自己適応型差分進化法におけるアルゴリズム構成の事前検証フレームワークによる性能の向上

西原 慧[1,a)]　中田 雅也[1]

概要：自己適応型差分進化法は，アルゴリズム構成を試行錯誤的に調整するため，少ない解評価回数では性能が十分に改善しない．本論文は，調整されたアルゴリズム構成の事前検証によって，試行錯誤的な調整を削減し，少ない解評価回数で高い性能を実現することを目的とする．また，提案する事前検証フレームワークは高い手法的汎用性があり，スケール係数，交叉率，突然変異・交叉戦略を個体ごとに調整する自己適応型差分進化法に適用できる．ベンチマーク問題を用いた実験では，代表手法である jDE と SaDE にそれぞれ提案手法を適用した結果，通常よりも少ない数千オーダの解評価回数において，その性能が改善することを示す．これは，自己適応型差分進化法が不得意とする高計算コストな問題において，提案手法がこれに展開できる汎用的な方法論となり得ることを示すものである．

Performance improvement with Prior-validation framework for Algorithmic configuration on Self-adaptive differential evolution

1. はじめに

実社会の連続最適化問題は，1 回の解評価に時間がかかる高計算コストな問題となる場合がある．このような問題では，最適化アルゴリズムの構成を試行錯誤的に調整する余裕はない．しかしながら，連続最適化問題を対象とした最適化アルゴリズムの 1 つである差分進化法（Differential Evolution: DE）[1] の性能は，ハイパーパラメータや遺伝的オペレータといったアルゴリズム構成に大きく依存する [2], [3]．そこで，DE のアルゴリズム構成をオンラインで調整する自己適応型 DE の研究が盛んに行われている [4]．

初期の研究として，アルゴリズム構成をランダム生成する jDE [5]，それらを確率分布を用いて生成する JADE [6] や SaDE [7] がある．これらの手法は，SHADE [8]，jSO [9] や SaNSDE [10] などの発展的な自己適応型 DE につながる重要な洞察を与えた [11]．なお，本段落で述べた自己適応型 DE を含め，一般的なフレームワークでは 1 つの個体に 1 つのアルゴリズム構成を割り当てる個体ベース調整を用いる．個体ベース調整の自己適応型 DE は，アルゴリズム構成の生成方法に違いがあり，調整したアルゴリズム構成をいかに生成方法へフィードバックするかが議論されてきた．一方で，生成されたアルゴリズム構成をそのまま使用し，その結果を生成方法にフィードバックする点では共通の方法を採用している．つまり，事後検証ベースの方法をもとに試行錯誤的な調整に強く依存することになる．例えば，ベンチマーク問題では，数万回から数十万回の解評価回数で性能評価を行うことが多く [12]，必ずしも高計算コストな最適化問題を想定してこなかった．実際，自己適応型 DE における試行錯誤的な調整フレームワークは，その最適化性能を改善するために多くの解評価回数を消費するという方法論的な欠点が指摘されている [13]．

そこで本論文では，「アルゴリズム構成をいかに生成するか」よりも，「生成されたアルゴリズム構成のうちどれを利用するか」という観点で，数百から数千の解評価回数でも性能を改善可能な自己適応型 DE の追加フレームワークを検討する．具体的には，生成されたアルゴリズム構成を使用する前に，好適なものを選択する汎用的な事前検証フレームワークを導入する．提案手法では，1 つの個体に対しアルゴリズム構成を選択候補として複数生成し，事前検証を経てこの中から 1 つ選択し出力する操作を行う．事前検証では，候補となるアルゴリズム構成を用いて子個体を

1　横浜国立大学大学院
79-1 Tokiwadai, Hodogaya-ku, Yokohama, Kanagawa 240–8501, Japan
a)　nishihara-kei-jv@ynu.jp

仮生成する．そして，その子個体と優良解との類似度を計算し，この指標に基づいて好適なアルゴリズム構成を出力する．最後に，出力されたアルゴリズム構成を用いて子個体を改めて本生成する．これは，1）優良解の周辺領域を局所探索するバイアスを高めたアルゴリズム構成によって収束速度を改善すること，一方で 2）子個体の本生成時に用いる突然変異の非決定性によって個体の多様性の著しい低下を抑制し早期収束を防ぐことを意図する．

様々な自己適応型 DE がある一方で，アルゴリズム構成の生成方法および調整するアルゴリズム構成の違いに着目し，基本的な自己適応型 DE に提案手法を適用する．具体的には，jDE と SaDE に提案手法を組み込む．jDE は，DE のスケール係数 F と交叉率 CR をランダムで生成し，好適なアルゴリズム構成はそのまま再利用する．これに比べて，SaDE は，F と CR に加えて突然変異戦略と交叉戦略を調整対象にし，好適なアルゴリズム構成から求めた確率分布を用いて生成する．したがって，jDE と SaDE は調整対象にするアルゴリズム構成に加えて，好適なアルゴリズム構成のフィードバック法も異なる．なお，JADE と SHADE は F と CR を確率分布を用いて生成するため，SaDE を対象にすることで（jDE と比べて）調整するアルゴリズム構成と生成方法の違いの両方を一度に検証できる．

本論文の構成は次の通りである．第 2 章では DE のアルゴリズムを紹介し，第 3 章では既存の自己適応型 DE を一般化してから，jDE と SaDE のアルゴリズムをまとめる．第 4 章では提案手法のメカニズムについて述べる．第 5 章では，提案手法を jDE と SaDE に組み込み，ベンチマーク問題において，その性能を評価する．最後に第 6 章で結論と今後の展望を述べる．なお，以降では個体ベース調整を用いる自己適応型 DE を単に「自己適応型 DE」と呼ぶことにする．

2. 差分進化法（DE）

本論文では，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 $\boldsymbol{x} \in \mathbb{R}^D$ とする目的関数 $f(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}$ に対し，$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ で与えられる D 次元の大域的最適解 $\boldsymbol{x}^* = [x_1, x_2, \ldots, x_D]$，あるいはその近似解を求めることが目的である．DE の各個体 $\boldsymbol{x}_i$（$i = 1, 2, \ldots, N$）も解ベクトル $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,D}]$ として表現される．ここで N は解集合サイズ（個体数）であり，各個体は解集合 $\mathcal{P}$ に属する（$\boldsymbol{x} \in \mathcal{P}$）．

■ 初期化　世代数 $t = 0$ および $\mathcal{P} = \emptyset$ において，N の数だけ初期解を生成し，$\mathcal{P}$ に追加する．具体的には，i 番目の個体 $\boldsymbol{x}_i$ （$i = 1, 2, \ldots, N$）に対し，j 次元目の決定変数 $x_{i,j} \in [x^l_{i,j}, x^u_{i,j}]$ を一様分布乱数を用いて，定義域からランダムに設定する（$j = 1, 2, \ldots, D$）．ここで，$x^l_{i,j}$ と $x^u_{i,j}$ はそれぞれ $x_{i,j}$ の定義域の最小値および最大値である．

■ 突然変異　次に，$t \leftarrow t + 1$ として，t 世代目の i 番目の個体の突然変異個体 $\boldsymbol{v}_i$ を解集合 $\mathcal{P}$ から生成する．例えば，突然変異戦略の一つである *rand/1* は次式で表される．

$$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}). \tag{1}$$

ここで，スケール係数 $F \in [0, 1]$ は差分ベクトルの大きさを調整するハイパーパラメータであり，その値に応じて大域探索と局所探索のバランスを制御する役割がある．上式において，$\boldsymbol{x}_{r_1}, \ldots, \boldsymbol{x}_{r_3}$ は現在の解集合 $\mathcal{P}$ より自身 $\boldsymbol{x}_i$ 以外でランダムに選択された個体を意味する．

■ 交叉　生成した突然変異個体 $\boldsymbol{v}_i$ と親個体 $\boldsymbol{x}_i$ を交叉させ，交叉後の子個体 $\boldsymbol{u}_i$ を生成する．交叉戦略も複数提案されており，代表的な方法として binomial 交叉と exponential 交叉がある．例えば，binomial 交叉は式 (2) で与えられる．

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand[0,1] \le CR \text{ or } j = j_{rand}, \\ x_{i,j} & \text{otherwise.} \end{cases} \tag{2}$$

なお，交叉率 $CR \in [0, 1]$ は，突然変異個体の決定変数の寄与具合を制御する役割がある．なお，全く交叉されない場合を避けるために $[1, D]$ から整数乱数 j_{rand} を決め，j_{rand} 次元目は必ず交叉させる．

■ 個体選択　式 (3) に示す通り，$\boldsymbol{u}_i$ の評価値が $\boldsymbol{x}_i$ の評価値よりも小さい場合にのみ $\boldsymbol{x}_i$ を $\boldsymbol{u}_i$ に設定する．

$$\boldsymbol{x}_i \leftarrow \begin{cases} \boldsymbol{u}_i & \text{if } f(\boldsymbol{u}_i) < f(\boldsymbol{x}_i), \\ \boldsymbol{x}_i & \text{otherwise.} \end{cases} \tag{3}$$

3. jDE と SaDE

本章では，まず自己適応型 DE を一般化した疑似アルゴリズムを定義する．この理由は，次章で導入する提案手法のフレームワークを特定の自己適応型 DE に限定せず議論するためである．その後，提案手法を組み込む jDE と SaDE のメカニズムについて一般化した疑似アルゴリズムに従いながら説明する．

3.1 自己適応型 DE の一般化アルゴリズム

各個体 $\boldsymbol{x}_i$ に割り当てるアルゴリズム構成を $\boldsymbol{\theta}_i = [\theta_{v,i}, \theta_{u,i}, \theta_{F,i}, \theta_{CR,i}]$ と表記する（$i = 1, 2, \ldots, N$）．また，Θ を $\boldsymbol{\theta}$ の集合とすると，$\boldsymbol{\theta} \in \Theta$ であり $|\Theta| = |\mathcal{P}| = N$ である．ここで，$\boldsymbol{\theta}_i$ の各要素 $\theta_{v,i}, \theta_{u,i}, \theta_{F,i}, \theta_{CR,i}$ をそれぞれ以下のように定義する．

■ θ_v　使用する突然変異戦略の種類に対応させたインデックスを表すカテゴリアル型変数．例えば，$\theta_v = 1, 2$ がそれぞれ *rand/1*, *best/2* に対応すると定義できる．突然変異戦略を自己適応しない場合は，用いる突然変異戦略を永続的に使うという意味で θ_v を 1 に固定する．

■ θ_u　使用する交叉戦略の種類に対応させたインデックスを表すカテゴリアル型変数．例えば，$\theta_u = 1, 2$ がそれぞ

Algorithm 1 一般化した自己適応型 DE (generalized self-adaptive DE)

```
t = 0
Initialize P = {x_1, x_2, ..., x_N}
Initialize Θ = {θ_1, θ_2, ..., θ_N}
while termination criteria are not met do
  t = t + 1
  for i = 1 to N do
    θ_i^{t-1} ← θ_i
    θ_i = [θ_{v,i}, θ_{u,i}, θ_{F,i}, θ_{CR,i}] ← Sample(θ_i^{t-1})
    v_i ← Mutation(P, θ_{v,i}, θ_{F,i})
    u_i ← Crossover(x_i, v_i, θ_{u,i}, θ_{CR,i})
  for i = 1 to N do
    x_i ← { u_i  if f(u_i) < f(x_i)
          { x_i  otherwise
    θ_i ← Update (θ_i, θ_i^{t-1})
```

れ binomial 交叉と exponential 交叉に対応すると定義できる．交叉戦略を自己適応しない場合は，用いる交叉戦略を永続的に使うという意味で θ_u を 1 に固定する．

■ θ_F　スケール係数 F を表す $\theta_F \in [0,1]$ の実数型変数．これを自己適応しない場合は，θ_F を規定値に固定する．

■ θ_{CR}　交叉率 CR を表す $\theta_{CR} \in [0,1]$ の実数型変数．これを自己適応しない場合は，θ_{CR} を規定値に固定する．

次に，自己適応型 DE のアルゴリズムを一般化した疑似コードを Algorithm 1 に示す．最初に解集合 $\mathcal{P} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ とアルゴリズム構成 $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N\}$ を初期化する．次にメインループとして，個体 $\boldsymbol{x}_i$ ごとに疑似関数 $\mathit{Sample}(\boldsymbol{\theta}_i^{t-1})$ でアルゴリズム構成 $\boldsymbol{\theta}_i$ を生成（サンプリング）する．具体的には，現在割り当てている $\boldsymbol{\theta}_i$ を $\boldsymbol{\theta}_i^{t-1} = [\theta_{v,i}^{t-1}, \theta_{u,i}^{t-1}, \theta_{F,i}^{t-1}, \theta_{CR,i}^{t-1}]$ として記憶し，必要であれば $\boldsymbol{\theta}_i^{t-1}$ を考慮して $\boldsymbol{\theta}_i$ を生成する．なお，$\mathit{Sample}(\boldsymbol{\theta}_i^{t-1})$ の定義は，自己適応 DE ごとのアルゴリズム構成の生成方法に依存するため疑似関数として表記している．生成された $\boldsymbol{\theta}_i$ を用いて，突然変異個体 $\boldsymbol{v}_i$，次いで交叉個体 $\boldsymbol{u}_i$ を生成し，解評価および個体選択を行う．そして，疑似関数 $\mathit{Update}\ (\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{t-1})$ を用いて，$\boldsymbol{\theta}_i$ をこのままに設定しておくか，$\boldsymbol{\theta}_i^{t-1}$ に戻すかを決定する．この操作は，自己適応型 DE ごとに異なる．なお，$\mathit{Sample}(\boldsymbol{\theta}_i^{t-1})$ と $\mathit{Update}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{t-1})$ で表記した引数の種類は，自己適応型 DE によって変わるが，最も単純な表記で記載している．

3.2 jDE

jDE [5] は解更新に成功（$f(\boldsymbol{u}_i) < f(\boldsymbol{x}_i)$）したときにその $\theta_{F,i}$ と $\theta_{CR,i}$ を次世代に継承する調整方法を用いる．jDE における $\mathit{Sample}(\boldsymbol{\theta}_i^{t-1})$ を $\mathit{Sample\text{-}jDE}(\boldsymbol{\theta}_i^{t-1})$ として Algorithm 2 に示す．全体のアルゴリズムとしては，最初に全ての個体 $\boldsymbol{x}_i \in \mathcal{P}$ を一様分布で初期化する．また，個体ごとのアルゴリズム構成 $\boldsymbol{\theta}_i$ のうち，$\theta_{F,i}, \theta_{CR,i}$ の値を $\theta_{F,i} = 0.5$, $\theta_{CR,i} = 0.9$ で初期化する．なお，jDE は，突然変異戦略 $\theta_{v,i}$ は *rand/1*，交叉戦略 $\theta_{u,i}$ は binomial 交叉を用い調整対象にしない．次に，Algorithm 2 に示したように，jDE のハイパーパラメータ τ_F, τ_{CR} の確率で新しい $\theta_{F,i}, \theta_{CR,i}$ の値を一様分布でサンプリングし，それ以外では前世代で用いた値を引き継ぐ．そして，これらを用いて一連の世代更新を行い，$\mathit{Update}\ (\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{t-1})$ に相当する操作として，解更新に失敗した場合（$f(\boldsymbol{u}_i^t) \geq f(\boldsymbol{x}_i^t)$）に，$\boldsymbol{\theta}_i$ を $\boldsymbol{\theta}_i^{t-1}$ に戻す（$\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i^{t-1}$）．このように，jDE では一定確率 τ_F, τ_{CR} で新しいハイパーパラメータを試しながら，問題や探索状況により好適なハイパーパラメータを継承する．

Algorithm 2 $\mathit{Sample\text{-}jDE}(\boldsymbol{\theta}_i^{t-1})$

$$\theta_{F,i} = \begin{cases} rand[0.1, 1] & \text{if } rand[0,1] \leq \tau_F \\ \theta_{F,i}^{t-1} & \text{otherwise} \end{cases}$$

$$\theta_{CR,i} = \begin{cases} rand[0, 1] & \text{if } rand[0,1] \leq \tau_{CR} \\ \theta_{CR,i}^{t-1} & \text{otherwise} \end{cases}$$

$\theta_{v,i} = 1$ (*rand/1*)
$\theta_{u,i} = 1$ (*binomial crossover*)
return $\boldsymbol{\theta}_i$

Algorithm 3 $\mathit{Sample\text{-}SaDE}(\boldsymbol{\theta}_i^{t-1})$

```
θ_{v,i}, θ_{u,i} ← Sample a pair of strategies from distribution p
θ_{F,i} = N(0.5, 0.3)
repeat
  θ_{CR,i} = N(CRm_k, 0.1)
until 0 ≤ θ_{CR,i} ≤ 1
return θ_i
```

Algorithm 4 SaDE のパラメータ調整 世代更新前 (adaptation of SaDE parameters before generation process)

```
if t > LP then
  for k = 1 to K do
    p_{k,t} ← eq. (4)
    Remove ns_{k,t-LP}, nf_{k,t-LP}
    CRm_k = median(CR_{Memory_k})
```

3.3 SaDE

SaDE [7] は，DE のハイパーパラメータに加えて突然変異・交叉戦略も調整する代表的な手法である．SaDE における $\mathit{Sample}(\boldsymbol{\theta}_i^{t-1},)$ を $\mathit{Sample\text{-}SaDE}(\boldsymbol{\theta}_i^{t-1})$ として Algorithm 3 に示す．$\theta_{F,i}$ については常に平均値 0.5, 標準偏差 0.3 の正規分布 $\mathcal{N}(0.5, 0.3)$ からサンプリングされる．$\theta_{CR,i}$ と突然変異・交叉戦略 $\theta_{v,i}, \theta_{u,i}$ は，ハイパーパラメータ LP を用いて過去 LP 世代での解更新の成功履歴を用

Algorithm 5 SaDE のパラメータ調整 世代更新後 (adaptation of SaDE parameters after generation process)

```
for i = 1 to N do
  if f(u_i) < f(x_i) then
    ns_{k,t} += 1, CR_{Memory_k} ← Add θ_{CR,i}
  else
    nf_{k,t} += 1
```

いて調整される．また，戦略は突然変異・交叉戦略の組合せを定義し，*rand/1/bin*, *rand/2/bin*, *current-to-rand/1*, *rand-to-best/2/bin* の 4 つが用意される．つまり，この組合せを 1 つ選べば，$\theta_{v,i}$ と $\theta_{u,i}$ が同時に設定される．4 つの組合せのインデックスをそれぞれ $k = 1, 2, 3, 4$（したがってインデックス総数 $K = 4$）とし，これらの組合せごとに過去 LP 世代の解更新の成功率から確率分布 p（歪ルーレット）を定義する．具体的には，Algorithm 5 のように世代更新の終わりの解更新時に戦略 k ごとかつ世代 t ごとの解更新の成功数 $ns_{k,t}$ と失敗数 $nf_{k,t}$ を過去 LP 世代に渡って記録する．そして世代更新の始めに Algorithm 4, 式 (5) に示すように戦略 k ごとの成功率 $S_{k,t}$ を求め，これに応じた戦略 k の選択確率 p_k を式 (4) で生成する．ここで式 (5) の ϵ は，全ての $S_{k,t}$ の第一項が 0 であるときに式 (4) で 0 での除算を回避するための定数である．

$$p_{k,t} = \frac{S_{k,t}}{\sum_{k=1}^{K} S_{k,t}}, \tag{4}$$

$$S_{k,t} = \frac{\sum_{g=t-LP}^{t-1} ns_{k,g}}{\sum_{g=t-LP}^{t-1} ns_{k,g} + \sum_{g=t-LP}^{t-1} nf_{k,g}} + \epsilon. \tag{5}$$

次に，Algorithm 3 のように，この確率分布 p を用いて，現在の世代で各個体 $\boldsymbol{x}_i$ が用いる組合せのインデックス k_i を決定する．$\theta_{CR,i}$ については，過去 LP 世代において解更新に成功した値が戦略ごとに CR_{Memory_k} に保存され，その中央値 CRm_k を平均値とした標準偏差 0.1 の正規分布 $\mathcal{N}(CRm_k, 0.1)$ より個体ごとにサンプリングされる．*Update* $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{t-1})$ に相当する操作として，解更新の可否に関わらず設定した $\boldsymbol{\theta}_i$ をそのまま利用する．

4. 提案手法

jDE や SaDE をはじめとする自己適応型 DE を対象とし，一般化された自己適応型 DE フレームワークに従って提案手法のメカニズムを説明する．

4.1 基本的な流れ

提案手法は，一般化された自己適応型 DE フレームワーク（Algorithm 1）において，アルゴリズム構成を生成する *Sample*$(\boldsymbol{\theta}_i^{t-1})$ を修正する．具体的なメカニズムは以下の通りとなる．

ある個体 $\boldsymbol{x}_i$ に割り当てる $\boldsymbol{\theta}_i$ を生成することを考える（$i = 1, 2, \ldots, N$）．まず，事前検証に用いる基準個体 $\boldsymbol{x}_i^*$ を定義する．具体的な定義方法は次節で述べる．そして，C 個のアルゴリズム構成候補 $\boldsymbol{\theta}'_{i,j} \in \Theta'_i$ を生成する（$j = 1, 2, \ldots, C$）．ここで，Θ'_i はアルゴリズム構成候補の集合とする．生成するアルゴリズム構成候補は，利用する自己適応型 DE の生成方法を C 回適用することで得られる．例えば，jDE では *Sample-jDE*$(\boldsymbol{\theta}_i^{t-1})$ を，SaDE では *Sample-SaDE*$(\boldsymbol{\theta}_i^{t-1})$ をそれぞれ C 回だけ繰り返して実行する．また，$C \in \mathbb{N}$ はハイパーパラメータである．

次に，自己適応型 DE の子個体生成プロセスを実行し，$\boldsymbol{\theta}'_{i,j}$ を用いて仮子個体 $\boldsymbol{u}'_{\boldsymbol{\theta}'_{i,j}}$ を生成する．例えば，jDE では，$\boldsymbol{\theta}'_{i,j}$ で指定される $\theta'_{F,i,j}$ と $\theta'_{CR,i,j}$ を用いて突然変異（*rand/1*）と交叉（binomial 交叉）を適用する．全ての $\boldsymbol{\theta}'_{i,j} \in \Theta'_i$ に対してこの操作を実行し，C 個の仮個体を生成する．ここで，$\boldsymbol{u}'_{\boldsymbol{\theta}'_{i,j}}$ の解評価は行わないことに留意されたい．そして，$\boldsymbol{u}'_{\boldsymbol{\theta}'_{i,j}}$ と $\boldsymbol{x}_i^*$ のユークリッド距離を計算し，その値が最小となるアルゴリズム構成候補を $\boldsymbol{\theta}_i$ と設定する．つまり，$\boldsymbol{\theta}_i$ は以下の式で得られる．

$$\boldsymbol{\theta}_i = \mathop{\arg\min}_{\boldsymbol{\theta}'_{i,j} \in \Theta'_i} ||\boldsymbol{x}_i^* - \boldsymbol{u}'_{\boldsymbol{\theta}'_{i,j}}||. \tag{6}$$

以上が提案手法の基本的な流れとなる．なお，$\boldsymbol{\theta}_i$ が決定した後は，$\boldsymbol{\theta}_i$ を用いて自己適応型 DE の子個体生成プロセスを適用し，$\boldsymbol{x}_i$ に対する $\boldsymbol{u}_i$ を本生成する．そして，次の個体の $\boldsymbol{\theta}_{i+1}$ も上記のメカニズムを経て決定される．

4.2 基準個体の定義

優良解の周辺に対する局所探索のバイアスを生むために，基準個体 $\boldsymbol{x}_i^*$ は現在までに発見した優良解に設定することが好ましいと考える．本論文では，現在の解集合 $\mathcal{P}$ における最良個体を選択し $\boldsymbol{x}_i^*$ に設定する．この場合，全てのアルゴリズム構成が同一の最良個体をもとに選択されることなる．しかしながら，提案手法は，基準個体の近傍に必ず解を生成できるアルゴリズム構成を生成することは意図していない．あくまでも自己適応型 DE の生成方法を踏襲しながら 1 つのアルゴリズム構成を選択するものであり，このように基準個体を設定することが，解の多様性の著しい低下を招く原因とはなりえない．

4.3 計算時間の削減

4.1 節に示したように，提案手法は実装が容易なメカニズムで実現できるが，このメカニズムを全個体に適用するため計算時間が増加することが懸念される．特に，個体ごとに C 個のアルゴリズム構成候補を生成する繰り返し処理に時間がかかる．高計算コストな最適化問題を想定すると，この個体あたりの計算時間の増加は無視できる可能性がある．しかしながら，解の評価時間が数分などの中程度の計

Algorithm 6 $Sample(\mathcal{P}, \boldsymbol{\theta}_i, \boldsymbol{x}_i^*)$

```
Require: P, θ_i, x*_i
  if f(u_i^{t-1}) ≥ f(x_i^{t-1}) then
    Θ'_i = ∅
    for j = 1 to C do
      Θ'_i ← Add sampled configuration θ'_{i,j}
      v'_{θ'_{i,j}} ← Mutation(P, θ'_{v,i,j}, θ'_{F,i,j})
      u'_{θ'_{i,j}} ← Crossover(x_i, v'_{θ'_{i,j}}, θ'_{u,i,j}, θ'_{CR,i,j})
    θ_i ← arg min_{θ'_{i,j} ∈ Θ'_i} ||x*_i − u'_{θ'_{i,j}}||
  return θ_i
```

算コストとなる最適化問題も見越し，計算時間を削減する方法を検討する．具体的には，提案手法のメカニズムにおいて，前世代で解更新に成功した（$f(\boldsymbol{u}_i) < f(\boldsymbol{x}_i)$）個体は，現在のアルゴリズム構成をそのまま出力し再利用することで，計算時間の削減を行う．したがって，前世代で解更新に失敗した（$f(\boldsymbol{u}_i) \geq f(\boldsymbol{x}_i)$）個体にのみ，アルゴリズム構成候補の生成から選択までを実行する．この計算時間の削減を適用した提案手法として，疑似関数 $Sample(\mathcal{P}, \boldsymbol{\theta}_i, \boldsymbol{x}_i^*)$ を Algorithm 6 に示す．

5. 実験

本章では，jDE と SaDE に提案手法を組み込み，ベンチマーク問題を用いてその性能を比較する．

5.1 実験設定

5.1.1 ベンチマーク問題

The special session and competition on Real-Parameter Single Objective Optimization at IEEE Congress on Evolutionary Computation 2013 [12] 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 $F_1, \ldots, F_{28}$ を用いる．関数の次元数は $D = 10, 30, 50, 100$ とし，次元数の増加に対する提案手法のスケーラビリティを評価する．したがって，合計 $112 (= 28 \times 4)$ の実験ケースを行う．コンペティションの規定通り，個体 $\boldsymbol{x}$ の定義域は $\boldsymbol{x} \in [-100, 100]^D$ とする [12].

5.1.2 比較手法と評価方法

提案手法を組み込んだ jDE と SaDE と組み込む前の jDE と SaDE を比較する．なお，全ての手法に対して，初期解は一様分布を用いて生成される．提案手法では，$C = 10$ とする．jDE のパラメータは，$F_{init} = 0.5$, $CR_{init} = 0.9$, $N = 100$, $\tau_F = 0.1$, $\tau_{CR} = 0.1$ とする [5]．SaDE のパラメータは，$p_{k,init} = 0.25$, $CRm_{k,init} = 0.5$, $LP = 50$, $\epsilon = 0.01$ とする [7]．解評価回数を 1,000 とし，異なる乱数シードを用いた独立した 51 試行で得られた性能（最良値の平均値）を計算する．また，jDE と提案手法を組み込んだ jDE，SaDE と提案手法を組み込んだ SaDE の 2 組を設定する．加えて，統計的有意差を確認するために次の 2 つのケースの検定を行う．

検定ケース 1 ある次元数 D のベンチマーク関数ごとに各組に対し Wilcoxon の符号順位検定を適用し，有意水準 0.05 の下で有意差を調べる．具体的には，$p \geq 0.05$ であれば有意差なしとして “∼”，$p < 0.05$ で提案手法を組み込んだ手法が優位ならば “+”，$p < 0.05$ で提案手法を組み込まない手法が優位ならば “–” とする．そして，解評価回数 1,000 でこれらの合計を “+/ – / ∼” として報告する．これにより，全ての試行を通して手法間に有意差が見られるかを関数ごとに確認できる。

検定ケース 2 次元数 D ごとの全ての関数の平均値について，各組に対し Wilcoxon の符号順位検定を適用し，有意水準 0.05 の下で有意差を調べる．全ての関数を通して手法間に有意差が見られるかを p 値で確認できる．

5.2 実験結果

提案手法を jDE, SaDE に適用し，評価回数 1,000 において，次元数を $D = 10, 30, 50, 100$ と変化させたときの性能を**表 1** に示す．網掛けした部分は手法がもう一方の手法と同じかより高い性能を導出していることを表す．全体を通して，提案手法を自己適応型 DE に適用することにより，多くのケースで性能が向上している．例として，jDE に提案手法を適用した場合，表 1 の左半分において $D = 10, 30, 50, 100$ の順にそれぞれ 28 個のうち 23, 23, 22, 22 個の関数で性能が向上している．検定ケース 1 でも 7, 10, 16, 17 個の関数で “+” となり，提案手法が優位である．特に次元数が増加し問題の難易度が上昇したときに，提案手法の有効性が高くなる．また，表の最後に示したように，検定ケース 2 で全ての次元数で $p < 0.05$ となり，関数全体での統計的有意差も見られている．すなわち，28 個の関数における全体的な提案手法の有効性が示されている．一方，$D = 10, 50, 100$ における F_9 や $D = 30, 50, 100$ における F_{26} のように提案手法が性能向上に失敗している．しかしながら，検定ケース 2 で $p < 0.05$ であることから，この性能低下に統計的な有意性はない．

SaDE に提案手法を適用した場合(表 1 の右半分)でも jDE の場合と同様の傾向であり，特に $D = 50$ では全てのケースで性能を向上または維持している．$D = 10, 30, 50, 100$ の順に全 28 個のうち 16, 17, 18, 19 個の関数で “+” となり，有意差をもって提案手法が性能を向上している．同じく SaDE でも全ての関数で，提案手法を用いない場合が優位であることを示す “–” は存在しない．さらに表の最後に示したように，全ての次元数で $p < 0.05$ となり問題全体の統計的有意差も見られている．これらの結果から，SaDE が遺伝的オペレータまで調整することを考えると，提案手法がより大きなアルゴリズム構成の探索空間を扱う場合にも高い有効性があるといえる．

表 1: jDE と提案手法を適用した jDE，SaDE と提案手法を適用した SaDE の性能比較
Table 1 Comparison of fitness derived from jDE, SaDE and jDE and SaDE with the proposed method.

	$D=10$		$D=30$		$D=50$		$D=100$		$D=10$		$D=30$		$D=50$		$D=100$	
	jDE	ours	jDE	ours	jDE	ours	jDE	ours	SaDE	ours	SaDE	ours	SaDE	ours	SaDE	ours
F1	5.04E+03	4.54E+03 ∼	5.92E+04	5.16E+04 +	1.25E+05	1.08E+05 +	3.28E+05	2.89E+05 +	3.68E+03	2.85E+03 +	4.60E+04	3.64E+04 +	9.45E+04	7.28E+04 +	2.53E+05	1.97E+05 +
F2	3.60E+07	3.86E+07 ∼	9.40E+08	9.15E+08 ∼	2.60E+09	2.57E+09 ∼	1.23E+10	1.09E+10 +	3.66E+07	3.36E+07 ∼	8.75E+08	7.72E+08 +	2.36E+09	1.91E+09 +	1.05E+10	8.81E+09 +
F3	1.32E+10	1.27E+10 ∼	5.36E+15	5.53E+15 ∼	2.89E+16	1.04E+16 +	1.30E+24	5.15E+23 +	1.12E+10	9.55E+09 ∼	4.62E+15	9.41E+14 ∼	5.37E+15	2.34E+15 +	3.09E+23	2.28E+22 +
F4	4.53E+04	4.14E+04 ∼	1.66E+05	1.57E+05 ∼	2.61E+05	2.49E+05 ∼	5.49E+05	5.30E+05 ∼	4.71E+04	4.82E+04 ∼	1.50E+05	1.49E+05 ∼	2.33E+05	2.25E+05 ∼	4.98E+05	4.81E+05 ∼
F5	3.18E+03	2.42E+03 +	4.73E+04	4.58E+04 ∼	1.24E+05	1.10E+05 +	4.15E+05	3.62E+05 +	2.60E+03	1.96E+03 +	4.14E+04	3.23E+04 +	8.70E+04	6.47E+04 +	2.92E+05	2.19E+05 +
F6	3.17E+02	3.00E+02 ∼	9.16E+03	8.16E+03 +	1.93E+04	1.51E+04 +	1.03E+05	8.77E+04 +	2.91E+02	2.28E+02 +	7.00E+03	5.26E+03 +	1.23E+04	8.52E+03 +	7.44E+04	5.23E+04 +
F7	1.70E+02	1.66E+02 ∼	6.16E+04	4.68E+04 ∼	8.65E+04	5.89E+04 +	3.94E+08	3.33E+08 ∼	1.57E+02	1.45E+02 +	2.58E+04	1.36E+04 +	4.40E+04	2.06E+04 +	2.02E+08	6.37E+07 +
F8	2.07E+01	2.07E+01 ∼	2.12E+01	2.12E+01 ∼	2.13E+01	2.13E+01 ∼	2.15E+01	2.15E+01 ∼	2.07E+01	2.08E+01 ∼	2.12E+01	2.12E+01 ∼	2.13E+01	2.13E+01 ∼	2.15E+01	2.15E+01 ∼
F9	1.11E+01	1.14E+01 ∼	4.50E+01	4.49E+01 ∼	7.98E+01	8.03E+01 ∼	1.72E+02	1.73E+02 ∼	1.13E+01	1.09E+01 +	4.49E+01	4.44E+01 ∼	8.01E+01	8.00E+01 ∼	1.73E+02	1.72E+02 ∼
F10	6.16E+02	5.71E+02 ∼	8.22E+03	7.54E+03 +	1.72E+04	1.59E+04 +	5.14E+04	4.71E+04 +	5.36E+02	4.18E+02 +	6.55E+03	5.66E+03 +	1.41E+04	1.14E+04 +	4.24E+04	3.44E+04 +
F11	1.36E+02	1.26E+02 +	9.48E+02	8.71E+02 +	1.95E+03	1.78E+03 +	5.35E+03	4.86E+03 +	1.18E+02	1.08E+02 +	8.12E+02	6.96E+02 +	1.56E+03	1.29E+03 +	4.27E+03	3.48E+03 +
F12	1.36E+02	1.35E+02 ∼	9.94E+02	9.14E+02 +	1.88E+03	1.73E+03 +	5.58E+03	5.00E+03 +	1.27E+02	1.10E+02 +	8.28E+02	7.08E+02 +	1.54E+03	1.32E+03 +	4.44E+03	3.52E+03 +
F13	1.36E+02	1.30E+02 ∼	9.67E+02	8.98E+02 +	1.90E+03	1.75E+03 +	5.66E+03	4.94E+03 +	1.25E+02	1.15E+02 +	8.10E+02	7.25E+02 +	1.54E+03	1.31E+03 +	4.39E+03	3.59E+03 +
F14	2.19E+03	2.14E+03 +	8.60E+03	8.44E+03 ∼	1.55E+04	1.52E+04 +	3.43E+04	3.39E+04 +	2.13E+03	2.06E+03 ∼	8.45E+03	8.47E+03 ∼	1.54E+04	1.53E+04 ∼	3.41E+04	3.41E+04 ∼
F15	2.19E+03	2.13E+03 ∼	8.97E+03	8.94E+03 ∼	1.64E+04	1.64E+04 ∼	3.43E+04	3.42E+04 ∼	2.17E+03	2.12E+03 ∼	8.88E+03	8.91E+03 ∼	1.65E+04	1.63E+04 ∼	3.41E+04	3.42E+04 ∼
F16	2.52E+00	2.49E+00 ∼	4.53E+00	4.47E+00 ∼	5.51E+00	5.44E+00 ∼	5.70E+00	5.81E+00 ∼	2.43E+00	2.42E+00 ∼	4.45E+00	4.56E+00 ∼	5.48E+00	5.43E+00 ∼	5.73E+00	5.65E+00 ∼
F17	2.25E+02	2.02E+02 +	1.94E+03	1.68E+03 +	4.31E+03	3.42E+03 +	1.04E+04	8.54E+03 +	1.87E+02	1.48E+02 +	1.44E+03	1.07E+03 +	2.99E+03	2.12E+03 +	7.40E+03	5.42E+03 +
F18	2.28E+02	2.03E+02 +	1.92E+03	1.66E+03 +	4.22E+03	3.50E+03 +	1.04E+04	8.47E+03 +	1.93E+02	1.58E+02 +	1.43E+03	1.07E+03 +	2.96E+03	2.15E+03 +	7.41E+03	5.31E+03 +
F19	3.09E+03	2.74E+03 ∼	1.94E+06	1.47E+06 +	6.65E+06	3.91E+06 +	7.66E+07	5.05E+07 +	1.43E+03	6.31E+02 +	9.11E+05	5.36E+05 +	2.55E+06	1.07E+06 +	3.27E+07	1.61E+07 +
F20	4.70E+00	4.66E+00 ∼	1.50E+01	1.50E+01 ∼	2.50E+01	2.50E+01 ∼	5.00E+01	5.00E+01 ∼	4.68E+00	4.62E+00 ∼	1.50E+01	1.50E+01 ∼	2.50E+01	2.50E+01 ∼	5.00E+01	5.00E+01 ∼
F21	7.16E+02	6.98E+02 ∼	4.63E+03	4.26E+03 +	1.02E+04	8.69E+03 +	2.29E+04	1.82E+04 +	6.48E+02	5.75E+02 +	3.75E+03	3.15E+03 +	8.06E+03	6.37E+03 +	1.71E+04	1.36E+04 +
F22	2.36E+03	2.34E+03 ∼	9.47E+03	9.31E+03 ∼	1.67E+04	1.66E+04 ∼	3.64E+04	3.60E+04 +	2.43E+03	2.28E+03 +	9.34E+03	9.24E+03 ∼	1.67E+04	1.65E+04 ∼	3.60E+04	3.59E+04 ∼
F23	2.47E+03	2.49E+03 ∼	9.55E+03	9.43E+03 ∼	1.73E+04	1.72E+04 ∼	3.60E+04	3.61E+04 ∼	2.48E+03	2.45E+03 ∼	9.57E+03	9.57E+03 ∼	1.73E+04	1.73E+04 ∼	3.60E+04	3.62E+04 ∼
F24	2.33E+02	2.31E+02 +	3.54E+02	3.52E+02 ∼	5.08E+02	5.01E+02 ∼	1.49E+03	1.41E+03 ∼	2.32E+02	2.31E+02 ∼	3.46E+02	3.38E+02 +	4.88E+02	4.67E+02 +	1.20E+03	9.84E+02 +
F25	2.32E+02	2.32E+02 ∼	3.61E+02	3.61E+02 ∼	5.05E+02	5.05E+02 ∼	9.82E+02	9.79E+02 ∼	2.33E+02	2.32E+02 ∼	3.56E+02	3.44E+02 +	5.03E+02	4.54E+02 +	9.41E+02	7.76E+02 +
F26	2.18E+02	2.16E+02 ∼	3.74E+02	3.76E+02 ∼	5.05E+02	5.06E+02 ∼	7.62E+02	7.64E+02 ∼	2.18E+02	2.15E+02 ∼	3.63E+02	3.41E+02 +	5.05E+02	5.04E+02 ∼	7.60E+02	7.51E+02 +
F27	7.12E+02	7.10E+02 ∼	1.55E+03	1.54E+03 ∼	2.65E+03	2.60E+03 +	5.97E+03	5.72E+03 +	6.89E+02	6.70E+02 +	1.52E+03	1.51E+03 ∼	2.59E+03	2.53E+03 +	5.66E+03	5.39E+03 +
F28	1.21E+03	1.14E+03 +	7.16E+03	7.07E+03 ∼	1.29E+04	1.23E+04 +	3.68E+04	3.48E+04 +	1.11E+03	1.04E+03 +	6.60E+03	6.09E+03 +	1.14E+04	1.03E+04 +	3.22E+04	2.95E+04 +
検定ケース 1: +/−/∼		7/0/21		10/0/18		16/0/12		17/0/11		16/0/12		17/0/11		18/0/10		19/0/9
検定ケース 2: p value		0.0006956		0.0003274		0.00003644		0.00003667		0.00007775		0.00007643		0.00001306		0.00005129

6. 結論

本論文は，自己適応型差分進化法（DE）のアルゴリズム構成を事前検証する，一般的な自己適応型 DE に適用可能なフレームワークを提案した．提案手法は実際に解を生成する前に，自己適応型 DE の方法に則りアルゴリズム構成候補を複数生成する．続いて，優良解への局所探索のバイアスを与えるアルゴリズム構成を 1 つ選択した後に，これを用いて実際の解生成を行う．実験では，提案手法を自己適応型 DE の代表手法である jDE や SaDE に組み込むことで，特に 1,000 回の解評価回数において，次元数に対するスケーラビリティを実現しつつ最適化性能を向上することを示した．したがって，従来の自己適応型 DE が試行錯誤的調整を行うために解評価回数を大量に消費していたのに対し，提案手法は試行錯誤的な調整要素を緩和して解評価回数を削減できる．

謝辞 本研究は JSPS 科研費 20H04254 の助成を受けた.

参考文献

[1] Storn, R. and Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization*, Vol. 11, No. 4, pp. 341–359 (1997).

[2] Karafotias, G., Hoogendoorn, M. and Eiben, Á. E.: Parameter control in evolutionary algorithms: Trends and challenges, *IEEE Trans. Evo. Comp.*, Vol. 19, No. 2, pp. 167–187 (2014).

[3] Huang, C., Li, Y. and Yao, X.: A Survey of Automatic Parameter Tuning Methods for Metaheuristics, *IEEE Trans. Evo. Comp.*, Vol. 24, No. 2, pp. 201–216 (2020).

[4] Tanabe, R. and Fukunaga, A.: Reviewing and benchmarking parameter control methods in differential evolution, *IEEE Trans. Cybernetics*, Vol. 50, No. 3, pp. 1170–1184 (2019).

[5] Brest, J., Greiner, S., Boskovic, B., Mernik, M. and Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, *IEEE Trans. Evo. Comp.*, Vol. 10, No. 6, pp. 646–657 (2006).

[6] Zhang, J. and Sanderson, A. C.: JADE: adaptive differential evolution with optional external archive, *IEEE Trans. Evo. Comp.*, Vol. 13, No. 5, pp. 945–958 (2009).

[7] Qin, A. K., Huang, V. L. and Suganthan, P. N.: Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Trans. Evo. Comp.*, Vol. 13, No. 2, pp. 398–417 (2008).

[8] Tanabe, R. and Fukunaga, A.: Success-history based parameter adaptation for differential evolution, *2013 IEEE Cong. Evo. Comp.*, IEEE, pp. 71–78 (2013).

[9] Brest, J., Maučec, M. S. and Bošković, B.: Single objective real-parameter optimization: algorithm jSO, *2017 IEEE Cong. Evo. Comp.*, IEEE, pp. 1311–1318 (2017).

[10] Yang, Z., Tang, K. and Yao, X.: Self-adaptive differential evolution with neighborhood search, *2008 IEEE Cong. Evo. Comp.*, IEEE, pp. 1110–1116 (2008).

[11] Das, S., Mullick, S. S. and Suganthan, P. N.: Recent advances in differential evolution–an updated survey, *Swarm and Evo. Comp.*, Vol. 27, pp. 1–30 (2016).

[12] Liang, J., Qu, B., Suganthan, P. and Hernández-Díaz, A. G.: Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, *Comp. Intell. Lab., Zhengzhou Univ., China and Nanyang Technological Univ., Singapore, Tech. Rep.*, Vol. 201212, No. 34, pp. 281–295 (2013).

[13] Lu, X., Tang, K., Sendhoff, B. and Yao, X.: A new self-adaptation scheme for differential evolution, *Neurocomputing*, Vol. 146, pp. 2–16 (2014).

