

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

西原 慧[†] 中田 雅也[†]

横浜国立大学[†]

1 はじめに

メタヒューリスティクスは、そのアルゴリズムの構成要素であるハイパーパラメータや遺伝的操作について選定方針が不明確である場合が多い。ノーフリーランチ定理¹²⁾が指摘するように、これらの構成要素を適切に選定することは、そのアルゴリズムを問題に特化させ最適化の性能を高めることが期待できる^{4, 6)}。そこで、ハイパーヒューリスティクスをはじめとして、最適化アルゴリズムの自動設計やアルゴリズムの適応的調整など、問題や探索過程に応じてアルゴリズムを自律的に特化させる研究が盛んに行われている。

これらの方法論は、アルゴリズムを試行錯誤的に特化させながら性能の改善を目指すという点で挑戦的である一方、通常メタヒューリスティクスよりもはるかに大きく複雑な探索空間を扱う困難さがある。したがって、メタヒューリスティクスと同一の基準(同じ評価回数)で性能を比較するとき、アルゴリズムを自律的に特化させる方法の性能が著しく劣る場合は極めて多い。そこで、ハイパーヒューリスティクスなどでは、調整用問題を設定し、評価回数の制限をなくした想定でアルゴリズムを自動生成および選定する場合もある。しかしながら、シミュレーションを用いて解を評価する高計算コストな最適化問題では評価回数が制限されるだけでなく、調整用問題(類似問題)の設計が困難である場合も実社会で多く存在する。したがって、実用的な観点から考えると、通常メタヒューリスティクスが適用される同一の基準のもとで、アルゴリズムを試行錯誤的に特化させながらメタヒューリスティクスを超える性能を導出することが究極的な目標となる。

この目標を達成するために、メタヒューリスティクスのハイパーパラメータの調整に焦点を当てることで、アルゴリズム調整の自由度を小さくする方法が多く存在する。例えばMetaEAのように、進化的な調整方法に加え、正規分布やコーシー分布からハイパーパラメータ値をサンプリングする方法^{13, 5, 10)}や、深層強化学習を用いた方法⁹⁾などが提案されている。これらの方法は、ハイパーパラメータを事前に定義する通常メタヒューリスティクスよりも優れた最適化性能を導出している。これらの成功を受け、近年ではハイパーパラメータだけでなく、突然変異戦略や交叉戦略などの遺伝的操作も適応的に選択するように、アルゴリズム調整の自由度を高める研究動向がある。特に、解生成に直接的に関連する遺伝的操作は、問題に特化する度合をさらに高めることができ、その適切な調整ができれば最適化の性能がさらに改善することが期待できる。例えば、文法構造に発想を得たGrammatical Evolution^{8, 7, 1)}などがあり、[Mirand+, 2018]ではPSOのアルゴリズムを遺伝的プログラミングで適応的に調整する方法が提案されている。本研究では、この動向を受けて、ハイパーパラメータと遺伝的操作の両方を対象とする自由度の高いアルゴリズム調整技術に焦点を当てる。

しかしながら、先に述べた通り、アルゴリズム調整の自由度を高めることで期待する性能の改善効果は、探索空間の膨大化によってその達成が困難となる。加えて、ハイパーパラメータと遺伝的操作の調整については、その操作で用いるハイパーパラメータの組合せによって挙動が変化するため、ハイパーパラメータのみを調整する場合と比べて、調整の複雑さは各段に増す⁷⁾。既存手法では、この解析困難な調整に関する複雑さに対し、遺伝的プログラミングなどの進化的な調整方法を用いている。しかしながら、進化的な調整方法は、1)アルゴリズムの調整に

Competitively Adaptive Algorithm Tuning inspired by Equilibrium Theory

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

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

Yokohama National University (†)

も解評価を用いるため評価回数が増える。また、2)他のアルゴリズムと比較しながら調整しないため効率的にアルゴリズム調整ができない。

そこで本研究では、進化的な調整法とは異なり、競争均衡原理¹¹⁾に着想を得た、複数のアルゴリズムを競合させる調整技術を提案する。競争均衡原理とは、「外乱のない一定条件で競争を繰り返すと均衡最適状態に収束する」という経済学における理論である。これを受けて、提案手法におけるアルゴリズム調整の原理は、「ある一定の世代で探索を中断し、新たな解評価を行わずとも複数のアルゴリズムを競合させて、より適切なアルゴリズムに調整する」というものである。具体的には、ある世代で複数のアルゴリズムによる探索を中断し、導出された解個体に基づき順位付けを行う。そして、順位が高いアルゴリズムは保存し、順位が低いアルゴリズムは上位個体を目標として自身の構成要素を調整する。この結果、1)アルゴリズム調整に追加の解評価を要さないこと、2)順位が低いアルゴリズムは教師データ(調整の目標となる上位個体)をもとに新しいアルゴリズムを生成できる。言い換えると、従来の進化的な調整方法に対し、解評価を用いずに試行錯誤的なアルゴリズム調整が行えるとともに、教師あり学習に近いフレームワークによって妥当性の高いアルゴリズム調整が行えることが期待できる。

本論文の構成は次の通りである。第2章ではソルバーとして機能するDEとそれに対する従来のパラメータ調整方法を要素技術として紹介し、アルゴリズムの構成要素の特徴も同時に述べる。第3章では、提案手法の詳細について述べる。第4章でベンチマーク問題における性能結果を示し、第5章で考察を述べる。最後に、第6章で結論と今後の展望を述べる。

2 差分進化法

本章では、提案手法で調整の対象となるDEについて説明する。なお、第4章で行う実験で用いた、ハイパーパラメータの適応的調整を行うSelf-Adaptive DE(jDE)²⁾についても説明する。

2.1 DEのアルゴリズム

初期化 まず、世代数 $t = 0$ において、個体数 N_P の数だけ初期解を生成する。具体的には、 i 番目の個体 $\mathbf{x}^i, i \in \{1 \dots N_P\}$ ($i = 1, \dots, N_P$) に対し、 j 次元目の要素 $x_j^i (j = 1, \dots, D)$ を一様分布乱数

Table 1 DEにおける代表的な突然変異戦略

名称	定義式
rand/1	$\mathbf{v}^{i,t} = \mathbf{x}^{r_1,t} + F(\mathbf{x}^{r_2,t} - \mathbf{x}^{r_3,t})$
rand/2	$\mathbf{v}^{i,t} = \mathbf{x}^{r_1,t} + F(\mathbf{x}^{r_2,t} - \mathbf{x}^{r_3,t}) + F(\mathbf{x}^{r_4,t} - \mathbf{x}^{r_5,t})$
best/1	$\mathbf{v}^{i,t} = \mathbf{x}^{best,t} + F(\mathbf{x}^{r_1,t} - \mathbf{x}^{r_2,t})$
best/2	$\mathbf{v}^{i,t} = \mathbf{x}^{best,t} + F(\mathbf{x}^{r_1,t} - \mathbf{x}^{r_2,t}) + F(\mathbf{x}^{r_3,t} - \mathbf{x}^{r_4,t})$
current-to-rand/1	$\mathbf{v}^{i,t} = \mathbf{x}^{i,t} + F(\mathbf{x}^{r_1,t} - \mathbf{x}^{i,t}) + F(\mathbf{x}^{r_2,t} - \mathbf{x}^{r_3,t})$
current-to-best/1	$\mathbf{v}^{i,t} = \mathbf{x}^{i,t} + F(\mathbf{x}^{best,t} - \mathbf{x}^{i,t}) + F(\mathbf{x}^{r_1,t} - \mathbf{x}^{r_2,t})$
current-to-pbest/1	$\mathbf{v}^{i,t} = \mathbf{x}^{i,t} + F(\mathbf{x}^{pbest,t} - \mathbf{x}^{i,t}) + F(\mathbf{x}^{r_1,t} - \mathbf{x}^{r_2,t})$

rand[0, 1]を用いて式(1)より生成する。ここで、 D は問題次元数である。

$$x_j^{i,t=0} = (x_{\max} - x_{\min})\text{rand}[0, 1] + x_{\min} \quad (1)$$

突然変異 次に、 t 世代目の i 番目の個体の突然変異個体 $\mathbf{v}^{i,t}$ を生成する。現在では、Table 1にまとめるように代表的な突然変異戦略は複数提案されている。ここで、Table 1にまとめた突然変異戦略について、パラメータ F は差分ベクトルの寄与率を調整するものであり、 F 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効果がある。本研究では、Table 1に示す7種類の突然変異戦略をDEの調整時における選定候補と定める。rand/1や、rand/2、current-to-rand/1は、ランダム性が強いいため大域探索を行う効果が強い。一方で、best/1、best/2、current-to-best/1は、世代 t における最良解に近づけ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、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、current-to-pbest/1は、親個体付近に個体を生成する傾向がある。なお、current-to-pbest/1で用いるパラメータ p は、局所探索と大域探索を調整可能にするパラメータであり、本論文では簡単のため1/2に固定する。

交叉 交叉戦略も複数ありbinomial交叉(Algorithm 1)とexponential交叉(Algorithm 2)について説明する。突然変異個体 $\mathbf{v}^{i,t}$ を生成後、次元カウンタ j を走らせながら親個体 $\mathbf{x}^{i,t}$ と交叉させ、完了後の

Algorithm 1 binomial 交叉

```
 $j_{rand} = \text{randint}[1, D]$ 
for  $j = 1$  to  $D$  do
  if  $\text{rand}[0, 1] < CR$  or  $j == j_{rand}$  then
     $u_j^{i,t} = v_j^{i,t}$ 
  else
     $u_j^{i,t} = x_j^{i,t}$ 
  end if
end for
```

Algorithm 2 exponential 交叉

```
 $j = \text{randint}[1, D]$ 
 $k = 1$ 
 $\mathbf{u}^{i,t} = \mathbf{x}^{i,t}$ 
repeat
   $u_j^{i,t} = v_j^{i,t}$ 
   $j = (j + 1) \bmod D$ 
   $k = k + 1$ 
until  $\text{rand}[0, 1] \geq CR$  or  $k \geq D$ 
```

ベクトルを $\mathbf{u}^{i,t}$ とする。その際、交叉率パラメータ CR を用いるが、全く交叉されない可能性を排除するために $[1, D]$ から整数乱数 j_{rand} を決め、 j_{rand} 次元目は必ず交叉させる。なお、 CR は突然変異個体の決定変数をどの程度寄与させるかを制御する役割がある。例えば、 $CR = 0.0$ では、Algorithm 1 の j_{rand} あるいは Algorithm 2 のループ初回の影響により、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¹⁴⁾。

選択 次に、式(2)に示す通り、個体 $\mathbf{u}^{i,t}$ の評価値が元の個体 $\mathbf{x}^{i,t}$ よりも小さい場合に $\mathbf{x}^{i,t+1}$ を $\mathbf{u}^{i,t}$ に設定する。そして、突然変異個体の生成に戻り一連の遺伝的操作を繰り返す。この操作を探索終了条件(例：最大評価回数)まで行う。

$$\mathbf{x}^{i,t+1} = \begin{cases} \mathbf{u}^{i,t}, & f(\mathbf{u}^{i,t}) \leq f(\mathbf{x}^{i,t}) \\ \mathbf{x}^{i,t}, & \text{otherwise} \end{cases} \quad (2)$$

2.2 jDE のアルゴリズム

通常の DE では F, CR が全個体かつ探索中に終始一定であったのに対し、jDE では F, CR のパラメータを適応的に変更する。具体的には、 F, CR が各個体 \mathbf{x}^i ごとに固有の値 F^i, CR^i を持

ち、それぞれの値が突然変異・交叉時に更新される。まず、すべての F^i, CR^i をそれぞれ初期値 $F^i = 0.5, CR^i = 0.9$ に設定し、各個体に対し一定確率 τ_F で F^i を一時的に $\text{rand}[0.1, 1]$ に変更した値を $F_{tmp}^{i,t}$ とする。なお、変更されない場合はそのまま $F^{i,t}$ を引き継ぐことになる。次に、 $F_{tmp}^{i,t}$ を用いて突然変異個体 $\mathbf{v}^{i,t}$ を生成する。同様に一定確率 τ_{CR} で CR^i を一時的に $\text{rand}[0, 1]$ に変更した $CR_{tmp}^{i,t}$ を用いて $\mathbf{u}^{i,t}$ を生成する。そして、 $\mathbf{u}^{i,t}$ の評価値が親個体 $\mathbf{x}^{i,t}$ よりも優れた場合に、 $F_{tmp}^{i,t}$ と $CR_{tmp}^{i,t}$ を採用し次世代に用いる。 $\mathbf{u}^{i,t}$ の評価値が改善されない場合は、 $F_{tmp}^{i,t}$ と $CR_{tmp}^{i,t}$ は破棄される。したがって、「問題や探索状況に適したパラメータを用いたことにより優れた子個体の生成が行えた」という仮定のもと、過去の探索において有用であったパラメータ F^i, CR^i を引き継ぎながら、一定確率で新しいパラメータを試行錯誤的に求める。

3 競争均衡原理に基づくアルゴリズム調整

提案手法では、複数のアルゴリズム (DE) を用意しそれらが生成する個体を定期的に競合させる点に着想を得ており、この競合の原理を活かすことで DE の構成要素であるハイパーパラメータおよび突然変異戦略、交叉戦略を適応的に調整する。具体的には、一定の世代ごとに全アルゴリズムが導出した解個体の順位をつける。そして、順位が高い個体を輩出したアルゴリズムは、評価値の高い解個体が生成できたことを根拠に、それらのアルゴリズムの構成要素が解探索に有効であると判定し調整しない。一方で、順位が低い個体を生成したアルゴリズムはその構成要素を調整することで、解探索に有効なアルゴリズムとなるように構築する。このために、順位が高い(調整の目標となる)解個体に近い解を現在の解集合から生成できるまで、アルゴリズムの構成要素を調整する。

3.1 メカニズム

提案手法では、DE のハイパーパラメータ F, CR 、Table 1 に示す7種類の突然変異戦略、binomial 交叉と exponential 交叉の2種類の交叉戦略を調整対象とする。 F, CR については実数値 $[0, 1]$ の範囲で調整を行う。また、突然変異戦略と交叉戦略については、扱う各戦略の種類についてそれぞれ1つの戦略を選定する。また、用いるアルゴリズムの数を n とする。本稿では、 n 個のアルゴリズム

Algorithm 3 提案手法

```
Initialize  $\mathbf{x}$ ,  $\mathbf{A}$ 
while TerminationCriteria == false do
  for  $i = 1$  to  $I$  do
     $\mathbf{x} = \text{GenerationUpdate}(\mathbf{A})$ 
  end for
   $\mathbf{w} = \text{Competition}(\mathbf{x})$ 
  for  $i = 1$  to  $n$  do
     $A_i = \text{Tuning}(A_i, \mathbf{x}, w_i)$ 
  end for
end while
```

ムは全てDEに設定する。

Algorithm 3に示す提案手法の流れを説明する。また、複数のアルゴリズムを集約した集合を \mathbf{A} とする。最初に、初期解 \mathbf{x} と複数のアルゴリズムをそれぞれ初期化する(Initialize)。次に、 $\text{GenerationUpdate}(\mathbf{A})$ として示すように、複数のアルゴリズムを用いて解集合を生成する。この操作を、 I 世代だけ繰り返すことで、調整されたアルゴリズムによって評価値が高い解個体が生成できるかを検証する。そして、 $\text{Competition}(\mathbf{x})$ として示すように、生成された解集合 \mathbf{x} を用いて優良解 \mathbf{w} を得る。続いて、 $\text{Tuning}()$ として表記されるように、順位が低いアルゴリズムは選定された優良解に近い解を生成できるまで、その構成要素の調整を行う。したがって、 I 世代ごとに検証期間としてのインターバルを設け、アルゴリズム調整を行うことになる。そして再び、 $\text{GenerationUpdate}(\mathbf{A})$ に戻り、調整されたアルゴリズムを用いて解生成を行う。この一連の操作を繰り返すことで、アルゴリズム調整を行いながら、評価値の高い解の生成を目指す。

以下には、提案手法のコンポーネントとなる1)初期化(Initialize)、2)検証(GenerationUpdate)、3)競合的調整(Competition, Tuning)について詳細を述べる。

3.1.1 初期化

初期解の生成は通常のDEと同様に、式(1)を用いて生成する。初期アルゴリズム \mathbf{A} の生成にあたり、ハイパーパラメータに関しては、それらの規定値($F = 0.5, CR = 0.9$)に設定する。ただし、突然変異戦略と交叉戦略については、アルゴリズムとしての多様性を担保するためにそれぞれ異なる戦略を設定する。そこで、 n 個のアルゴリズムに対し、突然変異戦略と交叉戦略の組合せが異なるように初期設定する。

3.1.2 検証

このコンポーネントは、通常のDEにおける解探索に対応するが、提案手法では、調整されたアルゴリズムが評価値の高い解が生成できるかを検証する役割がある。調整されたアルゴリズム $A_i \in \mathbf{A}$ 、すなわち、設定されたハイパーパラメータ、突然変異戦略ならびに交叉戦略に基づくDEに従って、個体数(N_P/n)だけ解を生成する。この操作を I 世代だけ繰り返す。 I 世代だけ繰り返した後、全てのアルゴリズムが生成した解を統合した解集合 \mathbf{x} を次に述べる競合的調整で述べる。したがって、提案手法が必要となる全体の解個体数は N_P となる。

3.1.3 競合的調整

優良アルゴリズムの保存 生成した解を統合した解集合 \mathbf{x} について、評価値の良い順に上位 n 個体 $\mathbf{w} = [w_1, w_2, \dots, w_n]$ を選出し、優良個体として扱う。なお、 w の添え字は順位を示している。したがって、 w_1 は最も評価が高い解個体を意味する。ここで、各上位個体 w_i は、どのアルゴリズムから生成されたかが記憶される。

その後、アルゴリズム集合 \mathbf{A} を空集合に設定し、保存すべきアルゴリズムを追加する。まず、 \mathbf{w} の中でもさらに上位 k ($0 \leq k \leq n$)の解個体 w_i を生成したアルゴリズムは、評価の高い解個体が生成できたことを根拠に、それらのアルゴリズムは調整せずに保存される。本稿では $k = 3$ と設定している。したがって、次に試す n 個のアルゴリズムのうち、 k 個のアルゴリズムは優良解を生成した既存のアルゴリズムが継承される(エリート保存)。また、上位 k 個の解個体 w_i が同一のアルゴリズムから生成された場合は、重複を許容して保存する。例えば、上位 k の解個体 w_i の全てが同じアルゴリズム A_i から生成されれば、 A_i を3つに複製しアルゴリズム集合に保存する。この場合、 A_i が最も有力なアルゴリズムであることが考えられるため、そのアルゴリズムの解探索をより促進するように、 $3 \times (N_P/n)$ に個体数を増やしたアドバンテージを与える。

一方で、残りの $n - k$ 個のアルゴリズムについては調整を経て新規に生成される。

アルゴリズムの調整 $n - k$ 個のアルゴリズムを生成するにあたり、調整目標となる優良解 \mathbf{w} を設定する。ただし、上位 k の優良個体 w_i ($i = 1, \dots, k$)については、それらを生成したアルゴリズムが保

Table 2 実数値連続単一目的ベンチマーク関数

関数	名称	探索領域	定義式
F1	Sphere	$[-100, 100]^D$	$F_1(x) = \sum_{i=1}^D x_i^2$
F2	Rosenbrock	$[-50, 50]^D$	$F_2(x) = \sum_{i=1}^{D-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right)$
F3	Ackley	$[-50, 50]^D$	$F_3(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$
F4	Rastrigin	$[-50, 50]^D$	$F_4(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
F5	Griewank	$[-100, 100]^D$	$F_5(x) = 1 + \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right)$
F6	Weierstrass	$[-0.5, 0.5]^D$	$F_6(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D, \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)],$ $a = 0.5, b = 3, k_{\max} = 20$
F7	Schwefel	$[-500, 500]^D$	$F_7(x) = 418.9829 \times D - \sum_{i=1}^D x_i \sin(\sqrt{ x_i })$
F8	Schwefel 1.2	$[-100, 100]^D$	$F_8(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$

存されているため、調整目標と設定しない。これは、調整するアルゴリズムの多様性を担保するためである。実際、事前実験では、アルゴリズムの調整目標を同一(例えば w_1 のみ)に設定する場合、それぞれのアルゴリズムは同じ決定変数の部分領域を探索する結果、局所解に陥る確率が高くなる。言い換えると、調整目標をアルゴリズムごとに変更し、多様に調整されたアルゴリズムを用いる方が、競合が頻繁に発生し最適化の性能が改善する。そこで、 $n-k$ 個のアルゴリズムの調整目標はそれぞれ $w_i (i = k+1, \dots, n)$ に設定する。したがって、 $n-k$ 個のアルゴリズムはそれぞれ異なる調整目標を持つ。ここで、アルゴリズムの調整終了条件は、調整目標 w_i を中心とし半径 d とした超球内に存在する子個体が生成されることである。この条件を達成するため、 F, CR 、突然変異戦略、交叉戦略をランダムに変更する。本稿では、各次元の決定変数 x_j に共通して $d = 0.1 \times (x_{j,\max} - x_{j,\min})$ と設定する。

4 実験

提案手法の有効性を評価するために、Table 2に示す8つの実数値連続単一目的ベンチマーク関数^{3, 2)}を用いる。そして、DEとハイパーパラメータを適応的に調整するjDEに対し、同一の評価回数において提案手法の性能を比較する。

4.1 実験設定

提案手法のパラメータ設定をTable 3に示す。なお、DEならびにjDEで用いるパラメータについても、Table 3に掲載するパラメータ値に設定する。なお、 N_P, F, CR のパラメータについては、jDEの原著²⁾と同じである。また、最大評価

Table 3 実験パラメータ

パラメータ	説明	数値
D	次元数	10, 20, 30
N_P	個体数	100
F	スケール係数(初期値)	0.5
CR	交叉率(初期値)	0.9
$evals_{\max}$	最大評価回数	100,000
n	アルゴリズム数	10
I	検証世代数	5

回数 $evals_{\max}$ は、DEおよびjDEも同じ値に設定しており、同一の評価回数のもとで提案手法の性能を比較できる。

各ベンチマーク問題において、乱数のシードが異なる30試行の実験を繰り返し、得られた最良解の評価値の中央値を比較する。また、次元数 $D = 10, 20, 30$ と変化させ、提案手法のスケラビリティを評価する。

4.2 実験結果

Table 4に、次元数 $D = 10, 20, 30$ と変化させたときの最良解の評価値を示す。なお、提案手法の収束速度を評価するために、最大評価回数である100,000回に加え、20,000回、50,000回の時点で導出した最良解の評価値も示している。 $D = 10$ においては、F1, F2, F3, F6, F7およびF8で提案手法が優れた性能を導出している。特に、 $D = 10$ に設定したF1, F2およびF6では、最良解のオーダーが大きく異なり、提案手法が極めて優れた性能を導出できていることがわかる。また、 $D = 20$ では、F1, F2およびF4で提案手法が優れている。同様に $D = 30$ では、F1, F2, F4およびF8で提案手法が優位であることがわかる。ま

Table 4 DE, jDEおよび提案手法の性能(最良解の中央値)の比較

評価回数		20,000			50,000			100,000		
関数	D	DE	jDE	proposal	DE	jDE	proposal	DE	jDE	proposal
F1	10	1.24E-04	3.82E-05	1.70E-13	1.23E-16	9.34E-18	5.70E-40	1.02E-36	8.66E-39	1.41E-84
	20	9.45E+00	1.70E+00	2.52E-03	3.70E-05	4.18E-07	5.22E-17	3.53E-14	4.14E-18	4.24E-40
	30	2.19E+02	6.06E+01	8.38E-01	6.08E-02	1.87E-03	8.01E-08	6.17E-08	5.75E-11	2.92E-19
F2	10	6.68E+00	9.09E+00	4.20E+00	5.76E-01	2.75E+00	1.03E-03	1.14E-11	6.25E-03	9.51E-18
	20	1.43E+03	8.00E+02	3.80E+01	1.55E+01	1.63E+01	1.16E+01	7.58E+00	1.15E+01	3.15E+00
	30	1.13E+05	3.91E+04	4.41E+02	6.00E+01	4.78E+01	4.80E+01	2.33E+01	2.42E+01	2.09E+01
F3	10	1.76E+01	1.97E+01	1.04E-04	7.04E-04	7.74E-03	4.00E-15	4.49E-14	1.11E-14	4.00E-15
	20	1.98E+01	2.00E+01	2.00E+01	1.97E+01	2.00E+01	2.00E+01	1.96E+01	1.99E+01	2.00E+01
	30	1.99E+01	2.00E+01	2.00E+01	1.99E+01	2.00E+01	2.00E+01	1.99E+01	2.00E+01	2.00E+01
F4	10	3.43E+01	1.27E+01	1.24E+01	2.46E+01	1.93E-05	2.46E+00	1.93E+01	0.00E+00	6.20E-01
	20	1.46E+02	9.14E+01	5.94E+01	1.11E+02	3.87E+01	3.39E+01	9.68E+01	1.76E+01	1.64E+01
	30	3.21E+02	2.28E+02	1.52E+02	2.10E+02	1.05E+02	6.84E+01	1.86E+02	6.14E+01	4.57E+01
F5	10	4.70E-01	1.72E-01	1.06E-01	3.37E-01	4.59E-03	3.20E-02	2.38E-01	1.41E-13	2.83E-02
	20	7.09E-01	3.48E-01	2.37E-02	2.44E-05	2.44E-07	1.97E-02	2.46E-14	0.00E+00	1.97E-02
	30	1.05E+00	9.76E-01	7.84E-02	5.43E-03	1.70E-04	7.40E-03	5.01E-09	4.36E-12	7.40E-03
F6	10	1.40E-01	1.45E-02	2.74E-03	8.76E-05	8.61E-10	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	20	4.89E+00	1.23E+00	6.38E-01	2.26E-01	6.99E-03	2.52E-01	1.27E-03	3.27E-10	2.52E-01
	30	1.48E+01	5.95E+00	2.95E+00	1.95E+00	1.72E-01	9.08E-01	6.41E-02	4.40E-04	9.08E-01
F7	10	1.53E+03	3.12E+02	7.03E+02	1.04E+03	1.27E-04	1.27E-04	1.17E-01	1.27E-04	1.27E-04
	20	4.61E+03	2.96E+03	3.57E+03	3.99E+03	1.62E+03	2.28E+03	3.66E+03	2.15E-02	4.83E+02
	30	7.89E+03	5.69E+03	6.69E+03	7.29E+03	4.39E+03	4.84E+03	6.86E+03	2.85E+03	3.06E+03
F8	10	2.48E-01	1.56E+02	1.83E+00	4.84E-08	9.92E-01	8.43E-08	2.97E-19	1.92E-04	1.14E-19
	20	4.06E+03	1.53E+04	3.32E+03	1.91E+02	3.78E+03	3.84E+02	8.95E+00	6.01E+02	2.00E+01
	30	8.79E+04	3.68E+04	3.06E+04	1.60E+04	5.07E+04	5.58E+03	2.68E+03	7.51E+03	1.86E+03

とめると、合計24個の実験ケースに対し、最大評価回数時点では提案手法が13個の実験ケースにおいてDEならびにjDEよりも優れた性能を導出している。なお、DEは3つの実験ケース、jDEは9つの実験ケースにおいて最良の性能を導出している。また、評価回数が20,000回であるとき、F3($D = 20, 30$), F7($D = 10, 20, 30$), F8($D = 10$)を除く全ての問題ケースにおいて、提案手法が優れた性能を導出していることがわかる。

一方で、 $D = 10$ におけるF4, F5や $D = 20$ におけるF5, F6、 $D = 30$ におけるF5, F6では提案手法が局所解に収束している。また、Figure 1に示すように、F4について四分位範囲も合わせた性能変化を見ると、提案手法の性能は安定していないことがわかる。一方で、30試行中14試行において、第一四分位数はjDEより少ない評価回数で最適解を発見している。

以上より、上記の多くの実験ケースにおいて提案手法がハイパーパラメータのみを調整するjDEと比較して優れた性能を導出していることから、ハイパーパラメータと遺伝的操作の両方を調整

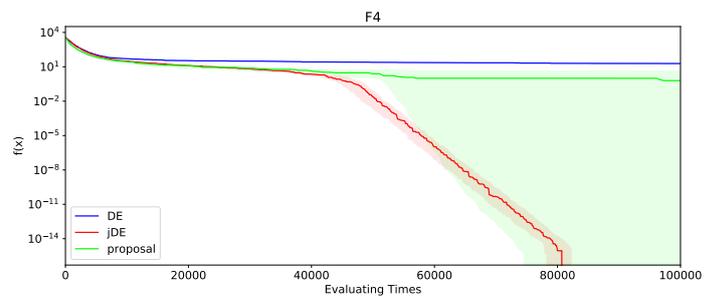


Fig. 1 F4($D = 10$)における性能の比較

する意義が確認できる。加えて、F4($D = 10$)では、jDEあるいはDEよりも劣る場合においても少ない試行数ながらも、これらを超える性能を導出している。jDEやDEに比べ性能が劣る問題が未だ存在するが、アルゴリズム調整の自由度を高める結果、複雑かつ探索空間が増大した場合においても、提案手法によって最適化性能を高めるようにアルゴリズム調整が成功している点に、本研究の意義がある。

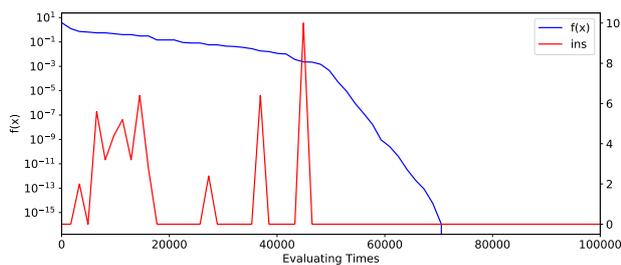


Fig. 2 提案手法の性能とアルゴリズム調整精度の関係

5 考察

5.1 競合的調整の有用性

前章で示された提案手法の優位性の理由は、アルゴリズムの問題および探索状況への特化を可能にするハイパーパラメータと遺伝的操作の両方の調整が適切に機能したことが考えられる。そこで、この調整が適切に機能したことが、競合的なアルゴリズム調整によってもたらされたことを明らかにする。一方で、問題や探索状況に応じて、どのようにアルゴリズムが調整された結果、性能が改善したかという理由を解析的に示すことは困難となる。したがって、ここでは、アルゴリズムの調整の妥当性を検証するために、アルゴリズム調整時に調整目標個体から半径 d の超球に入った個体数を示す。

この例として、F5($D = 10$)における個体数を Figure 2 に示す。青色の線が、提案手法の性能(最良解の評価値)であり、赤色の線が調整目標個体からのユークリッド距離が d の領域に入った個体数である。なお、個体数は各アルゴリズムの平均値を示している。ここで、この個体数が上方向に上昇すると、各アルゴリズムの調整が適切に行われていることを示している。図より、まず探索序盤では各アルゴリズムの調整が適切に行われ、それに伴って性能(青色の線)が改善している。さらに、評価回数44,900回あたりで性能が急激に改善しているが、このときにアルゴリズムの調整(赤色の線)が最も適切になされている。つまり、この競合的なアルゴリズム調整がきっかけとなり、提案手法の性能向上につながったと考えられる。

5.2 検証期間 I に対する性能変化

ここでは、調整されたアルゴリズムの検証期間を決定するパラメータ I を 5, 10, 15 と設定し、提

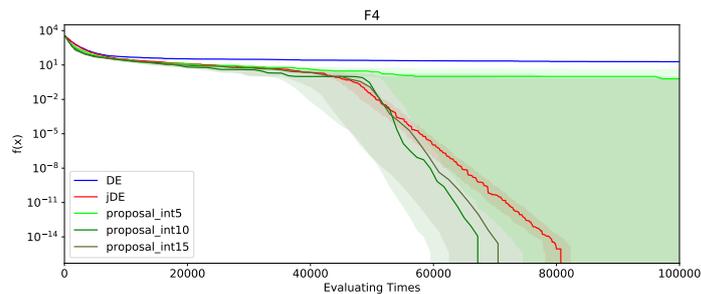


Fig. 3 検証期間 I に対する性能の変化

案手法の性能変化を明らかにする。Figure 3 に、F4($D = 10$)における各設定に従う提案手法の性能を示す。図より、 $I = 10$ では30試行中19試行で、 $I = 15$ では30試行中20試行でそれぞれ最適解を発見している。一方で、 $I = 5$ は、 $I = 10, 15$ に比べてその性能が大幅に劣る。前章では、アルゴリズムの調整頻度(競合頻度)を増やすために $I = 5$ と設定した。しかしながら、図に示すように、アルゴリズムの検証期間は、調整されたアルゴリズムの良し悪しを判断するのに、ある程度の世代数を費やすべきであることがわかる。一方で、検証期間は調整頻度とトレードオフの関係であり、このバランスを適切に設定することが今後の課題となる。

5.3 調整終了判定領域の半径 d に対する性能変化

アルゴリズム調整の終了条件に関するパラメータ d に対する提案手法の性能変化について検証する。ここで、前章における設定 $d = 0.1 \times (x_{j,\max} - x_{j,\min})$ に対し、 $d = 0.05 \times (x_{j,\max} - x_{j,\min})$ と設定したより厳密にアルゴリズム調整を行う場合を検証する。Figure 4 に、F4($D = 10$)における各設定に従う提案手法の性能を示す。図に示す通り、この問題においては d を小さく保ち、調整目標の近傍に生成するようにアルゴリズム調整を行う場合が、優れた性能を導出することがわかる。言い換えると、 $d = 0.05 \times (x_{j,\max} - x_{j,\min})$ と設定した場合は、 $d = 0.1 \times (x_{j,\max} - x_{j,\min})$ と比べて調整目標のより近傍を探索するようにアルゴリズム調整がなされる。つまり、提案手法の調整メカニズムでは、調整対象となるアルゴリズムの構成要素とは別に、アルゴリズム調整の仕方によってアルゴリズムの特性に影響を与える可能性がある。したがって、アルゴリズムの調整を通して問題に特化させるという目的を達成するために、パラメータ d を適応的に変化させることで、大域

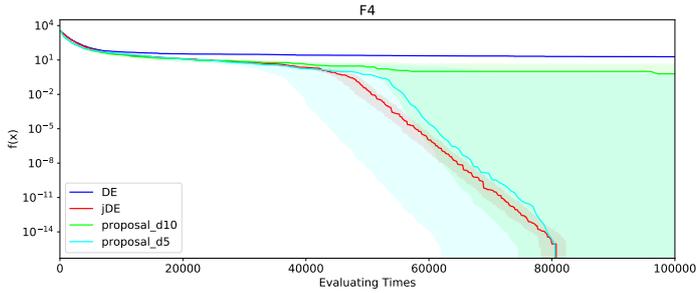


Fig. 4 パラメータ d に対する性能の変化

探索および局所探索を明示的に指向可能なアルゴリズム調整が実現できる可能性がある。

6 まとめ

本研究では、競争均衡原理¹¹⁾に着想を得たメタヒューリスティクスの適応的調整技術を提案した。具体的には、差分進化のハイパーパラメータと遺伝的操作を調整対象とし、複数のアルゴリズム(DE)を用意しそれらが生成する個体を定期的に競合させることで、1)アルゴリズム調整に追加の解評価を要せずに、2)順位が低いアルゴリズムは教師データ(調整の目標となる上位個体)をもとに調整できるフレームワークを構築した。実験結果より、多くの実験ケースにおいて、提案手法はDEやハイパーパラメータのみを調整するjDEよりも優れた性能を導出することを明らかにした。これは、ハイパーパラメータと遺伝的操作という高い自由度でアルゴリズム調整を行うことで、問題や探索状況に特化した適切なアルゴリズム調整がなされた結果といえる。これは、遺伝的操作で用いるハイパーパラメータの組合せによって挙動が変化することで調整の複雑さは各段に増す困難さに対して、これに対処可能な提案手法の有用性を示すものである。

今後の展望としては、調整(競合)頻度とアルゴリズムのトレードオフを制御するパラメータ I 、および、調整するアルゴリズムの特性を制御できる可能性があるパラメータ d について、実験的な検証をさらに進める。そして、これらのパラメータの適応的な調整法を導入することで、アルゴリズムの調整精度を改善し、本稿で明らかにした課題である提案手法の性能安定化を目指す。

参考文献

1) Anna Bogdanova, Jair Pereira Junior, and Claus Aranha. Franken-swarm: grammatical evolution for the automatic generation of swarm-like meta-heuristics. In *Proceedings of the Genetic and Evolutionary Compu-*

tation Conference Companion, pages 411–412. ACM, 2019.

- 2) 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 transactions on evolutionary computation*, 10(6):646–657, 2006.
- 3) Bingshui Da, Yew-Soon Ong, Liang Feng, A Kai Qin, Abhishek Gupta, Zexuan Zhu, Chuan-Kang Ting, Ke Tang, and Xin Yao. Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results. *arXiv preprint arXiv:1706.03470*, 2017.
- 4) Agoston E Eiben and Selmar K Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1(1):19–31, 2011.
- 5) 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 Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):482–500, 2011.
- 6) Giorgos Karafotias, Mark Hoogendoorn, and Ágoston E Eiben. Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 19(2):167–187, 2014.
- 7) Péricles BC Miranda and Ricardo BC Prudêncio. A novel context-free grammar for the generation of pso algorithms. *Natural Computing*, pages 1–19, 2018.
- 8) Michael O’Neill and Conor Ryan. Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358, 2001.
- 9) Mudita Sharma, Alexandros Komninos, Manuel Lopez Ibanez, and Dimitar Kazakov. Deep reinforcement learning based parameter control in differential evolution. *arXiv preprint arXiv:1905.08006*, 2019.
- 10) Ryoji Tanabe and Alex Fukunaga. Success-history based parameter adaptation for differential evolution. In *2013 IEEE congress on evolutionary computation*, pages 71–78. IEEE, 2013.
- 11) Leon Walras. *Elements of pure economics*. Routledge, 2013.
- 12) David H Wolpert, William G Macready, et al. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.
- 13) Jingqiao Zhang and Arthur C Sanderson. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13(5):945–958, 2009.
- 14) 田邊遼司. 関数最適化問題に対する適応型差分進化法の研究. PhD thesis, University of Tokyo(東京大学), 2016.