E+05	4.58E+05
F5	4.02E+04	8.51E+03	1.77E+04	3.64E+04	3.74E+03	5.66E+03	1.59E+05	5.25E+04	7.76E+04	1.49E+05	1.96E+04	2.64E+04
F6	3.92E+03	3.63E+02	4.70E+02	3.26E+03	1.07E+02	1.09E+02	4.24E+04	3.93E+03	6.68E+03	3.80E+04	7.93E+02	9.77E+02
F7	1.27E+03	5.95E+02	7.38E+02	9.76E+02	3.30E+02	4.00E+02	4.19E+06	3.15E+06	2.95E+06	3.64E+06	3.85E+05	3.67E+05
F8	2.14E+01	2.14E+01	2.14E+01	2.13E+01	2.14E+01	2.14E+01	2.15E+01	2.15E+01	2.15E+01	2.15E+01	2.15E+01	2.15E+01
F9	6.89E+01	5.53E+01	5.62E+01	6.44E+01	3.47E+01	4.60E+01	1.49E+02	1.41E+02	1.50E+02	1.44E+02	1.04E+02	1.35E+02
F10	7.66E+03	6.18E+02	1.28E+03	6.64E+03	7.90E+01	2.83E+02	2.50E+04	6.66E+03	1.01E+04	2.38E+04	1.43E+03	1.83E+03
F11	1.08E+03	6.12E+02	7.28E+02	9.79E+02	2.94E+02	4.22E+02	3.38E+03	1.77E+03	2.12E+03	3.21E+03	1.35E+03	1.58E+03
F12	1.20E+03	6.12E+02	7.78E+02	1.10E+03	3.70E+02	4.22E+02	3.44E+03	1.63E+03	2.18E+03	3.22E+03	1.31E+03	1.52E+03
F13	1.15E+03	6.33E+02	7.27E+02	1.12E+03	6.00E+02	6.32E+02	3.47E+03	1.72E+03	2.13E+03	3.37E+03	1.42E+03	1.88E+03
F14	1.40E+04	1.54E+04	1.38E+04	1.16E+04	1.29E+04	8.70E+03	3.15E+04	3.46E+04	3.27E+04	2.69E+04	3.10E+04	2.33E+04
F15	1.60E+04	1.62E+04	1.52E+04	1.39E+04	1.28E+04	9.52E+03	3.33E+04	3.42E+04	3.29E+04	3.01E+04	3.17E+04	2.26E+04
F16	6.17E+00	6.22E+00	5.82E+00	4.99E+00	5.15E+00	3.82E+00	6.20E+00	5.73E+00	6.11E+00	5.41E+00	5.51E+00	4.76E+00
F17	2.10E+03	5.70E+02	6.10E+02	1.89E+03	5.03E+02	4.36E+02	6.43E+03	1.33E+03	1.75E+03	5.94E+03	1.18E+03	1.45E+03
F18	2.22E+03	5.45E+02	6.24E+02	2.11E+03	5.31E+02	5.24E+02	6.34E+03	1.32E+03	1.65E+03	6.15E+03	1.17E+03	1.38E+03
F19	2.71E+05	6.75E+04	5.85E+04	2.17E+05	5.86E+03	3.92E+03	1.02E+07	2.60E+06	2.06E+06	8.97E+06	6.58E+05	5.16E+05
F20	2.49E+01	2.50E+01	2.50E+01	2.46E+01	2.50E+01	2.50E+01	5.00E+01	5.00E+01	5.00E+01	5.00E+01	5.00E+01	5.00E+01
F21	5.60E+03	5.39E+03	4.79E+03	5.34E+03	4.59E+03	3.75E+03	1.27E+04	1.10E+04	1.24E+04	1.23E+04	9.32E+03	1.02E+04
F22	1.55E+04	1.61E+04	1.53E+04	1.29E+04	1.21E+04	1.13E+04	3.48E+04	3.57E+04	3.52E+04	3.10E+04	3.42E+04	2.93E+04
F23	1.69E+04	1.74E+04	1.60E+04	1.61E+04	1.50E+04	1.15E+04	3.59E+04	3.60E+04	3.60E+04	3.46E+04	3.45E+04	2.91E+04
F24	4.19E+02	3.56E+02	3.95E+02	4.13E+02	3.26E+02	3.72E+02	7.98E+02	6.52E+02	6.66E+02	7.77E+02	5.68E+02	6.07E+02
F25	4.37E+02	3.97E+02	4.14E+02	4.33E+02	3.73E+02	4.02E+02	7.84E+02	6.93E+02	7.44E+02	7.74E+02	6.39E+02	7.11E+02
F26	4.73E+02	4.88E+02	4.68E+02	4.67E+02	4.69E+02	4.56E+02	7.06E+02	7.17E+02	6.86E+02	6.95E+02	6.83E+02	6.62E+02
F27	2.19E+03	1.95E+03	1.94E+03	2.10E+03	1.47E+03	1.77E+03	4.70E+03	4.35E+03	4.53E+03	4.64E+03	3.61E+03	4.12E+03
F28	7.95E+03	7.21E+03	7.33E+03	7.81E+03	6.47E+03	5.90E+03	2.53E+04	2.28E+04	2.60E+04	2.46E+04	1.89E+04	2.10E+04

Fig. 1 $D = 2$ におけるF14, F16の概形

優れた性能を導出している。全体として、多峰性関数(F7, F14, F15, F16, F19)や合成関数(F22, F23, F26)といった、複雑な形状の関数において提案手法は性能が向上している。これは、近似型サロゲートに基づくOUPSがPSOに対してほぼ性能が改善していないことに対し、近似型サロゲートに加え分類型サロゲートを組み込んだ提案手法の有効性を示すものである。

解評価回数を500回まで減らした場合でも、提案手法は56個中16個の実験ケースで最も高い性能を導出しており、特に複雑な形状の関数(F15, F19, F26)において性能が向上している。一方、OUPSは37個の実験ケースで最も優れた性能を

導出しており、極めて少ない解評価回数に限れば、近似型サロゲートのみを用いるOUPSが効率的な探索を行っている。例として、 $D = 100$ におけるFriedman検定において得られた全ての問題を通した平均順位はPSO, OUPS, 提案手法の順に2.57, 1.57, 1.86であり、OUPSが最も高い順位を出している。しかしながら、事後検定ではPSOとOUPS、PSOと提案手法の各組では有意差が見られる($p < 0.05$)ものの、OUPSと提案手法の組では有意差は検出されず($p \geq 0.05$)、提案手法とOUPSの性能は競合している。

同様の検定により、 $D = 100$ における最大解評価回数である1,000回時点での事後検定では、OUPSと提案手法の組で有意差は検出されていないものの、平均順位はPSO, OUPS, 提案手法の順に2.75, 1.64, 1.61である。つまり、極めて少ない解評価回数では、局所探索を促す近似型サロゲートを用いたOUPSが優位であったが、実験条件である1,000回程度の解評価回数では、大域探索としての分類型サロゲートも用いる提案手法が優位となっている。また、用いた28個の関数の中でも、F14とF16は強い多峰性をもつことから、提案手法が複雑な関数に対して有効であ

る可能性がある。参考として、Fig. 1(a)(b)に、それぞれ $D = 2$ におけるF14, F16の目的関数の形状を示す。

5 考察

提案手法では、PSOに対し $pbest$ を分類型サロゲートによって推測した優良解に置き換えることで、個体群を優良領域へ誘導するメカニズムを導入した。さらに、 $gbest$ 近傍において、近似関数モデル上で推測された最適解 $gbest'$ を現在の $gbest$ と比較し置き換えることで、 $gbest$ の評価値を改善するメカニズムを導入した。本章では、まず、PSOに分類型サロゲートを組み合わせたときの探索性能について考察する。続いて、近似関数モデル上の局所探索の有無が提案手法の探索性能に対してどのように影響しているかについて考察する。

具体的には、PSOに対し分類型サロゲートに基づく $pbest$ 置換を用いる場合と用いない場合について性能比較を行う。分類型サロゲートのみに基づく手法をPSO-SVMと表記する。また、PSO-SVMに加えて近似関数モデル上で $gbest$ 近傍を局所探索するメカニズムを用いる場合と用いない場合について性能比較を行う。PSO-SVMに対し局所探索を加える手法は提案手法と一致する。使用するベンチマーク関数、各種パラメータ、実験設定は前章と同じである。問題の次元数は $D = 100$ とし、性能の評価指標は10試行における最良解の中央値を用いる。なお、評価回数が1,000回時の結果に限定して報告する。

実験結果をTable 2に示す。ここで、提案手法とPSOの実行結果については、前章のTable 1の再掲である。

まず、分類型サロゲートを導入した効果について考察するため、PSOとPSO-SVMの性能比較を行う。実験結果より、F9, F22を除く28個中26個のベンチマーク問題においてPSO-SVMがPSOより優れた性能を導出している。この結果から、本稿で提案した $pbest$ の置換メカニズムによるPSOと分類型サロゲートの融合方法が、高次元問題に対して有効に動作していることが示された。また、個体群を優良領域へ引き寄せつつ、その優良領域を集中的に探索することの有効性が確認できる。

次に、提案手法とPSO-SVMの性能比較を行う。実験結果より、提案手法はF4, F8を除く28個

Table 2 提案手法の近似関数モデル上の $gbest$ 近傍探索の有無による性能比較 (10試行の中央値)。

関数	proposal	PSO-SVM	PSO
F1	2.09E-01	1.26E+05	1.95E+05
F2	4.91E+08	1.73E+09	3.87E+09
F3	4.60E+16	2.16E+17	8.99E+19
F4	4.58E+05	4.37E+05	4.40E+05
F5	2.64E+04	7.37E+04	1.49E+05
F6	9.77E+02	2.21E+04	3.80E+04
F7	3.67E+05	1.23E+06	3.64E+06
F8	2.15E+01	2.14E+01	2.15E+01
F9	1.35E+02	1.45E+02	1.44E+02
F10	1.83E+03	1.66E+04	2.38E+04
F11	1.58E+03	2.60E+03	3.21E+03
F12	1.52E+03	2.89E+03	3.22E+03
F13	1.88E+03	2.92E+03	3.37E+03
F14	2.33E+04	2.69E+04	2.69E+04
F15	2.26E+04	2.63E+04	3.01E+04
F16	4.76E+00	5.25E+00	5.41E+00
F17	1.45E+03	5.26E+03	5.94E+03
F18	1.38E+03	4.88E+03	6.15E+03
F19	5.16E+05	2.76E+06	8.97E+06
F20	5.00E+01	5.00E+01	5.00E+01
F21	1.02E+04	1.11E+04	1.23E+04
F22	2.93E+04	3.19E+04	3.10E+04
F23	2.91E+04	3.14E+04	3.46E+04
F24	6.07E+02	6.60E+02	7.77E+02
F25	7.11E+02	7.22E+02	7.74E+02
F26	6.62E+02	6.69E+02	6.95E+02
F27	4.12E+03	4.33E+03	4.64E+03
F28	2.10E+04	2.25E+04	2.46E+04

中26個のベンチマーク問題において、PSO-SVMの性能を上回っていることが確認できる。また、PSO-SVMにおいてPSOから性能が低下していたF9, F22についても性能を向上させることに成功している。特に、単峰性関数(F1, F2)やF6, F10などの問題では大幅な性能向上が見られる。これは、目的関数形状が比較的単純な問題に対しては近似関数モデルによる最適解の予測が成功しやすくなり、 $gbest$ の改善が大きく進んだためと考えられる。一方で、F4については、精度の低い近似関数モデルによって解の信頼性の低い領域に最適解を予測した可能性がある。その結果、個体群全体がその領域に引き寄せられしまい、解の精度が低下したと考えられる。しかしながら、近似関数モデルを利用したことによる精度の低下は小さく、ほぼ同等の性能であるといえる。

以上の結果より、*pbest*の置換メカニズムと近似関数モデル上で現在の*gbest*の近傍を探索するメカニズムは、PSOの性能向上に貢献していることがわかる。

6 まとめ

本稿では、高計算コストな最適化問題への適用を見据え、高次元複雑な最適化問題をより効率的に解くために、分類型・近似型サロゲートをハイブリッド化したサロゲートPSOアルゴリズムを提案した。提案手法は、このハイブリッド化とPSOフレームワークとの高い親和性に着眼した。つまり、*gbest*と*pbest*の独立性を活かせば、近似・分類型サロゲートのハイブリッド化による相乗効果に加え、サロゲートの精度低下リスクも管理できる。実験結果では、近似型サロゲートであるOUPSと競合する性能を導出できることを示し、特に高次元かつ複雑な関数で性能が改善する傾向を示した。

一方で、本稿で用いた問題次元は、近似型サロゲートも十分に有効に働くことを示した。なお、多目的最適化問題では100次元程度でも、分類問題の有効性が顕著に表れる。したがって、今後はさらに問題次元数を増加させ、提案手法が有効に働く問題クラスを明らかにする。

参考文献

- 1) Russell Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43. Ieee, 1995.
- 2) Yan-Liang Li, Wei Shao, Long You, and Bing-Zhong Wang. An improved pso algorithm and its application to uwb antenna design. *IEEE Antennas and wireless propagation letters*, 12:1236–1239, 2013.
- 3) Masoud Sharafi and Tarek Y ELMekkawy. Multi-objective optimal design of hybrid renewable energy systems using pso-simulation based approach. *Renewable energy*, 68:67–79, 2014.
- 4) Marian Nemec, David W Zingg, and Thomas H Pulliam. Multipoint and multi-objective aerodynamic shape optimization. *AIAA journal*, 42(6):1057–1065, 2004.
- 5) Dhanya M Dhanalakshmy, P Pranav, and G Jeyakumar. A survey on adaptation strategies for mutation and crossover rates of differential evolution algorithm. *International Journal on Advanced Science, Engineering and Information Technology*, 6(5):613–623, 2016.
- 6) Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.
- 7) Guohua Wu, Rammohan Mallipeddi, and Ponnuthurai Nagarathnam Suganthan. Ensemble strategies for population-based optimization algorithms—A survey. *Swarm and evolutionary computation*, 44:695–711, 2019.
- 8) Alan Díaz-Manríquez, Gregorio Toscano, Jose Hugo Barron-Zambrano, and Edgar Tello-Leal. A review of surrogate assisted multiobjective evolutionary algorithms. *Computational intelligence and neuroscience*, 2016:9420460, 2016.
- 9) Xiwen Cai, Haobo Qiu, Liang Gao, Chen Jiang, and Xinyu Shao. An efficient surrogate-assisted particle swarm optimization algorithm for high-dimensional expensive problems. *Knowledge-Based Systems*, 184:104901, 2019.
- 10) Chaoli Sun, Yaochu Jin, Ran Cheng, Jinliang Ding, and Jianchao Zeng. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 21(4):644–660, 2017.
- 11) Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- 12) Ge Gao, Chaoli Sun, Jianchao Zeng, and Songdong Xue. A constraint approximation assisted pso for computationally expensive constrained problems. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pages 1354–1359. IEEE, 2014.
- 13) David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
- 14) JJ Liang, BY Qu, PN Suganthan, and Alfredo G Hernández-Díaz. Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, 201212(34):281–295, 2013.
- 15) Rommel G Regis. Particle swarm with radial basis function surrogates for expensive black-box optimization. *Journal of Computational Science*, 5(1):12–23, 2014.
- 16) Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1):58–73, 2002.
- 17) Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.
- 18) Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies—a comprehensive introduction. *Natural computing*, 1(1):3–52, 2002.
- 19) Xinjing Wang, G Gary Wang, Baowei Song, Peng Wang, and Yang Wang. A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(5):815–827, 2019.