

# Cuckoo Search を用いた多数目的最適化手法

澤田 和磨<sup>1</sup> アランニャ・クラウス<sup>2</sup> 狩野 均<sup>2</sup>

**概要:** 本論文では Cuckoo Search(CS)を用いた多目的最適化手法を提案する. CS は連続値最適化手法として提案され高い性能を示しており, その応用として多目的最適化問題に対する手法も提案されている. しかし, 多数目的最適化問題(目的関数 4 以上)に対する研究はあまり行われていない. 本論文では CS をこの問題に適用する手法を提案し, ベンチマーク問題を用いた評価実験により, 本手法が従来手法よりも高い性能を得ることを示す.

**キーワード:** Cuckoo Search, NSGA-III, 多数目的最適化問題

## 1. はじめに

多目的最適化問題 (Mult-objective Optimization Problem : MOP) とは 1 つの目的関数ではなく, 複数の目的関数を取り扱う最適化問題である. MOP の最適化手法としてこれまでに数多くの進化計算, 群知能に基づく多目的最適化手法 (Mult-objective Evolutionary Algorithm : MOEA) が提案されている. MOEA の代表的な手法として NSGA-II[1]という遺伝的アルゴリズム (GA) をベースとした手法が存在する. しかし NSGA-II は多数目的最適化問題 (目的関数が 4 以上の MOP) に対して性能が悪いためその改良手法として NSGA-III[2]という手法が提案されている.

近年, 連続値最適化手法として Cuckoo Search (CS) [3] という群知能の手法が提案され, GA や PSO[4]などの他の進化計算, 群知能よりも良い性能を示している. CS を MOP に適用する研究は MOCS[5]をはじめとしていくつか行われているが多数目的最適化問題に対する研究はほとんど行われていない. そこで本研究では NSGA-III に CS を組み合わせた手法の提案を行う.

以下では, まず関連研究として, NSGA-III, CS, CS を使った多数目的最適化手法の従来手法の説明をする. 次に提案手法の説明をし, 最後にいくつかのベンチマーク問題を用いて提案手法と従来手法の比較実験を行い, 提案手法の有効性を評価する.

## 2. 関連研究

### 2.1 NSGA-III

NSGA-III は NSGA-II をベースとし, MOEA/D[6]の特徴である解空間上に広がる一様な重みベクトルに基づくスカラー化関数を用いた部分問題への分割方法を組み合わせた手法で 2013 年に Deb らによって提案された. アルゴリズムは図 1 のように表される.

```
1 初期集団  $P_0$  の生成;  
2 重みベクトルの生成;  
3 参照点の更新;  
4 while(終了条件)  
5   集団  $P_t$  から交叉により子集団  $Q$  を生成;  
6   子集団  $Q$  に対し突然変異を適用;  
7   参照点の更新;  
8    $P_t \cup Q$  を非劣ソートによりランク付け;  
9   ランク, スカラー関数の値から上位個体  
   を選択し, 選択した個体で  $P_{t+1}$  を生成;
```

図 1 NSGA-III のアルゴリズム

### 2.2 Cuckoo Search

Cuckoo Search (CS) は 2009 年に Yang らによって提案された連続値最適化手法である. CS はカッコウの繁殖行動である”托卵”と動物の飛行挙動や採餌行動に見られる”Lévy flight”を組み合わせた多点探索手法である.

GA や PSO などの他の進化計算, 群知能と比較してパラメータが少なく設定が容易で, 過去の研究ではこれらのアルゴリズムよりも良い性能を示すことが確認されている. CS は以下のルールに従う.

- 1) カッコウは一度に 1 つの卵を産み, 卵はランダムに選ばれた巣に産み落とされる.
- 2) 最も質の高い卵を持つ巣は次の世代に引き継がれる.

```
1  解候補の初期化;  
2  while(終了条件)  
3    for(すべての解候補)  
4      Lévy flight による新たな解候補の探索;  
5      適応度  $F_i$  を計算;  
6      ランダムに  $j$  を選択;  
7      if( $F_i' > F_j$ )  
8         $x_j = x_i$ ;  
9    for(すべての解候補)  
10   if(托卵発見) //最良解は保存
```

図 2 CS のアルゴリズム

<sup>1</sup> 筑波大学院システム情報工学研究科コンピュータサイエンス専攻  
Department of Computer Science, Graduate School of Systems and  
Information Engineering, University of Tsukuba

<sup>2</sup> 筑波大学システム情報系情報工学域  
Division of Information Engineering, Faculty of Engineering, Information and  
Systems, University of Tsukuba

3) 利用可能な巣は固定されており、卵は  $p_a \in [0,1]$  の確率で宿主の鳥に発見される。(このとき巣はランダムに選ばれる。)

これらの3つのルールに基づきアルゴリズムは図2のように定義される。

### 2.3 Lévy flight

Lévy flight は動物の飛行挙動や採餌行動にみられる動きで、Lévy 分布に従うランダムウォークである。特徴としては短距離移動の中にときおり長距離移動が挟まれるような移動方法となっており、これにより効率的な探索が可能となり、CS では Lévy flight による解の更新式は式(1)で定義される。ここで Lévy( $\beta$ ) は Mantegna's Algorithm を使って生成された Lévy 分布に従う乱数である。 $\alpha_0$  は定数で  $x_i$  は更新する個体、 $x_{best}$  は集団内で最も適応度の高い個体となる。また本研究での托卵発見時の解の生成は式(2)に従う。ここで(1)の $\oplus$ はアダマール積を表す。 $x'_j, x'_k$  は集団からランダムに選択した個体、 $rand$  は $[0,1]$ の一様乱数を意味する。

$$x'_i(t) = x_i(t) + \alpha \oplus Lévy(\beta) \quad (1)$$

$$\alpha = \alpha_0(x_i(t) - x_{best})$$

$$0.3 \leq \beta \leq 1.99$$

$$x_i(t+1) = x'_i(t) + rand \times (x'_k(t) - x'_j(t)) \quad (2)$$

### 2.4 hybrid many-objective cuckoo search

hybrid many-objective cuckoo search (HMaOCS) [6]は2019年に Cui らによって提案されたCSを多数目的最適化問題に適用した手法である。HMaOCSは本研究と同様にNSGA-IIIとCSを組み合わせた手法となっているがアルゴリズム全体の流れ、解の生成式が異なっている。アルゴリズムは図3に示す。Lévy flightの解の生成式は式(3)に従う。

$$x'_i(t) = x_i(t) + 0.01 \times \alpha \times r \oplus L \oplus (x_{upper} - x_{lower}) \times r_1 \oplus r_2 \quad (3)$$

```

1  解集団 P の初期化;
2  while(終了条件)
3    for( $P_i$  すべての解候補)
4      Lévy flight による新たな解候補の探索;
5      //新たな解候補の集団を Q とする
6     $P_i \cup Q$  を非劣ソートによりランク付け;
7    ランク, スカラー関数の値から上位個体
   を選択し, 選択した個体で P を更新;
8  for(P'のすべての解候補)
9    if(托卵発見)
10     新たな解候補の生成;
11     //新たな解候補の集団を Q' とする
12   $P' \cup Q'$  を非劣ソートによりランク付け;
   ランク, スカラー関数の値から上位個体

```

図3: HMaOCS のアルゴリズム

ここで、 $\alpha$  は定数で  $r$  は正規乱数、 $x_{upper}, x_{lower}$  はそれぞれ各変数の上限と下限を表している。 $r_1$  は式(4)で表現され、 $C_{evaluation}$  は総評価回数で  $C_{evaluated}$  は現在の評価回数である。

$$r_1 = 1 - \frac{C_{evaluated}}{C_{evaluation}} \quad (4)$$

$r_2$  は各成分が $-1,0,1$ からランダムに選択されたベクトルで、ここで0が選択されたときその成分は更新を行わないことを意味する。

托卵発見の操作は個体毎に行われ発見した場合、式(5)に従い新たな解の生成を行う。

$$x_i(t+1) = x'_i(t) + rand \oplus r_3 \oplus (x'_k(t) - x'_j(t)) \quad (5)$$

ここで  $r_3$  は  $0,1$  からなるベクトルで式(6)に従って選択される。

$$r_3^i = \begin{cases} 0, & 0 < rand \leq \frac{1}{M} \\ 1, & \frac{1}{M} < rand \leq 1 \end{cases} \quad (6)$$

$r_3^i$  は  $r_3$  の  $i$  番目の成分を表し、 $M$  は目的関数の数を表す。

### 3. 提案手法

本研究では連続値最適化手法においてCSがGAよりも良い性能を示すことに着目し、NSGA-IIIとCSを組み合わせた手法を提案する。

#### 3.1 NSGA-III と CS のハイブリッド化

NSGA-IIIにおける遺伝的操作(交叉, 突然変異)をCSのLévy flight, 托卵に置き換える手法を提案した。全体の流れは、まずLévy flightを使って子集団を生成し、その子集団に対し托卵の操作を行う。その後子集団と親集団を非劣ソート, スカラー化関数を用いたソートを行うことにより次世代の集団を決定する。アルゴリズムは図4のように表され、Lévy flight, 托卵発見時の解の生成式は(7), (8)に従う。

$$x'_i(t) = x_i(t) + \alpha_0(x_i(t) - x_j(t)) \oplus Lévy(\beta) \quad (7)$$

```

13 解集団  $P_0$  の初期化;
14 while(終了条件)
15   for( $P_i$  すべての解候補)
16     Lévy flight による新たな解候補の探索;
17     //新たな解候補の集団を Q とする
18   for(Q のすべての解候補)
19     if(托卵発見)
20       新たな解候補の生成;
21    $P_i \cup Q$  を非劣ソートによりランク付け;

```

図4: 提案手法のアルゴリズム

$$x_{ik}(t+1) = x'_{ik}(t) + 0.1r, \quad r \sim N(0,1) \quad (8)$$

式(1)を式(7)に変更した理由は多目的最適化問題において最良の個体の評価ができないため集団内からランダム選択した個体 $x_j$ を利用する。

托卵発見は個体毎ではなく成分毎に行い、式(8)の $x_{ik}$ は $x_i$ の $k$ 番目の成分を意味する。本来のCSであれば Lévy で探索を行った後、集団の更新を行ってから托卵発見の操作を行う。しかし、NSGA-III では集団の更新に大きな計算コストがかかるため Lévy flight の後、集団の更新を行わず、托卵発見の操作を行う変更を加えた。またその際に大きすぎる変更を加えると Lévy flight による探索を妨げるため成分毎に変更した。

### 3.2 マスクビットの導入

本研究ではNSGA-IIIとCSをハイブリッド化したもの(以下NSCSと呼ぶ)にさらに変更点を加えた手法の提案を行う。NSCSと提案手法での相違点は Lévy flight で解を生成する際の成分数にあり、通常は全成分に対し更新が行われているが、提案手法ではマスクビットを掛け合わせることでその数を調整しより細かい探索を行えるようにする。式を(9)に従う。

$$x'_i = x_i + \{\alpha_0(x_i - x_j) \oplus \text{Lévy}(\beta)\} \oplus m \quad (9)$$

$$m_k = \begin{cases} 0, & \text{rand} \leq p_b \\ 1, & \text{rand} > p_b \end{cases}, \quad k = 1, \dots, n(n: \text{成分数})$$

ここで $m$ はマスクビットを表し、0,1の生成にはランダムパラメータ $p_b$ を定義し、各成分確率 $p_b$ で0が選択されるようにする。

## 4. 実験

### 4.1 実験方法

評価実験ではHMaOCSの論文に条件を合わせて実験を行い提案手法とNSCS, HMaOCS, NSGA-IIIを比較する。対象問題はMOPのベンチマーク問題であるDTLZ1~4(目的関数の数 $M=2,4,6$ ) [8]を使用し、それぞれ20回実行する。また実際に得られた解を確認するために $M=3$ での実験も行う。各パラメータは表1~3に示し、反復回数は表4に示す。解の評価にはIGDという指標を用いる。IGDは式(10)に従い、 $A$ は探索によって得られた非劣解集合、 $Z$ はパレート最適解を表している。

$$IGD(A, Z) = \frac{1}{|Z|} \sum_{i=1}^{|Z|} \min(z_i, a_j) \quad (10)$$

この指標は各パレート最適解に最も近い解とのユークリッド距離の平均値で、値が小さいほどパレート最適解に近く幅広く均一に分布した良い解を得られていると言える。本論文でのIGDの計算にはHMaOCSの論文に合わせ、可能な限り500に近い数のパレート最適解を使用する。

表1 HMaOCS, 提案手法のパラメータ

パラメータ	HMaOCS	NSCS	提案手法
托卵発見率	0.3	$0.6/n$	$0.6/n$
$\alpha$	1	1	1
$\beta$	1.5	1.5	1.5
$p_b$	-	-	0.6

表2 NSGA-IIIのパラメータ

パラメータ	NSGA-III
SBX	1
Polynomial Mutation	$1/n$
$\eta_c$	30
$\eta_m$	20

表3 個体数

目的関数の数	個体数
2	100
3	92
4	120
6	132

表4 反復回数

問題	反復回数
DTLZ1	700
DTLZ2	250
DTLZ3	1000
DTLZ4	250

### 4.2 実験結果と考察

表5にIGDの平均、標準偏差を示し、図5~8にDTLZ1の $M=3$ でのIGDが最悪値の時に得られた解を示す。またIGDの最悪値で特徴的だったDTLZ4のとき解を図9~12に示す。さらにDTLZ1での進化のグラフを図13~14に示す。これらの図、表より以下のことがわかる。

- ・表5よりDTLZ2の $M=2,6$ とDTLZ3の $M=6$ 以外で提案手法が最も良い成績を示した。DTLZ3の $M=6$ については提案手法が収束しきっていない、または $p_b$ の値が最適でないことが原因で、探索回数を1500までもしくは $p_b$ の値を0.7とした場合表6の結果となり良い性能を示した。

- ・図5~8よりNSGA-IIIやHMaOCSでは最悪値の場合パレート最適解に対し、解の分布がばらついている。提案手法ではばらつきが小さくなっている。NSCSに関しては収束しきっていないことがわかる。このことから提案手法が安定して良い解が得られていることがわかる。

- ・図9~12よりNSGA-IIIはDTLZ4では局所解に陥り抜け出せない場合があることがわかる。HMaOCSやNSCSではパレート最適解とのずれが見られるが、局所解にはまらず

表 5 IGD の平均と標準偏差

問題	目的数		NSGA-III	HMaOCS	NSCS	提案手法
DTLZ1	2	平均	1.79e-3	1.79e-1	1.94e-3	<b>1.78e-3</b>
		標準偏差	8.59e-6	1.85e-1	1.91e-4	<b>7.48e-7</b>
	4	平均	4.13e-2	6.34e-2	3.76e-1	<b>3.94e-2</b>
		標準偏差	7.76e-5	7.01e-2	4.29e-1	<b>6.99e-6</b>
	6	平均	8.14e-2	8.18e-2	7.69e-1	<b>7.95e-2</b>
		標準偏差	2.55e-4	2.53e-4	9.75e-1	<b>1.60e-4</b>
DTLZ2	2	平均	4.17e-3	<b>4.04e-3</b>	6.39e-3	4.22e-3
		標準偏差	<b>5.15e-6</b>	6.03e-5	4.88e-4	9.96e-5
	4	平均	1.31e-1	1.25e-1	1.52e-1	<b>1.16e-1</b>
		標準偏差	<b>2.79e-5</b>	4.33e-4	3.89e-3	5.16e-4
	6	平均	<b>2.58e-1</b>	2.89e-1	3.19e-1	2.62e-1
		標準偏差	<b>9.41e-4</b>	8.03e-3	5.87e-3	2.94e-3
DTLZ3	2	平均	4.49e-3	3.30	2.41	<b>3.99e-3</b>
		標準偏差	7.82e-4	3.37	2.04	<b>4.31e-5</b>
	4	平均	1.22e-1	5.99e-1	3.51e-1	<b>1.21e-1</b>
		標準偏差	<b>4.50e-4</b>	4.58e-1	2.59e-1	5.24e-3
	6	平均	2.79e-1	<b>2.76e-1</b>	3.41	1.11
		標準偏差	<b>1.75e-3</b>	2.52e-2	2.88	1.30
DTLZ4	2	平均	7.78e-2	3.98e-3	6.69e-3	<b>3.97e-3</b>
		標準偏差	2.33e-1	<b>5.08e-6</b>	1.60e-4	2.72e-5
	4	平均	1.88e-1	1.24e-1	1.60e-1	<b>1.12e-1</b>
		標準偏差	1.40e-1	7.00e-4	6.44e-3	<b>2.28e-4</b>
	6	平均	2.96e-1	2.90e-1	3.16e-1	<b>2.60e-1</b>
		標準偏差	7.28e-2	4.61e-3	9.30e-3	<b>1.29e-3</b>

表 6 IGD の平均と標準偏差 2

	HMaOCS	提案手法(反復回数=1500)	提案手法( $p_b=0.7$ )
平均	2.79e-1	2.67e-1	2.73e-1
標準偏差	1.75e-3	2.07e-2	1.34e-1

探索ができています。それに対し提案手法ではパレート最適解に重なった良い解が得られていることがわかる。

・図 13,14 から提案手法は NSGA-III に比べ探索の進みが遅く、HMaOCS と比較すると早いことが分かる。また NSCS は探索の進みが一番遅くこれは移動量が大きすぎることが原因と考えられる。

## 5. おわりに

本研究では Cuckoo Search を用いた多数目的最適化手法を提案した。NSGA-III の遺伝的操作の部分に CS を置き換え、さらにマスクビットを導入した。従来手法との比較実験により、複数の問題において提案手法が最も良い性能を示し

た。今後の課題はさらに目的関数の数を増やした場合での実験を行うことと、問題と更新成分の関係性を調査することで適したパラメータの調整方法を探る。

## 参考文献

- [1] Kalyanmoy Deb, A Fast and Elitist Multiobjective Genetic Algorithm NSGA II, IEEE Transactions on Evolutionary Computation vol.6 no2, April, 2002.
- [2] Deb K, Jain H, An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints, IEEE Trans Evol Comput 18(4):577-601

- [3] Xin-she Yang, Suash Deb, Cuckoo Search via Lévy Flights, *Nature & Biologically Inspired Computing*, no.37, pp. 210–214, 2009.
- [4] Kennedy, J., Eberhart, R., Particle swarm optimization, *IEEE international conference on neural networks*, Vol.4, 1995, pp. 1942–1948.
- [5] Xin-she Yang, Suash Deb, Multiobjective cuckoo search for design optimization, *Computers & Operations Research*, 2013, 40(6): 1616-1624
- [6] Q.Zhang and H.Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.
- [7] Zhihua Cui, Maoqing Zhang, A hybrid many-objective cuckoo search algorithm, *Soft Computing*, April, 2019.
- [8] K Deb, L. Thiele, M. Laimanns, and E. Zizler, Scalable test problems for evolutionary multi-objective optimization, in *Evolutionary Multi-objective Optimization*, A. Abraham, L. Jain, London: Springer-Verlag, 2005, pp. 105-145.

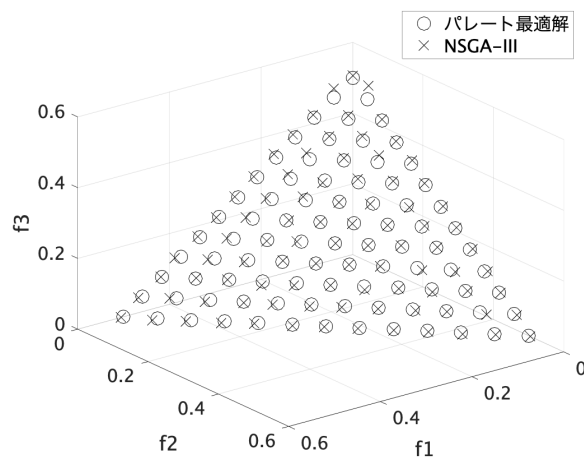


図 5 NSGA-III で得られた解 (DTLZ1)

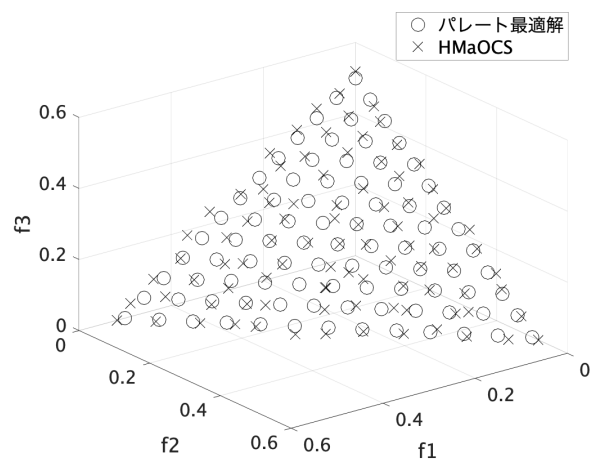


図 6 HMaOCS で得られた解 (DTLZ1)

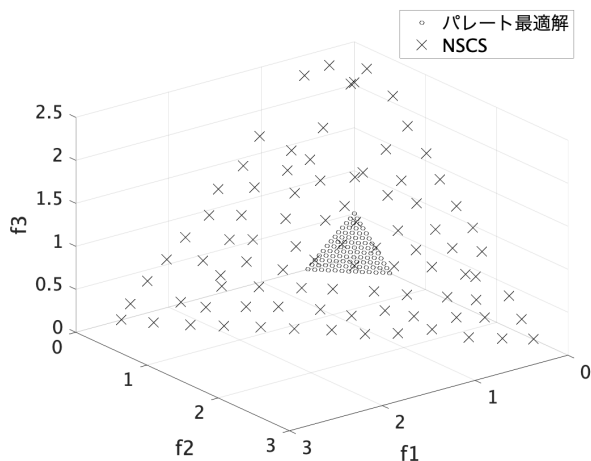


図 7 NSCS で得られた解 (DTLZ1)

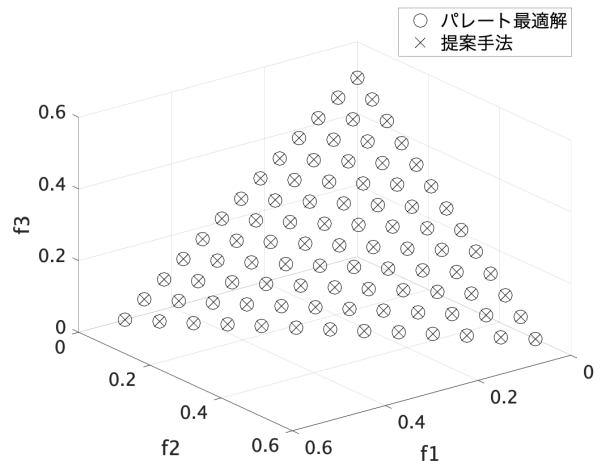


図 8 提案手法で得られた解 (DTLZ1)

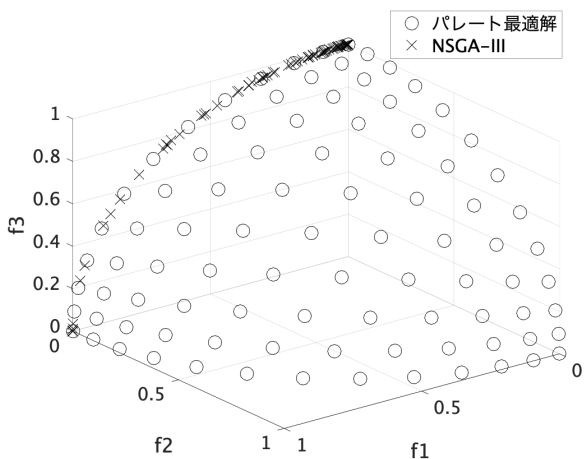


図 9 NSGA-IIIで得られた解 (DTLZ4)

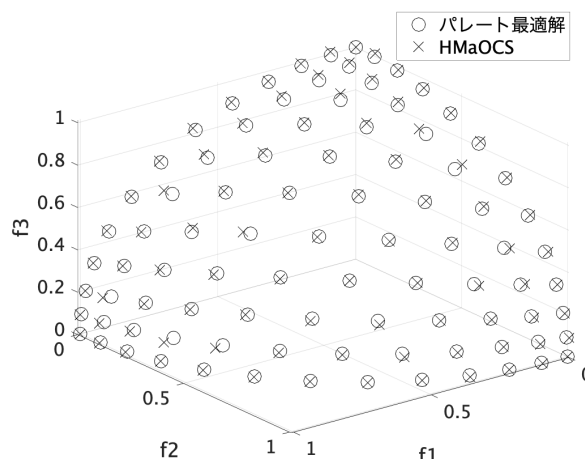


図 10 HMaOCSで得られた解(DTLZ4)

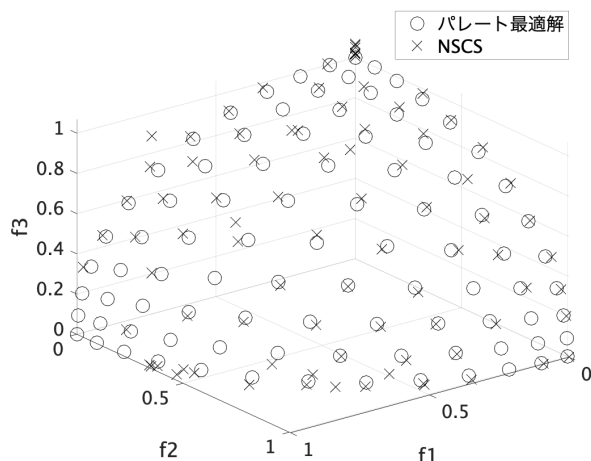


図 11 NSCSで得られた解 (DTLZ4)

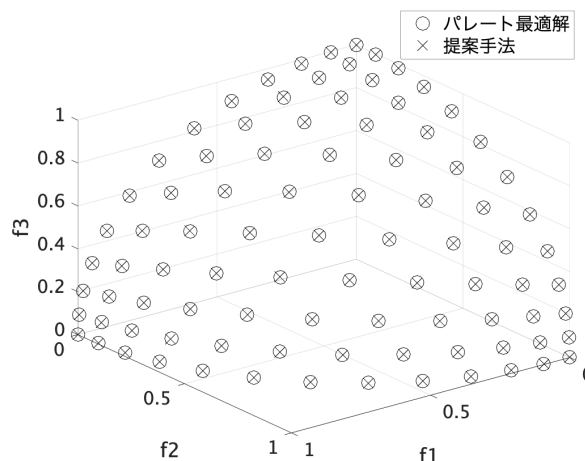


図 12 提案手法で得られた解 (DTLZ4)

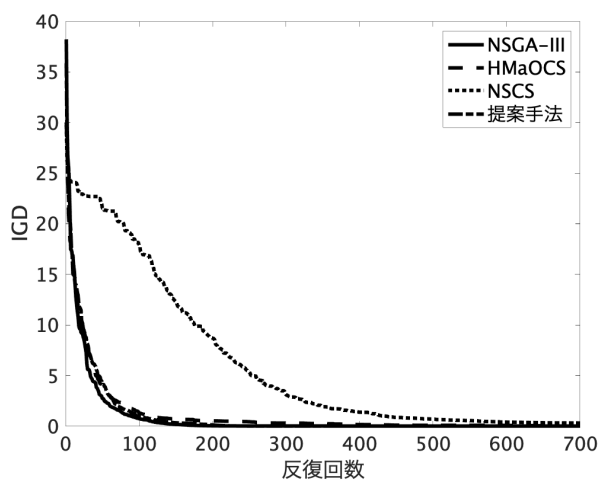


図 13 進化のグラフ (DTLZ1)

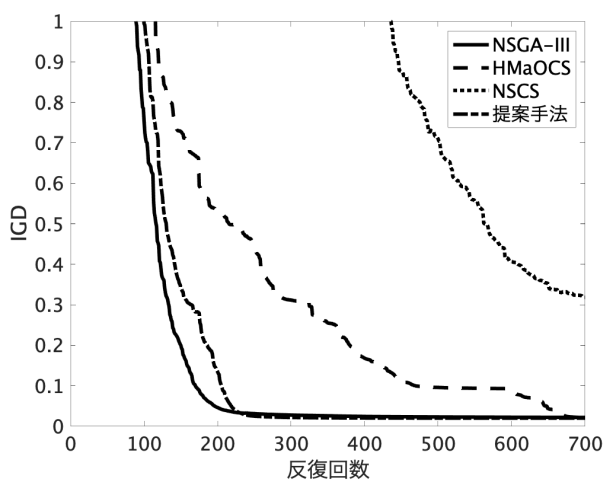


図 14 進化のグラフの拡大図 (DTLZ1)