横浜国立大学大学院 理工学府

令和三年度 修士論文

事前検証型パラメータ適応進化計算

学籍番号　20NC545
氏名　西原 慧

数物・電子情報系理工学専攻 電気電子ネットワーク教育分野
主任指導教員　中田 雅也 准教授

提出日　令和 4 年 3 月 11 日（金）

概要

実社会のあらゆるところに最適化問題が存在し，汎用的な最適化技術である進化計算は欠かせないものとなっている．中でも，1 回の解評価に時間がかかる，高計算コストな最適化問題が多く存在する．ここでは，可能な限り少ない解評価での高性能な解導出が求められる．一方で，進化計算の最適化性能はハイパーパラメータ設定に強く依存することが指摘されており，問題や探索状況に応じてパラメータを適切に調整することで性能は向上するが，問題や挙動の複雑性から調整指針のほとんどは不明である．

そこで，1 回の解探索中にオンラインにハイパーパラメータを自動調整する，適応進化計算が有効な手法となる．しかしながら，適応進化計算は，ハイパーパラメータを固定した進化計算に比べて高い性能を導出する一方で，性能の改善に数十万回以上の解評価を要するため，高計算コスト問題には適さない．具体的には，既存の適応進化計算は，ハイパーパラメータを選択した後にこれを使用して解を生成し，その評価値をフィードバックする，「事後検証型」の適応を行う．言い換えると，ハイパーパラメータの試行錯誤的な調整により，解評価回数が膨大に消費されることになる．また，調整したハイパーパラメータは有効性を検証されることなく，そのまま使用される．

これらを受けて本論文では，解生成と解評価に先立って，生成されたハイパーパラメータ候補群から好適な候補を解評価なしにスクリーニングする，「事前検証型」の適応進化計算を提案する．実験では，制約なし単一目的連続実数値最適化問題のベンチマークセットにおいて，提案手法を組み込んだ適応差分進化（適応進化計算の一手法群）が，従来より少ない数千回の解評価回数で性能を向上することを示す．

また，本論文の後半では，更なる性能向上を目指し，適応の対象をハイパーパラメータから適応進化計算の種類にまで拡張した研究を行う．同じく実験では，同様の適応差分進化をアンサンブルした既存手法よりも，提案手法が高い性能を導出することを示す．

併せて，本論文では，既存の適応差分進化をハイパーパラメータの適応方法で分類しながら紹介する．考察では提案手法の分析を行い，最後に今後の展望を述べる．

目次

図目次

表目次

第 1 章

序論

1.1 研究背景

今日の工学システムなど，実社会のあらゆるところに最適化問題は内包され，これを解く最適化技術の発展は重要である．最適化問題とは，所望の目的の達成度を表す指標を目的関数として定義し，目的を達成するために選択する量を表す決定変数を変化させて，目的関数を最小化あるいは最大化するような決定変数を求める問題を指す．

例えば，制約なし単一目的連続実数値最小化問題では，D 次元の決定変数を $\boldsymbol{x} \in \mathbb{R}^D$ とする目的関数 $f(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}$ に対し，の大域的最適解 $\boldsymbol{x}^* = [x_1, x_2, \cdots, x_D]$ は次のような式で表される．

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) \tag{1.1}$$

決定変数が連続値である連続最適化では解候補は無数にあり，離散値である組合せ最適化ではある次元数を超えると解候補の組合せ爆発が起きるように，これら最適化問題には NP クラスに分類される問題も多く，木やグラフ・その他の探索において実際に全ての解を試すことは現実的でない．そのため，最適化技術でも，現実的な時間で実用的な解を得る近似解法が有効となる．また，実社会における最適化問題は Black-box であることが多く [1]，ここでは，目的関数の形状や勾配といった問題の事前知識は利用できない．

そのため，多くの最適化問題に汎用的に利用可能な最適化技術群であるメタヒューリスティクスが，その有効性が認知されている [2]．中でも，生物の進化や群行動を模した最適化技術である進化計算は，その高い効率と性能から，実応用例も多い．進化計算では，問題の解候補集合を生物個体群（解集合）と考え，解候補に交叉や突然変異，選択といった遺伝的操作で探索を行う．基本的には，これら遺伝的操作を繰り返し適用しつつ，良い

評価値を持つ解をもとに，解集合内の解候補の相互作用で最適解を進化的に探索する．

実社会の最適化問題には，解評価に時間がかかる高計算コストな最適化問題（Computationally Expensive Optimization Problems: CEP）が多く存在する [1, 3]．例として，航空機形状設計 [4] やトラウマシステムの設計 [5, 6]，人工ニューラルネットワークモデルの構造設計 [7, 8] などが挙げられる．一方で，進化計算の最適化性能はハイパーパラメータ設定に強く依存することが指摘されている [9, 10]．よって，問題や探索状況に応じてハイパーパラメータを適切に調整することで性能の向上が期待されるが [11, 12]，問題の Black-box 性や，問題や挙動の複雑性から調整指針のほとんどは不明である．しかしながら，CEP では一回の試行に非常に長い時間を要するため，問題を繰り返し解くオフラインの方法でハイパーパラメータを試行錯誤的に調整する余裕はない．

そこで，一回の探索中にオンラインに，ハイパーパラメータを問題や探索状況に自動特化させる最適化技術である適応進化計算が実用的な手法になり得る [13, 14]．特に，CEP が想定する数百〜数千回などの少ない解評価回数 [15] で妥当な適応が成されるのであれば，適応進化計算は CEP に有効な方法論であると言える．

実数値連続最小化問題を対象とした差分進化法（Differential Evolution: DE）[16] は，適応技術を組み合わせることで飛躍的に性能が改善することが知られており [17, 18, 19]，これまで多くの適応差分進化（適応 DE）が提案されている [20]．この背景には，DE の性能もまたハイパーパラメータ（スケール係数 F，交叉率 CR，突然変異戦略，交叉戦略）に大きく左右されることがあげられる [21, 22]．加えて，問題に適切なハイパーパラメータを事前に決定することが困難である点や，どのハイパーパラメータを探索中のどのタイミングで適用すべきかが分からない点も挙げられる [12, 23, 24]．近年では，アンサンブル最適化 [25, 26, 27, 28, 29]，マルチモーダル最適化 [30, 31]，機械学習を用いたサロゲート型最適化 [32, 33, 34, 35, 36, 37] などに，適応 DE が利用されている．初期の研究として，ハイパーパラメータをランダム生成する jDE [38] や CoDE [39]，それらを確率分布を用いて生成する JADE [40] や SaDE [41, 42] がある．これらの手法は，SHADE [43]，jSO [44] や SaNSDE [45] などの発展的な適応 DE につながる重要な洞察を与えた [19, 46]．なお，本段落で述べた適応 DE を含め，一般的なフレームワークでは 1 つの個体に 1 つのハイパーパラメータを割り当てる個体ベース適応を用いる．個体ベース適応の適応 DE は，ハイパーパラメータの生成方法に違いがあり，調整したハイパーパラメータをいかに生成方法へフィードバックするかがこれまで議論されてきた．

一方で，生成されたハイパーパラメータをそのまま使用し，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，事後検証型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．たとえば，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 [47, 48]，必ずしも CEP を想定してこなかった．実際，適応 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 [33]．したがって，CEP への展開に向けて，少ない解評価回数でも有効に働く適応 DE の構築が求められる．

1.2 研究目的

そこで本論文では，「ハイパーパラメータをいかに生成するか」よりも，「生成されたハイパーパラメータ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適応 DE の追加フレームワークを検討する．具体的には，生成されたハイパーパラメータを使用する前に，好適なものをスクリーニング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 つの個体に対しハイパーパラメータ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，候補となるハイパーパラメータを用いて子個体を仮生成する．そして，その子個体と暫定的な優良解との類似度を計算し，この指標に基づいて好適なハイパーパラメータを出力する．最後に，出力されたハイパーパラメータを用いて子個体を改めて本生成する．これは，

- 優良解の周辺領域を局所探索可能なハイパーパラメータによって収束速度を改善すること，
- 一方で，子個体の本生成時に用いる突然変異や交叉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．上記で想定する効果は，粒子群最適化（Particle Swarm Optimization: PSO）[49] における関連研究と，著者の先行研究の知見を活かしている．具体的には，サロゲート PSO である OUPS は，1 つの粒子に対し複数の速度ベクトルを仮生成し，目的関数を近似したサロゲート（機械学習）を用いて近似評価値が最も高い速度ベクトルを利用する [50]．優良解の領域への局所探索が可能な速度ベクトルは，本論文で DE のハイパーパラメータとみなせる．また，著者が構築したアンサンブル型の適応 DE は，競合する DE が導出した優良解を模倣するようにハイパーパラメータを競合的に生成することで，収束速度が改善する [51]．したがって，サロゲートを構築しなくとも，優良解の模倣によって優良解の領域への局所探索を重視したハイパーパラメータを生成できる可能性がある．

1.3 研究方法

提案手法は，特定の適応 DE に限定せず，個体ベース適応で F，CR，突然変異戦略や交叉戦略を調整する適応 DE に利用できる利点がある．この手法的汎用性を高める理由は，適応 DE の問題依存性 [20] からユーザが用いる適応 DE を選択する余地を残すためである．加えて，ユーザが自動調整を所望するハイパーパラメータの違いに対応するためである．様々な適応 DE がある一方で，ハイパーパラメータの生成方法および適応するハイパーパラメータの違いに着目し，基本的な適応 DE に提案手法を適用する．具体的には，jDE と SaDE，JADE に提案手法を組み込む．jDE は，F と CR をランダムで生成し，好適なハイパーパラメータ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適応対象にし， 好適なハイパーパラメータ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 は適応対象にするハイパーパラメータに加えて，好適なハイパーパラメータのフィードバック法も異なる．また，F と CR を適応対象とする JADE は，評価値が上位の個体をもとにハイパーパラメータを適応する点に特徴があり，SaDE のフィードバック方法と異なる．

実験では，制約なし単一目的連続実数値最適化のベンチマークセット（IEEE CEC2013 real-parameter single-objective benchmark function suite） [47] において，提案手法を組み込んだ jDE と SaDE，JADE が提案手法を組み込まないオリジナルの手法よりも数千回の評価回数で性能が改善することを示す．したがって，本論文で実証する提案手法の有効性は jDE，SaDE ならびに JADE に限定される．この意味で，本論文は他の適応 DE における提案手法の有効性を主張するものではなく，提案手法の手法的汎用性を利点とすることに留意されたい．また，提案手法を組み込んだ場合の純粋な性能評価を行うために，少ない解評価回数を見越した jDE と SaDE，JADE の微調整（fine-tuning）は行わず，文献にしたがったフレームワークおよびパラメータ設定を用いる．つまり，CEP を見据えて，人手による調整は考慮しないこととする．

また，本論文の後半では，更なる性能向上を目指し，適応の対象をハイパーパラメータから，ハイパーパラメータと適応進化計算の種類の両方に拡張した研究を行う．この拡張研究の背景は，第 8 章で詳細に述べる．本論文では，[26] や [29] に見られるように，アンサンブル適応 DE で代表的な手法である JADE と CoDE をアンサンブルし，事前検証型適応を行う．同じく実験では，同様の適応 DE をアンサンブルした既存手法よりも，提案手法が高い性能を導出することを示す．

1.4 論文構成

本論文の構成は次の通りである．第 2 章で DE のアルゴリズムを紹介し，第 3 章では既存の適応 DE をハイパーパラメータの生成方法の観点でまとめる．続く第 4 章で，提案手法のメカニズムについて述べる．第 5 章では，実験で用いるベンチマーク問題の性質について述べる．第 6 章では，提案手法を jDE と SaDE，JADE に組み込み，前述のベンチマーク問題でその性能を評価し，第 7 章で提案手法の分析を行う．第 8〜10 章が，提案手法の拡張の研究部分となる．まず，第 8 章で，拡張研究を行う動機を述べる．次に，第 9 章で，JADE と CoDE をアンサンブルする提案手法を，同様の適応 DE をアンサンブルする HMJCDE [26] や EDEV [29] と比較する実験を行う．第 7 章で，拡張手法の考察を行い，最後に，第 11 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適応を用いる適応 DE を，単に「適応 DE」と呼ぶこととする．

第 2 章

差分進化法（DE）

2.1 概要

DE [16] は，解集合ベースの単一目的最適化技術であり，シンプルなアルゴリズムにして高い性能を持つことから [46]，近年でも盛んに研究が行われている [20].

本論文では，式（1.1）で示されるような，制約なし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，以下で与えられる 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った.

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（$\boldsymbol{x} \in \mathcal{P}$）. DE は，他の進化計算とは異なり，特定の生物の振る舞いを模倣したものではなく，Nelder-Mead 法 [52] に着想を得た手法である．Nelder-Mead 法は，勾配情報を利用しない決定的探索アルゴリズムであり，目的関数の次元数 D に対し $D + 1$ 個の探索点から構成される個体に，反射・拡張・縮小の操作を行い子個体を生成し，次いで比較と解更新を行う．同手法は $D \leq 2$ においては State-of-the-Art な進化計算に対しても比較的優位な性能を導出が，それ以上の次元数や多峰性関数においては性能は大きく低下することが知られている [53]．一方で，極端な高次元問題でない限り性能が低下しない DE [46] は，Nelder-Mead 法の反射操作を発展させ，遺伝的アルゴリズム（Genetic Algorithm: GA） [54] のように初期化・突然変異・交叉・選択からなるアルゴリズムとしている．以下に示すように，DE のフレームワークでは，初期化を行った後に，突然変異の適用から選択までの一連のプロセスを探索の終了条件を満たすまで繰り返す.

Algorithm 1 差分進化法（Differential Evolution: DE）

$t = 0$
Initialize $\mathcal{P} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$
while *termination criteria* are not met **do**
  $t = t + 1$
  for $i = 1$ **to** N **do**
    $\boldsymbol{v}_i \leftarrow Mutation(\mathcal{P})$
    $\boldsymbol{u}_i \leftarrow Crossover(\boldsymbol{x}_i, \boldsymbol{v}_i)$
  end for
  for $i = 1$ **to** N **do**
    if $f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i)$ **then**
      $\boldsymbol{x}_i \leftarrow \boldsymbol{u}_i$
    else
      $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i$
    end if
  end for
  $t = t + 1$
end while

2.2 メカニズム

DE の疑似コードを Algorithm 1 に示す．

2.2.1 初期化

世代数 $t = 0$ および $\mathcal{P} = \emptyset$ において，個体数 N の数だけ初期解を生成し，解集合 $\mathcal{P}$ 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．

表 2.1　DE における代表的な突然変異戦略

名称	定義
rand/1	$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$
rand/2	$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}) + F(\boldsymbol{x}_{r_4} - \boldsymbol{x}_{r_5})$
best/1	$\boldsymbol{v}_i = \boldsymbol{x}_{best} + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$
best/2	$\boldsymbol{v}_i = \boldsymbol{x}_{best} + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2}) + F(\boldsymbol{x}_{r_3} - \boldsymbol{x}_{r_4})$
current-to-rand/1	$\boldsymbol{v}_i = \boldsymbol{x}_i + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_i) + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$
current-to-best/1	$\boldsymbol{v}_i = \boldsymbol{x}_i + F(\boldsymbol{x}_{best} - \boldsymbol{x}_i) + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$
current-to-pbest/1	$\boldsymbol{v}_i = \boldsymbol{x}_i + F(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i) + F(\boldsymbol{x}_{r_1} - \tilde{\boldsymbol{x}}_{r_2})$.
rand-to-best/1	$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{best} - \boldsymbol{x}_{r_1}) + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$

2.2.2　突然変異

次に，t 世代目の i 番目の個体 $\boldsymbol{x}_i$ の突然変異個体 $\boldsymbol{v}_i$ を，現在の解集合 $\mathcal{P}$ から生成する．表 2.1 に示すように，これまでに様々な突然変異戦略が提案されている．例えば *rand/1* は次式で表される．*current-to-pbest/1* の説明は，3.3 節で行う．

$$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}). \tag{2.1}$$

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．上式ならびに表 2.1 において，$\boldsymbol{x}_{r_1}, \ldots, \boldsymbol{x}_{r_5}$ 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であり，$\boldsymbol{x}_{best}$ は $\mathcal{P}$ 内の最良個体を意味する．

rand/1 や *rand/2*，*current-to-rand/1* は，ランダム性が強いため大域探索を行う効果が強い．一方で，*best/1* や *best/2*，*current-to-best/1*，*rand-to-best/1* は，世代 t における最良解の周辺を探索するように突然変異が行われる．また，それぞれの突然変異戦略の中でも，末尾の数字が大きいほど加味される差分ベクトルの数が増えるため，*rand/1* や *best/1* よりも *rand/2* や *best/2* の方が生成される解の多様性が向上する傾向があると考えられる．さらに，*current-to-rand/1* や *current-to-best/1* は，親個体近傍に突然変異個体を生成する傾向がある．なお，[55] では，大域探索と局所探索の圧力（Tradeoff between Exploration and Exploitation: TEE）は，*best/2* を除いた 6 つの突然変異戦略について，*rand/2*，*current-to-rand/1*，*rand/1*，*rand-to-best/1*，*current-to-best/1*，*best/1* の順に，局所探索の度合いが高くなっていくことが報告されている．

Algorithm 2 exponential 交叉（exponential crossover）

$j = randint\ [1, D]$
$k = 1$
$\boldsymbol{u}_i \leftarrow \boldsymbol{x}_i$
repeat
  $u_{i,j} = v_{i,j}$
  $j = (j + 1) \bmod D$
  $k = k + 1$
until $rand\ [0, 1) \geq CR$ **or** $k \geq D$

2.2.3 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（2.2）で与えられる．

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand[0, 1] \leq CR \text{ or } j = j_{rand}, \\ x_{i,j} & \text{otherwise.} \end{cases} \tag{2.2}$$

ここで，交叉率 $CR \in [0, 1]$ は，突然変異個体の決定変数の寄与度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．$rand[0, 1]$ は，$[0, 1]$ の範囲から実数の一様分布乱数を返す擬似関数を表す．また，exponential 交叉は Algorithm 2 で与えられる．$randint[1, D]$ は，$[1, D]$ の範囲から整数の一様分布乱数を返す擬似関数を表す．

CR について，例えば，$CR = 0.0$ では，式（2.2）の j_{rand} あるいは Algorithm 2 の初回ループの影響により，親個体に対し決定変数は 1 つのみ変更されるので，1 次元ごとの垂直あるいは水平方向の探索が可能となる．一方，$CR = 1.0$ では垂直あるいは水平方向の探索ではなくなるものの，軸変換に対する回転不変性を持つため，変数分離不可能な関数に適した設定と言える [56].

2.2.4 選択

式（2.3）に示す通り，$\boldsymbol{u}_i$ の評価値が $\boldsymbol{x}_i$ の評価値以下である場合にのみ $\boldsymbol{x}_i$ を $\boldsymbol{u}_i$ に設定する．そして，突然変異（2.2.2 節）に戻り，一連のプロセスを探索終了条件（例：最大

世代数, 最大評価回数）まで繰り返す.

$$\boldsymbol{x}_i \leftarrow \begin{cases} \boldsymbol{u}_i & \text{if } f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i), \\ \boldsymbol{x}_i & \text{otherwise.} \end{cases} \tag{2.3}$$

第 3 章

適応 DE とその分類

本章では，まず適応 DE を一般化した擬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適応 DE に限定せず議論するためである．その後，ハイパーパラメータの生成方法の観点から適応 DE を 2 種類に大別する．また，提案手法を組み込む jDE と SaDE，JADE のメカニズムについても一般化した擬似アルゴリズムに従いながら説明する．なお，jDE や SaDE，JADE のアルゴリズム表記は原著と異なるが，その内容自体は変わらないことに留意されたい．また，DERSF，DETVSF [57] や，sinDE [58] などの決定的なスケジューリングを用いた調整は，適応的な調整とは異なるため本論文の対象としない．また，DE のハイパーパラメータには F, CR に加えて解集合サイズ N が存在するが，解生成プロセスに直接寄与する F と CR に着目する．

3.1 適応 DE の一般化アルゴリズム

各個体 $\boldsymbol{x}_i$ に割り当てるハイパーパラメータを $\boldsymbol{\theta}_i = [\theta_{v,i}, \theta_{u,i}, \theta_{F,i}, \theta_{CR,i}]$ 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カル型変数．例えば，$\theta_v = 1, 2, 3$ がそれぞれ *rand/1*, *best/2*, *current-to-rand/1* に対応すると定義できる．突然変異戦略を適応しない場合は，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カル型変数．例えば，$\theta_u = 1, 2$ がそれぞれ binomial 交叉と exponential 交叉に対応すると定義で

Algorithm 3 一般化した適応 DE（generalized self-adaptive DE）

$t = 0$
Initialize $\mathcal{P} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$
Initialize $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N\}$
while *termination criteria* are not met **do**
  $t = t + 1$
  for $i = 1$ **to** N **do**
    $\boldsymbol{\theta}_i^{t-1} \leftarrow \boldsymbol{\theta}_i$
    $\boldsymbol{\theta}_i = [\theta_{v,i}, \theta_{u,i}, \theta_{F,i}, \theta_{CR,i}] \leftarrow \mathit{Sample}(\boldsymbol{\theta}_i^{t-1})$
    $\boldsymbol{v}_i \leftarrow \mathit{Mutation}(\mathcal{P}, \theta_{v,i}, \theta_{F,i})$
    $\boldsymbol{u}_i \leftarrow \mathit{Crossover}(\boldsymbol{x}_i, \boldsymbol{v}_i, \theta_{u,i}, \theta_{CR,i})$
  end for
  for $i = 1$ **to** N **do**
    $\boldsymbol{x}_i \leftarrow \begin{cases} \boldsymbol{u}_i & \text{if } f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i) \\ \boldsymbol{x}_i & \text{otherwise} \end{cases}$
    $\boldsymbol{\theta}_i \leftarrow \mathit{Update}\ (\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{t-1})$
  end for
end while

きる．交叉戦略を適応しない場合は，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．スケール係数を適応しない場合は，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． 交叉率を適応しない場合は，θ_{CR} を規定値に固定する．

次に，適応 DE のアルゴリズムを一般化した擬似コードを Algorithm 3 に示す．最初に解集合 $\mathcal{P} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ とハイパーパラメータ $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N\}$ を初期化する．次にメインループとして，個体 $\boldsymbol{x}_i$ ごとに擬似関数 $\mathit{Sample}(\boldsymbol{\theta}_i^{t-1})$ でハイパーパラメータ $\boldsymbol{\theta}_i$ 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，適応 DE ごとのハイパーパラメータの生成方法に依存するため擬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，擬似関数 $\mathit{Update}\ (\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{t-1})$ 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，適応

Algorithm 4 *Sample-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 $\boldsymbol{\theta}_i$

DE ごとに異なる．なお，*Sample*($\boldsymbol{\theta}_i^{t-1}$) と *Update*($\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{t-1}$) で表記した引数の種類は，適応 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 ランダム生成による再利用モデル

このモデルでは，調整するハイパーパラメータ $\boldsymbol{\theta}_i$ をランダムに選択するため，試行錯誤的な要素が強い方法となる．また，解評価によってその個体を生成したハイパーパラメータ $\boldsymbol{\theta}_i$ が好適であると判断された場合には，$\boldsymbol{\theta}_i$ を次世代で再利用する形で継承する点に特徴がある．

jDE [38] は解更新に成功（$f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i)$）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．これは，GA や PSO の適応モデルを流用する傾向が強かった最初期の研究と比較し [59, 60, 61]，DE に特化した方法として位置づけられる [20]．jDE における *Sample*($\boldsymbol{\theta}_i^{t-1}$) を *Sample-jDE*($\boldsymbol{\theta}_i^{t-1}$) として Algorithm 4 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，個体ごとのハイパーパラメータ $\boldsymbol{\theta}_i$ 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 4 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，*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（$f(\boldsymbol{u}_i^t) > f(\boldsymbol{x}_i^t)$）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．この τ_F, τ_{CR} を動的に変化させた改良手法として FDSADE [62] や ISADE [63] が存在する．

他の例として，EPSDE [64] は，$\theta_{F,i}$ と $\theta_{CR,i}$ に加えて突然変異戦略 $\theta_{v,i}$ も適応的に

調整する．それぞれの選択肢を格納したプールを $P_F = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, $P_{CR} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, $P_v = \{rand/1, best/2, current\text{-}to\text{-}rand/1\}$ として事前に定義する．初期化プロセスにおいては，それぞれのプールから $\theta_{F,i}$, $\theta_{CR,i}$, $\theta_{v,i}$ を一つずつ一様分布で選択し，個体 $\boldsymbol{x}_i$ に割り当てる．その後の世代更新では前世代で解更新に成功した場合はそれらを継承し，失敗した場合にのみ $\theta_{F,i}$,$\theta_{CR,i}$, $\theta_{v,i}$ の全てを一様分布でランダムに割り当てる．なお，交叉戦略 $\theta_{u,i}$ は binomial 交叉である．比較的単純なアルゴリズムでありながら，優れた最適化性能を導出することが報告されている [20].

戦略まで適応対象とする，別のランダム生成モデルとして，CoDE [39] が挙げられる．CoDE では，事前に定義された $\theta_{F,i}$ と $\theta_{CR,i}$ のプールには $\{F, CR\} \in \{1.0, 0.1\}, \{1.0, 0.9\}, \{0.8, 0.2\}$ の 3 つの組が，突然変異戦略 $\theta_{v,i}$ と交叉戦略 $\theta_{u,i}$ を合わせた戦略プールには $\{rand/1/bin, rand/2/bin, current - to - rand/1\}$ が格納されている．各世代の各個体について，戦略プールの各要素に対しランダムに選出されたパラメータプールの組が割り当てられ 3 組のハイパーパラメータが生成される．これらを通して作られた 3 つの子個体が解評価された後，親個体との 4 個体の中から評価値が最小の個体が次世代に引き継がれる．ここでは各プールの要素は探索を通して固定であり，解更新に成功したハイパーパラメータの再利用はなく，各プールからのサンプリングは，毎世代各個体で行われる．

3.3 確率分布に基づく逐次更新モデル

このモデルでは，正規分布やコーシー分布といった，一様分布以外の確率分布からサンプリングすることで，ハイパーパラメータ $\boldsymbol{\theta}_i$ を適応的に調整する．この方法の特徴は，例えば過去に解更新に成功した情報から確率分布のメタパラメータを調整するなど，好適なハイパーパラメータを獲得するために世代を追って逐次的に確率分布を調整する点にある．過去のハイパーパラメータを再利用する 3.2 節のモデルとは異なり，毎世代調整される確率分布から逐次的にハイパーパラメータをサンプリングする点も特徴である．

JADE [40] では過去に解更新に成功した情報から有用と予想される範囲のハイパーパラメータを集中的にサンプリングする．まず，$\theta_{F,i}$ と $\theta_{CR,i}$ の値を生成するコーシー分布 C と正規分布 $\mathcal{N}$ の位置パラメータと平均値として，μ_F と μ_{CR} のメタパラメータを定義する．これらの値は毎回の世代更新終了時に解更新に成功した $\theta_{F,i}$ と $\theta_{CR,i}$ の値に基づいて式（3.1）（3.2）のように変更される．

$$\mu_F = (1 - c)\mu_F + c \cdot mean_L(S_F), \tag{3.1}$$

$$\mu_{CR} = (1 - c)\mu_{CR} + c \cdot mean(S_{CR}). \tag{3.2}$$

Algorithm 5 *Sample-JADE*($\boldsymbol{\theta}_i^{t-1}$)

repeat
  $\theta_{F,i} = C(\mu_F, 0.1)$
  $\theta_{F,i} = 1$ if $\theta_{F,i} > 1$
until $0 \leq \theta_{F,i} \leq 1$
$\theta_{CR,i} = \mathcal{N}(\mu_{CR}, 0.1)$
$\theta_{CR,i} = \begin{cases} 0 & \text{if } \theta_{CR,i} < 0 \\ 1 & \text{if } \theta_{CR,i} > 0 \end{cases}$
$\theta_{v,i} = 1$ (*current-to-pbest/1*)
$\theta_{u,i} = 1$ (*binomial crossover*)
return $\boldsymbol{\theta}_i$

ここで，c は学習率であり，S_F と S_{CR} はその世代で解更新に成功した $\theta_{F,i}$ と $\theta_{CR,i}$ を格納する集合をそれぞれ示す．また，$mean(S_{CR})$ は S_{CR} 内の全ての値の平均値であり，$mean_L(S_F)$ は S_F 内の全ての値の 2 階のレーマー平均 $\left(\sum_{\theta_{F,i} \in S_F} \theta_{F,i}^2 / \sum_{\theta_{F,i} \in S_F} \theta_{F,i}\right)$ である．Algorithm 5 に示す通り，各個体における $\theta_{F,i}$ と $\theta_{CR,i}$ のサンプリングは，それぞれ逐次的に調整された確率分布 $C(\mu_F, 0.1)$, $\mathcal{N}(\mu_{CR}, 0.1)$ より行われる．また，交叉戦略 $\theta_{u,i}$ は binomial 交叉であるが，JADE では新たな突然変異戦略 $\theta_{v,i}$ として *current-to-pbest/1* が用いられている．

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i) + F(\boldsymbol{x}_{r_1} - \tilde{\boldsymbol{x}}_{r_2}). \tag{3.3}$$

ここで，$\boldsymbol{x}_{pbest}$ は $\mathcal{P}$ 内で評価値が $p_i \times N$ 位 ($p_i \in [p_{\min}, p_{\max}]$) の個体からランダムに選ばれた個体であり，$\tilde{\boldsymbol{x}}_{r_2}$ は，予め用意したアーカイブ $\mathcal{A}$ からランダムに選ばれた個体である．$\mathcal{A}$ には解更新に成功した際の親個体 $\boldsymbol{x}_i$ が毎回保存される．なお，$\mathcal{A}$ のサイズが $\mathcal{P}$ のサイズと一致するように，世代更新の最後に $\mathcal{A}$ 内の個体がランダムに削除される．メタパラメータ μ_F，μ_{CR} とアーカイブ $\mathcal{A}$ の調整をまとめた擬似コードを Algorithm 6 に示す．JADE のコンセプトは，MDE_pBX [65] や SLADE [66]，後述の多くの手法の発展に大きな影響を与えた．

また，JADE を改良した手法として SHADE [43] がある．SHADE では，解更新に成功した際のハイパーパラメータ $\theta_{F,i}$ と $\theta_{CR,i}$ の 2 階のレーマー平均を世代毎に計算し，それぞれ H 個のメモリ $M_{F,h}$ と $M_{CR,h}$ にカウンタ $h = 1, 2, \ldots, H$ を巡回させながらその値を格納する．ハイパーパラメータ $\theta_{F,i}$ と $\theta_{CR,i}$ のサンプリングに，それぞれコーシー分布と正規分布を用いる点は JADE と同じである．一方で，$[1, H]$ よりランダムに選ばれた整数 r_i を用いて，M_{F,r_i} と M_{F,r_i} をそれぞれ位置パラメータと平均値として，これらの確率分布の調整が行われる．これにより，探索が経過するにつれて $M_{F,h}$ と $M_{CR,h}$ は

Algorithm 6 JADE のパラメータ調整 世代更新後

（adaptation of JADE parameters after generation process）

for $i = 1$ **to** N **do**
  if $f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i)$ **then**
    $S_F \leftarrow$ Add $\theta_{F,i}$, $S_{CR} \leftarrow$ Add $\theta_{CR,i}$, $\mathcal{A} \leftarrow$ Add $\boldsymbol{x}_i$
  end if
end for
$\mu_F, \mu_{CR} \leftarrow$ eq. (3.1), (3.2)
$S_F = \emptyset, S_{CR} = \emptyset$
Randomly remove $\boldsymbol{x} \in \mathcal{A}$ until $|\mathcal{A}| \leq |\mathcal{P}|$

Algorithm 7 *Sample-SaDE*($\boldsymbol{\theta}_i^{t-1}$)

$\theta_{v,i}, \theta_{u,i} \leftarrow$ Sample a pair of strategies from distribution p
$\theta_{F,i} = \mathcal{N}(0.5, 0.3)$
repeat
  $\theta_{CR,i} = \mathcal{N}(CRm_k, 0.1)$
until $0 \leq \theta_{CR,i} \leq 1$
return $\boldsymbol{\theta}_i$

多様性を保持しながら徐々に問題に適していき，ハイパーパラメータをサンプリングする確率分布も対応して調整されていく．なお，*current-to-pbest/1* における $\boldsymbol{x}_{pbest}$ は評価値が $p_i \times N$ 位 $(2/N \leq p_i \leq 0.2)$ の個体に JADE の $\boldsymbol{x}_{pbest}$ から変更され，これは評価値が上位 2 割の個体のうちランダムに選ばれることを意味する．改良手法としては，解集合サイズを線形的に減少させることで大域探索から局所探索へ徐々にシフトする L-SHADE [67] やこれに重みなし再帰ニューラルネットワークのようなニューロダイナミクスを加えた L-SHADE-ND [68]，L-SHADE にガウス分布を用いたローカルサーチと sinDE を参考にしたハイパーパラメータ調整を加えた LSHADE-Epsin [69] やこれにニッチングベースの解集合サイズ削減を加えた EsDE$_r$-NR [70]，L-SHADE において事前にスケジューリングされたルールで探索フェーズに応じた細かい調整を行う iL-SHADE [71] や jSO [44] 等が存在する．

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に SaDE [41, 42] がある．SaDE における *Sample*($\boldsymbol{\theta}_i^{t-1}$) を *Sample-SaDE*($\boldsymbol{\theta}_i^{t-1}$) として Algorithm 7 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いて調整される．また，戦略は突然変異・交

Algorithm 8 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**
    $p_{k,t} \leftarrow$ eq. (3.4)
    Remove $ns_{k,t-LP}, nf_{k,t-LP}$
    $CRm_k = median(CR_{Memory_k})$
  end for
end if

Algorithm 9 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 $f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i)$ **then**
    $ns_{k,t} += 1$, $CR_{Memory_k} \leftarrow$ Add $\theta_{CR,i}$
  else
    $nf_{k,t} += 1$
  end if
end for

又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 9 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 8, 式（3.5）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（3.4）で生成する．ここで式（3.5）の ϵ は，全ての $S_{k,t}$ の第一項が 0 であるときに式（3.4）で 0 での除算を回避するための定数である．

$$p_{k,t} = \frac{S_{k,t}}{\sum_{k=1}^{K} S_{k,t}}, \tag{3.4}$$

$$S_{k,t} = \frac{\sum_{g=t-LP}^{t-1} ns_{k,g}}{\sum_{g=t-LP}^{t-1} ns_{k,g} + \sum_{g=t-LP}^{t-1} nf_{k,g}} + \epsilon. \tag{3.5}$$

次に，Algorithm 7 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．なお，SaDE の改良手法としての SaNSDE [45] では，$\theta_{F,i}$ についても適応的に調整している．他の改良手法として，教育における教師からの指導（Teaching）と学習者相互の学習（Learning）を模擬した多点探索アルゴリズムを導入した TLBSaDE [72] や，GPU を活用したアルゴリズムである cuSaDE [73] などが提案されている．

3.4 適応 DE の分類のまとめ

本章のまとめとして，以上で紹介した適応 DE を，

行　ハイパーパラメータの生成方法
列　適応対象の違い

の観点で分類して，表 3.1 に示す．

表 3.1　適応 DE の分類

	適応対象	
	F と CR	F，CR，突然変異戦略，交叉戦略
ランダム生成による再利用モデル	jDE [38]，FDSADE [62]，ISADE [63]	EPSDE [64]，MPEDE [27]（CoDE [39] ※再利用はしない）
確率分布に基づく逐次更新モデル	JADE [40]，MDE$_{_p}$BX [65]，SLADE [66]，SHADE [43]，L-SHADE [67]，L-SHADE-ND [68] ，LSHADE-Epsin [69]，iL-SHADE [71]，jSO [44]，EsDE$_r$-NR [70]	SaDE [41, 42]，SaNSDE [45]，TLBSaDE [72]，CoBiDE [74]，cuSaDE [73]

第 4 章

提案手法

jDE や SaDE，JADE をはじめとする適応 DE を対象とし，一般化された適応 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 概要

適応 DE は，解更新に成功した場合にハイパーパラメータを再利用する方法や逐次的にサンプリング元の確率分布を調整する方法を用いることで，好適なハイパーパラメータを生成する改良が成されている．一方で，生成するハイパーパラメータは乱数に依存するため，乱数によっては性能改善に寄与しないアルゴリズムが生成される可能性もある．そこで，適応 DE は生成されたハイパーパラメータをそのまま使用する点では共通の方法を採用していることから，試行錯誤しながら調整する傾向が強くなる．しかしながら，少ない解評価回数を想定した場合は，この試行錯誤的な調整を可能な限り排除することが，最適化性能を改善する上で重要と考えられる．

そこで提案手法は，ハイパーパラメータ候補を複数生成し，その中から好適と考えられるハイパーパラメータを選択することで，試行錯誤的な調整要素を緩和することを狙う．一般的な適応 DE では，1 つの個体につき 1 つのハイパーパラメータのみをサンプリングする．一方で，提案手法を組み込むと，1 つの個体につきハイパーパラメータを一度に複数生成することになる．そして，これらのハイパーパラメータ候補から生成した仮子個体と，既に発見された優良解とのユークリッド距離をそれぞれ計算し，これが最小となるハイパーパラメータを出力する．つまり，優良解に最も近づいた仮子個体を生成できたハイパーパラメータを好適と定義する．これは，第 1 章で述べたように，優良解の周辺領域を局所探索する傾向を高めたハイパーパラメータによって，収束速度を改善することを意図

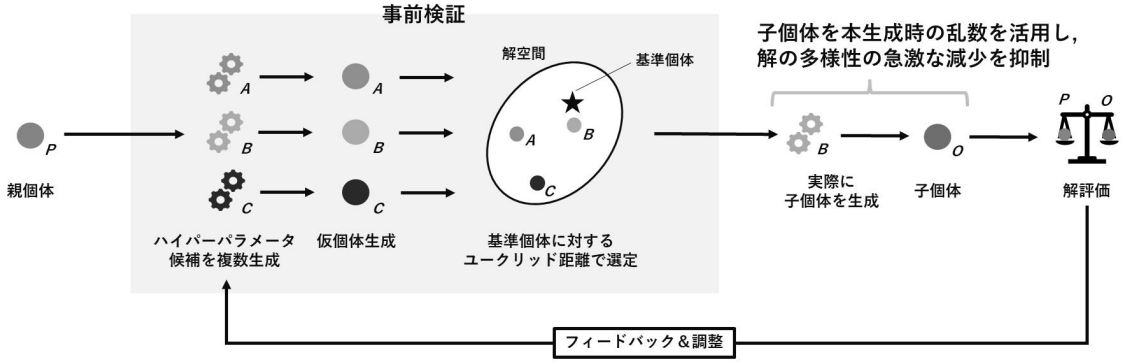


図 4.1　事前検証の概念図

している．加えて，早期収束を抑制する工夫として，出力されたハイパーパラメータを用いて子個体を改めて生成する．これは，局所探索を傾向を高めながらも，解の多様性の急速な低下を突然変異や交叉の非決定性によって抑制することを意図する．

これらをまとめた事前検証の概念図を，図 4.1 に示す．なお，ハイパーパラメータ候補の個数は簡単のために 3 としているが，この個数は，以降で提案手法自体のハイパーパラメータとして再定義する．

なお，著者の分析から，基準個体と仮個体のユークリッド距離と採用されるハイパーパラメータの間には明確な相関が確認できていない．この理由としては，基準個体に近い個体を生成するハイパーパラメータは複数種類存在すること，仮個体と子個体は乱数に依存して生成されることが挙げられる．したがって，提案手法は，基準個体に近い仮個体を生成したハイパーパラメータを用いるという簡略的な検証方法を採用し，個体ごとに独立してハイパーパラメータを調整する．

4.2　メカニズム

4.2.1　基本的な流れ

提案手法は，適応 DE が 1 つの個体に割り当てるハイパーパラメータを生成するプロセスに適用される．すなわち，一般化された適応 DE フレームワーク（Algorithm 3）において，ハイパーパラメータを生成する $Sample(\boldsymbol{\theta}_i^{t-1})$ 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 $\boldsymbol{x}_i$ に割り当てるハイパーパラメータ $\boldsymbol{\theta}_i$ 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項で述べる．そして，C 個のハイパーパラメータ候補 $\boldsymbol{\theta}'_{i,j} \in \Theta'_i$ を生成する（$j = 1, 2, \ldots, C$）．ここで，Θ'_i はハイパーパラメータ候補の集合とする．生成するハイパーパラメータ候補は，利用する適応 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，JADE では *Sample-JADE*($\boldsymbol{\theta}_i^{t-1}$) をそれぞれ C 回だけ繰り返して実行する．また，$C \in \mathbb{N}$ は提案手法で用いるハイパーパラメータである．

次に，適応 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ハイパーパラメータ候補を $\boldsymbol{\theta}_i$ と設定する．つまり，$\boldsymbol{\theta}_i$ は以下の式で得られる．

$$\boldsymbol{\theta}_i = \underset{\boldsymbol{\theta}'_{i,j} \in \Theta'_i}{\arg\min} ||\boldsymbol{x}_i^* - \boldsymbol{u}'_{\boldsymbol{\theta}'_{i,j}}||. \tag{4.1}$$

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，$\boldsymbol{\theta}_i$ を用いて適応 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．

4.2.2 基準個体の定義

優良解の周辺に対する局所探索を促すために，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，以下の 4 つの戦略を考える．

***rand* 戦略**　現在の解集合 $\mathcal{P}$ からランダムに選択し $\boldsymbol{x}_i^*$ に設定する．

***greedy* 戦略**　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．

***p-best* 戦略**　パラメータ $p \in [0, 1]$ を用いて評価値が上位 $p \times N$ 位までの個体からランダムに選択し $\boldsymbol{x}_i^*$ に設定する．

***ϵ-greedy* 戦略**　閾値 $\epsilon \in [0, 1]$ を設け，確率 ϵ で *rand* 戦略を，確率 $1 - \epsilon$ で *greedy* 戦略を適用し $\boldsymbol{x}_i^*$ に設定する．

なお，*p-best* 戦略は，突然変異戦略 *current-to-pbest* を参考にしている．*rand* 戦略は，より解の多様性を重視した傾向にあるといえる．*greedy* 戦略では，全ての個体に同じ基準個体が設定され，収束速度を改善する可能性がある．これらに対し，パラメータ p や ϵ の値

に依存するが，*p-best* 戦略と ϵ-*greedy* 戦略は，より保守的な戦略である．

これらの戦略を用いた性能比較実験は第 7 章で行うが，単純かつ決定的な操作である *greedy* 戦略を既定の戦略として設定する．*greedy* 戦略を用いた提案手法が jDE と SaDE, JADE の性能を安定して改善するならば，追加のパラメータ p や ϵ を設定する必要がなくなる．この場合，全てのハイパーパラメータ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ハイパーパラメータを生成することは意図していない．あくまでも適応 DE の生成方法を踏襲しながら 1 つのハイパーパラメータを選択するものであり，*greedy* 戦略を用いることが，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

4.2.1 項に示したように，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 個のハイパーパラメータ候補を生成する繰り返し処理に時間がかかる．CEP 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（$f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i)$）個体は，現在のハイパーパラメータ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（$f(\boldsymbol{u}_i) > f(\boldsymbol{x}_i)$）個体にのみ，ハイパーパラメータ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，擬似関数 $\mathit{Sample}(\mathcal{P}, \boldsymbol{\theta}_i, \boldsymbol{x}_i^*)$ を Algorithm 10 に示す．

Algorithm 10 *Sample*$(\mathcal{P}, \boldsymbol{\theta}_i, \boldsymbol{x}_i^*)$

Require: $\mathcal{P}, \boldsymbol{\theta}_i, \boldsymbol{x}_i^*$
  if $f(\boldsymbol{u}_i^{t-1}) > f(\boldsymbol{x}_i^{t-1})$ **then**
    $\Theta_i' = \emptyset$
    for $j = 1$ **to** C **do**
      $\Theta_i' \leftarrow$ Add sampled configuration $\boldsymbol{\theta}_{i,j}'$
      $\boldsymbol{v}_{\boldsymbol{\theta}_{i,j}'}' \leftarrow$ *Mutation*$(\mathcal{P}, \theta_{v,i,j}', \theta_{F,i,j}')$
      $\boldsymbol{u}_{\boldsymbol{\theta}_{i,j}'}' \leftarrow$ *Crossover*$(\boldsymbol{x}_i, \boldsymbol{v}_{\boldsymbol{\theta}_{i,j}'}', \theta_{u,i,j}', \theta_{CR,i,j}')$
    end for
    $\boldsymbol{\theta}_i \leftarrow \underset{\boldsymbol{\theta}_{i,j}' \in \Theta_i'}{\arg\min} ||\boldsymbol{x}_i^* - \boldsymbol{u}_{\boldsymbol{\theta}_{i,j}'}'||$
  end if
  return $\boldsymbol{\theta}_i$

第 5 章

問題設定

本章では，提案手法の有効性を評価するために実験で用いる問題について説明する．

5.1 ベンチマーク問題

5.1.1 概要

本論文では，制約なし（境界制約つき）単一目的連続実数値最適化のベンチマークセットである，IEEE CEC2013 real-parameter single-objective benchmark function suite [47] を用いる．進化計算分野の代表的な国際会議である ACM The Genetic and Evolutionary Computation Conference（ACM GECCO）や，IEEE Congress on Evolutionary Computation（IEEE CEC）では，Special Session 内でアルゴリズムの性能を競うコンペティションが，問題ドメイン毎に毎年開催されている．本ベンチマークセットは，IEEE CEC 2013 で提案・使用されたものである．他のベンチマークセットと同様に，セット内の各ベンチマーク関数は異なる問題性質を持つため，本ベンチマークセットを用いることで，アルゴリズムの性能を総合的に評価できる可能性が期待されている．

本ベンチマークセットのソースコードは，2022 年 3 月 4 日現在，Web 上で入手可能である[*1]．公開されているソースコードのプログラミング言語は，C 言語，java, MATLAB であるが，本研究では，これを Python 3 に書き直したプログラムを使用する．

[*1] https://github.com/P-N-Suganthan/CEC2013

5.1.2 定義

■準備 表記を簡潔にするために，各関数で共通して使用される予備関数を以下に示す．

シフト済の大域的最適解

$$\boldsymbol{o} = [o_1, o_2, \ldots, o_D]^T \in [-80, 80]^D$$

グラムシュミットの正規直交化法用の回転行列 $D \times D$

$$\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{10}$$

対角行列 Λ^α の要素 λ_{ii}

$$\lambda_{ii} = \alpha^{\frac{i-1}{2(D-1)}}, i \in \{1, 2, \ldots, D\}$$

解空間を非対称にする変換

$$T_{\text{asy}}^{\beta} \ : \ \text{if } x_i > 0, x_i = x_i^{1+\beta \frac{i-1}{D-1}\sqrt{x_i}}, i \in \{1, 2, \ldots, D\}$$

局所的な不規則性を与える変換

$$T_{\text{osz}} : x_i = \text{sign}\,(x_i) \exp\left(\hat{x}_i + 0.049\left(\sin\left(c_1 \hat{x}_i\right) + \sin\left(c_2 \hat{x}_i\right)\right)\right), i \in \{1, 2, \ldots, D\}$$

$$\text{where } \hat{x}_i = \begin{cases} \log\left(|x_i|\right) & \text{if } x_i \neq 0 \\ 0 & \text{otherwise} \end{cases}, \text{sign}\,(x_i) = \begin{cases} -1 & \text{if } x_i < 0 \\ 0 & \text{if } x_i = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$c_1 = \begin{cases} 10 & \text{if } x_i > 0 \\ 5.5 & \text{otherwise} \end{cases}, \text{and } c_2 = \begin{cases} 7.9 & \text{if } x_i > 0 \\ 3.1 & \text{otherwise} \end{cases}$$

■各関数の詳細 ベンチマーク関数の分類を表 5.1 に示す．ベンチマークセットは 28 個の関数 F1〜F28 で構成され，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．表中の n は，合成関数における合成する関数の個数を表す．なお，全ての関数は最小値問題とし，問題次元数 D に対し，その評価値 $f(\boldsymbol{x})$ を最小化する解（大域的最適解）$\boldsymbol{x}^* = [x_1, x_2, \cdots, x_D]$ を発見することが目的となる．各関数は，解を入力すると，以下の各式における $f(\boldsymbol{x})$（最適解では $f(\boldsymbol{x}^*) = 0$ となる）に，バイアスと呼ばれる $-1400 \sim 1400$ の関数毎に異なる定数を加えた値を真の評価値として返す．しかしながら，ベンチマークセットを用いた研究では，理解を容易にするために，最適解と解の真の評価値の差を誤差と呼び，これでアルゴリズムの性能を議論することが多い．すなわち，バイアスが相殺されることになり，最適解の評

表 5.1　ベンチマーク関数の分類

分類		関数名	バイアス
単峰性関数	F1	Sphere Function	−1400
	F2	Rotated High Conditioned Elliptic Function	−1300
	F3	Rotated Bent Cigar Function	−1200
	F4	Rotated Discus Function	−1100
	F5	Different Powers Function	−1000
多峰性関数	F6	Rotated Rosenbrock's Function	−900
	F7	Rotated Schaffers F7 Function	−800
	F8	Rotated Ackley's Function	−700
	F9	Rotated Weierstrass Function	−600
	F10	Rotated Griewank's Function	−500
	F11	Rastrigin's Function	−400
	F12	Rotated Rastrigin's Function	−300
	F13	Non-Continuous Rotated Rastrigin's Function	−200
	F14	Schwefel's Function	−100
	F15	Rotated Schwefel's Function	100
	F16	Rotated Katsuura Function	200
	F17	Lunacek Bi-Rastrigin Function	300
	F18	Rotated Lunacek Bi-Rastrigin Function	400
	F19	Rotated Expanded Griewank's plus Rosenbrock's Function	500
	F20	Rotated Expanded Scaffer's F6 Function	600
合成関数	F21	Composition Function 1 (n = 5, Rotated)	700
	F22	Composition Function 2 (n = 3, Unrotated)	800
	F23	Composition Function 3 (n = 3, Rotated)	900
	F24	Composition Function 4 (n = 3, Rotated)	1000
	F25	Composition Function 5 (n = 3, Rotated)	1100
	F26	Composition Function 6 (n = 5, Rotated)	1200
	F27	Composition Function 7 (n = 5, Rotated)	1300
	F28	Composition Function 8 (n = 5, Rotated)	1400

価値は常に 0 と見なすことができる．本論文でも，これに倣い，以降では最適解の評価値が 0 となるようにバイアスを無視し，この値を単に評価値と呼ぶ．

解 $\boldsymbol{x}$ の定義域は，全ての関数で $\boldsymbol{x} \in [-100, 100]^D$ である．このように，定義域が設定されている最適化問題を，境界制約つき最適化問題と呼ぶことがある．

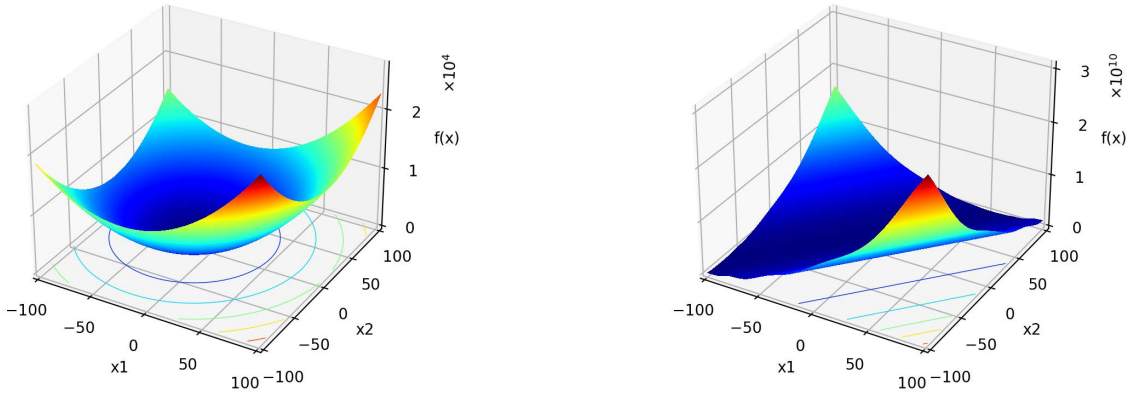
以降では，各関数の定義について述べる．また，$D = 2$ とした際の関数の概形を併せて示す．なお，最適化にあたっては，全ての問題は Black-box とされていることに注意されたい．

F1　Sphere Function

$$f_1(x) = \sum_{i=1}^{D} z_i^2, z = x - o$$

F2　Rotated High Conditioned Elliptic Function

$$f_2(x) = \sum_{i=1}^{D} \left(10^6\right)^{\frac{i-1}{D-1}} z_i^2, z = T_{\text{osz}}\left(\mathbf{M}_1(x - o)\right)$$



a) F1　　b) F2

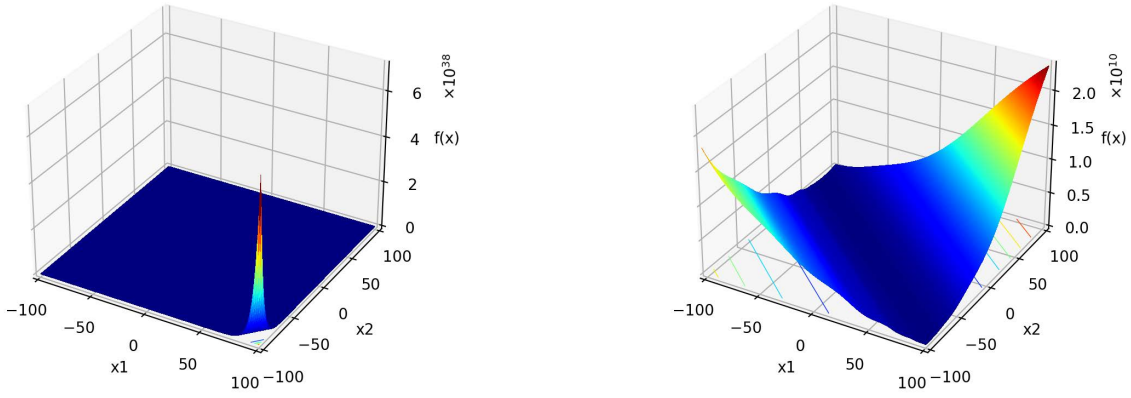
図 5.1　F1 と F2 の $D = 2$ での概形

F3　Rotated Bent Cigar Function

$$f_3(x) = z_1^2 + 10^6 \sum_{i=2}^{D} z_i^2, \quad z = \mathbf{M}_2 T_{\text{asy}}^{0.5}\left(\mathbf{M}_1(x - o)\right)$$

F4　Rotated Discus Function

$$f_4(x) = 10^6 z_1^2 + \sum_{i=2}^{D} z_i^2, \quad z = T_{\mathrm{osz}}\left(\mathbf{M}_1(x - o)\right)$$



a) F3　　b) F4

図 5.2　F3 と F4 の $D = 2$ での概形

F5　Different Powers Function

$$f_5(x) = \sqrt{\sum_{i=1}^{D} |z_i|^{2+4\frac{i-1}{D-1}}}, \quad z = x - o$$

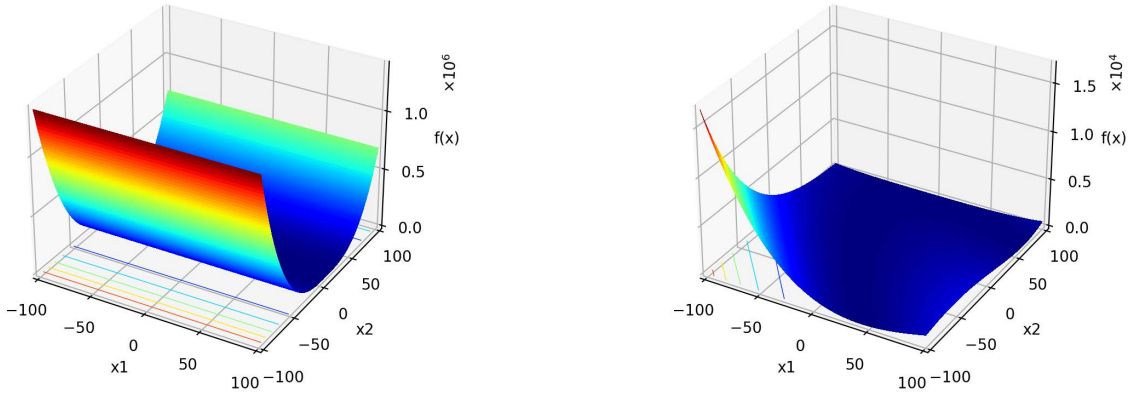
F6　Rotated Rosenbrock's Function

$$f_6(x) = \sum_{i=1}^{D-1} \left(100\left(z_i^2 - z_{i+1}\right)^2 + (z_i - 1)^2\right), \quad z = \mathbf{M}_1\left(\frac{2.048(x - o)}{100}\right) + 1$$

F7　Rotated Schaffers F7 Function

$$f_7(\mathbf{x}) = \left(\frac{1}{D-1}\sum_{i=1}^{D-1}\left(\sqrt{z_i} + \sqrt{z_i}\sin^2\left(50 z_i^{0.2}\right)\right)\right)^2$$

$$z_i = \sqrt{y_i^2 + y_{i+1}^2} \text{ for } i = 1, \ldots, D, y = \Lambda^{10}\mathbf{M}_2 T_{\mathrm{asy}}^{0.5}\left(\mathbf{M}_1(x - o)\right)$$



a) F5　　b) F6

図 5.3　F5 と F6 の $D = 2$ での概形

F8　Rotated Ackley's Function

$$f_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi z_i\right)\right) + 20 + e$$

$$z = \Lambda^{10}\mathbf{M}_2 T_{\text{asy}}^{0.5}\left(\mathbf{M}_1(x-o)\right)$$

F9　Rotated Weierstrass Function

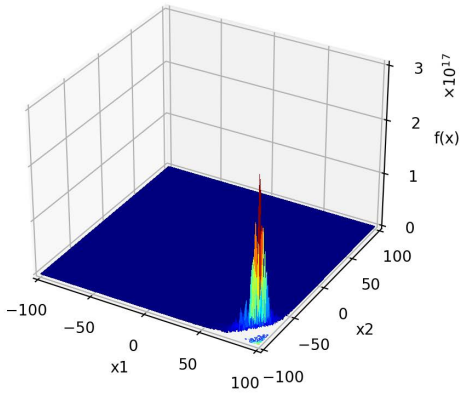
$$f_9(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{kmax}\left[a^k\cos\left(2\pi b^k\left(z_i+0.5\right)\right)\right]\right) - D\sum_{k=0}^{kmax}\left[a^k\cos\left(2\pi b^k\cdot 0.5\right)\right]$$

$$a = 0.5, b = 3, kmax = 20, \quad z = \Lambda^{10}\mathbf{M}_2 T_{\text{asy}}^{0.5}\left(\mathbf{M}_1\frac{0.5(x-o)}{100}\right)$$

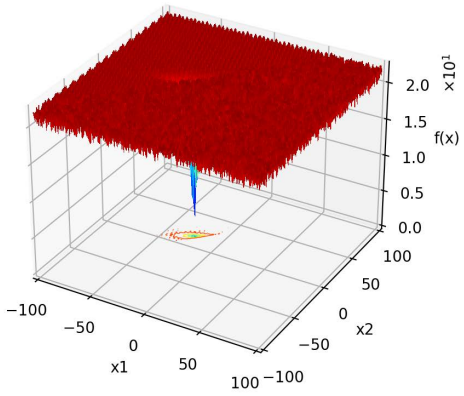
F10　Rotated Griewank's Function

$$f_{10}(x) = \sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1, \quad z = \Lambda^{100}\mathbf{M}_1\frac{600(x-o)}{100}$$

F11　Rastrigin's Function

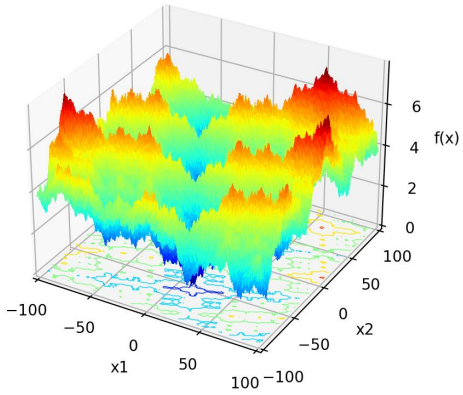


a) F7

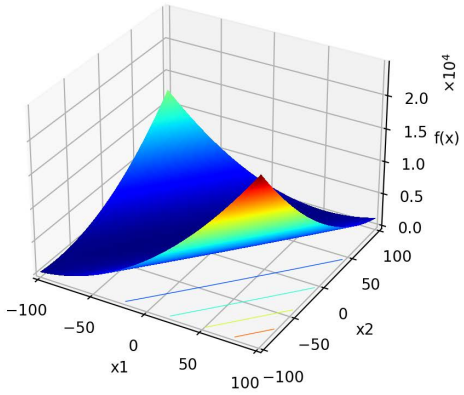


b) F8

図 5.4　F7 と F8 の $D = 2$ での概形



a) F9



b) F10

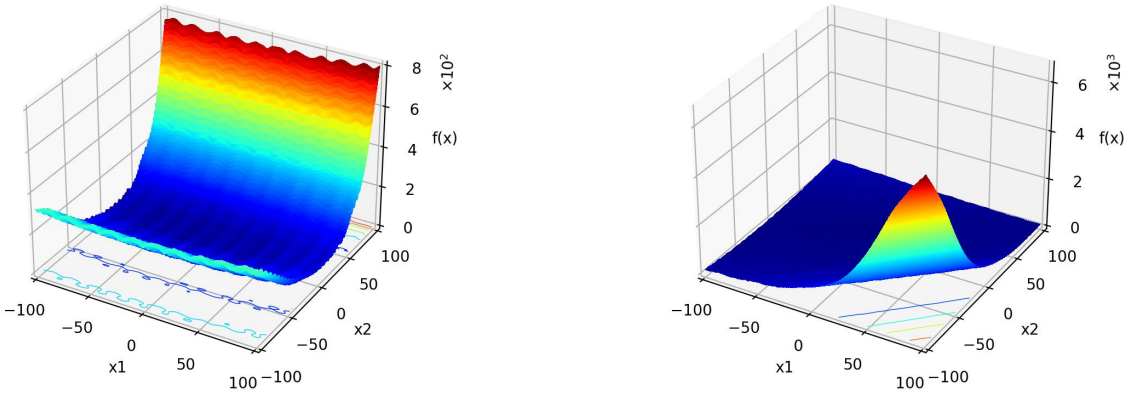
図 5.5　F9 と F10 の $D = 2$ での概形

$$f_{11}(x) = \sum_{i=1}^{D} \left(z_i^2 - 10\cos(2\pi z_i) + 10 \right)$$

$$z = \Lambda^{10} T_{\text{asy}}^{0.2} \left(T_{\text{osz}} \left(\frac{5.12(x-o)}{100} \right) \right)$$

F12　Rotated Rastrigin's Function

$$f_{12}(x) = \sum_{i=1}^{D} \left(z_i^2 - 10\cos(2\pi z_i) + 10 \right)$$
$$z = \mathbf{M}_1 \Lambda^{10} \mathbf{M}_2 T_{\text{asy}}^{0.2} \left(T_{\text{osz}} \left(\mathbf{M}_1 \frac{5.12(x-o)}{100} \right) \right)$$



a) F11　　b) F12

図 5.6　F11 と F12 の $D = 2$ での概形

F13　Non-continuous Rotated Rastrigin's Function

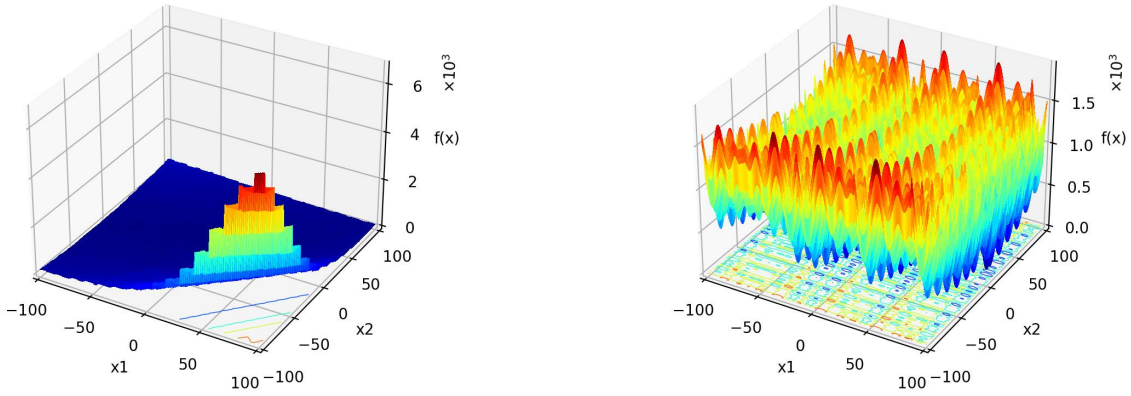
$$f_{13}(x) = \sum_{i=1}^{D} \left(z_i^2 - 10\cos(2\pi z_i) + 10 \right)$$
$$\widehat{x} = \mathbf{M}_1 \frac{5.12(x-o)}{100}, y_i = \begin{cases} \widehat{x}_i & \text{if } |\widehat{x}_i| \le 0.5 \\ \text{round}(2\widehat{x}_i)/2 & \text{if } |\widehat{x}_i| > 0.5 \end{cases} \quad \text{for } i = 1, 2, \ldots, D$$
$$z = \mathbf{M}_1 \Lambda^{10} \mathbf{M}_2 T_{\text{asy}}^{0.2} \left(T_{\text{osz}}(y) \right)$$

F14　Schwefel's Function

$$f_{14}(z) = 418.9829 \times D - \sum_{i=1}^{D} g\left(z_i\right)$$

$$z = \Lambda^{10}\left(\frac{1000(x-o)}{100}\right) + 4.209687462275036\text{e+}2$$

$$g\left(z_i\right) = \begin{cases} z_i \sin\left(|z_i|^{1/2}\right) & \text{if } |z_i| \leq 500 \\ \left(500 - \bmod\left(z_i, 500\right)\right) \sin\left(\sqrt{|500 - \bmod(z_i, 500)|}\right) - \frac{(z_i - 500)^2}{10000D} & \text{if } z_i > 500 \\ \left(\bmod\left(|z_i|, 500\right) - 500\right) \sin\left(\sqrt{|\bmod\left(|z_i|, 500\right) - 500|}\right) - \frac{(z_i + 500)^2}{10000D} & \text{if } z_i < -500 \end{cases}$$



a) F13　　b) F14

図 5.7　F13 と F14 の $D = 2$ での概形

F15　Rotated Schwefel's Function

$$f_{15}(z) = 418.9829 \times D - \sum_{i=1}^{D} g\left(z_i\right)$$

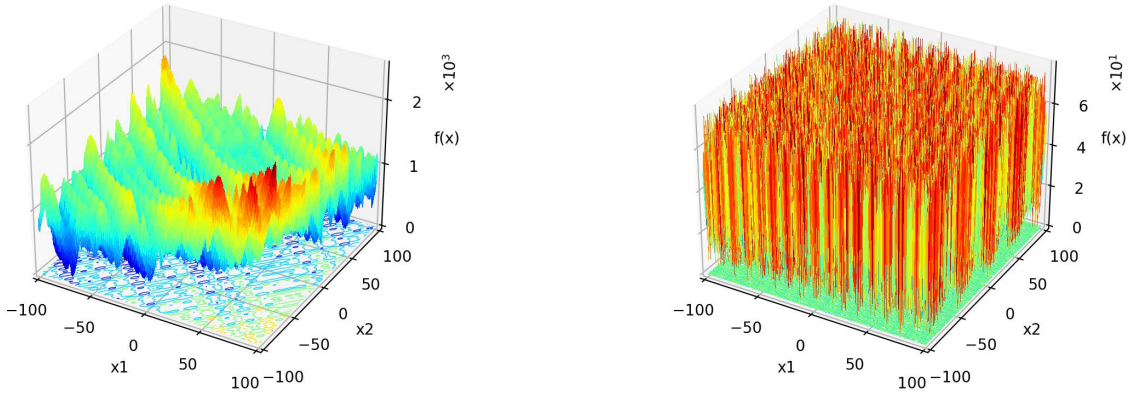
$$z = \Lambda^{10}\mathbf{M}_1\left(\frac{1000(x-o)}{100}\right) + 4.209687462275036\text{e+}2$$

$$g\left(z_i\right) = \begin{cases} z_i \sin\left(|z_i|^{1/2}\right) & \text{if } |z_i| \leq 500 \\ \left(500 - \bmod\left(z_i, 500\right)\right) \sin\left(\sqrt{|500 - \bmod(z_i, 500)|}\right) - \frac{(z_i - 500)^2}{10000D} & \text{if } z_i > 500 \\ \left(\bmod\left(|z_i|, 500\right) - 500\right) \sin\left(\sqrt{|\bmod\left(|z_i|, 500\right) - 500|}\right) - \frac{(z_i + 500)^2}{10000D} & \text{if } z_i < -500 \end{cases}$$

F16　Rotated Katsuura Function

$$f_{16}(x) = \frac{10}{D^2} \prod_{i=1}^{D} \left(1 + i \sum_{j=1}^{32} \frac{2^j z_i - \text{round}\left(2^j z_i\right) |}{2^j} \right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2}$$

$$z = \mathbf{M}_2 \Lambda^{100} \left(\mathbf{M}_1 \frac{5(x - o)}{100} \right)$$



a) F15　　b) F16

図 5.8　F15 と F16 の $D = 2$ での概形

F17　Lunacek bi-Rastrigin Function

$$f_{17}(x) = \min \left(\sum_{i=1}^{D} (\widehat{x}_i - \mu_0)^2 , dD + s \sum_{i=1}^{D} (\widehat{x}_i - \mu_1)^2 \right) + 10 \left(D - \sum_{i=1}^{D} \cos\left(2\pi \widehat{z}_i\right) \right)$$

$$\mu_0 = 2.5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D + 20} - 8.2}, d = 1$$

$$y = \frac{10(x - o)}{100}, \hat{x}_i = 2 \,\text{sign}\left(x_i^*\right) y_i + \mu_0, \text{ for } i = 1, 2, \ldots, D$$

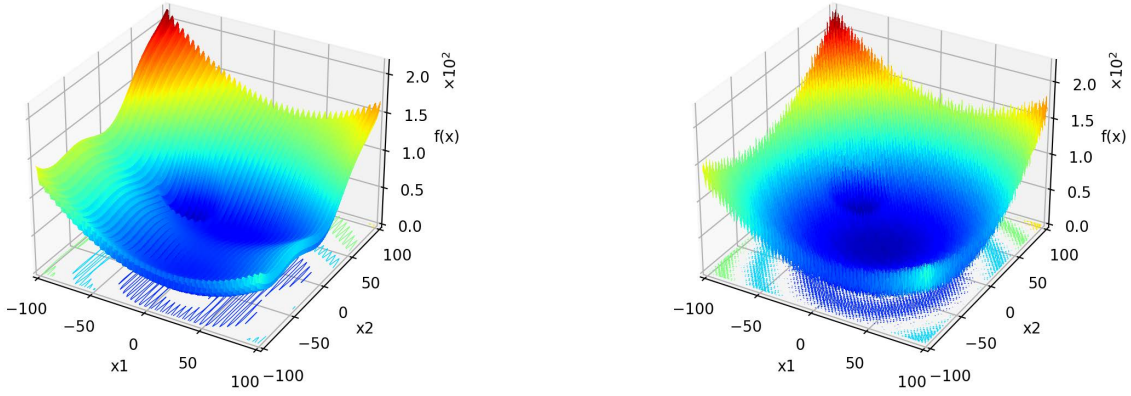
$$z = \Lambda^{100} \left(\hat{x} - \mu_0 \right)$$

F18　Rotated Lunacek bi-Rastrigin Function

$$f_{18}(x) = \min\left(\sum_{i=1}^{D}\left(\widehat{x}_i - \mu_0\right)^2, dD + s\sum_{i=1}^{D}\left(\widehat{x}_i - \mu_1\right)^2\right) + 10\left(D - \sum_{i=1}^{D}\cos\left(2\pi\widehat{z}_i\right)\right)$$

$$\mu_0 = 2.5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D+20} - 8.2}, d = 1$$

$$y = \frac{10(x-o)}{100}, \hat{x}_i = 2\,\mathrm{sign}\left(x_i^*\right) y_i + \mu_0, \text{ for } i = 1, 2, \ldots, D, z = \mathbf{M}_2 \Lambda^{100}\left(\mathbf{M}_1\left(\hat{x} - \mu_0\right)\right)$$



a) F17　　b) F18

図 5.9　F17 と F18 の $D = 2$ での概形

F19　Rotated Expanded Griewank's plus Rosenbrock's Function

Basic Griewank's Function:

$$g_1(x) = \sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Basic Rosenbrock's Function:

$$g_2(x) = \sum_{i=1}^{D-1}\left(100\left(x_i^2 - x_{i+1}\right)^2 + \left(x_i - 1\right)^2\right)$$

$$f_{19}(x) = g_1\left(g_2\left(z_1, z_2\right)\right) + g_1\left(g_2\left(z_2, z_3\right)\right) + \ldots + g_1\left(g_2\left(z_{D-1}, z_D\right)\right) + g_1\left(g_2\left(z_D, z_1\right)\right)$$

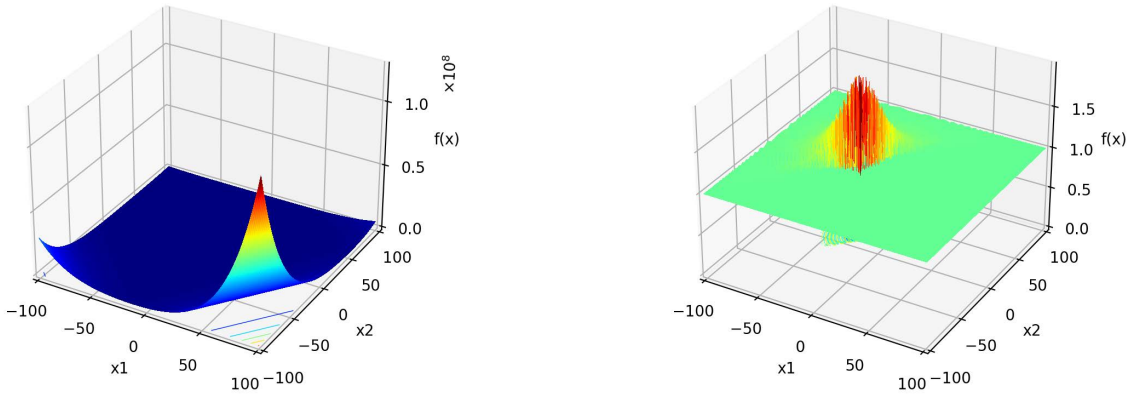
$$z = \mathbf{M}_1\left(\frac{5(x-o)}{100}\right) + 1$$

F20　Rotated Expanded Scaffer's F6 Function

$$\text{Scaffer's F6 Function: } g(x,y) = 0.5 + \frac{\left(\sin^2\left(\sqrt{x^2+y^2}\right) - 0.5\right)}{\left(1 + 0.001\left(x^2+y^2\right)\right)^2}$$

$$f_{20}(x) = g\left(z_1, z_2\right) + g\left(z_2, z_3\right) + \ldots + g\left(z_{D-1}, z_D\right) + g\left(z_D, z_1\right)$$

$$z = \mathbf{M}_2 T_{\text{asy}}^{0.5}\left(\mathbf{M}_1(x-0)\right)$$



a) F19　　b) F20

図 5.10　F19 と F20 の $D = 2$ での概形

合成関数　定義

$$f(x) = \sum_{i=1}^{n}\left\{\omega_i * \left[\lambda_i g_i(x) + bias_i\right]\right\}$$

$g_i(x)$: i 番目の合成関数を構成する基本関数

n : 基本関数の個数

o_i : 新たな最適解の位置 $g_i(x)$．これにより，大域的最適解または局所的最適解の位置が定義される．

$bias_i$: 局所的最適解のうち，大域的最適解を決定する定数.

σ_i : 各 $g_i(x)$ のカバー範囲を制御するために使用される定数．小さな σ_i の値は，$g_i(x)$ の範囲を狭くする．

λ_i : 各 $g_i(x)$ の高さを制御する定数

w_1 : 各 $g_i(x)$ の重み係数．以下のように計算される．

$$w_i = \frac{1}{\sqrt{\sum_{j=1}^{D}\left(x_j - o_{ij}\right)^2}} \exp\left(-\frac{\sum_{j=1}^{D}\left(x_j - o_{ij}\right)^2}{2D\sigma_i^2}\right)$$

続いて，$\omega_i = w_i / \sum_{i=1}^{n} w_i$ で，重みを正規化する．したがって，$x = 0$ のとき，$\omega_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}$ for $j = 1, 2, \ldots, n, f(x) = bias_1$ となる．

つまり，バイアス値が最小となる局所的最適解を大域的最適解としている．

F21 Composition Function 1

$n = 5, \quad \sigma = [10, 20, 30, 40, 50]$

$\lambda = [1, 1e\text{-}6, 1e\text{-}26, 1e\text{-}6, 0.1]$

$bias = [0, 100, 200, 300, 400]$

g_1: Rotated Rosenbrock's Function f_6

g_2: Rotated Different Powers Function f_5

g_3: Rotated Bent Cigar Function f_3

g_4: Rotated Discus Function f_4

g_5: Sphere Function f_1

F22 Composition Function 2

$n = 3$

$\sigma = [20, 20, 20]$

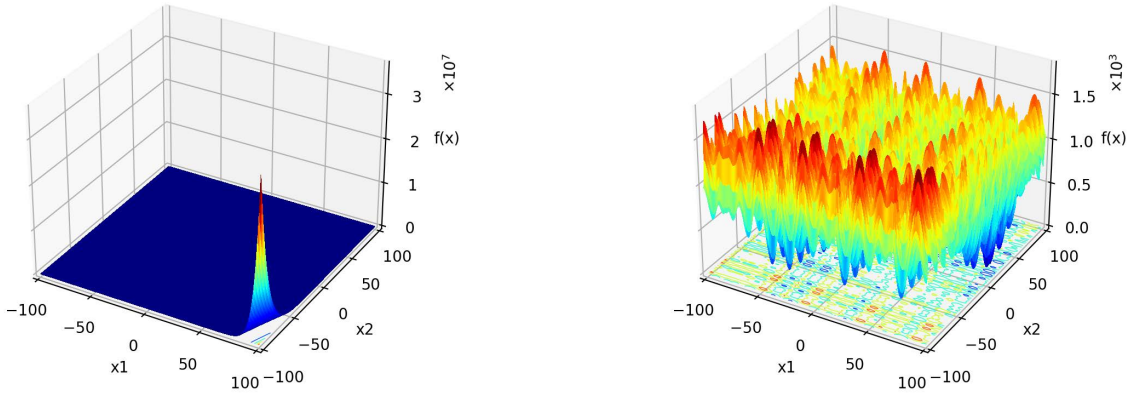
$\lambda = [1, 1, 1]$

$bias = [0, 100, 200]$

$g_{1\sim3}$: Schwefel's Function f_{14}

F23 Composition Function 3

$n = 3$



a) F21　　b) F22

図 5.11　F21 と F22 の $D = 2$ での概形

$\sigma = [20, 20, 20]$

$\lambda = [1, 1, 1]$

$bias = [0, 100, 200]$

$g_{1\sim3}$: Rotated Schwefel's Function f_{15}

F24　Composition Function 4

$n = 3$

$\sigma = [20, 20, 20]$

$\lambda = [0.25, 1, 2.5]$

$bias = [0, 100, 200]$

g_1: Rotated Schwefel's Function f_{15}

g_2: Rotated Rastrigin's Function f_{12}

g_3: Rotated Weierstrass Function f_9

F25　Composition Function 5

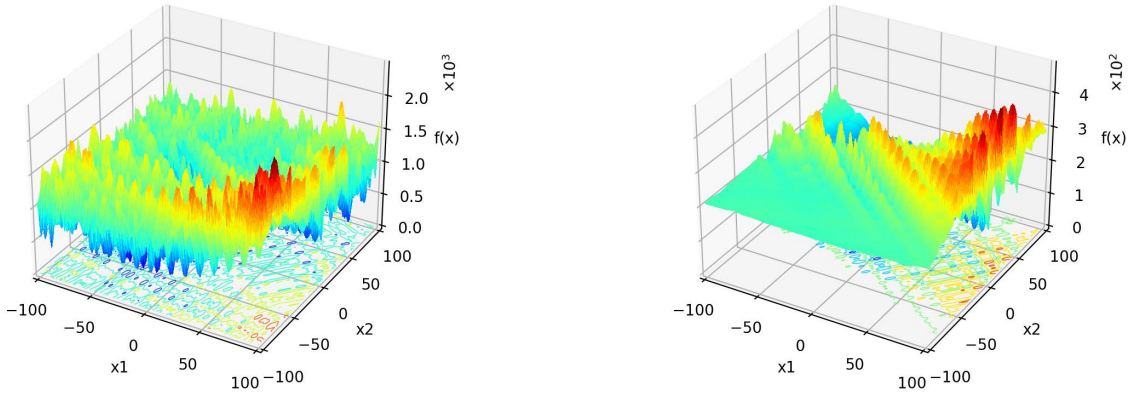
$\sigma = [10, 30, 50]$ であることを除いて，Composition Function 4 と同じ設定である．

F26　Composition Function 6

$n = 5$

$\sigma = [10, 10, 10, 10, 10]$

$\lambda = [0.25, 1, 1e\text{-}7, 2.5, 10]$



a) F23　　b) F24

図 5.12　F24 と F24 の $D = 2$ での概形

$bias = [0, 100, 200, 300, 400]$

g_1: Rotated Schwefel's Function f_{15}

g_2: Rotated Rastrigin's Function f_{12}

g_3: Rotated High Conditioned Elliptic Function f_2

g_4: Rotated Weierstrass Function f_9

g_5: Rotated Griewank's Function f_{10}

F27　Composition Function 7

$n = 5$

$\sigma = [10, 10, 10, 20, 20]$

$\lambda = [100, 10, 2.5, 25, 0.1]$

$bias = [0, 100, 200, 300, 400]$

g_1: Rotated Griewank's Function f_{10}

g_2: Rotated Rastrigin's Function f_{12}

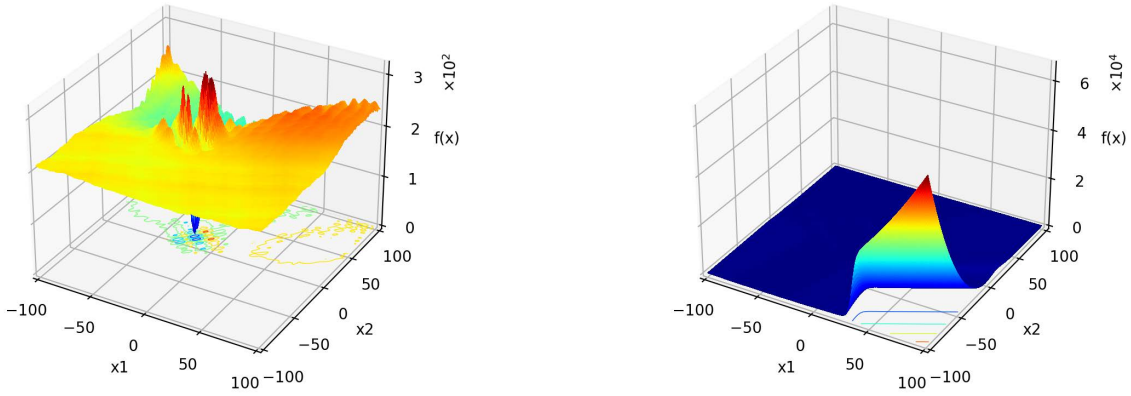
g_3: Rotated Schwefel's Function f_{15}

g_4: Rotated Weierstrass Function f_9

g_5: Sphere Function f_1

F28　Composition Function 8

$n = 5$



a) F25　　b) F26

図 5.13　F25 と F26 の $D = 2$ での概形

$\sigma = [10, 20, 30, 40, 50]$

$\lambda = [2.5, 2.5\text{e-}3, 2.5, 5\text{e-}4, 0.1]$

$bias = [0, 100, 200, 300, 400]$

g_1: Rotated Expanded Griewank's plus Rosenbrock's Function f_{19}

g_2: Rotated Schaffers F7 Function f_7

g_3: Rotated Schwefel's Function f_{15}

g_4: Rotated Expanded Scaffer's F6 Function f_{20}

g_5: Sphere Function f_1

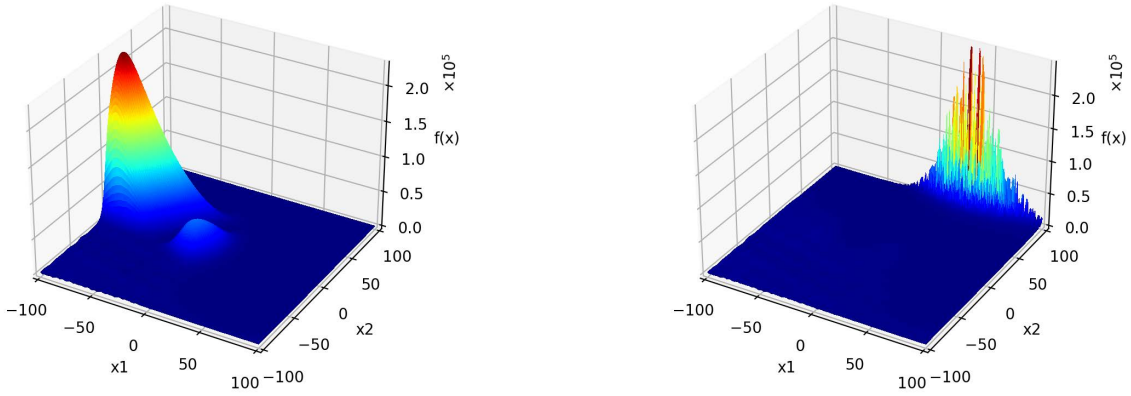
5.2　関数の性質

ベンチマーク関数の性質のうち，代表的なものを述べる．

5.2.1　単峰性・多峰性（Modality）

まず，大域的最適解 $\boldsymbol{x}^*$ を再定義する．ここで，S は探索領域を示す．

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \forall \boldsymbol{x} \in S \tag{5.1}$$



a) F27　　b) F28

図 5.14　F27 と F28 の $D = 2$ での概形

また，ある解 $\boldsymbol{x}^l \in S$ とその近傍 $S^n\left(\boldsymbol{x}^l\right)$ について次式が成り立つとき，$\boldsymbol{x}^l$ は局所解 (局所的最適解) として次で定義される．ここで，任意の正数 ϵ に対し，$S^n\left(\boldsymbol{x}^l\right) = \left\{\boldsymbol{x} | \epsilon > \left\|\boldsymbol{x}^l - \boldsymbol{x}\right\|\right\}$ とする．

$$f\left(\boldsymbol{x}^l\right) \leq f(\boldsymbol{x}), \forall \boldsymbol{x} \in S \cap S^n\left(\boldsymbol{x}^l\right) \tag{5.2}$$

つまり，周辺に対してその評価値が小さい点を局所解 $\boldsymbol{x}^l$ としている．したがって，この局所解 $\boldsymbol{x}^l$ の中でも最も評価値が小さいものを最適解 $\boldsymbol{x}^*$ と呼ぶことになる．

本題に戻って，単峰性・多峰性の定義とは，単峰性とは最適解 $\boldsymbol{x}^*$ 以外の局所解 $\boldsymbol{x}^l$ が存在しないものを単峰性とし，そうでないものを多峰性とされる．多峰性関数の中でも，局所解が多数存在したりその深さが大きいものを定性的に「強い多峰性」や「弱い多峰性」と呼ぶこともある．特に強い多峰性では，局所解にトラップされそこから脱出することが難しいため，比較的単純とされる単峰性関数よりも難しい問題クラスであると言える [56].

5.2.2　変数分離可能性（Separable）

決定変数を各次元 $j \in \{1, 2, \cdots, D\}$ ごとに分離することによって，目的関数も $\sum_{j=1}^{D} f_j\left(x_j\right)$ のように分解でき，1 次元ごと，あるいは j 次元のまとまりごとに分けて問題を扱うことができる [56]．この状態のことを変数分離可能という．このようにする

ことで，次元数増加による計算量増加といった困難さを回避しながら全体として問題を扱うことができる．一方で，$f(x) = \sum_{i=1}^{D-1} \left(100 \left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right)$ のように，異なる次元の決定変数を同時に用いて組み合わせることで構成される関数は変数分離不可能と呼ばれ，次元ごとの計算が行えないことになる．特に CEP では，計算量の観点から，変数分離不可能性はボトルネックとなる．実問題では変数分離可能な問題は少ないとされ [75]，全ての決定変数が完全に依存している（つまり，部分ごとにさえ分離できない）ケースである，完全分離不可能な関数も不自然であるとされている [76].

5.2.3 悪スケール性（ill-scale）

決定変数ごとに目的関数の感度が異なる問題を指す．一般に，進化計算では全ての次元を同時・同様に動かすため，悪スケール性を持つ問題は，最適化が困難とされている [56]. また，F8 のように，決定変数の値によっては急激に傾きに変化が生じる関数にも，この表現が用いられることがある.

5.2.4 CEC 2013 ベンチマーク関数性質のまとめ

上記を踏まえて，今回用いた問題の性質を表 5.2 に示す．表中の Y は各項目にその関数が該当することを，数字は回転行列を乗算した回数（2 回以上の場合）を表す.

表 5.2　ベンチマーク関数の性質

	変数分離不可能	局所解多数	移動	回転	対角化	非対称	不規則	その他
F1			Y					
F2	Y		Y	Y			Y	二次悪条件，滑らかながら局所的には凹凸
F3	Y		Y	2		Y		滑らかながら深い溝
F4			Y	Y			Y	1 方向に強い感度，滑らかながら局所的には凹凸
F5			Y					決定変数の感度にばらつき
F6	Y		Y	Y				局所的最適解と大域的最適解の間に長い溝
F7	Y	Y	Y	2	Y	Y		
F8			Y	2	Y	Y		
F9	Y		Y	2	Y	Y		連続ながら至る所で微分不可能
F10	Y		Y	Y	Y			
F11		Y	Y		Y	Y	Y	
F12	Y	Y	Y	3	Y	Y	Y	
F13	Y	Y	Y	3	Y	Y	Y	不連続
F14	Y	Y	Y	Y	Y	Y		第二大域的最適解が大域的最適解から遠い
F15	Y	Y	Y	Y	Y	Y		第二大域的最適解が大域的最適解から遠い
F16	Y		Y	2	Y	Y		連続ながら至る所で微分不可能
F17	Y		Y		Y	Y		連続ながら至る所で微分不可能
F18	Y		Y	Y	Y	Y		連続ながら至る所で微分不可能
F19	Y		Y	Y				
F20	Y		Y	2		Y		
F21	Y		Y	Y		Y		局所的最適解ごとに異なる性質
F22			Y			Y		局所的最適解ごとに異なる性質
F23	Y		Y	Y		Y		局所的最適解ごとに異なる性質
F24	Y		Y	Y		Y		局所的最適解ごとに異なる性質
F25	Y		Y	Y		Y		局所的最適解ごとに異なる性質
F26	Y		Y	Y		Y		局所的最適解ごとに異なる性質
F27	Y		Y	Y		Y		局所的最適解ごとに異なる性質
F28	Y		Y	Y		Y		局所的最適解ごとに異なる性質

第 6 章

実験

本章では，jDE と SaDE，JADE 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．

6.1 実験設定

6.1.1 問題設定

前章で述べたベンチマークセットを用いる．問題の次元数は $D = 10, 30, 50, 100$ 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．

6.1.2 比較手法とハイパーパラメータ設定

提案手法を組み込んだ jDE と SaDE，JADE と組み込む前の jDE と SaDE，JADE 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 を大きく設定するほど好適なハイパーパラメータが見つかる可能性は高くなるが，計算時間の増加を避けるために $C = 10$ とする．jDE のハイパーパラメータは，$F_{init} = 0.5$, $CR_{init} = 0.9$, $N = 100$, $\tau_F = 0.1$, $\tau_{CR} = 0.1$ とする [38]．SaDE のハイパーパラメータは，$p_{k,init} = 0.25$, $CRm_{k,init} = 0.5$, $LP = 50$, $\epsilon = 0.01$ とする [42]．JADE のハイパーパラメータは，$\mu_{F,init} = 0.5, \mu_{CR,init} = 0.5, c = 0.1, p_{\min} = 0.05, p_{\max} = 0.2$ とする [40].

6.1.3 評価指標

解評価回数を最大 10,000 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．この平均値の比較は，500, 1,000, 3,000, 10,000 の解評価回数ごとに行う．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，JADE と提案手法を組み込んだ JADE の 3 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，$p \geq 0.05$ であれば有意差があるとは言えないとして “~”，$p < 0.05$ 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，500, 1,000, 3,000, 10,000 の解評価回数ごとにこれらの合計を “+/−/~” 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．

なお，1,000 の解評価回数を規定値として性能を表にまとめる．一方で，500, 3,000, 10,000 の解評価回数ごとの詳細な性能の表については，紙面の都合上省略し，検定結果のみを示す．

6.2 実験結果

提案手法を jDE, SaDE，JADE 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 6.1，表 6.2，表 6.3 にそれぞれ示す． 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，提案手法を適応 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 6.1 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 6.1 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 F9 や $D = 30, 50, 100$ における F26 のように提案手法が性能向上に失敗していることから，提案手法の適用によって性能が低下する場合もある．しかしながら，検定ケース 1 における提案手法を組み込まない jDE が統計的に優位であることを示す “–” は存在せず，検定ケース 2 でも $p < 0.05$ 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 6.2）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 6.2 の最後に示したように，全ての次元数で $p < 0.05$ となり関数全体の統計的有意差も見られている．これらの結果から，ハイパーパラメータのみを調整する jDE に対し，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ハイパーパラメータの探索空間を扱う場合にも高い有効性があるといえる．

JADE に提案手法を適用した場合（表 6.3）においても，提案手法は JADE の性能を向上している．具体的には，$D = 10, 30, 50, 100$ の順に 14, 13, 17, 16 個の関数で “+” となっている．一方，JADE が優位であることを示す “–” は $D = 50, 100$ でそれぞれ 1，2 個見られているが，検定ケース 2 では全ての次元数で $p < 0.05$ となり，全 28 個の関数を総合的に見ると提案手法が優位である．

次に，表 6.1 と表 6.2，表 6.3 では解評価回数 1,000 であったが，これを 500, 1,000, 3,000, 10,000 としたときの検定ケース 1 の結果を “+/−/~” で表 6.4 に示す．全体として限られた解評価回数であっても性能が統計的に向上していることから提案手法の有効性が確認できる．$D = 10, 30, 50$ では jDE と SaDE が優位であることを示す “–” の数は 0 であり，次元数が $D = 100$ と上がってもその数は高々 1 個である．特に，解評価回数を 500 に減らした場合においても，jDE，SaDE，JADE に提案手法を用いた場合は “+” の数が多いように，性能の向上が見られる．次元数 $D = 100$，解評価回数 10,000 の JADE における実験ケースを除いて，次元数が $D = 50, 100$ と増加した際は $D = 10, 30$ のときに比べて “+” の数が増えていることから，問題の難易度が上がったときに提案手法の適用はより一層効果的である．解評価回数 3,000 のときも提案手法を適用した適応 DE が優位である傾向は同様である．解評価回数を 10,000 まで緩和すると，jDE と SaDE に提案手法を用いた場合は各次元数ごとに 28 個中 21~24 個の関数で，JADE に提案手法を用いた場

合は 10~23 個の関数で “+” となり，その合計数は 112 実験ケース中，jDE では 95 ケース，SaDE では 92 ケース，JADE では 69 ケースである．したがって，提案手法を jDE と SaDE，JADE に組み込むと，解評価回数を 500 と制限しても性能が向上し，10,000 まで緩和すると $D = 10, 30, 50, 100$ の多くの次元数で “+” の数が顕著に増加し，提案手法の有効性が確認できる．

なお，全ての関数において最適解の評価値は 0 であることを考慮すると，提案手法ならびに比較手法の性能は最適解の評価値から大きく乖離する．まず，state-of-the-art な手法を用いても，かつ，コンペティションの規定の解評価回数（$D \times 10,000$）を以てしても，関数によっては同じ傾向にある [77] ことから，ベンチマークセットは容易に最適化できないように設計されていることがわかる．また，本論文が設定した評価回数のもとで提案手法の有効性が確認できていることに，本研究の意義がある．

表 6.1　jDE と提案手法を適用した jDE の性能比較

	$D = 10$		$D = 30$		$D = 50$		$D = 100$	
	jDE	ours	jDE	ours	jDE	ours	jDE	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** +
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** +
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** +
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** ~
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** +
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** +
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** ~
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** ~
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 ~
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** +
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** +
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** +
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** +
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** +
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** ~
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 ~
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** +
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** +
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** +
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** ~
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** +
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** +
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 ~
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** ~
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** ~
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 ~
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** +
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** +
検定ケース 1: +/−/~		7/0/21		10/0/18		16/0/12		17/0/11
検定ケース 2: p value		0.0006956		0.0003274		0.00003644		0.00003667

表 6.2　SaDE と提案手法を適用した SaDE の性能比較

	$D = 10$		$D = 30$		$D = 50$		$D = 100$	
	SaDE	ours	SaDE	ours	SaDE	ours	SaDE	ours
F1	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	3.66E+07	**3.36E+07** ~	8.75E+08	**7.72E+08** +	2.36E+09	**1.91E+09** +	1.05E+10	**8.81E+09** +
F3	1.12E+10	**9.55E+09** ~	4.62E+15	**9.41E+14** ~	5.37E+15	**2.34E+15** +	3.09E+23	**2.28E+22** +
F4	**4.71E+04**	4.82E+04 ~	1.50E+05	**1.49E+05** ~	2.33E+05	**2.25E+05** ~	4.98E+05	**4.81E+05** ~
F5	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	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	1.57E+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**	2.08E+01 ~	**2.12E+01**	**2.12E+01** ~	**2.13E+01**	**2.13E+01** ~	**2.15E+01**	**2.15E+01** ~
F9	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	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	1.18E+02	**1.08E+02** +	8.12E+02	**6.96E+02** +	1.56E+03	**1.29E+03** +	4.27E+03	**3.48E+03** +
F12	1.27E+02	**1.10E+02** +	8.28E+02	**7.08E+02** +	1.54E+03	**1.32E+03** +	4.44E+03	**3.52E+03** +
F13	1.25E+02	**1.15E+02** +	8.10E+02	**7.25E+02** +	1.54E+03	**1.31E+03** +	4.39E+03	**3.59E+03** +
F14	2.13E+03	**2.06E+03** ~	**8.45E+03**	8.47E+03 ~	1.54E+04	**1.53E+04** ~	**3.41E+04**	**3.41E+04** ~
F15	2.17E+03	**2.12E+03** ~	**8.88E+03**	8.91E+03 ~	1.65E+04	**1.63E+04** ~	**3.41E+04**	3.42E+04 ~
F16	2.43E+00	**2.42E+00** ~	**4.45E+00**	4.56E+00 ~	5.48E+00	**5.43E+00** ~	5.73E+00	**5.65E+00** ~
F17	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	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	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	4.68E+00	**4.62E+00** ~	**1.50E+01**	**1.50E+01** ~	**2.50E+01**	**2.50E+01** ~	**5.00E+01**	**5.00E+01** ~
F21	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	2.43E+03	**2.28E+03** +	9.34E+03	**9.24E+03** ~	1.67E+04	**1.65E+04** ~	3.60E+04	**3.59E+04** ~
F23	2.48E+03	**2.45E+03** ~	**9.57E+03**	**9.57E+03** ~	**1.73E+04**	**1.73E+04** ~	**3.60E+04**	3.62E+04 ~
F24	2.32E+02	**2.31E+02** ~	3.46E+02	**3.38E+02** +	4.88E+02	**4.67E+02** +	1.20E+03	**9.84E+02** +
F25	2.33E+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	**2.15E+02** ~	3.63E+02	**3.41E+02** +	5.05E+02	**5.04E+02** ~	7.60E+02	**7.51E+02** +
F27	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	1.11E+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: +/−/~		16/0/12		17/0/11		18/0/10		19/0/9
検定ケース 2: p value		0.00007775		0.00007643		0.00001306		0.00005129

表 6.3　JADE と提案手法を適用した JADE の性能比較

	$D = 10$		$D = 30$		$D = 50$		$D = 100$	
	JADE	ours	JADE	ours	JADE	ours	JADE	ours
F1	2.43E+03	**1.65E+03** +	2.98E+04	**2.58E+04** +	7.00E+04	**5.79E+04** +	2.06E+05	**1.81E+05** +
F2	3.00E+07	**2.38E+07** +	6.11E+08	**5.29E+08** +	1.61E+09	**1.36E+09** +	6.84E+09	**5.94E+09** +
F3	6.47E+09	**4.99E+09** +	2.82E+13	**1.77E+13** ~	2.56E+14	**4.38E+13** +	7.74E+21	**2.69E+21** +
F4	3.98E+04	**3.89E+04** ~	1.36E+05	**1.31E+05** ~	2.10E+05	**2.09E+05** ~	4.55E+05	**4.16E+05** +
F5	1.47E+03	**1.18E+03** +	2.44E+04	**2.04E+04** +	5.69E+04	**4.39E+04** +	2.02E+05	**1.67E+05** +
F6	1.52E+02	**1.35E+02** ~	3.29E+03	**2.80E+03** +	7.30E+03	**5.63E+03** +	5.32E+04	**4.46E+04** +
F7	1.13E+02	**1.06E+02** ~	3.14E+03	**2.59E+03** ~	7.71E+03	**4.85E+03** +	2.93E+07	**1.66E+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** ~
F9	1.13E+01	**1.10E+01** ~	4.45E+01	**4.44E+01** ~	7.98E+01	**7.97E+01** ~	**1.71E+02**	**1.71E+02** ~
F10	3.29E+02	**2.56E+02** +	4.57E+03	**3.74E+03** +	1.04E+04	**8.37E+03** +	3.25E+04	**2.82E+04** +
F11	9.73E+01	**8.69E+01** +	6.15E+02	**5.55E+02** +	1.27E+03	**1.13E+03** +	3.50E+03	**3.14E+03** +
F12	1.00E+02	**9.28E+01** +	6.53E+02	**5.85E+02** +	1.26E+03	**1.11E+03** +	3.59E+03	**3.21E+03** +
F13	**9.64E+01**	9.65E+01 ~	6.53E+02	**5.84E+02** +	1.28E+03	**1.11E+03** +	3.71E+03	**3.19E+03** +
F14	**1.95E+03**	1.98E+03 ~	8.42E+03	**8.39E+03** ~	1.52E+04	**1.51E+04** ~	**3.37E+04**	3.38E+04 ~
F15	2.12E+03	**2.10E+03** ~	8.89E+03	**8.88E+03** ~	1.64E+04	**1.62E+04** ~	**3.40E+04**	**3.40E+04** ~
F16	**2.49E+00**	2.53E+00 ~	**4.39E+00**	4.43E+00 ~	**5.41E+00**	5.56E+00 ~	**5.74E+00**	5.80E+00 ~
F17	1.53E+02	**1.33E+02** +	1.10E+03	**9.65E+02** +	2.46E+03	**2.08E+03** +	6.61E+03	**5.64E+03** +
F18	1.49E+02	**1.34E+02** +	1.11E+03	**9.54E+02** +	2.49E+03	**2.05E+03** +	6.56E+03	**5.57E+03** +
F19	1.79E+02	**1.03E+02** +	2.03E+05	**1.22E+05** +	9.75E+05	**4.65E+05** +	1.65E+07	**1.11E+07** +
F20	4.55E+00	**4.39E+00** +	**1.50E+01**	**1.50E+01** ~	**2.50E+01**	**2.50E+01** +	**5.00E+01**	**5.00E+01** ~
F21	5.71E+02	**5.32E+02** +	3.23E+03	**2.93E+03** +	6.95E+03	**6.14E+03** +	1.57E+04	**1.33E+04** +
F22	2.23E+03	**2.18E+03** ~	9.15E+03	**9.12E+03** ~	**1.64E+04**	**1.64E+04** ~	**3.56E+04**	3.57E+04 ~
F23	**2.46E+03**	2.47E+03 ~	9.51E+03	**9.50E+03** ~	1.73E+04	**1.72E+04** ~	3.61E+04	**3.60E+04** ~
F24	**2.30E+02**	**2.30E+02** ~	**3.30E+02**	**3.30E+02** ~	**4.43E+02**	4.45E+02 ~	**8.80E+02**	9.52E+02 −
F25	**2.30E+02**	**2.30E+02** ~	**3.48E+02**	**3.48E+02** ~	**4.82E+02**	4.89E+02 −	**8.91E+02**	9.25E+02 −
F26	2.25E+02	**2.20E+02** ~	3.57E+02	**3.43E+02** ~	5.01E+02	**4.95E+02** +	**7.40E+02**	**7.40E+02** ~
F27	6.50E+02	**6.39E+02** +	**1.46E+03**	**1.46E+03** ~	2.45E+03	**2.43E+03** ~	**5.13E+03**	5.14E+03 ~
F28	1.02E+03	**9.54E+02** +	5.35E+03	**5.01E+03** +	9.41E+03	**8.67E+03** +	2.84E+04	**2.57E+04** +
検定ケース 1: +/−/~		14/0/14		13/0/15		17/1/10		16/2/10
検定ケース 2: p value		0.0002527		0.00003291		0.00004313		0.0004418

表 6.4　jDE, SaDE, JADE と提案手法を適用した jDE, SaDE, JADE とのそれぞれの Wilcoxon の符号順位検定の結果（+/−/~）

D	10			30			50			100		
解評価回数	vs jDE	vs SaDE	vs JADE	vs jDE	vs SaDE	vs JADE	vs jDE	vs SaDE	vs JADE	vs jDE	vs SaDE	vs JADE
500	1/0/27	6/0/22	7/0/21	7/0/21	12/0/16	7/1/20	12/0/16	12/0/16	10/1/17	12/0/16	14/0/14	10/ 3/15
1,000	7/0/21	16/0/12	14/0/14	10/0/18	17/0/11	13/0/15	16/0/12	18/0/10	17/1/10	17/0/11	19/0/ 9	16/ 2/10
3,000	17/0/11	20/0/ 8	22/0/ 6	18/0/10	22/0/ 6	21/0/ 7	20/0/ 8	21/0/ 7	21/1/ 6	22/1/ 5	21/0/ 7	21/ 0/ 7
10,000	24/0/ 4	24/0/ 4	20/0/ 8	24/0/ 4	24/0/ 4	23/2/ 3	24/0/ 4	23/0/ 5	16/6/ 6	23/0/ 5	21/0/ 7	10/12/ 6

第 7 章

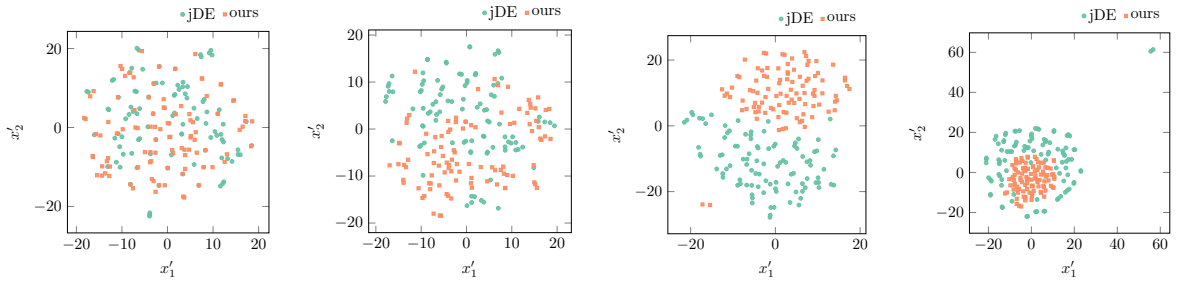
考察

本章では，提案手法の分析を行う．まず，提案手法が獲得した解集合の分布を通して，局所探索傾向が強い提案手法において，解の多様性が損なわれていないかを確認する．次に，提案手法を設計する際の方針の妥当性を，事前検証を行う頻度と，事前検証における基準個体の選び方の観点から検証する．

7.1 解集合の多様性の分析

提案手法では，子個体の本生成メカニズムにより解の多様性の低下を抑制することを意図していた．この効果を検証するために，本節では jDE, SaDE ならびに提案手法が生成した解集合内の個体分布を解評価回数 500, 1,000, 3,000, 10,000 回ごとに比較する．具体的には，単峰性関数 F1（Sphere Function，$D = 100$）と多峰性の合成関数 F28（Composition Function 8，$D = 100$）において，各手法が生成した個体を t-SNE で 2 次元に写像した分布図をそれぞれ図 7.1 と図 7.2 に示す．なお，図中の ours は jDE と SaDE に提案手法をそれぞれ組み込んだ手法を示す．

図 7.1 の d) と g) を除いて，F1 において提案手法が生成した解集合は jDE と SaDE と同程度の分散がある．一方で，同図の d) や g) に示すように，解評価回数が増えると提案手法の解集合の多様性は低下する傾向にある．しかし，単峰性関数である F1 において，この傾向は探索性能を改善する上で妥当である．また，図 7.2 の全ての図に示すように，提案手法が生成した解集合は jDE および SaDE と同程度の分散がある．以上より，少ない解評価回数における単峰性関数 F1 と，多峰性の合成関数 F28 において提案手法は解の多様性の低下を抑制できていることがわかる．なお，他の実験ケースにおいても同様の傾向が確認できている．



a) jDE，500 回　b) jDE，1,000 回　c) jDE，3,000 回　d) jDE，10,000 回

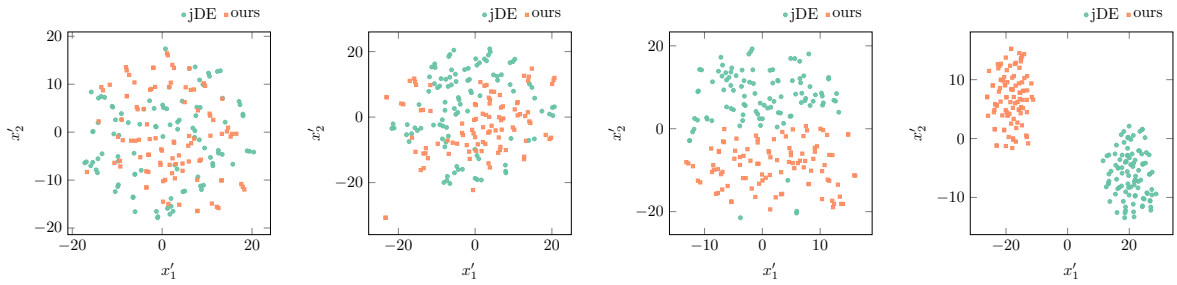
e) SaDE，500 回　f) SaDE，1,000 回　g) SaDE，3,000 回　h) SaDE，10,000 回

図 7.1　F1（$D = 100$）での jDE, SaDE と提案手法を適用した jDE, SaDE における解集合の分布

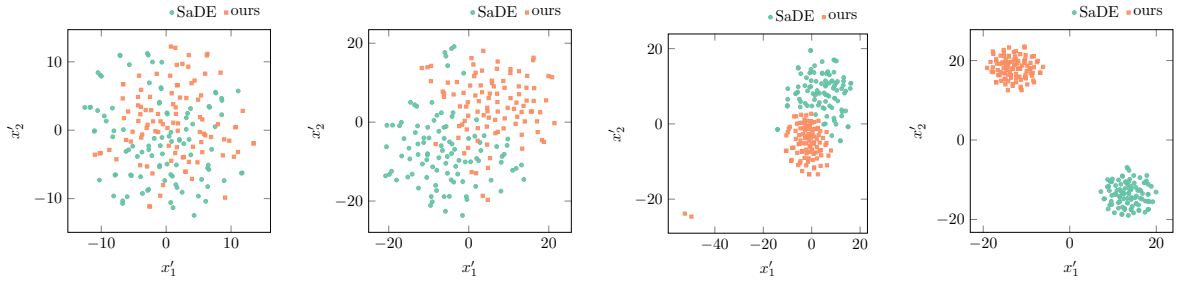
7.2　計算時間の削減を行わない場合との比較

4.3 節において述べたように，計算時間の削減のため，提案手法では前世代に解更新に失敗した個体のハイパーパラメータのみを調整した．ここでは，全ての個体についてハイパーパラメータを調整する方針をとった場合に，性能と計算時間がどのように変化するかを考察する．表 7.1 に，有意水準 0.05 の下で Friedman 検定を行った際に得られた，全問題の平均順位を示す．また，Friedman 検定でいずれかの手法間に有意差が見られたときに，事後検定として行う Wilcoxon の符号順位検定結果を Holm 法で調整した後に，有意差が見られた組を表 7.2 に示す．ここで，表 7.1 と表 7.2 において，ours (F) は第 6 章で用いた設定であり，解更新に失敗した個体のハイパーパラメータを調整する方法を表す．ours (E) は全ての個体のハイパーパラメータを調整する方法を表す．

jDE では解評価回数 500 など極端に評価回数が限られた場合においては，ours (E) が ours (F) より比較的高い性能を導出しており，なおかつ jDE に対して優位である．これは，全個体でハイパーパラメータの調整を行う ours (E) が，ours (F) よりも多くの調整を行うため，特に少ない評価回数ではより適切にハイパーパラメータを調整できているもの



a) jDE，500 回　b) jDE，1,000 回　c) jDE，3,000 回　d) jDE，10,000 回



e) SaDE，500 回　f) SaDE，1,000 回　g) SaDE，3,000 回　h) SaDE，10,000 回

図 7.2　F28（$D = 100$）での jDE, SaDE と提案手法を適用した jDE, SaDE における解集合の分布

と考えられる．一方で，解評価回数が増えると ours (F) の再利用モデルとしての機能が働き，例えば，次元数 $D = 10, 30$ における解評価回数 10,000 では ours (F) が ours (E) に対し優れている．

一方，SaDE では全体として解更新失敗時のみハイパーパラメータの調整を行う ours (F) が高い性能を導出している．これは，ハイパーパラメータをランダムにサンプリングする jDE のランダム生成による再利用モデルよりも，確率分布を調整してサンプリングする SaDE の確率分布に基づく逐次更新モデルが，解更新失敗時のみハイパーパラメータを調整する設定に相応しい可能性を示している．この理由は，SaDE は解更新の成功率をもとに確率分布を調整するため，解更新に失敗したハイパーパラメータを効率的に調整でき，解更新に成功したハイパーパラメータは長い世代に渡って探索に有効であると考えられるからである．加えて，ours (F) では，擬似関数 $Sample(\mathcal{P}, \boldsymbol{\theta}_i, \boldsymbol{x}_i^*)$ に解評価が事後検証結果として与えられることになるため，提案手法の事前検証と対になって効率的に働いていることが確認できる．しかしながら，表 7.2 の検定結果では，ours (F) と ours (E) はどちらも SaDE に全て統計的に優位であるため，両手法のどちらを用いても性能を向上する．

JADE では，次元数 $D = 100$ かつ解評価回数 10,000 をに設定した場合を除いた全ての実験ケースにおいて，ours (F) と ours (E) のどちらも性能を向上している．なお，表 7.2

より，次元数 $D = 100$ かつ解評価回数 10,000 の実験ケースにおいて有意差が検出された組はないことから，JADE と ours(F)，ours(E) は競合する性能を導出している．また，表 7.1 の網掛けした部分を見ると，次元数が $D = 10$ と低いときには ours(F) が高い順位を導出しているのに対し，次元数が増加すると徐々に ours(E) の順位が次第に改善する．これより，優良解の位置情報を利用する JADE に関しては，高次元により問題の難易度が上がった場合，全ての個体について事前検証と調整を行う方が高い性能を導出することがわかる．

次に，ours (E) から ours (F) に変更することによる計算時間の削減効果を確認する．具体的には，表 7.3 に示すように，intel(R) Core(TM) i7-9700(3.0GHz) の CPU における全 28 関数の平均計算時間を次元数 $D = 10, 30, 50, 100$ ごとに比較する．

表 7.3 より，提案手法の実行時間は長くなる傾向にある．ours (F) が次元数 $D = 10$ で解評価回数 $500, 1,000$ で逆転している原因としては，ours (F) が解更新失敗時のみ調整を行っているため，一定確率でこれを行う jDE よりも短時間になっていることが考えられる．つまり，解更新に成功したときは前世代のハイパーパラメータが問題や探索状況に相応しいとし，これを引き継ぐこととすればハイパーパラメータの事前検証と更新の時間が省かれる．

また，計算時間についても次元数ごとに有意水準 0.05 の下で Friedman 検定を行い，同様の事後検定を行った．検定結果としては，表 7.3 に掲載した全ての解評価回数において全組で有意差が検出され，表 7.3 と同じ順位が得られた．したがって，ours (F) と ours (E) を比べたときは実行時間は削減されていることがわかる．一方で，比較的実行時間が短い ours (F) も jDE や SaDE，JADE に比べれば約 2 倍の計算時間がかかっている．しかしながら，提案手法は，元の手法より計算時間がかかるものの，解評価 1 回あたりの計算時間は小さい．例として，解評価回数 1,000 時点で jDE の ours (F) が 1 回の解評価あたりに要する計算時間は，次元数 $D = 10, 30, 50, 100$ の順に 5.861×10^{-3} 秒, 13.04×10^{-3} 秒, 20.14×10^{-3} 秒, 34.52×10^{-3} 秒である．加えて，提案手法は限られた解評価回数で性能を向上しており，数分から数時間，あるいは数日と，1 回の解評価に時間のかかる CEP を解く際には，この問題点は緩和されると考えられる．

7.3 基準個体の選択戦略

4.2.2 項において，ハイパーパラメータの事前検証をする際の基準個体 $\boldsymbol{x}_i^*$ は現在の解集合 $\mathcal{P}$ における最良個体とした（*greedy* 戦略）．ここでは，基準個体の選択戦略が提案手法の最適化性能にどのような影響を与えるかを jDE を例に考察する．今回比較する基準

表 7.1　jDE, SaDE, JADE と提案手法を適用した jDE, SaDE, JADE においてハイパーパラメータの調整頻度を変更したときの平均順位比較

D	10			30			50			100		
解評価回数	jDE	ours (F)	ours (E)	jDE	ours (F)	ours (E)	jDE	ours (F)	ours (E)	jDE	ours (F)	ours (E)
500	2.286	2.036	1.679	2.536	1.679	1.786	2.643	1.732	1.625	2.786	1.643	1.571
1,000	2.500	1.821	1.679	2.750	1.714	1.536	2.571	1.714	1.714	2.607	1.607	1.786
3,000	2.786	1.571	1.643	2.893	1.679	1.429	2.821	1.536	1.643	2.786	1.643	1.571
10,000	2.893	1.357	1.750	2.857	1.321	1.821	2.786	1.571	1.643	2.893	1.393	1.714
解評価回数	SaDE	ours (F)	ours (E)	SaDE	ours (F)	ours (E)	SaDE	ours (F)	ours (E)	SaDE	ours (F)	ours (E)
500	2.571	1.714	1.714	2.518	1.714	1.768	2.679	1.607	1.714	2.679	1.571	1.750
1,000	2.714	1.679	1.607	2.679	1.571	1.750	2.821	1.321	1.857	2.714	1.393	1.893
3,000	2.786	1.357	1.857	2.857	1.143	2.000	2.821	1.357	1.821	2.750	1.214	2.036
10,000	2.857	1.214	1.929	2.964	1.179	1.857	2.821	1.393	1.786	2.714	1.643	1.643
解評価回数	JADE	ours (F)	ours (E)	JADE	ours (F)	ours (E)	JADE	ours (F)	ours (E)	JADE	ours (F)	ours (E)
500	2.607	1.679	1.714	2.375	1.964	1.661	2.536	1.714	1.750	2.429	1.893	1.679
1,000	2.750	1.500	1.750	2.679	1.536	1.786	2.571	1.607	1.821	2.643	1.607	1.750
3,000	2.714	1.464	1.821	2.786	1.393	1.821	2.679	1.357	1.964	2.643	1.679	1.679
10,000	2.714	1.321	1.964	2.786	1.571	1.643	2.571	1.857	1.571	1.964	2.214	1.821

表 7.2　jDE, SaDE, JADE と提案手法を適用した jDE, SaDE, JADE においてハイパーパラメータの調整頻度を変更したときの比較における有意差検出組

解評価回数	$D = 10$	$D = 30$	$D = 50$	$D = 100$
500	(E) - j	(E) - j	(E) - j, (F) - j	(E) - j, (F) - j
1,000	(E) - j	(E) - j, (F) - j	(E) - j, (F) - j	(E) - j, (F) - j
3,000	(E) - j, (F) - j	(E) - j, (F) - j	(E) - j, (F) - j	(E) - j, (F) - j
10,000	(E) - (F), (E) - j, (F) - j	(E) - (F), (E) - j, (F) - j	(E) - j, (F) - j	(E) - j, (F) - j
500	(E) - S, (F) - S	(E) - S, (F) - S	(E) - S, (F) - S	(E) - S, (F) - S
1,000	(E) - S, (F) - S	(E) - S, (F) - S	(E) - (F), (E) - S, (F) - S	(E) - (F), (E) - S, (F) - S
3,000	(E) - (F), (E) - S, (F) - S	(E) - (F), (E) - S, (F) - S	(E) - (F), (E) - S, (F) - S	(E) - (F), (E) - S, (F) - S
10,000	(E) - (F), (E) - S, (F) - S	(E) - (F), (E) - S, (F) - S	(E) - S, (F) - S	(E) - S, (F) - S
500	(E) - J, (F) - J	(E) - J	(E) - J, (F) - J	(E) - J, (F) - J
1,000	(E) - J, (F) - J	(E) - J, (F) - J	(E) - (F), (E) - J, (F) - J	(E) - J, (F) - J
3,000	(E) - (F), (E) - J, (F) - J	(E) - (F), (E) - J, (F) - J	(E) - (F), (E) - J, (F) - J	(E) - J, (F) - J
10,000	(E) - (F), (E) - J, (F) - J	(E) - J, (F) - J	(E) - J	–

辞書順に記載, j: jDE, S: SaDE, J: JADE, (E): ours (E), (F): ours (F)

個体の選択戦略は 4.2.2 項に示した 4 つである．その他の実験条件は第 6 章と同じとし，*p-best* 戦略や ϵ-*greedy* 戦略のパラメータは $p = \epsilon = 0.2$ とする．得られた性能に対しては Friedman 検定を行い，これまでと同様の事後検定を行う．

表 7.4 に平均順位，表 7.5 に Holm 法適用後に有意水準 0.05 で有意差が見られた組を示す．なお，提案手法は全て統計的有意差をもって jDE より高い平均順位を導出していることから，jDE は表から外した．表 7.4 について，*rand* 戦略でありながら，$D = 50$, 解評価回数 1,000 で高い性能を導出している．この理由として，DE のアルゴリズム自体が

表 7.3　jDE, SaDE, JADE と提案手法を適用した jDE, SaDE, JADE の全実験ケースでの平均実行時間比較 （単位：秒）

D	10			30			50			100		
解評価回数	jDE	ours (F)	ours (E)	jDE	ours (F)	ours (E)	jDE	ours (F)	ours (E)	jDE	ours (F)	ours (E)
500	3.297	2.612	3.278	4.890	5.663	7.045	6.796	8.901	11.27	10.18	15.42	19.69
1,000	6.736	5.861	7.499	9.938	13.04	15.96	13.83	20.14	25.59	20.98	34.52	43.29
3,000	19.21	19.91	23.56	29.39	45.79	49.72	40.95	68.56	78.88	64.04	110.7	131.6
10,000	63.09	72.26	79.10	100.4	163.8	163.5	136.6	248.9	260.9	214.8	411.3	437.7
解評価回数	SaDE	ours (F)	ours (E)	SaDE	ours (F)	ours (E)	SaDE	ours (F)	ours (E)	SaDE	ours (F)	ours (E)
500	1.085	1.908	2.939	2.097	3.543	5.649	3.103	4.888	8.367	5.635	8.639	14.95
1,000	2.236	4.313	7.096	4.370	8.970	13.82	6.478	12.26	20.52	11.77	22.26	36.71
3,000	6.576	13.26	22.81	13.38	29.76	46.16	19.92	38.82	68.33	36.49	76.55	122.6
10,000	22.87	44.01	84.88	44.98	97.83	170.3	66.72	130.8	257.4	122.3	263.2	470.7
解評価回数	JADE	ours (F)	ours (E)	JADE	ours (F)	ours (E)	JADE	ours (F)	ours (E)	JADE	ours (F)	ours (E)
500	0.809	1.219	1.973	1.632	2.396	3.554	2.453	3.552	5.161	4.384	6.489	9.227
1,000	1.683	3.011	4.782	3.342	5.989	8.764	5.011	8.484	12.78	9.075	16.59	22.77
3,000	5.023	11.19	16.12	10.14	18.93	30.08	15.29	25.49	44.11	27.82	53.57	79.15
10,000	16.70	35.71	58.03	33.93	56.53	111.4	51.05	81.14	166.4	93.54	180.7	306.2

非決定的に子個体を生成し，好適な個体が生成されればそれを次世代の親個体とするため，一定確率で高い性能を出す可能性があることが考えられる．全体的には，現在の最良個体の周辺に解生成する能力を持たせる *greedy* 戦略が高い性能を導出する．一方で，*ϵ-greedy* 戦略の順位も高いことから，多峰性関数などでは多様な個体を基準として，個体だけでなくハイパーパラメータの多様性を担保することも重要になると推測できる．しかしながら，表 7.5 にも見られるように多くの実験ケースで有意差が見られず，有意差がある組では *greedy* 戦略や *ϵ-greedy* 戦略が優位である．ここで，*p-best* 戦略や *ϵ-greedy* 戦略と *greedy* 戦略を比較したとき，*p-best* 戦略や *ϵ-greedy* 戦略では提案手法のハイパーパラメータ数が 1 つ増加することになる．したがって，jDE からの性能向上という点では基本的にはどの基準個体選択戦略を使用しても大きな差異はないものの，ハイパーパラメータ数の観点からは，*greedy* 戦略が *ϵ-greedy* 戦略よりも良い選択肢になり得る．

表 7.4　jDE に提案手法を適用した際の基準個体の選択戦略による平均順位比較

D	10				30			
解評価回数	*rand*	*greedy*	*p-best*	ϵ-*greedy*	*rand*	*greedy*	*p-best*	ϵ-*greedy*
500	2.393	3.000	2.429	2.179	2.875	2.411	2.411	2.304
1,000	2.571	2.500	2.393	2.536	2.964	2.036	2.714	2.286
3,000	2.929	2.214	2.786	2.071	2.964	2.357	2.679	2.000
10,000	3.143	2.036	2.857	1.964	3.393	1.714	2.821	2.071
D	50				100			
解評価回数	*rand*	*greedy*	*p-best*	ϵ-*greedy*	*rand*	*greedy*	*p-best*	ϵ-*greedy*
500	2.661	2.429	2.554	2.357	2.661	2.554	2.482	2.304
1,000	2.304	2.429	2.482	2.786	2.518	2.482	2.554	2.446
3,000	3.000	2.071	2.429	2.500	2.875	2.268	2.946	1.911
10,000	3.500	1.857	2.607	2.036	3.196	1.732	2.804	2.268

表 7.5　jDE に提案手法を適用した際の基準個体の選択戦略による比較における有意差検出組

解評価回数	$D = 10$	$D = 30$	$D = 50$	$D = 100$
500	e - g	–	–	–
1,000	–	g - r	–	–
3,000	–	e - r	g - r	–
10,000	e - r	e - p, e - r, g - p, g - r	e - r, g - r, p - r	g - p, g - r

辞書順に記載, r: *rand*, g: *greedy*, p: *p-best*, e: ϵ-*greedy*

第 8 章

提案手法の拡張：
事前検証型アンサンブル適応 DE

本章以降では，性能をさらに向上すべく，第 4 章の提案手法を拡張する．

8.1 背景

適応進化計算では，あらかじめ定義されたハイパーパラメータを，適応の選択肢としてアンサンブルして好適な候補を選択する．そのため，適応進化計算は，アンサンブル手法とみなすことができる．このアンサンブルの方法論は二つに大別できる [78]．一つは，進化計算のハイパーパラメータ（突然変異戦略や交叉戦略などの遺伝的操作を含む）をアンサンブルしてこれらを適応する Low-level アンサンブルであり，もう一つは，一つのフレームワークが複数の進化計算を内包して適応的に使用する High-level アンサンブルである．引き続き，適応進化計算のうち盛んに改良手法が提案されている手法群である適応差分進化（適応 DE）を扱う．以降では，Low-level アンサンブルの適応 DE を単に適応 DE と，異種の適応 DE を High-level にアンサンブルする手法をアンサンブル適応 DE と，それぞれ呼ぶ．

一般に，適応 DE をアンサンブル適応 DE に拡張した際には，最適化性能の向上の可能性と，アルゴリズム空間を探索する困難さの増加のトレードオフが発生する．具体的には，No Free Lunch Theorem [79] が示唆するように，アンサンブル適応 DE への拡張により適応の選択肢が広がると適応可能な最適化問題が増え，性能を大きく向上する可能性がある．一方で，膨大化したアルゴリズム探索空間から好適なアルゴリズムやハイパーパラメータを見つける困難さが増加する．実際，著者の調査の限りでは，異種の適応 DE で構

成されたアンサンブル適応 DE の既存手法は以下の三つのみと非常に少なく，前述のトレードオフを克服して性能向上する手法の実現は困難であることが窺える．特に少ない解評価回数でこれは顕著になる．

HMJCDE [26] は，適応 DE の中で比較的に局所探索指向である JADE [40] と，比較的に大域探索指向である CoDE [39] を，これらの指向をそれぞれ更に高める形で改良してアンサンブルする．適応法として，HMJCDE は一方の適応 DE を採用し，最良解の評価値改善率（Fitness Improvement Rate: FIR）が一定世代間に渡り閾値以下となり探索が停滞する度に，もう一方の適応 DE に切り替える．MVC-* [28] では EPSDE [64]，SHADE [43]，CoBiDE [74] とその改良手法をアンサンブルし，バリエーション毎に使用する適応 DE の名称が*に入る．探索フェーズを学習世代 LG と実行世代 EG が繰り返すように等分割し，LG では適応 DE 毎に独立して探索を行い，最良解を導出した適応 DE を使用して EG で探索する．EDEV [29] は JADE，CoDE，EPSDE をアンサンブルし，解集合をそれぞれの適応 DE のサブ解集合と報酬解集合に分割する．そして，一定世代毎に FIR が最大の適応 DE に報酬解集合を与えて探索を行う．

これら既存のアンサンブル適応 DE は共通して，過去一定世代における適応 DE の使用結果（解評価値）をフィードバックして，次世代で使用する適応 DE を選択する．すなわち，第 1 章で述べた，事後検証型の適応法に該当する．この適応法は，一定世代の探索を終えるまで適応 DE の妥当性を判断できない．つまり，好適な適応 DE を見つけるために試行錯誤的に一定世代の探索を繰り返し，大量の解評価回数を消費する．よって，事後検証型の適応法は CEP には不適である．加えて，選択した適応 DE の現在の解に対する妥当性は検証されずにそのまま使用される．

したがって，拡張研究の目的は，現在の解に割り当てる適応 DE の妥当性を解生成に先立って検証しながら，好適な適応 DE を追加の解評価なしに毎世代推定する，事前検証型のアンサンブル適応 DE の提案である．具体的には，CEP が対象とする少ない解評価回数での最適化では，優良解近傍の局所探索が有効であることから，現在の解から優良解に近い解を生成可能な適応 DE を好適な候補として現在の解に割り当てる．これにより，膨大なアルゴリズム空間でも少ない解評価回数で好適な候補を絞り込み，前述のトレードオフを緩和して性能向上を目指す．

実験では，第 5 章で述べた，制約なし単一目的連続実数値最適化のベンチマークセット（CEC2013 benchmark suite [47]）において，アンサンブル適応 DE の代表手法と拡張手法の性能を比較する．

8.2 概要

拡張手法は，JADE，CoDE，EPSDE をアンサンブルする．この点は EDEV と同様である．しかしながら，EDEV を含む既存のアンサンブル適応 DE は，事後検証型の適応法により，好適な適応 DE を見つけるために一定世代の探索を繰り返さなければならず，大量の解評価回数を消費する．また，割り当てた適応 DE の，現在の探索状況での個体 $\boldsymbol{x}_i$ に対する妥当性を検証せずにそのまま用いる．

そこで，拡張手法における事前検証型の適応法は，現在の探索状況において，$\boldsymbol{x}_i$ 毎に，

- 適応 DE のスクリーニングにより，割り当ての妥当性を検証しながら，
- 好適な適応 DE を解の本生成に先立って追加の解評価なしに推定する．

ここで，拡張手法は事後検証を行わないが，これは，各適応 DE が事後検証の要領で，Low-level に自身のハイパーパラメータを既に適応しているためである．

事前検証におけるアイデアは，次の通りである．

- 収束速度を改善するために，優良解の周辺領域を局所探索可能な適応 DE をスクリーニングする．具体的には，既に発見された優良解に最も近づく仮子個体を生成可能な適応 DE を好適と定義し，各適応 DE による解生成のシミュレーションを行う．つまり，各適応 DE の方法で決定されたハイパーパラメータ持つ適応 DE の設定を一つずつ生成し，更にこれらから一つずつ生成した仮子個体と，優良解とのユークリッド距離をそれぞれ計算し，これが最小となる適応 DE の設定を選択する．
- 一方で，早期収束を抑制する工夫として，世代更新時には，選択された適応 DE を用いて子個体を改めて生成する．これは，突然変異や交叉の非決定性によって，解の多様性の急速な低下を抑制することを意図する．

8.3 メカニズム

8.3.1 初期化

通常の DE と同様に，一様乱数を用いて生成した個体 $\boldsymbol{x}_i$（$i = 1, 2, \ldots, N$）からなる解集合 $\mathcal{P}$ を用意する．続いて，$\mathcal{P}$ を適応 DE の個数（=3）で均等に分割し，$\mathcal{P}_j$ とする．

$j \in \{1,2,3\}$ は, それぞれ JADE，CoDE，EPSDE を表すインデックスである．ここで，$\mathcal{P}_j$ は互いに素な集合であり，$\mathcal{P} = \bigcup_j \mathcal{P}_j$ を満たす．$N \not\equiv 0 \pmod 3$ の場合は端数となる個体をランダムに $\mathcal{P}_j$ に割り振る．次に，解評価を伴う解の本生成として，1 世代分の世代更新をそれぞれの適応 DE の方法で $\mathcal{P}_j$ 内で行う．具体的な世代更新の方法は，第 2・3 章を参照されたい．以降は，事前検証と世代更新の組を，探索終了条件まで繰り返す．

8.3.2 事前検証

疑似コードを Algorithm 11 に示す．コード中の ♣ は，第 10 章の考察で用いるための準備である．最初に，現在の解集合 $\mathcal{P}$ 内の最良解を事前検証の基準個体 $\boldsymbol{x}^*$ とする．

続いて，各 $\boldsymbol{x}_i \in \mathcal{P}$ について，以下の操作を行う．適応 DE の候補を格納する集合 $\mathcal{D}'_i$ を空集合 $\emptyset$ で用意する．現在の世代における適応 DE により決定されたスケール係数，交叉率，突然変異戦略，交叉戦略を持つ適応 DE の設定を $aDE'_{i,j}$ として，各 j について一つずつ生成し，$\mathcal{D}'_i$ に追加する．具体的には，JADE（$j = 1$）であれば，$F_i = C(\mu_F, 0.1)$, $CR_i = \mathcal{N}(\mu_{CR}, 0.1)$ で新たにサンプリングした DE ハイパーパラメータを持つ JADE の設定を生成する．CoDE（$j = 2$）であれば第 3 章で述べた各プールからランダムに組み合わされた三組を持つ CoDE を生成し，各組の設定を $aDE'_{i,j,k}$（$k \in \{1,2,3\}$）とする．EPSDE（$j = 3$）であれば，前世代の $\boldsymbol{x}_i$ が解更新に成功した場合（$f(\boldsymbol{x}_i) \leq f(\boldsymbol{u}_i)$）は前世代の設定を，失敗した場合（$f(\boldsymbol{x}_i) > f(\boldsymbol{u}_i)$）は P_F，P_{CR}，P_v よりランダムにそれぞれ選んだ F_i，CR_i，突然変異戦略を持つ EPSDE の設定を生成する．

次に，$aDE'_{i,j}$ を用いて，$\boldsymbol{x}_i$ を親個体として $\mathcal{P}_j$ より仮子個体 $\boldsymbol{u}'_{aDE'_{i,j}}$ を生成する．なお，CoDE の場合は全ての $aDE'_{i,j,k}$ で仮子個体 $\boldsymbol{u}'_{aDE'_{i,j,k}}$ を生成し，その重心を $\boldsymbol{u}'_{aDE'_{i,j}}$ とする．これは，後の世代更新で，CoDE は解評価回数を 1 個体あたり JADE や EPSDE の 3 倍消費するため，CoDE の三組の設定が平均的に好適な解生成を実現することを事前検証し，世代更新時の解評価回数の浪費を抑えることを意図している．最後に，基準個体 $\boldsymbol{x}^*$ と仮子個体 $\boldsymbol{u}'_{aDE'_{i,j}}$ のユークリッド距離の最小値を与える $aDE'_{i,j}$ を $\boldsymbol{x}_i$ に割り当てる適応 DE の設定（aDE_i）とする．

全ての個体 $\boldsymbol{x}_i \in \mathcal{P}$ を事前検証した後に，aDE_i に基づき $\boldsymbol{x}_i$ を $\mathcal{P}_j$ に割り当て直す．ここで，第 2・3 章より，全ての突然変異戦略を実行可能にする最小解集合サイズは 6 であるため，$|\mathcal{P}_j| < 6$ となった場合は，$|\mathcal{P}_j| \geq 6$ となるまで他のサブ解集合よりランダムに選んだ個体を $\mathcal{P}_j$ に割り当てる．

Algorithm 11 *Prior-validation* $(\mathcal{P}, \mathcal{A})$

$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ $\quad \boldsymbol{x} \in \mathcal{P}$
for $i = 1$ **to** N **do**
  $\mathcal{D}'_i = \emptyset$
  for $j = 1$ **to** 3 **do**
    $\mathcal{D}'_i \leftarrow aDE'_{i,j}$ including sampled settings $\cdots$ ♣
    if $j \in \{1, 3\}$ **then**
      $\boldsymbol{v}'_{aDE'_{i,j}} = Mutation\left(\mathcal{P}_j, \mathcal{A}, aDE'_{i,j}\right)$
      $\boldsymbol{u}'_{aDE'_{i,j}} = Crossover\left(\boldsymbol{x}_i, \boldsymbol{v}'_{aDE'_{i,j}}, aDE'_{i,j}\right)$
    else
      for $k = 1$ **to** 3 **do**
        $\boldsymbol{v}'_{aDE'_{i,j,k}} = Mutation\left(\mathcal{P}_j, aDE'_{i,j,k}\right)$
        $\boldsymbol{u}'_{aDE'_{i,j,k}} = Crossover\left(\boldsymbol{x}_i, \boldsymbol{v}'_{aDE'_{i,j,k}}, aDE'_{i,j,k}\right)$
        $\boldsymbol{u}'_{aDE'_{i,j}} = \frac{1}{3}\sum_k \boldsymbol{u}'_{aDE'_{i,j,k}}$
      end for
    end if
  end for
  $aDE_i = \arg\min_{aDE'_{i,j}} \min_{aDE'_{i,j} \in \mathcal{D}'_i} \left\| \boldsymbol{x}^* - \boldsymbol{u}'_{aDE'_{i,j}} \right\|$
end for

8.3.3 世代更新

全ての $\boldsymbol{x}_i \in \mathcal{P}$ を，1 世代のみ世代更新する．ここでは，aDE_i によって $\mathcal{P}_j$ より改めて解生成を行い，解評価と解更新を行う．その後，JADE のメタパラメータ μ_F，μ_{CR} とアーカイブ $\mathcal{A}$ を更新し，再び事前検証へと戻る．

第 9 章

拡張手法の実験

拡張手法の有効性を確認するために，以下の実験を行う．

9.1 実験設定

9.1.1 問題設定

引き続き，CEC 2013 benchmark suite 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を用いる．問題の次元数は $D \in \{10, 30, 50, 100\}$ と設定して，次元数の増加に対する拡張手法の性能のスケーラビリティを評価する．したがって，合計 112（28×4）の実験ケースを扱うことになる．

9.1.2 比較手法とハイパーパラメータ設定

アンサンブル適応 DE の代表手法である HMJCDE と EDEV を比較手法とする．なお，MVC-*は原著で八つのバリエーションが同等に提案されており，全てを比較することは現実的でないため，比較手法から除外した．比較手法はいずれも事後検証型の適応法であり，比較を通して拡張手法の事前検証型の適応法の有効性が確認できる．

全手法で解集合サイズは $N = 100$ とし，初期解は一様分布を用いて生成される．原著の設定通り，HMJCDE のハイパーパラメータは，$Q_1 = 10, Q_2 = 5, m = 30, \epsilon = 0.05, \mu_{F,init} = 0.5, \mu_{CR,init} = 0.5, c = 0.1, p \in [0, 0.05]$ とする [26]．同じく原著通り，EDEV のハイパーパラメータは $\lambda_1 = \lambda_2 = \lambda_3 = 0.1, \lambda_4 = 0.7, ng = 20$ とし [29]，EDEV と拡張手法で JADE のハイパーパラメータ（$\mu_{F,init} = 0.5, \mu_{CR,init} = 0.5, c = 0.1, p_{\min} = 0.05, p_{\max} = 0.2$），CoDE と EPSDE の設定（第 3 章を参照）は共通である．拡張手法は適応 DE の設定以外

のハイパーパラメータを持たない.

9.1.3 評価指標

解評価回数を最大 100,000 回とし，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）で比較する．この平均値の比較は，300，500，1,000，3,000，5,000，10,000，30,000，50,000，100,000 回の解評価回数毎に行う．加えて，統計的有意差を確認するために，ある次元数 D の関数毎に，有意水準 0.05 の下で Wilcoxon の符号順位検定を行う．具体的には，$p \geq 0.05$ であれば有意差があるとは言えないとして “~”，$p < 0.05$ で拡張手法が優位ならば “+”，$p < 0.05$ で比較手法が優位ならば “−” とする．各解評価回数・次元数毎に合計を “+/−/~” として報告する．すなわち，これは第 6 章の検定ケース 1 に相当する．さらに，全体の性能を比較するために有意水準 0.05 の下で Friedman 検定を行う．

9.2 実験結果

解評価回数 1,000 回と 100,000 回において，次元数を $D \in \{10, 30, 50, 100\}$ と変化させたときの性能をそれぞれ表 9.1 と 9.2 に示す．緑の網掛けは手法が最良値を導出したことを表す．全体を通して，事前検証を行う拡張手法が多くのケースで高い性能を導出している．解評価回数を 1,000 回に制限した表 9.1 において D =10，30，50，100 の順にそれぞれ 28 個のうち 23，25，24，25 個の関数で最良値となっている．検定でも対 HMJCDE で 23，20，21，22，対 EDEV で 10，15，17，18 個の関数で “+” となり，拡張手法が優位である．特に 224 個の検定結果のうち，比較手法が優位であることを示す “−” は一つのみであり，少ない解評価回数で拡張手法の有効性が確認できる．

表 9.2 に示すように，解評価回数を 100,000 回まで緩和しても拡張手法の有効性は示されている．D =10，30，50，100 でそれぞれ 15，18，20，19 回の最良値を導出しており，拡張手法は次元数に対する性能のスケーラビリティを有する．一方で，$D = 100$ における合成関数のうち $F_{24} \sim F_{28}$ は HMJCDE が最良値を導出し，うち四つの関数では統計的有意差を持って HMJCDE が優位であることから，拡張手法は大域探索圧力が小さい．しかしながら，全ての次元数において拡張手法の最悪値の導出回数は高々 3 であり，拡張手法は安定した性能を導出している．また，検定でも対 HMJCDE で 12，16，17，21 個，対 EDEV で 15，19，16，18 個の関数で “+” となり，拡張手法の優位性は失われない．

次に，解評価回数を 300, 500, 1,000, 3,000, 5,000, 10,000, 30,000, 50,000, 100,000 回と

表 9.1　解評価回数 1,000 回時点の性能比較（$D \in \{10, 30, 50, 100\}$）

	$D = 10$			$D = 30$			$D = 50$			$D = 100$		
ID	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours
F1	7.76E+03 +	5.34E+03 +	**4.03E+03**	6.83E+04 +	5.54E+04 +	**4.59E+04**	1.26E+05 +	1.16E+05 +	**9.45E+04**	3.06E+05 +	2.91E+05 +	**2.57E+05**
F2	5.45E+07 +	4.35E+07 ~	**4.20E+07**	1.21E+09 +	1.01E+09 +	**8.89E+08**	3.15E+09 +	2.75E+09 +	**2.29E+09**	1.33E+10 +	1.12E+10 +	**9.36E+09**
F3	2.24E+10 +	1.50E+10 ~	**1.14E+10**	3.86E+17 +	5.83E+16 ~	**7.15E+15**	1.20E+17 +	4.17E+16 +	**3.20E+15**	2.95E+24 +	3.71E+24 +	**3.39E+23**
F4	**5.16E+04** ~	5.27E+04 ~	5.22E+04	1.65E+05 ~	1.73E+05 +	**1.58E+05**	2.59E+05 ~	**2.49E+05** ~	2.53E+05	5.50E+05 +	5.50E+05 +	**5.19E+05**
F5	5.31E+03 +	3.53E+03 ~	**3.03E+03**	6.82E+04 +	5.48E+04 +	**4.07E+04**	1.40E+05 +	1.17E+05 +	**8.99E+04**	4.12E+05 +	3.86E+05 +	**3.07E+05**
F6	5.02E+02 +	3.89E+02 +	**2.64E+02**	1.23E+04 +	9.03E+03 +	**6.14E+03**	1.88E+04 +	1.58E+04 +	**1.21E+04**	9.97E+04 +	9.12E+04 +	**7.43E+04**
F7	2.24E+02 +	1.82E+02 +	**1.49E+02**	2.39E+05 +	1.02E+05 +	**1.61E+04**	1.77E+05 +	8.86E+04 +	**3.51E+04**	6.54E+08 +	5.56E+08 +	**1.37E+08**
F8	**2.07E+01** ~	**2.07E+01** ~	**2.07E+01**	**2.12E+01** ~	**2.12E+01** ~	**2.12E+01**	**2.13E+01** ~	**2.13E+01** ~	**2.13E+01**	2.15E+01 ~	2.15E+01 ~	**2.14E+01**
F9	1.17E+01 +	1.13E+01 ~	**1.12E+01**	4.49E+01 ~	**4.44E+01** ~	4.49E+01	8.08E+01 ~	**8.01E+01** ~	8.06E+01	1.73E+02 ~	**1.72E+02** ~	**1.72E+02**
F10	8.41E+02 +	6.63E+02 ~	**5.69E+02**	9.13E+03 +	7.66E+03 +	**6.53E+03**	1.91E+04 +	1.63E+04 +	**1.45E+04**	5.42E+04 +	4.81E+04 +	**4.12E+04**
F11	1.55E+02 +	1.34E+02 ~	**1.29E+02**	1.07E+03 +	9.12E+02 +	**8.04E+02**	1.98E+03 +	1.80E+03 +	**1.55E+03**	5.14E+03 +	4.97E+03 +	**4.35E+03**
F12	1.66E+02 +	1.48E+02 +	**1.26E+02**	1.05E+03 +	9.37E+02 +	**7.94E+02**	1.87E+03 +	1.77E+03 +	**1.57E+03**	5.25E+03 +	5.04E+03 +	**4.43E+03**
F13	1.60E+02 +	1.48E+02 +	**1.27E+02**	1.06E+03 +	9.33E+02 +	**8.27E+02**	1.91E+03 +	1.77E+03 +	**1.56E+03**	5.49E+03 +	5.12E+03 +	**4.41E+03**
F14	2.27E+03 +	2.20E+03 ~	**2.18E+03**	8.69E+03 +	8.55E+03 ~	**8.53E+03**	1.57E+04 +	**1.54E+04** ~	**1.54E+04**	3.46E+04 +	3.42E+04 ~	**3.40E+04**
F15	2.20E+03 ~	**2.17E+03** ~	2.20E+03	9.04E+03 ~	8.94E+03 ~	**8.93E+03**	1.64E+04 ~	1.64E+04 ~	**1.63E+04**	**3.42E+04** ~	3.43E+04 ~	3.43E+04
F16	2.54E+00 ~	2.59E+00 +	**2.38E+00**	4.47E+00 ~	**4.36E+00** ~	4.46E+00	**5.45E+00** ~	5.49E+00 ~	5.48E+00	**5.71E+00** ~	5.87E+00 ~	5.78E+00
F17	2.74E+02 +	2.29E+02 ~	**2.04E+02**	1.89E+03 +	1.74E+03 +	**1.52E+03**	3.69E+03 +	3.56E+03 +	**3.13E+03**	8.78E+03 +	8.81E+03 +	**7.82E+03**
F18	2.61E+02 +	2.35E+02 +	**1.96E+02**	1.92E+03 +	1.80E+03 +	**1.51E+03**	3.75E+03 +	3.59E+03 +	**3.21E+03**	8.94E+03 +	8.75E+03 +	**7.94E+03**
F19	1.19E+04 +	8.41E+03 +	**1.73E+03**	2.96E+06 +	1.77E+06 +	**7.98E+05**	5.44E+06 +	5.08E+06 +	**2.50E+06**	5.56E+07 +	4.73E+07 +	**3.35E+07**
F20	4.79E+00 +	4.68E+00 ~	**4.65E+00**	**1.50E+01** ~	**1.50E+01** ~	**1.50E+01**	**2.50E+01** –	**2.50E+01** ~	**2.50E+01**	**5.00E+01** ~	**5.00E+01** ~	**5.00E+01**
F21	8.17E+02 +	7.38E+02 +	**6.92E+02**	4.50E+03 +	4.29E+03 +	**3.89E+03**	9.08E+03 +	8.98E+03 +	**8.09E+03**	1.90E+04 +	1.93E+04 +	**1.74E+04**
F22	2.47E+03 +	**2.33E+03** ~	2.37E+03	9.57E+03 +	9.36E+03 ~	**9.29E+03**	1.70E+04 +	**1.67E+04** ~	**1.67E+04**	3.65E+04 +	3.62E+04 +	**3.59E+04**
F23	2.49E+03 ~	**2.45E+03** ~	2.47E+03	9.55E+03 ~	**9.50E+03** ~	9.59E+03	1.73E+04 ~	**1.72E+04** ~	**1.72E+04**	3.63E+04 ~	**3.61E+04** ~	**3.61E+04**
F24	2.34E+02 +	2.32E+02 +	**2.31E+02**	3.76E+02 +	3.45E+02 ~	**3.41E+02**	5.90E+02 +	4.98E+02 +	**4.71E+02**	1.97E+03 +	1.26E+03 ~	**1.12E+03**
F25	2.33E+02 +	**2.31E+02** –	2.33E+02	3.72E+02 +	3.55E+02 ~	**3.54E+02**	5.22E+02 +	4.95E+02 ~	**4.90E+02**	1.03E+03 +	**9.19E+02** ~	9.26E+02
F26	2.25E+02 +	2.19E+02 ~	**2.16E+02**	3.88E+02 ~	3.83E+02 ~	**3.68E+02**	5.13E+02 +	**5.03E+02** ~	5.09E+02	7.95E+02 +	7.62E+02 ~	**7.56E+02**
F27	7.47E+02 +	6.99E+02 ~	**6.85E+02**	1.59E+03 +	1.53E+03 ~	**1.52E+03**	2.77E+03 +	2.61E+03 +	**2.56E+03**	6.59E+03 +	5.74E+03 +	**5.51E+03**
F28	1.31E+03 +	1.17E+03 ~	**1.13E+03**	7.66E+03 +	7.04E+03 +	**6.33E+03**	1.30E+04 +	1.24E+04 +	**1.12E+04**	3.54E+04 +	3.54E+04 +	**3.22E+04**
+/−/~	23/0/5	10/1/17		20/0/8	15/0/13		21/1/6	17/0/11		22/0/6	18/0/10	

したときに，Friedman 検定で得られた平均順位を表 9.3 に示す．なお，全ケースで有意差を検出した．全体の傾向として，拡張手法が最高順位を多く導出し．$D = 10$ を除き拡張手法は最高順位を譲っていないことから，限られた解評価回数であっても，また十分な解評価回数まで許容しても，拡張手法の有効性が確認できる．特に，$D = 10, 30, 50, 100$ と次元数が増加するにつれて平均順位の差が大きくなることから，拡張手法は次元数に対する性能のスケーラビリティを十分に有すると言える．

表 9.2　解評価回数 100,000 回時点の性能比較（$D \in \{10, 30, 50, 100\}$）

	$D = 10$			$D = 30$			$D = 50$			$D = 100$		
ID	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours
F1	**0.00E+00** ~	2.26E-16 +	2.35E-30	6.34E-19 +	3.74E-10 +	**4.69E-22**	2.08E-13 +	3.53E-11 +	**9.52E-20**	4.83E-06 +	**2.19E-10** −	9.89E-10
F2	7.14E-03 +	5.27E+01 +	**2.53E-17**	2.97E+05 +	4.71E+05 ~	**1.45E+05**	3.45E+06 +	1.99E+06 +	**8.34E+05**	1.68E+07 +	6.19E+06 +	**5.03E+06**
F3	3.90E+02 +	6.87E+02 +	**3.28E+01**	8.29E+01 ~	8.08E+01 ~	**3.06E+01**	4.46E+04 ~	5.89E+03 ~	**1.92E+03**	2.07E+06 +	3.78E+04 +	**9.07E+02**
F4	**2.87E-04** −	9.74E-01 +	1.55E-02	**7.05E+02** ~	7.41E+03 +	1.54E+03	8.96E+03 +	2.68E+04 +	**7.09E+03**	6.28E+04 +	1.11E+05 +	**3.97E+04**
F5	**1.66E-35** −	1.56E-12 +	1.65E-25	6.43E-14 +	2.07E-05 +	**2.12E-14**	2.20E-08 +	3.56E-04 ~	**4.65E-10**	2.03E-02 +	4.25E-04 ~	**3.57E-04**
F6	7.49E+00 +	**1.93E-01** −	6.34E+00	1.93E+01 +	1.47E+01 +	**1.44E+01**	4.58E+01 +	4.42E+01 ~	**4.41E+01**	2.42E+02 +	**1.97E+02** ~	2.12E+02
F7	5.43E-01 ~	**3.95E-01** ~	4.61E-01	**8.80E+00** ~	2.78E+01 +	1.18E+01	3.68E+01 ~	7.68E+01 +	**3.53E+01**	1.17E+02 +	1.24E+02 +	**9.25E+01**
F8	2.04E+01 +	**2.02E+01** ~	2.03E+01	2.10E+01 +	**2.09E+01** ~	**2.09E+01**	2.12E+01 +	**2.11E+01** ~	**2.11E+01**	**2.13E+01** +	**2.13E+01** ~	**2.13E+01**
F9	1.58E+00 ~	1.52E+00 ~	**1.37E+00**	**1.07E+01** −	1.59E+01 −	1.79E+01	**3.04E+01** −	3.91E+01 −	4.46E+01	**9.15E+01** −	1.05E+02 −	1.14E+02
F10	1.56E-01 +	6.29E-02 +	**3.12E-02**	3.30E-02 ~	**3.27E-02** ~	**3.27E-02**	3.84E-01 +	3.33E-02 ~	**2.99E-02**	1.81E+01 +	1.87E+00 ~	**1.85E+00**
F11	**1.49E-25** −	3.30E-10 +	3.74E-24	7.14E+00 +	2.52E+01 +	**1.16E-03**	4.41E+01 +	9.67E+01 +	**8.01E+00**	2.54E+02 +	4.15E+02 +	**1.27E+02**
F12	1.23E+01 +	8.69E+00 +	**5.93E+00**	4.51E+01 +	4.25E+01 +	**2.97E+01**	9.57E+01 ~	1.23E+02 +	**8.00E+01**	4.70E+02 +	4.39E+02 +	**3.34E+02**
F13	1.51E+01 +	1.50E+01 +	**6.57E+00**	9.26E+01 +	8.88E+01 +	**6.69E+01**	2.37E+02 +	2.21E+02 +	**1.82E+02**	8.52E+02 +	6.68E+02 +	**5.88E+02**
F14	2.87E-01 +	4.10E-01 +	**6.00E-02**	3.30E+02 +	6.42E+02 +	**5.62E+01**	1.18E+03 +	2.72E+03 +	**4.32E+02**	9.00E+03 +	1.45E+04 +	**6.22E+03**
F15	9.33E+02 +	7.88E+02 ~	**7.11E+02**	**4.13E+03** −	5.02E+03 +	4.54E+03	8.79E+03 ~	9.79E+03 +	**8.69E+03**	2.33E+04 +	1.98E+04 +	**1.83E+04**
F16	2.88E-01 ~	2.00E-01 ~	**1.88E-01**	2.27E+00 +	**5.01E-01** ~	5.17E-01	3.21E+00 +	**9.16E-01** ~	9.65E-01	3.96E+00 +	1.68E+00 ~	**1.57E+00**
F17	1.05E+01 +	**1.01E+01** +	**1.01E+01**	5.93E+01 +	5.64E+01 +	**3.15E+01**	1.51E+02 +	1.51E+02 +	**6.42E+01**	5.61E+02 +	5.37E+02 +	**2.49E+02**
F18	1.93E+01 ~	2.24E+01 +	**1.78E+01**	1.91E+02 +	1.17E+02 +	**7.04E+01**	3.92E+02 +	2.11E+02 +	**1.31E+02**	9.89E+02 +	6.11E+02 +	**3.41E+02**
F19	7.88E-01 ~	8.26E-01 +	**6.55E-01**	8.19E+00 +	5.43E+00 +	**3.76E+00**	2.41E+01 +	1.35E+01 +	**1.08E+01**	7.81E+01 +	5.95E+01 +	**4.16E+01**
F20	**2.39E+00** ~	2.69E+00 ~	2.54E+00	1.18E+01 ~	1.24E+01 +	**1.17E+01**	2.18E+01 +	2.20E+01 +	**2.13E+01**	**5.00E+01** ~	**5.00E+01** ~	**5.00E+01**
F21	3.96E+02 ~	**3.51E+02** ~	3.92E+02	3.14E+02 +	2.98E+02 +	**2.84E+02**	8.39E+02 ~	**7.56E+02** ~	8.24E+02	4.37E+02 +	**3.75E+02** −	3.95E+02
F22	9.88E+01 +	1.02E+02 +	**7.73E+01**	3.86E+02 +	1.11E+03 +	**2.59E+02**	1.49E+03 +	3.20E+03 +	**6.08E+02**	1.02E+04 +	1.55E+04 +	**7.73E+03**
F23	1.03E+03 +	9.29E+02 +	**7.98E+02**	**4.55E+03** ~	5.47E+03 +	4.74E+03	**8.80E+03** ~	1.07E+04 +	9.05E+03	2.41E+04 +	2.32E+04 +	**2.07E+04**
F24	**2.01E+02** ~	2.03E+02 ~	2.05E+02	**2.19E+02** −	2.43E+02 +	2.35E+02	**2.37E+02** −	2.95E+02 +	2.74E+02	**3.22E+02** −	4.30E+02 +	3.42E+02
F25	**2.02E+02** ~	**2.02E+02** ~	2.03E+02	**2.41E+02** −	2.61E+02 ~	2.61E+02	**2.94E+02** −	3.39E+02 ~	3.37E+02	**4.34E+02** −	5.59E+02 +	5.49E+02
F26	1.88E+02 −	**1.85E+02** −	1.89E+02	**2.02E+02** −	2.11E+02 ~	2.11E+02	**3.07E+02** −	3.63E+02 ~	3.63E+02	**4.41E+02** −	5.63E+02 ~	5.42E+02
F27	**4.06E+02** ~	4.10E+02 ~	**4.06E+02**	**5.14E+02** −	7.29E+02 ~	7.15E+02	**7.95E+02** −	1.28E+03 ~	1.31E+03	**1.55E+03** −	3.07E+03 +	2.76E+03
F28	2.96E+02 ~	**2.65E+02** −	2.73E+02	**3.00E+02** +	**3.00E+02** +	**3.00E+02**	7.49E+02 +	8.78E+02 ~	**4.57E+02**	**3.25E+03** ~	3.83E+03 +	3.28E+03
+/−/~	12/4/12	15/3/10		16/6/6	19/1/8		17/5/6	16/1/11		21/5/2	18/3/7	

表 9.3　全問題での平均順位（$D \in \{10, 30, 50, 100\}$）

	$D = 10$			$D = 30$			$D = 50$			$D = 100$		
解評価回数	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours	HMJCDE	EDEV	Ours
300	2.679	**1.393**	1.929	2.625	1.804	**1.571**	2.750	1.714	**1.536**	2.500	1.911	**1.589**
500	2.821	1.679	**1.500**	2.839	1.643	**1.518**	2.750	1.857	**1.393**	2.607	2.071	**1.321**
1,000	2.750	1.857	**1.393**	2.786	1.964	**1.250**	2.929	1.821	**1.250**	2.821	2.036	**1.143**
3,000	2.643	2.071	**1.286**	2.536	2.107	**1.357**	2.571	2.071	**1.357**	2.643	2.143	**1.214**
5,000	2.000	2.714	**1.286**	2.071	2.643	**1.286**	2.143	2.536	**1.321**	2.214	2.643	**1.143**
10,000	**1.536**	2.929	**1.536**	1.750	2.893	**1.357**	2.143	2.571	**1.286**	2.214	2.607	**1.179**
30,000	**1.536**	2.857	1.607	1.679	2.964	**1.357**	1.929	2.714	**1.357**	2.214	2.571	**1.214**
50,000	1.875	2.750	**1.375**	1.857	2.857	**1.286**	2.071	2.643	**1.286**	2.286	2.536	**1.179**
100,000	2.125	2.536	**1.339**	2.214	2.321	**1.464**	2.071	2.321	**1.607**	2.393	2.179	**1.429**

第 10 章

拡張手法の考察

10.1　インターバルの導入

拡張手法では，毎世代で事前検証を行った．一方で，既存のアンサンブル適応 DE は全て，解への適応 DE の割り当てを一定世代おきに行う．ここでは，拡張手法に事前検証を実行するインターバル世代数を表すハイパーパラメータ I を設けて，その性能を比較する．アルゴリズムは，初期化で全ての解を評価した後に I 世代の更新をし，以降は事前検証と I 世代の更新を繰り返すように変更される．$I \in \{1, 2, 3, 4, 5, 10, 15, 20\}$ で実験を行うが，第 9 章の実験での設定は $I = 1$ と表せる．これにより，

- 例えば，解評価回数 1,000 回における $I \in \{10, 15, 20\}$ など，事前検証がまだ成されていない世代における事前検証の有無が与える性能への影響
- 事前検証よりも各適応 DE による事後検証の頻度を相対的に高めた際の性能変化

の確認ができる．

実験結果を表 10.1 に示す．300 回という極めて少ない解評価回数では，$D \in \{10, 30, 50\}$ で $I = 1$ が低い順位となっている．これは, 事前検証型の適応法が局所探索を促進したとしても, 探索序盤で解同士の距離があまりに遠いと収束速度を高められない可能性を示し, 今後はこの点の手法改良に取り組む．一方で，解評価回数 500 回での $I \in \{3, 4, 5, 10, 15, 20\}$ や 1,000 回での $I \in \{10, 15, 20\}$ と比較した際には，$I \in \{1, 2\}$ が高い順位であることから，事前検証の効果を確認できる．全体としては，デフォルト設定である $I = 1$ が高い順位となる傾向にある．特に，次元数 D が高くなるほど $I = 1$ が最も高い順位を導出することから，次元数が高くなり問題が複雑になるほど，頻繁に事前検証を行うべきであることがわかる．また，この結果は，第 8 章で述べたように，事後検証は High-level アンサンブ

表 10.1　事前検証のインターバル I に関する全問題での平均順位（$D \in \{10, 30, 50, 100\}$）

	$D = 10$								$D = 30$							
解評価回数	$I = 1$	2	3	4	5	10	15	20	$I = 1$	2	3	4	5	10	15	20
300	5.500	**4.357**	**4.357**	**4.357**	**4.357**	**4.357**	**4.357**	**4.357**	5.000	**4.429**	**4.429**	**4.429**	**4.429**	**4.429**	**4.429**	**4.429**
500	**3.375**	4.446	4.696	4.696	4.696	4.696	4.696	4.696	**2.821**	4.143	4.839	4.839	4.839	4.839	4.839	4.839
1,000	**2.911**	3.054	3.321	4.179	5.607	5.643	5.643	5.643	**2.321**	2.429	3.196	4.411	5.268	6.125	6.125	6.125
3,000	**2.786**	3.107	4.107	3.643	4.857	5.214	6.536	5.75	**2.357**	3.214	4.018	4.500	4.679	4.429	6.446	6.357
5,000	**3.000**	3.429	3.714	4.357	4.786	5.929	5.25	5.536	**2.536**	3.679	4.357	3.786	4.964	5.893	4.929	5.857
10,000	**3.500**	4.071	3.643	4.464	4.571	5.286	5.607	4.857	**2.679**	2.929	3.964	4.214	5.607	6.107	5.357	5.143
30,000	4.107	**3.607**	4.179	4.000	4.321	4.679	5.393	5.714	3.179	**3.036**	4.250	3.679	4.857	5.964	5.679	5.357
50,000	**3.036**	3.214	3.607	4.179	4.536	5.286	6.071	6.071	3.857	**3.107**	4.393	3.964	4.429	5.679	5.607	4.964
100,000	**3.036**	3.286	3.571	4.357	4.679	5.393	5.821	5.857	3.286	**3.179**	3.964	4.214	5.036	5.357	5.786	5.179

	$D = 50$								$D = 100$							
解評価回数	$I = 1$	2	3	4	5	10	15	20	$I = 1$	2	3	4	5	10	15	20
300	5.000	**4.429**	**4.429**	**4.429**	**4.429**	**4.429**	**4.429**	**4.429**	**3.625**	4.625	4.625	4.625	4.625	4.625	4.625	4.625
500	**2.571**	4.607	4.804	4.804	4.804	4.804	4.804	4.804	**2.018**	4.304	4.946	4.946	4.946	4.946	4.946	4.946
1,000	2.625	**2.464**	2.875	4.161	5.714	6.054	6.054	6.054	**1.911**	3.161	3.304	4.571	5.643	5.804	5.804	5.804
3,000	3.429	**3.107**	3.536	4.143	4.643	4.643	5.875	6.625	**2.196**	3.482	3.661	4.857	4.643	3.768	6.518	6.875
5,000	**3.321**	3.500	3.429	4.643	5.375	5.732	4.393	5.607	**2.446**	3.875	4.054	4.839	5.946	5.875	3.946	5.018
10,000	**2.929**	3.286	4.179	4.429	5.393	6.107	5.250	4.429	**2.768**	3.304	3.661	4.982	5.732	6.482	5.089	3.982
30,000	**2.750**	3.214	3.964	4.393	5.107	6.357	5.821	4.393	**2.518**	3.589	3.625	4.554	5.125	6.625	5.125	4.839
50,000	**3.143**	3.857	4.143	4.321	5.214	6.286	5.107	3.929	**3.268**	3.304	3.625	4.589	5.089	6.482	5.196	4.446
100,000	**3.714**	3.875	4.089	4.946	5.000	5.250	4.964	4.161	**3.554**	3.946	4.196	4.911	5.125	5.375	4.554	4.339

ルのフレームワークではなく，Low-level アンサンブルである各適応 DE のみが行えば良い可能性を示唆している．

10.2　適応 DE によるハイパーパラメータ生成の確率的揺らぎの考慮

拡張研究では，アンサンブル適応 DE において，事前検証が純粋な High-level アンサンブルに与える効果を明らかにするために，各適応 DE を一つずつ生成してスクリーニングした．しかしながら，各適応 DE は DE のハイパーパラメータを確率的に生成するため，そもそも生成された適応 DE が持つハイパーパラメータが好適でない可能性がある．そこで，ハイパーパラメータの確率的揺らぎを軽減するために，Algorithm 11 における ♣ で，各適応 DE あたり追加ハイパーパラメータ M 個の候補を生成して，拡張手法を改良する．$M \in \{1, 2, 3, 4, 5\}$ で実験を行うが，第 9 章の実験での設定は $M = 1$ と表せる．

実験結果を表 10.2 に示す．表 10.2 より，適応 DE を複数生成することで，好適でないパラメータの適応 DE のみが生成されるリスクを軽減し，様々な適応の選択肢を事前検証することが可能になり，$M = 1$ に設定した時より性能が向上することがわかる．なお，最

表 10.2　適応 DE 一つあたりの適応候補数 M に関する全問題での平均順位（$D \in \{10, 30, 50, 100\}$）

	$D = 10$					$D = 30$				
解評価回数	$M = 1$	2	3	4	5	$M = 1$	2	3	4	5
300	3.000	2.946	3.411	2.929	**2.714**	**2.661**	3.357	3.232	2.804	2.946
500	3.429	2.786	3.089	**2.643**	3.054	3.339	2.929	3.036	**2.661**	3.036
1,000	3.786	3.214	**2.464**	2.964	2.571	3.429	2.893	2.964	**2.857**	**2.857**
3,000	4.000	3.143	2.679	**2.429**	2.750	3.321	**2.607**	3.250	2.964	2.857
5,000	4.071	2.857	2.857	2.821	**2.393**	3.321	**2.750**	3.036	3.036	2.857
10,000	4.071	2.821	2.821	2.750	**2.536**	3.786	3.071	2.857	**2.571**	2.714
30,000	4.393	2.821	2.821	**2.357**	2.607	2.893	**2.786**	3.036	3.393	2.893
50,000	3.714	3.107	3.179	**2.429**	2.571	3.321	2.821	**2.679**	3.071	3.107
100,000	3.536	2.893	3.196	2.732	**2.643**	3.643	**2.607**	2.893	2.982	2.875

	$D = 50$					$D = 100$				
解評価回数	$M = 1$	2	3	4	5	$M = 1$	2	3	4	5
300	3.125	3.161	2.839	3.429	**2.446**	**2.589**	3.089	3.071	2.946	3.304
500	2.893	**2.625**	2.893	3.607	2.982	3.107	3.107	**2.393**	3.518	2.875
1,000	3.536	2.911	2.946	2.929	**2.679**	**2.571**	2.964	3.357	3.107	3.000
3,000	4.000	**2.571**	3.071	2.643	2.714	3.000	**2.250**	3.107	3.214	3.429
5,000	3.929	**2.214**	3.107	2.821	2.929	2.821	**2.393**	2.607	3.214	3.964
10,000	3.643	**2.429**	3.071	2.571	3.286	3.107	**2.321**	2.821	3.107	3.643
30,000	3.107	**2.286**	3.036	2.929	3.643	**2.500**	2.571	2.821	3.643	3.464
50,000	3.143	**2.250**	3.143	2.964	3.500	2.714	2.750	**2.536**	3.357	3.643
100,000	3.357	**2.750**	2.964	2.821	3.107	3.357	2.679	3.143	**2.607**	3.214

高順位を導出した回数の観点から，$D = 10$，30，50，100 でそれぞれ $M = 4$ または 5，2，2，1 または 2 が適切な設定と言える．次元数 D が増加するほど適応 DE の必要なサンプリング数が減る理由としては，適応 DE と仮子個体を数多くサンプリングしても，高次元空間では球面集中現象により各仮子個体間の距離の差異が大きく現れず，適応 DE 間の差異が大きく反映されないことが考えられる．この点に関しては，解のサンプリングをしたり，ミンコフスキー距離を用いて制御パラメータを変化させたりするなどの追加の検証が必要である．しかしながら，表 9.3 が示すように，高次元空間であっても事前検証そのものが有効である点は変わらない．

第 11 章

結論

11.1　本論文のまとめ

本論文では，「ハイパーパラメータやこれを持つアルゴリズムをいかに生成すべきか」よりも「生成されたハイパーパラメータやこれを持つアルゴリズムのうちどれを利用すべきか」という観点に着目した．

具体的には，前半では，適応差分進化法（適応 DE）のハイパーパラメータを事前検証する，一般的な適応 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，適応 DE の方法に則りハイパーパラメータ候補を複数生成する．続いて，優良解への局所探索の傾向を高めるハイパーパラメータを 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，提案手法を適応 DE の代表手法である jDE と SaDE, JADE に組み込むことで，特に 500 から 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，従来の適応 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．これは，適応 DE が不得意とする CEP のような問題に対しても，提案手法が展開できる可能性を示すものである．

後半では，前半の提案手法の拡張として，適応対象をハイパーパラメータだけでなく，ハイパーパラメータと適応 DE の種類の両方とした．具体的には，少ない解評価で高性能に働く事前検証型アンサンブル適応 DE を提案した．事前検証プロセスでは，既に発見した優良解をおよそ複製可能な適応 DE を，実際の解生成に先立って追加の解評価なしにスクリーニングした．これにより，High-level アンサンブルにおける，Low-level アンサンブルと比べた際の最適化性能の向上の可能性と，膨大化したアルゴリズム空間における探

索の困難さのトレードオフを緩和し，実験では解評価回数の多寡に関わらず高い性能を導出した．また，拡張手法の次元数に対するスケーラビリティも確認した．

11.2 今後の展望

まず，前半の提案手法で新たに導入されたハイパーパラメータ C に関する実験的な検証を行う．

また，CEP に対する代表的アプローチであるサロゲート進化計算を提案手法や拡張手法に融合することで，性能向上に必要な解評価回数をさらに削減することを目指す．さらに，実応用への展開を図るべく，提案手法と拡張手法を高計算コストな実問題に適用し有効性を検証するとともに，実問題に多く存在する多目的最適化や制約つき最適化へ拡張する．

研究の他の展開方法として，本論文で明らかにした事前検証型の適応の有効性を，サロゲート進化計算に応用する．進化計算にサロゲートを組み込んだ際，ハイパーパラメータの数は元の進化計算から大きく増加することが考えられる．例として，サロゲートや内部カーネルの種類，サロゲート内部のハイパーパラメータ，データセットサイズ，サロゲートの使用頻度・回数，データセットサイズ，サロゲートを構築する解空間の定義などが存在する．ここで，著者の事前検証型の適応法の利点として，解評価を用いない評価指標が決まれば，適応するハイパーパラメータの種類や型を問わない点が挙げられる．そこで，例えば，評価指標を「サロゲートモデルの精度」などとすることにより，事前検証型の適応法をサロゲート進化計算のハイパーパラメータ適応に応用することが可能である．

謝辞

本論文をまとめるにあたり，終始多大なるご指導とご教示をくださった，主任指導教員の中田雅也准教授に，心より感謝の意を表します．また，本論文の研究を進める上で，適切な助言をくださった研究室の皆様にも，この場を借りて感謝いたします．

本論文の副査をお引き受けくださった，落合秀樹教授，濱上知樹教授，島圭介准教授に深く感謝いたします．

公益財団法人 日新電機グループ社会貢献基金，公益財団法人 山田育英会，公益財団法人 村田学術振興財団，独立行政法人 日本学生支援機構には，主に経済面で温かいご支援をくださったことに御礼申し上げます。

最後に，陰ながらいつも支えてくれた家族に感謝します．

参考文献

[1] Songqing Shan and G Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Struct. Multidiscip. Optim.*, 41(2):219–241, March 2010.

[2] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. John Wiley & Sons, May 2009.

[3] Yoel Tenne and Chi-Keong Goh. *Computational Intelligence in Expensive Optimization Problems*. Springer Science & Business Media, March 2010.

[4] Marian Nemec, David W Zingg, and Thomas H Pulliam. Multipoint and Multi-Objective Aerodynamic Shape Optimization. *AIAA J.*, 42(6):1057–1065, June 2004.

[5] Handing Wang and Yaochu Jin. A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. *IEEE Trans. Cybern.*, 50(2):536–549, February 2020.

[6] Handing Wang, Yaochu Jin, and Jan O Jansen. Data-Driven Surrogate-Assisted Multiobjective Evolutionary Optimization of a Trauma System. *IEEE Trans. Evol. Comput.*, 20(6):939–952, December 2016.

[7] Chaoyue Wang, Chang Xu, Xin Yao, and Dacheng Tao. Evolutionary Generative Adversarial Networks. *IEEE Trans. Evol. Comput.*, 23(6):921–934, December 2019.

[8] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized Evolution for Image Classifier Architecture Search. In *AAAI Conf. Artif. Intell.*, volume 33, pages 4780–4789, July 2019.

[9] A E Eiben, R Hinterding, and Z Michalewicz. Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.*, 3(2):124–141, July 1999.

[10] A E Eiben, Zbigniew Michalewicz, M Schoenauer, and J E Smith. Parameter control in evolutionary algorithms. In *Parameter Setting in Evolutionary Algorithms*, pages 19–46.

Springer, Berlin, Heidelberg, 2007.

[11] A E Eiben and S K Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm Evol. Comput.*, 1(1):19–31, 2011.

[12] Giorgos Karafotias, Mark Hoogendoorn, and A E Eiben. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Trans. Evol. Comput.*, 19(2):167–187, April 2015.

[13] F J Lobo, Cláudio F Lima, and Zbigniew Michalewicz. *Parameter Setting in Evolutionary Algorithms*. Springer Science & Business Media, March 2007.

[14] James E Smith. Self-Adaptation in Evolutionary Algorithms for Combinatorial Optimisation. In Carlos Cotta, Marc Sevaux, and Kenneth Sörensen, editors, *Adaptive and Multilevel Metaheuristics*, pages 31–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[15] Chaoli Sun, Yaochu Jin, Ran Cheng, Jinliang Ding, and Jianchao Zeng. Surrogate-Assisted Cooperative Swarm Optimization of High-Dimensional Expensive Problems. *IEEE Trans. Evol. Comput.*, 21(4):644–660, August 2017.

[16] Rainer Storn. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Global Optimiz.*, 11:341–359, 1997.

[17] Ferrante Neri and Ville Tirronen. Recent advances in differential evolution: a survey and experimental analysis. *Artif. Intell. Rev.*, 33(1):61–106, February 2010.

[18] Kenneth Price, Rainer M Storn, and Jouni A Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer Science & Business Media, March 2006.

[19] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evol. Comput.*, 15(1):4–31, February 2011.

[20] Ryoji Tanabe and Alex Fukunaga. Reviewing and Benchmarking Parameter Control Methods in Differential Evolution. *IEEE Trans. Cybern.*, 50(3):1170–1184, March 2020.

[21] Roger Gämperle, Sibylle D Müller, and Petros Koumoutsakos. A parameter study for differential evolution. *Adv. Intell. Syst. Fuzzy Syst. Evol. Comput.*, 10(10):293–298, 2002.

[22] Efrñn Mezura-Montes, Jesús Velázquez-Reyes, and Carlos A Coello Coello. A comparative study of differential evolution variants for global optimization. In *Annu. Conf. Genet. Evol. Comput. (GECCO)*, GECCO '06, pages 485–492, New York, NY, USA,

July 2006. Association for Computing Machinery.

[23] Noha M Hamza, Daryl L Essam, and Ruhul A Sarker. Constraint Consensus Mutation-Based Differential Evolution for Constrained Optimization. *IEEE Trans. Evol. Comput.*, 20(3):447–459, June 2016.

[24] Nasser R Sabar and Masri Ayob. Examination timetabling using scatter search hyper-heuristic. In *Conf. Data. Min. Optimiz.*, pages 127–131, October 2009.

[25] Nandar Lynn, Rammohan Mallipeddi, and Ponnuthurai Nagaratnam Suganthan. Differential Evolution with Two Subpopulations. In *Swarm Evol. Memet. Comput.*, pages 1–13. Springer International Publishing, 2015.

[26] Genghui Li, Qiuzhen Lin, Laizhong Cui, Zhihua Du, Zhengping Liang, Jianyong Chen, Nan Lu, and Zhong Ming. A novel hybrid differential evolution algorithm with modified CoDE and JADE. *Appl. Soft Comput.*, 47:577–599, October 2016.

[27] Guohua Wu, Rammohan Mallipeddi, P N Suganthan, Rui Wang, and Huangke Chen. Differential evolution with multi-population based ensemble of mutation strategies. *Inf. Sci.*, 329:329–345, February 2016.

[28] Sheng Xin Zhang, Shao Yong Zheng, and Li Ming Zheng. An Efficient Multiple Variants Coordination Framework for Differential Evolution. *IEEE Trans Cybern*, 47(9):2780–2793, September 2017.

[29] Guohua Wu, Xin Shen, Haifeng Li, Huangke Chen, Anping Lin, and P N Suganthan. Ensemble of differential evolution variants. *Inf. Sci.*, 423:172–186, January 2018.

[30] Weifeng Gao, Gary G Yen, and Sanyang Liu. A cluster-based differential evolution with self-adaptive strategy for multimodal optimization. *IEEE Trans Cybern*, 44(8):1314–1327, August 2014.

[31] Zi-Jia Wang, Zhi-Hui Zhan, Ying Lin, Wei-Jie Yu, Hua Wang, Sam Kwong, and Jun Zhang. Automatic Niching Differential Evolution With Contour Prediction Approach for Multimodal Optimization Problems. *IEEE Trans. Evol. Comput.*, 24(1):114–128, February 2020.

[32] Rammohan Mallipeddi and Minho Lee. Surrogate model assisted ensemble differential evolution algorithm. In *IEEE Congr. Evol. Comput. (CEC)*, pages 1–8, June 2012.

[33] Xiaofen Lu, Ke Tang, Bernhard Sendhoff, and Xin Yao. A new self-adaptation scheme for differential evolution. *Neurocomputing*, 146:2–16, December 2014.

[34] Rammohan Mallipeddi and Minho Lee. An evolving surrogate model-based differential evolution algorithm. *Appl. Soft Comput.*, 34:770–787, September 2015.

[35] Mudita Sharma, Alexandros Komninos, Manuel López-Ibáñez, and Dimitar Kazakov. Deep reinforcement learning based parameter control in differential evolution. In *Annu. Conf. Genet. Evol. Comput. (GECCO)*, GECCO '19, pages 709–717, New York, NY, USA, July 2019. ACM.

[36] Xiao-Gen Zhou, Chun-Xiang Peng, Jun Liu, Yang Zhang, and Gui-Jun Zhang. Underestimation-Assisted Global-Local Cooperative Differential Evolution and the Application to Protein Structure Prediction. *IEEE Trans. Evol. Comput.*, 24(3):536–550, June 2020.

[37] Jianyong Sun, Xin Liu, Thomas Bäck, and Zongben Xu. Learning Adaptive Differential Evolution Algorithm From Optimization Experiences by Policy Gradient. *IEEE Trans. Evol. Comput.*, 25(4):666–680, August 2021.

[38] Janez Brest, Sao Greiner, Borko Boskovic, Marjan Mernik, and Viljem Zumer. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Trans. Evol. Comput.*, 10(6):646–657, December 2006.

[39] Yong Wang, Zixing Cai, and Qingfu Zhang. Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters. *IEEE Trans. Evol. Comput.*, 15(1):55–66, February 2011.

[40] Jingqiao Zhang and Arthur C Sanderson. JADE: Adaptive Differential Evolution With Optional External Archive. *IEEE Trans. Evol. Comput.*, 13(5):945–958, October 2009.

[41] A K Qin and P N Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *IEEE Congr. Evol. Comput. (CEC)*, volume 2, pages 1785–1791 Vol. 2, September 2005.

[42] A K Qin, V L Huang, and P N Suganthan. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Trans. Evol. Comput.*, 13(2):398–417, April 2009.

[43] Ryoji Tanabe and Alex Fukunaga. Success-history based parameter adaptation for Differential Evolution. In *IEEE Congr. Evol. Comput. (CEC)*, pages 71–78, June 2013.

[44] Janez Brest, Mirjam Sepesy Maučec, and Borko Bošković. Single objective real-parameter optimization: Algorithm jSO. In *IEEE Congr. Evol. Comput. (CEC)*, pages 1311–1318, June 2017.

[45] Zhenyu Yang, Ke Tang, and Xin Yao. Self-adaptive differential evolution with neighborhood search. In *IEEE Congr. Evol. Comput. (CEC)*, pages 1110–1116, June 2008.

[46] Swagatam Das, Sankha Subhra Mullick, and P N Suganthan. Recent advances in differential evolution – An updated survey. *Swarm Evol. Comput.*, 27:1–30, April 2016.

[47] J J Liang, B Y Qu, P N Suganthan, and Alfredo G Hernández-Díaz. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Technical report, Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China and Nanyang Tech. Univ., Singapore, 2013.

[48] P N Suganthan, N Hansen, J J Liang, K Deb, Y P Chen, A Auger, and S Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, omput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China and Nanyang Tech. Univ., Singapore, 2005.

[49] J Kennedy and R Eberhart. Particle swarm optimization. In *Int. Conf. Neural Netw.*, volume 4, pages 1942–1948 vol.4, November 1995.

[50] Rommel G Regis. Particle swarm with radial basis function surrogates for expensive black-box optimization. *J. Comput. Sci.*, 5(1):12–23, January 2014.

[51] Kei Nishihara and Masaya Nakata. Competitive-Adaptive Algorithm-Tuning of Metaheuristics inspired by the Equilibrium Theory: A Case Study. In *IEEE Congr. Evol. Comput. (CEC)*, pages 1–8, July 2020.

[52] J A Nelder and R Mead. A Simplex Method for Function Minimization. *Comput. J.*, 7(4):308–313, January 1965.

[53] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Annu. Conf. Genet. Evol. Comput. (GECCO)*, GECCO '10, pages 1689–1696, New York, NY, USA, July 2010. ACM.

[54] John H Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, April 1992.

[55] Yiqiao Cai and Jiahai Wang. Differential evolution with neighborhood and direction information for numerical optimization. *IEEE Trans Cybern*, 43(6):2202–2215, December 2013.

[56] 遼司 田邊. **関数最適化問題に対する適応型差分進化法の研究**. PhD thesis, 東京大学, 2016.

[57] Swagatam Das, Amit Konar, and Uday K Chakraborty. Two improved differential evolution schemes for faster global search. In *Annu. Conf. Genet. Evol. Comput. (GECCO)*, GECCO '05, pages 991–998, New York, NY, USA, June 2005. ACM.

[58] Amer Draa, Samira Bouzoubia, and Imene Boukhalfa. A sinusoidal differential evolution algorithm for numerical optimisation. *Appl. Soft Comput.*, 27:99–126, February 2015.

[59] Y Shi and R C Eberhart. Empirical study of particle swarm optimization. In *IEEE Congr. Evol. Comput. (CEC)*, volume 3, pages 1945–1950 Vol. 3, July 1999.

[60] J Liu and J Lampinen. A fuzzy adaptive differential evolution algorithm. *Soft Comput.*, 9(6):448–462, June 2005.

[61] P Kaelo and M M Ali. A numerical study of some modified differential evolution algorithms. *Eur. J. Oper. Res.*, 169(3):1176–1184, March 2006.

[62] Ville Tirronen and Ferrante Neri. Differential Evolution with Fitness Diversity Self-adaptation. In Raymond Chiong, editor, *Nature-Inspired Algorithms for Optimisation*, pages 199–234. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[63] Liyuan Jia, Wenyin Gong, and Hongbin Wu. An Improved Self-adaptive Control Parameter of Differential Evolution for Global Optimization. In *Comput. Intell. Intell. Syst.*, pages 215–224. Springer Berlin Heidelberg, 2009.

[64] R Mallipeddi, P N Suganthan, Q K Pan, and M F Tasgetiren. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput.*, 11(2):1679–1696, March 2011.

[65] Sk Minhazul Islam, Swagatam Das, Saurav Ghosh, Subhrajit Roy, and Ponnuthurai Nagaratnam Suganthan. An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Trans. Syst. Man Cybern. B Cybern.*, 42(2):482–500, April 2012.

[66] Zhiwei Zhao, Jingming Yang, Ziyu Hu, and Haijun Che. A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems. *Eur. J. Oper. Res.*, 250(1):30–45, 2016.

[67] Ryoji Tanabe and Alex S Fukunaga. Improving the search performance of SHADE using linear population size reduction. In *IEEE Congr. Evol. Comput. (CEC)*, pages 1658–1665, July 2014.

[68] Karam M Sallam, Ruhul A Sarker, Daryl L Essam, and Saber M Elsayed. Neurodynamic differential evolution algorithm and solving CEC2015 competition problems. In *IEEE Congr. Evol. Comput. (CEC)*, pages 1033–1040, May 2015.

[69] Noor H Awad, Mostafa Z Ali, Ponnuthurai N Suganthan, and Robert G Reynolds. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving

CEC2014 benchmark problems. In *IEEE Congr. Evol. Comput. (CEC)*, pages 2958–2965, July 2016.

[70] Noor H Awad, Mostafa Z Ali, and Ponnuthurai N Suganthan. Ensemble of parameters in a sinusoidal differential evolution with niching-based population reduction. *Swarm Evol. Comput.*, 39:141–156, April 2018.

[71] Janez Brest, Mirjam Sepesy Maučec, and Borko Bošković. iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization. In *IEEE Congr. Evol. Comput. (CEC)*, pages 1188–1195, July 2016.

[72] Subhodip Biswas, Souvik Kundu, Swagatam Das, and Athanasios V Vasilakos. Teaching and learning best Differential Evoltuion with self adaptation for real parameter optimization. In *IEEE Congr. Evol. Comput. (CEC)*, pages 1115–1122, June 2013.

[73] Tsz Ho Wong, A K Qin, Shengchun Wang, and Yuhui Shi. cuSaDE: A CUDA-Based Parallel Self-adaptive Differential Evolution Algorithm. In *Asia Pacific Symp. Intell. Evol. Syst.*, 2, pages 375–388. Springer International Publishing, 2015.

[74] Yong Wang, Han-Xiong Li, Tingwen Huang, and Long Li. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. *Appl. Soft Comput.*, 18:232–247, May 2014.

[75] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. *Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions*. PhD thesis, INRIA, 2009.

[76] K Tang, X Yao, P N Suganthan, C MacNish, Y P Chen, C M Chen, and Z Yang. Benchmark Functions for the CEC 2010 Special Session and Competition on Large Scale Global Optimization. Technical report, University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), Hefei, Anhui, 2010.

[77] Zhenyu Meng, Yuxin Zhong, Guojun Mao, and Yan Liang. PSO-sono: A novel PSO variant for single-objective numerical optimization. *Inf. Sci.*, 586:176–191, March 2022.

[78] Guohua Wu, Rammohan Mallipeddi, and Ponnuthurai Nagaratnam Suganthan. Ensemble strategies for population-based optimization algorithms – A survey. *Swarm Evol. Comput.*, 44:695–711, February 2019.

[79] D H Wolpert and W G Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, 1(1):67–82, April 1997.

発表文献

学術論文（全て査読あり）

1. **西原慧**，中田雅也："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"，情報処理学会論文誌「数理モデル化と応用」（TOM），Vol.14, No.3 pp.51–67, 2021.
2. 蛭田悠介, **西原慧**, 小熊祐司, 藤井正和, 中田雅也："Cartesian Genetic Programmingを用いた転用可能な積み付けアルゴリズムの自動生成"，情報処理学会論文誌「数理モデル化と応用」（TOM），Vol.14, No.3, pp.11–26, 2021.

国際会議における発表（全て口頭発表・査読あり）

3. **K.Nishihara** and M.Nakata:“Competitive Adaptive Algorithm Tuning of Metaheuristics inspired by the Equilibrium Theory: A Case Study”, IEEE Congr. Evol. Comput. (CEC), E-24156, July 2020.
4. **K.Nishihara** and M.Nakata: “Comparison of Adaptive Differential Evolution Algorithms on the MOEA/D-DE Framework”, IEEE Congr. Evol. Comput. (CEC), pp.161–168, June 2021.

国内学会・シンポジウム等における発表（全て査読なし）

5. **西原慧**，中田雅也："競争均衡原理に着想を得たメタヒューリスティクスの適応的アルゴリズム調整"，進化計算シンポジウム 2019, pp.37-44, 2019 年 12 月.（ポスター発表：ポスター発表のみ募集）
6. **西原慧**，中田雅也："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"，情報処理学会 第 131 回「数理モデル化と問題

解決」研究会（MPS），No.3, pp.1–6, 2020 年 12 月.（口頭発表）

7. 蛭田悠介，**西原慧**，小熊祐司，藤井正和，中田雅也："Cartesian genetic programming を用いた転移利用可能な積み付けアルゴリズムの自動生成"，情報処理学会 第 131 回「数理モデル化と問題解決」研究会（MPS），No.10, pp.1–6, 2020 年 12 月.（口頭発表）
8. 宮原悠司，園田拓海，**西原慧**，中田雅也："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"，進化計算シンポジウム 2020，pp.227–234, 2020 年 12 月.（口頭発表）
9. 蛭田悠介，**西原慧**，小熊祐司，藤井正和，中田雅也："遺伝的プログラミングを用いた積み付けアルゴリズムの対話的生成"，電気学会 電子・情報・システム部門（C 部門）大会，pp.197–202, 2021 年 9 月.（口頭発表）
10. **西原慧**，中田雅也："高計算コストな最適化問題に向けた事前検証型アンサンブル適応差分進化"，インテリジェント・システム・シンポジウム（FAN 2021），No.54, pp.132–137, 2021 年 9 月.（口頭発表，**優秀論文賞 受賞**）
11. 針谷亘輝，**西原慧**，中田雅也："視覚情報に基づく進化的関数同定の検討"，進化計算シンポジウム 2021，pp.1–8, 2021 年 12 月.（口頭発表）