

有制約最適化のための制約条件の目的関数化と 適応的重み調整を用いた MOEA/D

学生員 安田 雄佑* 正 員 熊谷 渉**
正 員 田村 健一* フェロー 安田恵一郎*

MOEA/D with Constraint Objectivization and Adaptive Weight Adjustment for Constrained Optimization

Yusuke Yasuda*, Student Member, Wataru Kumagai**, Member, Kenichi Tamura*, Member, Keiichiro Yasuda*, Fellow

(2022 年 2 月 28 日 受付)

In this paper, MOEA/D is extended to constrained optimization by making it a constraint objectivization. An adaptive adjustment method is proposed to introduce parameters for varying weights. The parameters for varying the weights are given in such a way that the bias of the search towards feasible and infeasible regions can be adjusted. The parameters are tuned based on two guiding principles to properly utilize the infeasible solutions. The first is to actively utilize infeasible solutions with large constraint violations, and to encourage global search including infeasible regions. The second is to actively utilize infeasible solutions with small constraint violations, and to encourage search on the boundary of the feasible region. This is expected to improve the global optimization performance to the feasible region, which is a non-convex set, and the convergence performance to the feasible solution. We verify the usefulness of the proposed method for problems where the feasible region is a convex set and a nonconvex set.

キーワード：有制約最適化，メタヒューリスティクス，制約対処法，MOEA/D，パラメータ調整

Keywords: constrained optimization, metaheuristics, constraint handling technique, MOEA/D, parameter adjustment

1. はじめに

近年のシステムの大規模・複雑化に伴い，実用的な最適化手法の枠組みとしてメタヒューリスティクスが注目されている⁽¹⁾。メタヒューリスティクスは生物の進化過程や群知能などの類推によって構築されており，代表的手法として，Genetic Algorithm (GA)⁽²⁾，Evolution Strategy (ES)⁽³⁾，Particle Swarm Optimization (PSO)⁽⁴⁾，Differential Evolution (DE)⁽⁵⁾などが知られている。メタヒューリスティクスは目的関数の勾配などの解析的情報を必要としない直接探索法に分類され，目的関数が陽に定式化されていなくとも最適化可能という高い汎用性を有する。また，多くのメタヒューリスティクスは複数の個体の相互作用を主とした

発見的な探索戦略であるため，多峰性関数においても局所解に陥りにくい特徴がある。

実システムでは何らかの制約条件が課されるが，メタヒューリスティクスは無制約最適化を主な対象として設計されているため，有制約最適化に適用するためには，制約を取り扱うための手法である制約対処法 (Constraint Handling Technique: CHT) を必要とする^{(6)~(8)}。有制約最適化におけるメタヒューリスティクスの探索効率は，CHT のメカニズムに大きく依存することが知られている⁽⁸⁾。この理由の一つには，有制約最適化においては実行可能領域の境界上に大域的最適解が存在することが一般的であるため⁽⁶⁾，実行可能領域の境界上を効率的に探索するメカニズムが求められることがある。また，制約条件に非凸関数が含まれている場合には，実行可能領域が非凸集合となるために多数の局所解が存在し，大域的最適解を獲得することは非常に困難となる。そのため，直接探索や多点探索といったメタヒューリスティクスの特徴を保持・活用することで，これらの困難さを解消した効果的な CHT の開発が求められている。

メタヒューリスティクスにおける CHT には，ペナルティ

* 東京都立大学

〒192-0397 東京都八王子市南大沢 1-1

Tokyo Metropolitan University

1-1, Minamiosawa, Hachioji-shi, Tokyo 192-0397, Japan

** 横河電機株式会社

〒180-0006 東京都武蔵野市中町 2-9-32

Yokogawa Electric Corporation, Ltd

2-9-32, Naka-cho, Musashino-shi, Tokyo 180-0006, Japan

に基づくアプローチ^{(6)(9)~(13)}, 分離に基づくアプローチ^{(14)~(20)}, 多目的に基づくアプローチ^{(21)~(23)}などが挙げられる。一方, 近年では探索効率の向上には, 実行不可能解の活用が重要であると指摘されている⁽²²⁾⁽²⁴⁾。実行不可能解を適切に活用することで, 実行可能領域の境界上の探索性能や, 実行不可能領域における大域的探索性能に寄与するためである。特に多目的に基づくアプローチは, 目的関数値と制約違反量を同時に最適化するメカニズムを有しているため, 実行不可能解を積極的かつ明示的に活用する。従って, 高次元かつ多峰構造を有する有制約最適化に対しても高い探索性能を発揮することが期待される。しかしながら, これまでに提案された多目的に基づくアプローチは, NSGA-II⁽²⁵⁾に代表されるパレートランキングや混雑距離による探索戦略が一般的に利用されるため⁽²¹⁾⁽²²⁾, 実行不可能領域に対する探索の多様性が過度に高まるあまり, 得られる解の実行可能性や収束性が低いなどの課題があった。

著者らの先行研究⁽²⁶⁾では, 非凸集合となる実行可能領域に対しても頑健性を維持できる多目的に基づくアプローチに着目し, 得られる解の実行可能性や収束性を向上させることを目的として, Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D)⁽²⁷⁾を有制約最適化へと拡張し, 有制約最適化のための適応的重み調整を用いた MOEA/D を提案した。MOEA/D は, 異なる重みを用いたスカラー化関数によって, 多目的最適化問題を複数の単一目的最適化問題に分割することで多様なパレート解を求めるため, 問題に応じて重みを適切に調整することで良好な実行可能解への収束性を高めることが期待される。しかしながら, 実行可能領域が凸集合となる問題での数値実験に限定されており, 実行可能領域が非凸集合となる問題や, より高次元の問題での有用性は検証されていない⁽²⁶⁾。

本論文の目的は, 著者らの先行研究⁽²⁶⁾に基づき, 実行可能解への収束性能と非凸集合となる実行可能領域に対する大域的最適化性能の両立の可能な CHT の開発である。そのため, 実行不可能解を明示的に活用するアプローチの下で, 凸集合及び非凸集合となる実行可能領域を考慮した新たな重みの適応的調整法を提案し, 実行可能領域が凸集合及び非凸集合となる問題に対して, 提案手法の有用性を検証する。

以上を踏まえると, 本論文は著者らの先行研究⁽²⁶⁾に対して, 以下の点で新規性がある。

- 凸集合及び非凸集合となる実行可能領域を考慮した重みの適応的調整法を提案すること。
- 問題や次元数を増やした数値実験により提案手法の有用性をより詳しく検証すること。

本論文の構成を述べる。2 章では, 有制約最適化問題を定式化する。さらに, 既存の CHT の特徴を述べ, 実行不可能解の活用度の観点から細分化してまとめる。3 章では, 本論文の提案手法の基礎となる MOEA/D を説明し, 有制約最適化への拡張について検討する。4 章では, 3 章にて有制約最適化へと拡張された MOEA/D を基礎とし, 重みを

可変するためのパラメータを導入し, 探索に与える影響を検証する。さらに, 検証結果に基づいてパラメータの適応的調整法を提案する。5 章では, 実行可能領域が凸集合及び非凸集合となる問題に対して提案手法を適用し, 実行可能解への収束性及び大域的最適化性能を検証する。6 章では, 本論文のまとめと今後の課題を述べる。

2. 有制約最適化

〈2・1〉 有制約最適化問題の定式化 本稿では, 式 (1)(2)(3) で定義される有制約最適化問題を対象とする。

$$\min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) \dots\dots\dots (1)$$

$$\text{subj. to } g_k(\mathbf{x}) \leq 0, k = 1, \dots, K \dots\dots\dots (2)$$

$$a_j \leq x_j \leq b_j, j = 1, \dots, N \dots\dots\dots (3)$$

$f(\mathbf{x})$ は目的関数, $g_k(\mathbf{x}), k = 1, \dots, K$ は K 個の不等式制約関数であり, $f, g_k : \mathbb{R}^N \rightarrow \mathbb{R}$ なる写像である。 $[a_j, b_j]^N, j = 1, \dots, N$ は各変数 x_j に課せられる上下限制約とする。全ての制約を満たす領域を実行可能領域 \mathcal{F} とし, 実行可能領域 \mathcal{F} に包含される解を実行可能解と呼び, そうでなければ実行不可能解と呼ぶ。

ある解 \mathbf{x}^o において $g_k(\mathbf{x}^o) = 0$ が成り立つとき, $g_k(\mathbf{x}^o)$ を有効制約と呼び, \mathbf{x}^o はその制約の境界上に存在している。有制約最適化では, 大域的最適解 \mathbf{x}^* が実行可能領域の境界上に存在することが一般的であると報告されており⁽⁶⁾, その場合には \mathbf{x}^* において少なくとも 1 個の制約は有効制約となる。

〈2・2〉 既存の制約対処法 本節ではメタヒューリスティクスのために設計された CHT をペナルティに基づくアプローチ, 分離に基づくアプローチ, 多目的に基づくアプローチに大別し, 概説する。

〈2・2・1〉 ペナルティに基づくアプローチ ペナルティに基づくアプローチでは, 目的関数に対して制約を逸脱した際の制約違反量を付加し, 無制約最適化問題へと変換することで制約に対処する。古典的には, ペナルティ係数を無限大として与え, 実行可能解のみで探索を行うデス・ペナルティ⁽⁶⁾, ペナルティ係数を定数として与える静的なペナルティ⁽⁹⁾, ペナルティ係数を探索過程でスケジュール的に増加させる動的なペナルティ⁽¹⁰⁾などが提案されている。

一方, 適切なペナルティ係数は問題に大きく依存することが知られており, ペナルティ係数を適切に設定・調整しなければ, 実行可能解が得られない, あるいは局所解に陥る可能性がある。そのため, ペナルティ係数を探索状況に応じて適応的に変化させる方法がしばしば用いられる。Lemongeraらは探索過程で得られた個体群の目的関数値・制約違反量をフィードバックする Adaptive penalty Method (APM)⁽¹¹⁾。それ以外には, Fuzzy rule based penalty⁽¹²⁾や ASCHEA⁽¹³⁾などがペナルティに基づくアプローチとして挙げられる。

〈2・2・2〉 分離に基づくアプローチ 分離に基づくアプローチでは, 目的関数値と制約違反量を分離して評価・活用

Table 1. Classification of constraint handling techniques

Use of infeasible solutions	Constraint handling technique		
	Penalty	Separatist	Multiobjective
None	Death penalty (Resampling) ⁽⁶⁾	-	-
Inexplicit	Static penalty ⁽⁹⁾ , Dynamic penalty ⁽¹⁰⁾ , APM ⁽¹¹⁾ , Fuzzy rule based penalty ⁽¹²⁾	Feasibility rule ⁽⁷⁾⁽¹⁴⁾ , Stochastic ranking ⁽¹⁵⁾ , ε constraint method ⁽¹⁶⁾⁽¹⁷⁾ , MCR ⁽¹⁸⁾	-
Explicit	ASCHEA ⁽¹³⁾	MCODE ⁽¹⁹⁾ , TNSDM ⁽²⁰⁾	Two phase framework ⁽²¹⁾ , IDEA ⁽²²⁾ , DeCODE ⁽²³⁾ , Proposed method

することで制約に対処する。Runarsson らは生存選択における適合度の大小比較の際に、制約違反量を用いた適合度による比較と目的関数値を用いた適合度による比較を確率的に切り替える Stochastic ranking を提案している⁽¹⁵⁾。Stochastic ranking では、目的関数値と制約違反量の優先度を決定する確率的なパラメータを有するが、目的関数値と制約違反量の加重和を用いるペナルティに基づくアプローチに対して、その評価が分離されるため、ペナルティ係数のようなパラメータの問題依存性が緩和されている⁽¹⁵⁾。Paula Garcia らは目的関数値・制約違反量・制約違反数のランクに変換し、この総和を解の適合度に用いる Multiple constraint ranking (MCR) を提案している⁽¹⁸⁾。MCR は、目的関数値や制約違反量をランクに変換することで、それぞれの関数のスケールから独立しており、パラメータレスでありながらペナルティに基づくアプローチよりも高い探索性能が報告されている⁽¹⁸⁾。それ以外には、Feasibility rule⁽⁷⁾⁽¹⁴⁾、 ε constraint method⁽¹⁶⁾⁽¹⁷⁾、MCODE⁽¹⁹⁾、TNSDM⁽²⁰⁾ などが分離に基づくアプローチとして挙げられる。

〈2・2・3〉多目的に基づくアプローチ 多目的に基づくアプローチでは、目的関数に対して、制約違反量を追加の目的関数と見なし、有制約最適化問題を多目的最適化問題へと変換することで制約に対処する。Venkatraman らは探索を第一段階と第二段階に分け、第一段階では実行可能解を求めるため、制約違反量のみを最適化し、実行可能解が得られると第二段階に移り、目的関数値と制約違反量のパレートランキング・混雑距離によって生存選択を行う方法（以下、Two phase framework と呼ぶ）を提案している⁽²¹⁾。Ray らは、個体群における実行不可能解を保持する割合をパラメータとして設定することで、実行可能解と実行不可能解の両方を保持し、実行可能解の適合度として目的関数値、実行不可能解の適合度として目的関数値と制約違反量のパレートランキング・混雑距離を用いる Infeasibility Driven Evolutionary Algorithm (IDEA) を提案している⁽²²⁾。これらの手法は、NSGA-II⁽²⁵⁾ に代表されるパレートランキング・混雑距離を主とした探索戦略が利用されるため、探索空間

に対して実行可能領域が狭い場合に実行不可能領域に対する探索の多様性が過度に高まることで、パレートランキング・混雑距離による生存選択がほとんど機能しないことや、そもそも実行可能解が得られないなどの弊害が起こりうる。

一方、Wang らは MOEA/D⁽²⁷⁾ を基礎とし、重みをスケジューリング的に調整することで実行不可能領域を含めた多様な探索から、実行可能解へと収束させる DeCODE を提案している⁽²³⁾。著者らも MOEA/D を基礎とした手法を提案しており⁽²⁶⁾、実行可能領域と実行不可能領域の探索のバランスを取るように重みを適応的に調整することで、探索性能の向上を試みている。MOEA/D を主とした探索戦略は、異なる重みを用いたスカラー化関数によって、多目的最適化問題を複数の単一目的最適化問題に分割することで多様なパレート解を求めるため、問題に応じて重みを適切に調整することで良好な実行可能解の獲得が期待される。しかしながら、著者らの先行研究⁽²⁶⁾ では、実行不可能領域を含めた大域的な探索を促すような重みの調整戦略は検討されておらず、非凸集合となる実行可能領域に対しては局所解に陥る可能性が高いと考えられる。

〈2・3〉制約対処法の分類 前節では制約対処法のアプローチについて述べたが、近年では実行不可能解を探索に積極的に活用することが重要視されている⁽²²⁾⁽²⁴⁾。これは、有制約最適化における大域的最適解が実行可能領域の境界上に存在することが多いこと、制約が非凸関数である場合に制約に起因する局所解が多数存在することに起因する。以上から、前節で述べたアプローチを実行不可能解の活用度の観点から細分化した表を Table 1 に示す。

デス・ペナルティ⁽⁶⁾ は実行不可能解を全く用いず、実行可能解のみで探索を行うため、“None” に分類される。静的なペナルティ⁽⁹⁾、動的なペナルティ⁽¹⁰⁾、適応的なペナルティ⁽¹¹⁾、Fuzzy rule based penalty⁽¹²⁾、Feasibility rule⁽²⁵⁾、Stochastic ranking⁽¹⁵⁾、 ε constraint method⁽¹⁶⁾⁽¹⁷⁾、MCR⁽¹⁸⁾ は実行不可能解を活用することはあるものの非明示的であるため、“Inexplicit” に分類される。ASCHEA⁽¹³⁾、MCODE⁽¹⁹⁾、TNSDM⁽²⁰⁾、Two phase framework⁽²¹⁾、IDEA⁽²²⁾、DeCODE⁽²³⁾ は構

造・機能上で実行不可能解を明示的に保持・活用する機構が含まれているため, “Explicit” に分類される。

Table 1 より, 多目的に基づくアプローチは目的関数値と制約違反量を同時に最適化するメカニズムであるために, “Explicit” に集中して分類されていることがわかる。従って, 多目的に基づくアプローチは実行不可能解の活用度との関係性が強いといえる。本論文の提案手法も多目的に基づくアプローチを採用していることから “Explicit” に分類され, 実行不可能解を適切に活用することで高い探索性能が期待される。

3. Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D)

〈3・1〉 MOEA/D の概要 MOEA/D⁽²⁷⁾ は Zhang らによって提案され, 近年有力な多目的最適化手法の一つとされる。MOEA/D の探索メカニズムを Fig. 1 に示す。Fig. 1 の青い矢印は重みベクトルを表しており, 赤い曲線はパレートフロンティアを表している。MOEA/D は, 各個体 $\mathbf{x}^i, i = 1, \dots, m$ に対して異なる重みベクトル \mathbf{w}^i を用いたスカラー化関数 $S(\mathbf{x}^i|\mathbf{w}^i)$ により, 多目的最適化問題を複数の単一目的最適化問題に分割する。さらに, i 番目の個体に対する重みベクトル間のユークリッド距離に基づく T 個の近傍 $B(i) = \{i_1, \dots, i_T\}$ を定義し, 近傍内で進化操作を行うことで複数の単一目的最適化問題を同時に解く。後で述べるように, 重みベクトルは目的関数空間に対して均一に配置され, 広大な目的関数空間においても高い精度でパレートフロンティアを近似可能であることが示されている。MOEA/D の疑似コードを Algorithm 1 に示す。

重みベクトルは式 (4) を満たすように, 式 (5) で与えられる。

$$\sum_{r=1}^R w_r = 1, \quad w_r \geq 0 \dots\dots\dots (4)$$

$$w_r \in \left\{0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H}\right\} \dots\dots\dots (5)$$

ただし, $H \in \mathbb{N}$, R は目的数を表す。重みベクトルは個体数 m と同数生成され $m =_{H+R-1} C_{R-1}$ で与えられる。例えば, $R = 2$ であれば $m =_{H+1} C_1$ となり, $H + 1$ 個の重みベクトルが生成される。

MOEA/D においてスカラー化関数は, パレートフロンティアを近似する際に重要な役割を果たす。代表的なスカラー化関数には, 加重和, チェビシェフノルム, Penalty-based Boundary Intersection (PBI) などが知られているが, パレートフロンティアの形状によって得意なスカラー化関数が異なることが知られている⁽²⁷⁾。本稿では, 有制約最適化へと拡張する基礎検討として加重和を採用する。加重和スカラー化関数 $S^{\text{WS}}(\mathbf{x}|\mathbf{w})$ は式 (6) で与えられる。

$$S^{\text{WS}}(\mathbf{x}|\mathbf{w}) = \sum_{r=1}^R w_r f_r(\mathbf{x}) \dots\dots\dots (6)$$

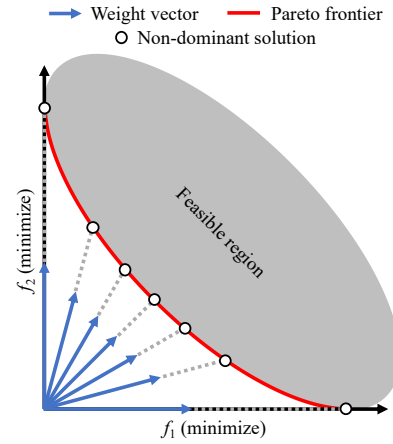


Fig. 1. MOEA/D search mechanism

Algorithm 1 MOEA/D

```

1: procedure MOEA/D( $m, T, g_{\max}$ )
2:   Initialize  $\mathbf{x}^{i(1)} (i = 1, \dots, m)$ .
3:   Set  $\mathbf{w}^{i(1)} (i = 1, \dots, m)$  by Eq. (5).
4:    $g := 1$ 
5:   while  $g \leq g_{\max}$  do
6:     for  $i = 1$  to  $m$  do
7:       Calculate the Euclidean distance between  $\mathbf{w}^{i(1)}$  and
        $\mathbf{w}^{j(1)} (j = 1, \dots, m)$ .
8:       Set the neighborhood index  $B(i) = \{i_1, \dots, i_T\}$ .
9:     end for
10:    for  $i = 1$  to  $m$  do
11:      Randomly select  $a, b \in B(i)$ .
12:      Crossover and mutation on  $\mathbf{x}^{a(g)}$  and  $\mathbf{x}^{b(g)}$  to produce  $\mathbf{y}^{i(g)}$ .
13:      for  $j \in B(i)$  do
14:        if  $S(\mathbf{y}^{i(g)}|\mathbf{w}^{j(g)}) \leq S(\mathbf{x}^{j(g)}|\mathbf{w}^{j(g)})$  then
15:           $\mathbf{x}^{j(g)} := \mathbf{y}^{i(g)}$ 
16:        end if
17:      end for
18:    end for
19:     $g := g + 1$ 
20:  end while
21:  return  $\mathbf{x}^{i(g_{\max})} (i = 1, \dots, m)$ 
22: end procedure
    
```

〈3・2〉 有制約最適化への拡張 MOEA/D は多目的最適化問題に対して, 収束性と多様性に優れたパレート解を求めるために設計されており, そのまま有制約最適化問題に適用することはできない。しかしながら, 制約条件を逸脱した量を追加の目的関数と見なすことで多目的最適化問題と同様に適用することができる^{(21)~(23)(26)}。

ある解 \mathbf{x} が制約条件を逸脱する量の総和を制約違反量 $v(\mathbf{x})$ とし, 目的関数 $f(\mathbf{x}) (= f_1(\mathbf{x}))$ 及び制約違反量 $v(\mathbf{x}) (= f_2(\mathbf{x}))$ で定義される二目的最適化問題 ($R = 2$) へと変換する。Fig. 2 に有制約最適化におけるパレートフロンティアの例を示す。Fig. 2 の左図は目的関数の等高線と制約の課された決定変数空間, Fig. 2 の右図は f と v を軸とする目的関数空間 (f - v 空間) である。黒い線は制約関数, 灰色の領域

は実行可能領域, 赤い線はパレートフロンティアを表している。大域的最適解が実行可能領域の境界上に存在する場合には, 実行不可能領域において目的関数値が改善し, かつ制約違反量が改善するトレードオフとなる領域が表れる。上述の二目的最適化問題に対して, MOEA/D を適用することで, Fig. 2 の赤い線で表される有制約最適化における f - v 空間のパレート解集合 (パレートフロンティア) の獲得が期待される。

制約違反量 $v(\mathbf{x})$ は柔軟な定義を用いることができるが, 単純に逸脱量の総和とする式 (7) や個体 $\mathbf{x}^i \in X; i = 1, \dots, m$ で正規化を行う式 (8) が考えられる。

$$v(\mathbf{x}) = \sum_{k=1}^K \Omega_k(\mathbf{x}) \dots \dots \dots (7)$$

$$v^{\text{nor}}(\mathbf{x}) = \sum_{k=1}^K \frac{\Omega_k(\mathbf{x}) - \min_{\mathbf{x}' \in X} \Omega_k(\mathbf{x}')}{\max_{\mathbf{x}' \in X} \Omega_k(\mathbf{x}') - \min_{\mathbf{x}' \in X} \Omega_k(\mathbf{x}')} \dots \dots \dots (8)$$

ただし, $\Omega_k(\mathbf{x}) = \max\{0, g_k(\mathbf{x})\}$ である。

4. 提案手法

〈4・1〉 提案手法の概要 本論文では, 実行可能解への収束性能と非凸集合となる実行可能領域に対する大域的最適化性能の両立するため, 有制約最適化へと拡張された MOEA/D を基礎とし, 重みを可変するためのパラメータ α の導入とそのパラメータ α の適応的調整法を提案する。

重みを可変するためのパラメータ α は, 実行可能領域及び実行不可能領域への探索の偏り状況を調整できるように与えられる。パラメータ α は実行不可能解を適切に活用するように二つの指針に基づいて調整される。

指針 1 目的関数値の優れた制約違反量の大きな実行不可能解を積極的に活用し, 実行不可能領域を含めた大域的な探索を促すこと。

指針 2 目的関数値の優れた制約違反量の小さな実行不可能解を積極的に活用し, 実行可能領域の境界上の探索を促すこと。

指針 1 に従うことで非凸集合となる実行可能領域に対する大域的最適化性能の向上が期待され, 指針 2 に従うことで実行可能解への収束性能の向上が期待される。以下, 〈4・2〉節ではパラメータ α の導入方法とその検証, 〈4・3〉節ではパラメータ α の適応的調整法を述べる。

〈4・2〉 パラメータ α の導入と数値実験的検証

〈4・2・1〉 パラメータ α の導入 多くの有制約最適化では大域的最適解が制約境界上 (Fig. 2 におけるパレートフロンティアの端) に位置するため, 制約境界付近の目的関数値が良い領域が有望領域となる。そのため, 重みベクトルはパレートフロンティア全体 (式 (5)) ではなく, 実行可能領域側に偏った解に対応するように与えることが適切だと考えられる。そこで, パラメータ $\alpha \in [0, 1]$ を用いて重みベクトル $\mathbf{w}^i, i = 1, \dots, m$ を式 (9) で定義する。

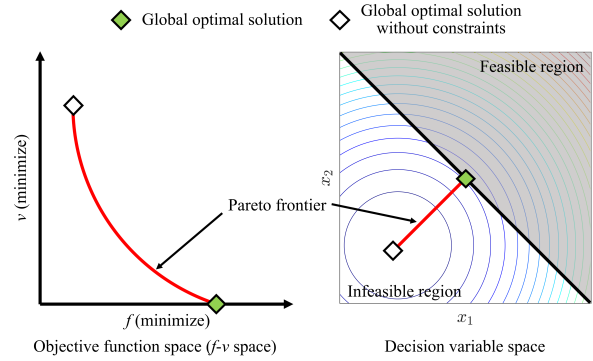


Fig. 2. Pareto frontier in constrained optimization

$$\mathbf{w}^i = \left[\alpha \frac{i-1}{m-1}, 1 - \alpha \frac{i-1}{m-1} \right]^T \dots \dots \dots (9)$$

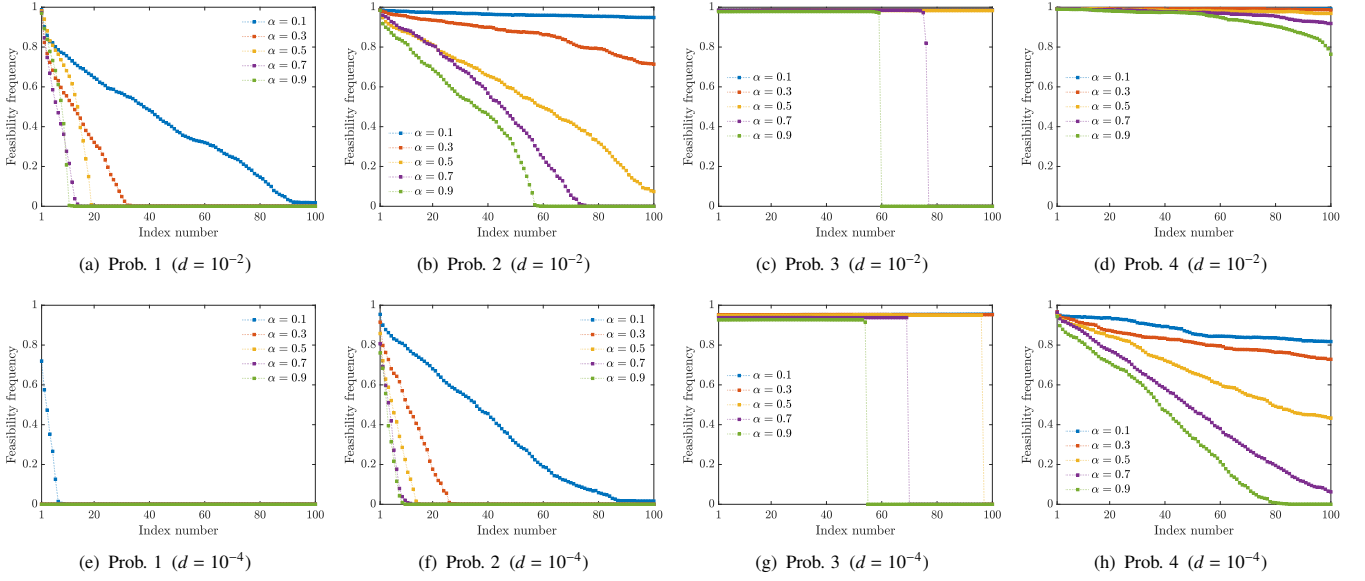
ただし, $w_1^i = 0$ または $w_2^i = 0$ の場合には, 目的関数値・制約違反量を無視しないように, 微小値 $\delta > 0$ で置き換える。 $\alpha = 1$ の場合には目的関数値と制約違反量を対等に扱い, α が 0 に近づくにつれて目的関数値の貢献度が減少する。従って, α を変化させることでパレートフロンティアの近似部位を調整し, 実行可能領域及び実行不可能領域に対する探索の偏りを調整可能となることが期待される。

〈4・2・2〉 パラメータ α の探索への影響の検証 パラメータ α による探索への影響を検証する。Test 問題 (app.Table 1 を参照) に対して, $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$ とした 5 通りの α を用いた MOEA/D を適用し, 全世代において各個体 $\mathbf{x}^i, i = 1, \dots, m$ が実行可能となった相対頻度 (Feasibility frequency) を観測する。これにより, 各個体が実行可能となる確率を観察できる。MOEA/D の条件は, 交叉に Simulated Binary Crossover (SBX) ⁽²⁸⁾, 交叉率 $p_c = 1$, 分布定数 $\eta_c = 20$, 変異に Polynomial Mutation (PM) ⁽²⁵⁾, 変異率 $p_m = 1/N$, 分布定数 $\eta_m = 20$, 個体数 $m = 100$, 最大世代数 $g_{\max} = 5000$, 近傍数 $T = m/10$, 制約違反量は式 (7) とする。

検証結果を Fig. 3 に示す。結果より以下のことが確認される。

- 個体番号が小さいほど Feasibility frequency が高く, 個体番号が大きいくほど Feasibility frequency が低い。
- α が小さいほど Feasibility frequency の高い個体が増加し, α が大きいほど Feasibility frequency の高い個体が減少する。

紙面の都合上割愛したが, 様々な問題において同様の傾向が観られた。以上から, パラメータ α を導入することで個体群における実行可能解と実行不可能解のバランス調整が可能となること, 同じ α を用いても各個体の Feasibility frequency は制約の厳しさや景観に応じて大きく異なることが確認された。なお, Prob. 3 において Feasibility frequency の高い個体と低い個体の差が顕著である理由は, 制約の境界付近で f - v 空間におけるパレートフロンティアが凹型の形状となり, 加重和では凹型のパレートフロンティアを適


 Fig. 3. Feasibility frequency when $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$ is set in Test problems ($N = 100$)

切に近似できないためだと考えられる。

〈4・3〉 パラメータ α の適応的調整 問題に応じて制約の凸性や厳しさは異なるため、適切な α を事前に与えることは困難である。そこで、目的 1 である非凸集合となる実行可能領域への大域的最適化性能の向上を達成するために大域的な探索を十分に行った後、目的 2 である実行可能解への収束性能の向上を達成するため、実行可能領域の境界上の探索へと推移させる探索戦略をとる。指針 1 から指針 2 への推移は条件 1 に従って判断され、指針 2 の実現は条件 2 に従って判断される。

条件 1 個体群の目的関数値と制約違反量の改善がトレードオフとなる領域 (パレートフロンティア) への収束状況。

条件 2 個体群の実行可能領域と実行不可能領域への偏り状況。

条件 1 については、目的関数値と制約違反量の改善がトレードオフとなる領域に対して個体群が十分に収束していなければ大域的な探索を行っている判断でき、十分に収束していれば実行可能領域の境界上の探索を行っている判断できるためである。条件 2 については、個体群が実行可能領域に偏りを持てば実行不可能解を活用できていないと判断でき、実行不可能領域に偏りを持てば制約違反量の小さな実行不可能解を活用できていないと判断できるためである。

提案手法では、条件 1 を個体群における非劣解の割合[†]、条件 2 を指定の重みベクトル \mathbf{w}^t ($t \in \{1, \dots, m\}$) に対応する個体 \mathbf{x}^t の実行可能性に対応させる。目的関数値と制約違反量の改善がトレードオフとなる領域に収束した際、個体群における非劣解の割合は自然と増加するため指針 1 から指針 2 への推移が期待される。また、〈4・2・2〉項の検証結果か

ら、個体番号と Feasibility frequency は相関関係を持つため、指針 2 の実現が期待される。以上の検討をまとめ、世代数 g のパラメータ $\alpha^{(g)}$ を式 (10) で更新する。

$$\alpha^{(g+1)} = \begin{cases} \gamma_d \alpha^{(g)} & \text{if } \mathbf{x}^{s,(g)} \in \mathcal{P} \wedge \mathbf{x}^{t,(g)} \notin \mathcal{F} \\ \min\{\gamma_u \alpha^{(g)}, 1\} & \text{otherwise} \end{cases} \quad (10)$$

ただし、 $\gamma_u > 1$ は増加係数、 $\gamma_d \in (0, 1)$ は減少係数、 \mathcal{P} は個体群中の非劣解集合である。 $\mathbf{x}^{s,(g)}$ が非劣解かつ $\mathbf{x}^{t,(g)}$ が実行不可能な場合のみ α を減少させ、それ以外の場合は α を増加させる。これにより、指針 1 及び指針 2 の適切に実現し、非凸集合となる実行可能領域への大域的最適化性能と実行可能解への収束性能の向上が期待される。提案手法では、 t, γ_u, γ_d のパラメータを追加しているが、予備実験による推奨値として、 $t = [0.8m], \gamma_u = 1.001, \gamma_d = 0.999$ を与える。

提案手法の疑似コードを **Algorithm 2** に示す。オリジナルの MOEA/D (**Algorithm 1**) との相違点は以下の通りである。

- α を初期化し、重みベクトルを式 (9) で与える (3-4 行目)。
- 式 (10) で α を更新した後、式 (9) で重みベクトルを更新する (20-21 行目)。

5. 数値実験

〈5・1〉 探索性能の評価 数値実験を通じて、提案手法の探索性能を評価する。制約違反量を式 (7) で定義し、目的関数及び制約違反量に正規化を行わないモデルを Proposal1、制約違反量を式 (8) で定義し、目的関数に対しても同様に正規化を行うモデルを Proposal2 とする。ベンチマーク問題には、Test 問題 (app.Table 1 を参照) を使用する。Test 問題では、次元数 $N \in \{10, 50, 100, 300, 500\}$ 、制

[†] 個体群における非劣解の割合は、個体群からランダムに個体 \mathbf{x}^s ($s \in \{1, \dots, m\}$) を選び出した際に、 \mathbf{x}^s が非劣解である確率に等しいため、非劣解の割合を実際に計算する必要はない。

Algorithm 2 Proposed method

```

1: procedure PROPOSED METHOD( $m, T, g_{\max}$ )
2:   Initialize  $\mathbf{x}^{i(1)}$  ( $i = 1, \dots, m$ ).
3:   Initialize  $\alpha^{(1)}$ .
4:   Set  $\mathbf{w}^{i(1)}$  ( $i = 1, \dots, m$ ) by Eq. (9).
5:    $g := 1$ 
6:   while  $g \leq g_{\max}$  do
7:     for  $i = 1$  to  $m$  do
8:       Calculate the Euclidean distance between  $\mathbf{w}^{i(1)}$  and
        $\mathbf{w}^{j(1)}$  ( $j = 1, \dots, m$ ).
9:       Set the neighborhood index  $B(i) = \{i_1, \dots, i_T\}$ .
10:    end for
11:    for  $i = 1$  to  $m$  do
12:      Randomly select  $a, b \in B(i)$ .
13:      Crossover and mutation on  $\mathbf{x}^{a(g)}$  and  $\mathbf{x}^{b(g)}$  to pro-
      duce  $\mathbf{y}^{i(g)}$ .
14:      for  $j \in B(i)$  do
15:        if  $S(\mathbf{y}^{i(g)} | \mathbf{w}^{j(g)}) \leq S(\mathbf{x}^{j(g)} | \mathbf{w}^{j(g)})$  then
16:           $\mathbf{x}^{j(g)} := \mathbf{y}^{i(g)}$ 
17:        end if
18:      end for
19:    end for
20:    Update  $\alpha^{(g)}$  by Eq. (10).
21:    Update  $\mathbf{w}^{i(g)}$  ( $i = 1, \dots, m$ ) by Eq. (9).
22:     $g := g + 1$ 
23:  end while
24:  return  $\mathbf{x}^{i(g_{\max})}$  ( $i = 1, \dots, m$ )
25: end procedure
    
```

約の厳しさ $d \in \{10^{-2}, 10^{-4}\}$ とする。個体群の初期化領域は、探索領域 \mathcal{S} とする。比較手法には、実行不可能解を明示的に活用し、かつパレートランキングを用いる IDEA⁽²²⁾、及び実行不可能解を非明示的に活用し、かつペナルティに基づくアプローチよりも優れた探索性能が報告されている MCR⁽¹⁸⁾ を用いる。

手法の共通条件は基本的に $\langle 4 \cdot 2 \cdot 2 \rangle$ 項と同様とするが、終了条件は目的関数の最大評価回数 5.0×10^5 とする。提案手法のパラメータとして、近傍数 $T = 1/m$ 、 $\alpha^{(1)} = 1$ 、微小値は $\delta = 10^{-15}$ とする。IDEA の固有パラメータは文献⁽²²⁾ を参考に 0.2 とする。50 通りの初期値を用いて、最大評価回数に達した際の $f(\mathbf{x}^*) - f(\mathbf{x}^{\text{best}})$ の平均値 (Mean)、標準偏差 (Std)、実行可能解を得られた試行数 (Feasibility runs) を算出し、比較する。ただし、 \mathbf{x}^{best} は式 (11) で定義される。

$$\mathbf{x}^{\text{best}} = \underset{\mathbf{x}^i \in X}{\operatorname{argmin}} \{f(\mathbf{x}^i) \mid v(\mathbf{x}^i) = 0\} \dots \dots \dots (11)$$

Test 問題に対する実験結果を Table 2 に示す。実験結果より以下のことが確認される。

Proposal1 Prob. 1,2,4 では探索性能が最も優れていることが多いが、Prob. 3 では IDEA や MCR に探索性能が劣っている。

Proposal2 Prob. 1,2,4 では Proposal1 に次いで探索性能が優れていることが多く、全ての問題で実行可能解を得られているが、Prob. 3 では Proposal1 と同様に IDEA や MCR に探索性能が劣っている。

IDEA $N = 10$ では全ての問題で実行可能解を得られており、Prob. 3 ($N = 10, d = 10^{-2}$) では最も探索性能が優れているが、 $N \geq 100$ の場合には Prob. 4 ($d = 10^{-2}$) を除き全ての問題で実行可能解が得られていない。

MCR 全ての問題で実行可能解を得られており、Prob. 3 では最も探索性能が優れていることが多いが、Prob. 1,2,4 では Proposal1 及び Proposal2 に劣っていることが多い。また、Prob. 4 では探索性能が大きく劣っている。

MCR は実行不可能解を明示的に活用しないため探索効率が低く、IDEA は実行不可能解を明示的に活用するものの、パレートランキングを利用するため実行可能解への収束性が低かったと考えられる。一方、提案手法は MOEA/D を基礎として α の調整側を付加することで実行不可能解を適切に活用し、探索性能を向上できたと考えられる。Prob. 3 において提案手法の探索性能が劣る原因は、 $\langle 4 \cdot 2 \cdot 2 \rangle$ 項で考察したように加重和では凹型のパレートフロンティアを近似することが困難であるため、個体群中の実行可能解と実行不可能解の距離が離れていき、実行可能領域の境界上の探索が促進されなかったためだと考えられる。

各手法の Test 問題 ($N = 100$) における $f(\mathbf{x}^*) - f(\mathbf{x}^{\text{best}})$ 及び α の推移を Fig. 4 に示す。なお、 $f(\mathbf{x}^*) - f(\mathbf{x}^{\text{best}})$ の推移は実行可能解が得られた地点から描写している。また、IDEA はほとんどの問題で実行可能解が得られていないため、Fig. 4 から除いている。実験結果から以下のことが確認される。

- Prob. 1,2,4 では Proposal1 及び Proposal2 は MCR よりも \mathbf{x}^{best} が早期に大きく改善される傾向がある。Prob. 3 では Proposal1 及び Proposal2 も \mathbf{x}^{best} が改善されにくい傾向がある。また、Proposal1 は Proposal2 よりも \mathbf{x}^{best} が大きく改善される傾向がある。
- Proposal2 は Proposal1 よりも比較的少ない評価回数で実行可能解が得られている。
- Prob. 4 では Proposal2 は探索序盤に α が増加した後に減少に転じ、それに伴って \mathbf{x}^{best} が改善されている。

Proposal1 及び Proposal2 では探索序盤から実行不可能解を探索に積極的に活用することで、実行可能領域の凸及び非凸に依存せず、MCR よりも早期に解改善を促進できていることが確認できる。また、Proposal1 は実行可能解を得るまでに Proposal1 よりも多くの評価回数を必要とする傾向があるため、高次元かつ制約の厳しい Prob. 1 ($N = 500, d = 10^{-4}$) では実行可能解を得られなかったと考えられる。一方、Proposal2 は比較的少ない評価回数で実行可能解が得られる傾向があるが、実行可能領域が非凸集合となる Prob. 4 では実行可能解を得た後にも大域的な探索を継続するため探索効率が低下し、高次元の Prob. 4 ($N = 500, d = 10^{-4}$) では探索性能が劣っていたと考えられる。さらに、Proposal2 は Proposal1 よりも解改善が停滞しやすいのは、正規化した際に制約違反量の小さい個体が選択操作で淘汰され、実行可能解と実行不可能解の制約違反量 $v(\mathbf{x})$ の差が開いていったためだと考えられる。しかしながら、正規化を施すことで

実行可能解を得られやすくなる傾向があるため、適切な正規化法の検討は今後の課題とする。

以上から、提案手法は全般的に、高次元や制約が厳しい問題に対して IDEA よりも実行可能解への収束性能を向上させ、実行可能領域が非凸集合となる問題に対して MCR よりも大域的最適化性能を向上させたといえる。

〈5・2〉 パラメータ α の適応的調整の評価 提案手法におけるパラメータ α の適応的調整の効果を評価する。Proposal1 及び $\alpha = 10^{-6}, 0.1, 0.2, \dots, 1$ の 11 通りの固定パラメータを用いた MOEA/D (MOEA/D with α) を比較する。ベンチマーク問題には、Test 問題を使用し、次元数 $N = 100$ とする。他の条件は〈5・1〉節と同様とする。

実験結果を Fig. 5 に示す。実験結果より以下のことが確認される。

- Prob. 1,2,4 において MOEA/D with α は α の値に応じて探索性能が変動しているが、Prob. 3 では探索性能の大きな変動は見られない。また、Prob. 1 ($d = 10^{-4}$) では $\alpha \geq 0.2$ では実行可能解を得られていない。
- Proposal1 は最も探索性能が高くなる MOEA/D with α と同等程度の探索性能となっている。

以上から、 α を適切な値に設定しなければ、実行可能解を得られないことや、探索性能が悪化することがわかる。〈4・2〉節で述べたように、固定の α を用いると問題に応じて Feasibility frequency は大きく変化するが、提案手法では実行可能領域と実行不可能領域への偏り状況を判断することで、個体群中の実行可能解と実行不可能解をバランス良く維持し、多くの問題で安定して良好な実行可能解を得られたと考えられる。また、Fig. 4 の α の推移と比較すると、MOEA/D with α において最も高い探索性能を引き出す α の値は、Proposal1 が適応される α の値と同等程度となっている。以上から、提案手法は α が適切に調整された結果、良好な実行可能解への収束性能を向上させたといえる。

6. おわりに

本論文では、実行可能解への収束性と非凸集合となる実行可能領域に対する大域的最適化性能の向上を目的として、有制約最適化のための適応的重み調整を用いた MOEA/D を提案した。目的関数及び制約違反量に対して正規化を施すモデルとそうでないモデルを構築し、実行可能領域が凸集合及び非凸集合となる問題に適用し、分離に基づくアプローチである MCR や、多目的に基づくアプローチである IDEA よりも実行可能解への収束性能及び大域的最適化性能の向上を確認した。

提案手法では正規化を施すことで実行可能解を早期に発見可能となるが、解改善が停滞しやすくなるため、適切な正規化法の開発が求められる。また、提案手法では交叉として SBX を用いているが、Simplex Crossover (SPX)⁽²⁹⁾ や Unimodal Normal Distribution Crossover (UNDX)⁽³⁰⁾ などの交叉法、MOEA/D よりも優れた探索性能が報告されている MOEA/D-DE⁽³¹⁾ などとの比較実験により、適切な近傍

生成法の選択・開発も必要である。さらに、本論文では制約が 1 個の人工的な関数を対象としたが、多数制約の問題や実問題における提案手法の性能検証も今後の課題として挙げられる。

文 献

- (1) 相吉・安田 編著：メタヒューリスティクスと応用，オーム社 (2007)
- (2) D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley (1989)
- (3) H.-P. Schwefel: "Evolution and Optimum Seeking", Wiley (1995)
- (4) J. Kennedy and R.C. Eberhart: "Particle Swarm Optimization", Proc. of IEEE Inter. Conf. on Neural Networks, Vol. 4, pp. 1942-1948 (1995)
- (5) R.M. Storn and K.V. Price: "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Space", Journal of Global Optimization, Vol. 11, No. 4, pp. 341-359 (1997)
- (6) C. A. C. Coello: "Theoretical and Numerical Constraint-handling Techniques Used with Evolutionary Algorithms: A Survey of The State of The Art", Computer Methods in Applied Mechanics and Engineering, Vol. 191, No. 11-12, pp. 1245-1287 (2002)
- (7) E. Mezura-Montes and C. A. C. Coello: "Constraint-handling in Nature-inspired Numerical Optimization: Past, Present and Future, Swarm and Evolutionary Computation, Vol. 1, No. 4, pp. 173-194 (2011)
- (8) Z. Michalewicz: "A Survey of Constraint Handling Techniques in Evolutionary Computation Methods", Proc. of the Fourth Annual Conference on Evolutionary Programming, pp. 135-155 (1995)
- (9) A. Homaidan, C. X. Qi, and S. H. Lai: "Constrained optimization via genetic algorithms", Simulation, Vol. 62, No. 4, pp. 242-253 (1994)
- (10) J. Joines and C. Houck: "On The Use of Non-stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with Gas", Proc. of the First IEEE Conf. on Evolutionary Computation, pp. 579-584 (1994)
- (11) A. C. Lemonge and H. J. Barbosa: "An Adaptive Penalty Scheme for Genetic Algorithms in Structural Optimization", International Journal for Numerical Methods in Engineering, Vol. 59, No. 5 pp. 703-736 (2004)
- (12) C. Saha, S. Das, K. Pal, and S. Mukherjee: "A Fuzzy Rule-based Penalty Function Approach for Constrained Evolutionary Optimization", IEEE Trans. on Cybernetics, Vol. 46, No. 12, pp. 2953-2965 (2014)
- (13) S. B. Hamida and M. Schoenauer: "An Adaptive Algorithm for Constrained Optimization Problems", Inter. Conf. on Parallel Problem Solving from Nature, pp. 529-538 (2000)
- (14) K. Deb: "An Efficient Constraint Handling Method for Genetic Algorithms", Computer Methods in Applied Mechanics and Engineering, Vol. 186 No. 2-4, pp. 311-338 (2000)
- (15) T. P. Runarsson and X. Yao: "Stochastic Ranking for Constrained Evolutionary Optimization", IEEE Trans. on Evolutionary Computation, Vol. 4, No. 3, pp. 284-294 (2000)
- (16) T. Takahama, S. Sakai, and N. Iwano: "Constrained Optimization by The ϵ Constrained Hybrid Algorithm of Particle Swarm Optimization and Genetic Algorithm", Proc. of Australasian Joint Conf. on Artificial Intelligence, pp. 389-400 (2005)
- (17) T. Takahama and S. Sakai: "Constrained Optimization by The ϵ Constrained Differential Evolution with Gradient-based Mutation and Feasible Elites", Proc. of 2006 IEEE Inter. Conf. on Evolutionary Computation, pp. 1-8 (2006)
- (18) R. Paula Garcia, B. S. L. P. de Lima, A. C. Castro, A. C. Lemonge, and B. P. Jacob: "A Rank-based Constraint Handling Technique for Engineering Design Optimization Problems Solved by Genetic Algorithms", Computers and Structures, Vol. 187, pp. 77-87 (2017)
- (19) B. Liu, H. Ma, X. Zhang, and Y. Zhou: "A Memetic Co-evolutionary Differential Evolution Algorithm for Constrained Optimization", Proc. of 2007 IEEE Congress on Evolutionary Computation, pp. 2996-3002 (2007)
- (20) M. Miyakawa, K. Takadama, and H. Sato: "Two-stage Non-dominated Sorting and Directed Mating for Solving Problems with Multi-Objectives and Constraints", Proc. of 2013 Genetic and Evolutionary Computation Conf., pp. 647-654 (2013)
- (21) S. Venktraman and G. G. Yen: "A Generic Framework for Constrained Optimization Using Genetic Algorithms", IEEE Trans. on Evolutionary Computation, Vol. 9, No. 4, pp. 424-435 (2005)
- (22) T. Ray, H. K. Singh, A. Isaacs, and W. Smith: "Infeasibility Driven Evolutionary Algorithm for Constrained Optimization", Constraint-handling in Evolutionary Optimization, pp. 145-165 (2009)
- (23) B. Wang, H. Li, Q. Zhang, and Y. Wang: "Decomposition-Based Multi-

Table 2. Experimental results in the Test problems, where the best MEAN are shown in bold

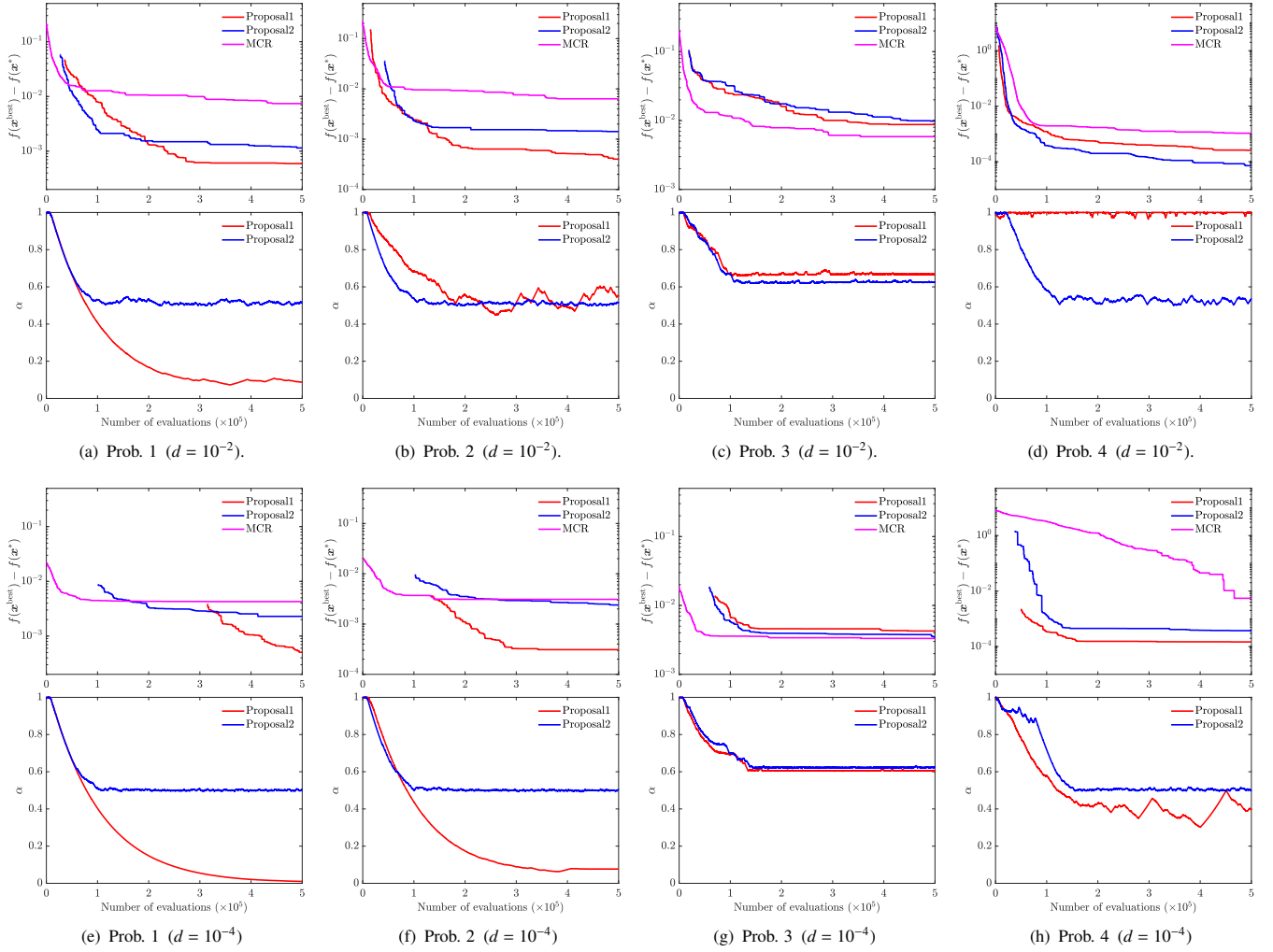
Prob.	Dim.	Proposal1			Proposal2			IDEA			MCR		
		Mean	Std	Feas. runs	Mean	Std	Feas. runs	Mean	Std	Feas. runs	Mean	Std	Feas. runs
Prob. 1 ($d = 10^{-2}$)	10	3.41E-05	2.87E-05	50	1.12E-04	8.28E-05	50	2.28E-04	9.98E-05	50	1.32E-03	6.48E-04	50
	50	1.93E-04	6.93E-05	50	6.68E-04	2.73E-04	50	1.98E-02	0	1	4.24E-03	8.90E-04	50
	100	4.41E-04	9.75E-05	50	1.36E-03	4.58E-04	50	-	-	0	6.55E-03	9.73E-04	50
	300	1.36E-03	1.51E-04	50	4.41E-03	1.19E-03	50	-	-	0	1.17E-02	9.99E-04	50
	500	2.38E-03	2.46E-04	50	7.30E-03	1.55E-03	50	-	-	0	1.56E-02	1.33E-03	50
Prob. 2 ($d = 10^{-2}$)	10	4.16E-05	3.06E-05	50	1.19E-04	8.23E-05	50	2.47E-04	1.08E-04	50	1.23E-03	6.31E-04	50
	50	2.16E-04	7.75E-05	50	5.64E-04	2.36E-04	50	-	-	0	4.18E-03	8.38E-04	50
	100	4.71E-04	1.47E-04	50	1.21E-03	3.78E-04	50	-	-	0	6.27E-03	1.04E-03	50
	300	1.50E-03	3.75E-04	50	3.22E-03	9.96E-04	50	-	-	0	1.18E-02	1.03E-03	50
	500	2.38E-03	4.10E-04	50	6.38E-03	1.78E-03	50	-	-	0	1.54E-02	1.20E-03	50
Prob. 3 ($d = 10^{-2}$)	10	1.74E-03	8.57E-04	50	1.66E-03	7.33E-04	50	2.48E-04	1.10E-04	50	1.19E-03	5.93E-04	50
	50	6.66E-03	1.46E-03	50	6.52E-03	1.19E-03	50	-	-	0	3.89E-03	8.27E-04	50
	100	1.06E-02	1.50E-03	50	1.04E-02	1.36E-03	50	-	-	0	6.34E-03	8.40E-04	50
	300	2.01E-02	1.86E-03	50	1.96E-02	1.69E-03	50	-	-	0	1.21E-02	1.01E-03	50
	500	2.54E-02	1.81E-03	50	2.57E-02	1.34E-03	50	-	-	0	1.49E-02	1.16E-03	50
Prob. 4 ($d = 10^{-2}$)	10	1.92E-05	1.06E-05	50	7.92E-06	6.07E-06	50	2.79E-05	1.68E-05	50	2.63E-04	1.07E-04	50
	50	1.51E-04	4.10E-05	50	3.26E-05	7.67E-06	50	3.25E-04	7.89E-05	50	7.22E-04	1.27E-04	50
	100	2.89E-04	4.64E-05	50	7.52E-05	1.91E-05	50	5.16E-04	7.25E-05	50	1.17E-03	1.82E-04	50
	300	8.45E-04	8.44E-05	50	2.50E-04	4.16E-05	50	9.87E-04	7.35E-05	50	2.14E-03	1.91E-04	50
	500	1.45E-03	9.40E-05	50	4.27E-04	4.92E-05	50	1.42E-03	8.78E-05	50	3.02E-03	2.46E-04	50
Prob. 1 ($d = 10^{-4}$)	10	3.86E-05	2.54E-05	50	3.92E-04	1.93E-04	50	3.58E-04	3.06E-04	50	1.33E-03	5.74E-04	50
	50	1.72E-04	5.21E-05	50	1.77E-03	3.75E-04	50	-	-	0	3.08E-03	4.46E-04	50
	100	3.47E-04	7.11E-05	50	2.54E-03	3.11E-04	50	-	-	0	3.45E-03	4.12E-04	50
	300	1.65E-03	2.26E-04	50	4.08E-03	3.29E-04	50	-	-	0	3.82E-03	2.16E-04	50
	500	-	-	0	5.91E-03	3.22E-04	50	-	-	0	4.15E-03	2.47E-04	50
Prob. 2 ($d = 10^{-4}$)	10	3.78E-05	2.90E-05	50	3.95E-04	2.01E-04	50	4.64E-04	3.83E-04	50	1.41E-03	6.55E-04	50
	50	1.79E-04	7.05E-05	50	1.72E-03	4.40E-04	50	-	-	0	3.18E-03	5.46E-04	50
	100	3.20E-04	8.59E-05	50	2.49E-03	3.58E-04	50	-	-	0	3.43E-03	4.43E-04	50
	300	1.10E-03	1.45E-04	50	4.24E-03	3.43E-04	50	-	-	0	3.84E-03	2.69E-04	50
	500	2.42E-03	1.85E-04	50	5.99E-03	3.85E-04	50	-	-	0	4.07E-03	2.20E-04	50
Prob. 3 ($d = 10^{-4}$)	10	2.18E-03	7.49E-04	50	2.17E-03	9.11E-04	50	4.43E-04	2.29E-04	50	1.44E-03	6.54E-04	50
	50	4.13E-03	7.28E-04	50	4.25E-03	5.69E-04	50	-	-	0	3.25E-03	4.80E-04	50
	100	4.50E-03	4.91E-04	50	4.34E-03	4.53E-04	50	-	-	0	3.51E-03	4.28E-04	50
	300	4.86E-03	3.54E-04	50	4.94E-03	3.43E-04	50	-	-	0	3.82E-03	2.81E-04	50
	500	6.37E-03	4.02E-04	50	6.17E-03	3.39E-04	50	-	-	0	4.10E-03	2.27E-04	50
Prob. 4 ($d = 10^{-4}$)	10	1.11E-05	9.98E-06	50	5.06E-05	3.08E-05	50	1.91E-05	1.04E-05	50	2.83E-04	1.49E-04	50
	50	7.16E-05	2.58E-05	50	2.17E-04	5.85E-05	50	-	-	0	1.35E-03	4.44E-03	50
	100	1.45E-04	3.41E-05	50	3.66E-04	7.32E-05	50	-	-	0	1.33E-02	1.75E-02	50
	300	5.56E-04	6.53E-05	50	9.41E-04	9.96E-05	50	-	-	0	1.20E+00	2.01E-01	50
	500	8.10E-04	1.57E-04	50	1.03E-01	8.70E-02	50	-	-	0	2.82E+00	2.28E-01	50

objective Optimization for Constrained Evolutionary Optimization”, IEEE Trans. on Systems, Man, and Cybernetics: Systems, Vol. 51, No. 1, pp. 574-587 (2021)

- (24) H. K. Singh, K. Alam, and T. Ray: “Use of Infeasible Solutions During Constrained Evolutionary Search: A Short Survey”, Proc. of Australasian Conf. on Artificial Life and Computational Intelligence, pp. 193-205 (2016)
- (25) K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan: “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, IEEE Trans. on Evolutionary Computation, Vol. 6, No. 2, pp. 182-197 (2002)
- (26) Y. Yasuda, W. Kumagai, K. Tamura, and K. Yasuda: “An Extension of MOEA/D to Constrained Optimization and Adaptive Weight Adjustment”, IEEE Trans. on Electronics, Information and Systems, Vol. 142, No. 1 pp. 108-109 (2022) (in Japanese)
安田雄佑・熊谷渉・田村健一・安田恵一郎: 「MOEA/D の有制約最適化への拡張と適応的重み調整に関する基礎検討」, 電学論 C, Vol. 142, No. 1 pp. 108-109 (2022)
- (27) Q. Zhang and H. Li: “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”, IEEE Trans. on Evolutionary Computation, Vol.

11, No. 6 pp. 712-731 (2007)

- (28) K. Deb and R. B. Agrawal: “Simulated Binary Crossover for Continuous Search Space”, Complex Systems, Vol. 9, No. 2, pp.115-148 (1995)
- (29) S. Tsutsui, M. Yamamura, and T. Higuchi: “Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms”, Proc. of 1999 Genetic and Evolutionary Computation Conf., pp. 657-664 (1999)
- (30) H. Kita, I. Ono, and S. Kobayashi: “Multi-parental Extension of The Unimodal Normal Distribution Crossover for Real-coded Genetic Algorithms”, Trans. of the society of Instrument and Control Engineers, Vol. 36, No. 10, pp. 875-883 (2000)
- (31) H. Li and Q. Zhang: “Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II”, IEEE Trans. on Evolutionary Computation, Vol. 13, No. 2, pp. 284-302 (2008)


 Fig. 4. Transitions of $f(\mathbf{x}^*) - f(\mathbf{x}^{\text{best}})$ and α in Test problems ($N = 100$)

付 録

1. Test 問題

本稿で用いた Test 問題を説明する。Test 問題の定義を app.Table 1 に示す。 $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^N$, $\text{sgn} : \mathbb{R} \rightarrow \{-1, 1\}$ は符号関数, d は制約の厳しさを設定するパラメータである。Prob. 1,2,3,4 は目的関数及び大域的最適解は共通するが, 制約関数がそれぞれ異なるスケール・景観となる。制約関数が $g^1(\mathbf{x})$ (Prob. 1) は 2 次関数, $g^2(\mathbf{x})$ (Prob. 2) は指数関数, $g^3(\mathbf{x})$ (Prob. 3) は累乗根を含む無理関数であり, 凸集合の実行可能領域を持つ。 $g^4(\mathbf{x})$ (Prob. 4) は多峰性関数であり非凸集合 (非連結) となる実行可能領域を持つ。

安 田 雄 佑 (学生員) 2020 年 3 月首都大学東京都市教養学部都市教養学科電気電子工学コース卒業。同年 4 月東京都立大学大学院システムデザイン研究科電子情報システム工学域博士前期課程進学, 現在に至る。システム最適化, 主としてメタヒューリスティクスに関する研究に従事。

熊 谷 渉 (正員) 2016 年 4 月横河電機株式会社入社。2020 年 9 月東京都立大学大学院理工学研究科電気電子工学専攻博士後期課程を修了し, 現在に至る。システム最適化 (主にメタヒューリスティクス), 機械学習, およびエネルギー最適化に関する研究に従事。博士 (工学)。

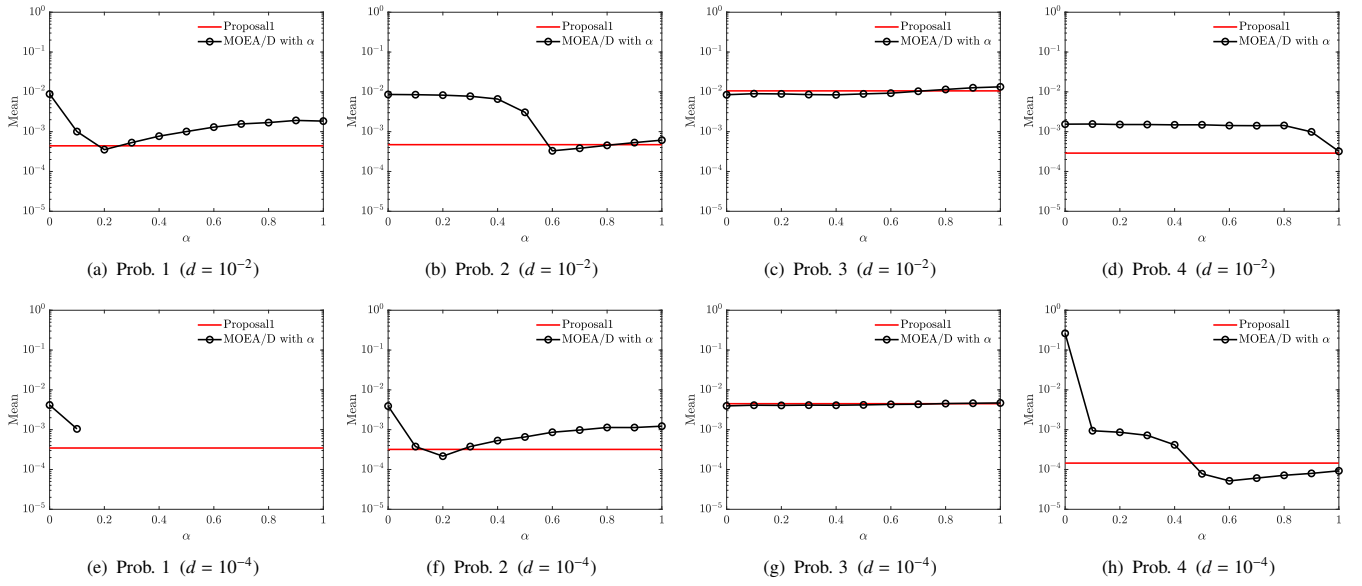


Fig.5. Search performance between Proposal1 and MOEA/D with α when $\alpha = 10^{-6}, 0.1, 0.2, \dots, 1$ is set ($N = 100$)

app. Table 1. Test problems

Prob.	Function	Definition	Optimal solution \mathbf{x}^*	Optimal value $f(\mathbf{x}^*)$	Search space \mathcal{S}
Prob. 1	Obj.	$f^1(\mathbf{x}) = \sum_{j=1}^N x_j^2/N$	$[1 - \sqrt{d}, \dots, 1 - \sqrt{d}]^T$	$(1 - \sqrt{d})^2$	$[-5, 5]^N$
	Cons.	$g^1(\mathbf{x}) = \sum_{j=1}^N (x_j - 1)^2/N - (\sqrt{d})^2$			
Prob. 2	Obj.	$f^2(\mathbf{x}) = \sum_{j=1}^N x_j^2/N$	$[1 - \sqrt{d}, \dots, 1 - \sqrt{d}]^T$	$(1 - \sqrt{d})^2$	$[-5, 5]^N$
	Cons.	$g^2(\mathbf{x}) = \exp(10g^1(\mathbf{x})) - 1$			
Prob. 3	Obj.	$f^3(\mathbf{x}) = \sum_{j=1}^N x_j^2/N$	$[1 - \sqrt{d}, \dots, 1 - \sqrt{d}]^T$	$(1 - \sqrt{d})^2$	$[-5, 5]^N$
	Cons.	$g^3(\mathbf{x}) = \text{sgn}(g^1(\mathbf{x})) g^1(\mathbf{x}) ^{1/4}$			
Prob. 4	Obj.	$f^4(\mathbf{x}) = \sum_{j=1}^N x_j^2/N$	$[0.25 - \sqrt{d}, \dots, 0.25 - \sqrt{d}]^T$	$(0.25 - \sqrt{d})^2$	$[-5, 5]^N$
	Cons.	$g^4(\mathbf{x}) = -\sum_{j=1}^N \cos(2\pi(x_j - 0.25))/N + \cos(2\pi\sqrt{d})$			

田村 健一 (正員) 2008 年 9 月慶應義塾大学大学院理工学研究科総合デザイン工学専攻後期博士課程修了。同年 10 月首都大学東京理工学研究科助教となり、現在に至る。システム最適化およびシステム制御の研究に従事。博士 (工学)。

安田 恵一郎 (フェロー) 1989 年 3 月北海道大学大学院工学研究科電気工学専攻博士課程修了。同年 4 月東京都立大学工学部助手, 91 年同大助教授, 2006 年首都大学東京理工学研究科教授となり、現在に至る。システム最適化および電力系統工学の研究に従事。工学博士。