

多目的制約最適化問題：ユーザとの対話型解法の提案

Interactive Algorithm for Multi-objective Constraint Optimization

沖本 天太*
Tenda Okimoto

ジョ ヨンジュン*
Yongjoon Joe

岩崎 敦*
Atsushi Iwasaki

横尾 真*
Makoto Yokoo

1 序論

実世界に存在する様々な最適化問題では、複数の異なる評価基準を同時に考慮する場合が存在する。多目的制約最適化問題 (Multi-objective Constraint Optimization Problem, MO-COP) [5, 6, 7, 12] は、異なる評価基準をもつ複数の目的関数が存在する制約最適化問題 (Constraint Optimization Problem, COP) [3, 14] である。制約最適化問題とは、有限で離散的な領域から値をとる複数の変数に、ある目的関数を最大化するように値を割当てる問題である。多目的制約最適化問題は、単一目的の制約最適化問題を多目的へと拡張した問題である。この問題では、一般には、複数の異なる目的関数間にトレードオフの関係があるため、すべての目的関数を同時に最大化するような割当は存在しない。そのため、多目的制約最適化問題では、パレート最適性の概念を用いて最適解を特徴づける。多目的制約最適化問題を解くとはパレートフロントを求めることである。パレートフロントとは、パレート解によって得られる利得ベクトルの集合である。ある割当がパレート解であるとは、すべての評価基準において、その割当によって得られる利得ベクトルを改善するような他の割当が存在しないことを意味する。制約最適化問題 / 多目的制約最適化問題は、変数をノードに、制約をノード間のエッジに対応させることにより、グラフ (制約グラフ) を用いて表現できる。

多目的制約最適化問題の解法は完全解法と近似解法に大別される。完全解法では、すべてのパレートフロントが求解可能であり、Multi-objective Russian Doll Search algorithm (MO-RDS) [13], Multi-objective AND/OR Branch-and-Bound search algorithm (MO-AOBB) [6], MultiObjective Bucket Elimination (MO-BE) [12] 等がある。多目的制約最適化問題では、最も簡単な木構造の問題でも、パレートフロントの個数が変数の数に対して、指数関数的に増加する場合がある。このような問題では、すべてのパレートフロントを求めることは現実的ではない。そのため、制約最適化問題の近似手法として知られている Mini-Bucket Elimination [3] に基づいた Multi-Objective Mini-Bucket Elimination (MO-MBE) [12] や、 ϵ -dominance [10] を用いて緩いパレートフロントを求める Multi-objective Best-First AND/OR search algorithm (MO-AOBF) [7] や Multiobjective A* search al-

gorithm (MOA*) [11] 等の近似解法が提案されている。

多目的最適化問題のパレート解を求める古典的手法に線形化加重和法 (Aggregate Objective Function, AOF) がある [9]。AOF は、各目的関数に重みを与えることにより、単一の重み付き目的関数を作り、その最適解を求める。このとき得られる最適な割当は、元の問題のパレート解となることが示されている [9]。AOF は、考え方が極めて容易であり、実問題に対して簡単に適用できる。

従来、様々な多目的最適化問題が定義され、多くの解法が提案されてきた [1, 2, 4, 8]。これらのほとんどは、変数が連続値を取る最適化問題を対象としている。このことは、離散最適化問題が NP 完全であることから実問題に対する有効な解法の開発が難しいことに起因する。本論文では、変数が離散値を取る制約最適化問題 (離散最適化問題) を対象としている。

多目的制約最適化問題では、数多くの最適解の候補からユーザの選好に最もあった解を選び出す必要がある。本論文では、パレート解が存在する領域を、ユーザとのインタラクションにより段階的に狭めていく対話型解法 (Interactive Algorithm for MO-COP, MO-IA) を提案する。本解法は、擬似木に基づく対話型解法である。擬似木とは、制約最適化問題の解法で広く用いられているグラフ構造である。本解法の計算 / 空間複雑度は問題の誘導幅で抑えられる。誘導幅とは、制約最適化問題の解法の複雑度を決定する指標である。実験では、ミクロ経済学で広く用いられているユーザの選好に関する既存のモデル、Constraint Elasticity of Substitution (CES) 型効用関数 [16]、を用いて本解法を評価し、その有効性を示す。さらに、本解法の拡張として、パレート解が存在する領域を、あらかじめ狭めた状態でユーザに提示する方法を提案する。

本解法と類似した対話型解法に Physical Programming (PP) [8] や Directed Search Domain algorithm (DSD) [4] がある。PP や DSD は変数が連続値を取る多目的最適化問題の解法である。また本解法と遺伝的解法 [1] の違いは、本解法では、パレート解が得られることを保証している点である。

本論文は、序論と結論を含めて全体を 6 章で構成している。2 章では準備として関連研究について述べる。3 章では多目的制約最適化問題における対話型解法を提案し、4 章ではユーザの選好に関する既存のモデルを用いて本解法を評価する。5 章では考察、6 章では結論と今後の課題について述べる。

* 九州大学大学院システム情報科学府

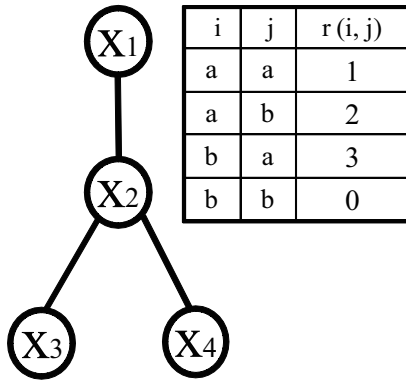


図 1 4 個の変数からなる制約最適化問題の例。各制約の利得は図中の利得表で与えられているものとする ($i < j$)。この問題の解、すなわち、利得の総和を最大化するような最適な割当 (最適解) は $\{(x_1, a), (x_2, b), (x_3, a), (x_4, a)\}$ であり、このとき得られる利得の総和 (最適値) は 8 である。

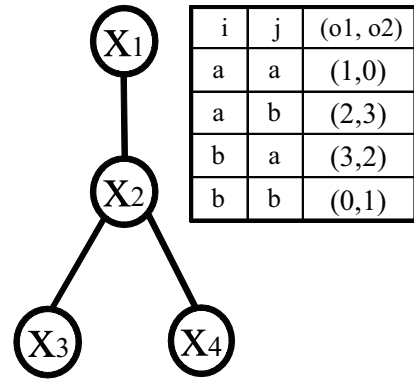


図 2 4 個の変数からなる 2 目的制約最適化問題の例。この問題のパレート解は $\{(x_1, b), (x_2, a), (x_3, b), (x_4, b)\}, \{(x_1, a), (x_2, b), (x_3, a), (x_4, a)\}$ であり、パレートフロントは $\{(7, 8), (8, 7)\}$ である。

2 準備

本章では、制約最適化問題および多目的制約最適化問題の定義を示し、多目的制約最適化問題を解く古典的手法について概説する。また制約最適化問題の解法で広く用いられているグラフ構造を示す。さらに、ミクロ経済学で広く用いられている、ユーザの選好に関する既存のモデルについて概説する。

2.1 制約最適化問題

制約最適化問題 (Constraint Optimization Problem, COP) [3, 14] とは、有限で離散的な領域から値をとる複数の変数に、ある目的関数を最大化するように値を割当てて問題である。制約最適化問題は、変数の集合 X 、二項制約の集合 C 、二項利得関数の集合 F により定義される。各変数 x_i は離散有限集合 D_i に含まれる変数の値を取る。制約 (i, j) は x_i と x_j の間に制約があることを示す ($i < j$)。制約で関係する 2 変数についての、ある割当 $\{(x_i, d_i), (x_j, d_j)\}$ の利得は、二項利得関数 $r_{i,j} : D_i \times D_j \rightarrow \mathbb{R}$ により定義される。すべての変数への割当 A に関して、

$$R(A) = \sum_{(i,j) \in C, \{(x_i, d_i), (x_j, d_j)\} \subseteq A} r_{i,j}(d_i, d_j)$$

を利得関数の合計値として、問題の解は $\operatorname{argmax}_A R(A)$ 、すなわち、 $R(A)$ を最大化する値の割当である。制約最適化問題は変数をノードに、制約をノード間のエッジに対応させることにより、グラフ (制約グラフ) を用いて表すことができる。

例 1 (制約最適化問題)。図 1 は 4 個の変数からなる制約最適化問題 (制約グラフ) を表す。各変数は a または b の値を取る。各制約の利得は図中の利得表で与えられているものとする。たとえば、すべての変数の割当を a とするとき、得られる利得の総和は 3 である。この問題の解、すなわち、利得の総和を最大化するような最適な割当は $\{(x_1, a), (x_2, b), (x_3, a), (x_4, a)\}$ であり、このとき得られる利得の総和は 8 である。

2.2 多目的制約最適化問題

多目的制約最適化問題 (Multi-objective Constraint Optimization Problem, MO-COP) [5, 6, 7, 12] とは、異なる評価基準をもつ複数の目的関数が存在する制約最適化問題である。この問題は、単一目的の制約最適化問題を多目的へと拡張した問題である。多目的制約最適化問題は、変数の集合 X 、二項制約の集合からなる集合 $C = \{C^1, \dots, C^m\}$ 、目的関数 (二項利得関数) の集合からなる集合 $O = \{O^1, \dots, O^m\}$ により定義される。各 C^l ($1 \leq l \leq m$) は目的 l に関する二項制約の集合を表す。各 O^l は目的 l に関する二項利得関数の集合を表す。変数 x_i は離散有限集合 D_i に含まれる変数の値を取る。制約 (i, j) は x_i と x_j の間に制約があることを示す ($i < j$)。各目的 l に関して、制約で関係する 2 変数についての、ある割当 $\{(x_i, d_i), (x_j, d_j)\}$ の利得は、二項利得関数 $r_{i,j}^l : D_i \times D_j \rightarrow \mathbb{R}$ により定義される。すべての変数への割当 A に関して、

$$R^l(A) = \sum_{(i,j) \in C^l, \{(x_i, d_i), (x_j, d_j)\} \subseteq A} r_{i,j}^l(d_i, d_j)$$

を目的 l に関する利得関数の合計値として、問題の解は利得ベクトル $R(A) = (R^1(A), \dots, R^m(A))$ により定義される。すべての目的関数を同時に最大化するような割当が存在すれば理想的であるが、一般には、目的関数間にトレードオフの関係があるため、そのような割当は存在しない。そのため、多目的制約最適化問題では、パレート最適性の概念を用いて最適解を特徴づける。本論文では、すべての利得は非負とする。

定義 1 (支配)。多目的制約最適化問題に関して、 $R(A)$ および $R(A')$ を割当 A および A' によって得られる利得ベクトルとする。 $R(A), R(A')$ に関して、以下が成立するとき、 $R(A)$ は $R(A')$ を支配するといい、 $R(A') \prec R(A)$ と記述する。

- (i) すべての目的 l に関して $R^l(A') \leq R^l(A)$ かつ、
- (ii) 少なくとも 1 つの l に関して $R^l(A') < R^l(A)$ 。

定義 2 (パレート解). 多目的制約最適化問題に関して, ある割当 A がパレート最適であるとは, $R(A) \prec R(A')$ が成立するような割当 A' が存在しないことを意味する. また, パレート最適な割当をパレート解と呼ぶ.

定義 3 (パレートフロント). 多目的制約最適化問題に関して, パレート解によって得られる利得ベクトルの集合をパレートフロントと呼ぶ. 多目的制約最適化問題を解くとは, パレートフロントを求めることである.

例 2 (多目的制約最適化問題). 図 2 は図 1 の問題を 2 目的へと拡張した 2 目的制約最適化問題を表す. 各変数は a または b の値を取る. o_1 および o_2 は, それぞれ目的 1 および 2 における利得を表す. 各制約の利得は図中の利得表で与えられているものとする ($i < j$). この問題のパレート解は $\{(x_1, b), (x_2, a), (x_3, b), (x_4, b)\}$, $\{(x_1, a), (x_2, b), (x_3, a), (x_4, a)\}$ であり, パレートフロントは $\{(7, 8), (8, 7)\}$ である.

2.3 擬似木

擬似木 [14] とは, 最適化問題の解法で広く用いられているグラフ構造である. 擬似木は元のグラフと同一のエッジおよびノードからなり, 元のグラフの生成木のいずれか 1 つに対応する. グラフ内のエッジは, 生成木の木辺または, それ以外の後退辺に分類される. 擬似木では, ルートノードから, いずれか 1 つの葉ノードへのパスに含まれる 2 ノード間のみ後退辺があり, 異なるパスに含まれる 2 ノード間には後退辺はない. 本論文では, 擬似木を以下のように構成する. 変数の優先順位を $\langle x_1, x_2, \dots, x_n \rangle$ (x_1 の優先順位が最も高く, x_n の優先順位が最も低い) として, (i) x_n の親ノードを, x_n とエッジで関連する変数中で, 最も優先順位の低い変数とする. (ii) 以下, x_{n-1}, x_{n-2} の順に, ノード x_i の親ノードを以下のように決定する. x_i および x_i の子孫ノードが関連する, x_i より優先順位の高い変数中で, 最も優先順位の低い変数を x_i の親ノードとする. ノード x_i と制約辺で関連するノードおよびノードの集合を次の表記で示す.

- P_i : 木辺で x_i と関連する x_i の親ノード.
- PP_i : 後退辺で x_i と関連する x_i の祖先ノードの集合.
- C_i : 木辺で x_i と関連する x_i の子ノードの集合.

2.4 線形化加重和法 (AOF)

多目的最適化問題のパレート解を求める古典的手法に線形化加重和法 (Aggregate Objective Function, AOF) がある [9]. AOF は, 各目的関数に重みを与えることにより, 単一の重み付き目的関数を作り, その最適解を求める. 具体的には, 多目的最適化問題の m 個の目的関数 (o^1, \dots, o^m) に関して, 重み α を以下のように定義する.

$$\alpha = (\alpha_1, \dots, \alpha_m), \sum_{1 \leq i \leq m} \alpha_i = 1, \alpha_i > 0$$

次に, m 個の重み付き目的関数 $\alpha_1 o^1, \dots, \alpha_m o^m$ を用いて,

単一の重み付き目的関数 $\alpha_1 o^1 + \dots + \alpha_m o^m$ を作り, その最適解を求める. このとき以下が成立する [9].

定理 1. 多目的最適化問題に関して, AOF を用いて得られる最適な割当は元の問題のパレート解である.

本論文では, AOF を多目的制約最適化問題に適用する. 紙面の都合上, 証明は省くが 定理 1 は変数が離散値を取る多目的制約最適化問題でも成立する.

2.5 CES 型効用関数

効用関数とは, ユーザの選好 (価値観) を定量的に表現するための数学モデルである. 本論文では, ユーザの選好をミクロ経済学で広く用いられている Constraint Elasticity of Substitution (CES) 型効用関数 [16] で表現する. CES 型効用関数 u は以下の式で与えられる.

$$u(x_1, \dots, x_m) = (\alpha_1 x_1^p + \dots + \alpha_m x_m^p)^{1/p}, \text{ where} \\ \sum_{1 \leq i \leq m} \alpha_i = 1, \alpha_i > 0$$

ここで, パラメータ p は $p < 1$ を満たす任意の実数である. CES 型効用関数は, 効用関数の典型的な例として用いられる線形, コブ・ダグラスおよびレオンチェフ関数を特殊ケースとして含む. たとえば, CES 型効用関数は, パラメータ p が 1 に近づくとき, 線形の関数に収束する.

$$u(x_1, \dots, x_m) = \alpha_1 x_1 + \dots + \alpha_m x_m$$

また p が 0 に近づくときは, コブ・ダグラス関数に収束し,

$$u(x_1, \dots, x_m) = x_1^{\alpha_1} \times \dots \times x_m^{\alpha_m}$$

p が $-\infty$ に近づくときは, レオンチェフ関数に収束する.

$$u(x_1, \dots, x_m) = \min(x_1, \dots, x_m)$$

2.6 無差別曲線

無差別曲線 [15] とは, 同じ効用水準をもたらす, 各目的関数の値の組合せ全体をさす. 異なる 2 つの組合せ, たとえば, 10 枚の CD と 150 本のキャンディの組と 12 枚の CD と 130 本のキャンディの組が同じ無差別曲線上にあるとは, ユーザが, どちらを選択してもユーザの満足度は同じ (無差別) であることを意味する. 図 3(a) および 図 3(b) に目的数が 2 のコブ・ダグラスおよびレオンチェフ型効用関数の無差別曲線を示す. X 軸は目的 1 における利得を表し, Y 軸は目的 2 における利得を表す. 無差別曲線の性質を以下にまとめる [15].

- 原点に対して凸である.
- 個々の無差別曲線は互いに交わらない.
- 原点から遠い無差別曲線ほど効用水準は高い.

無差別曲線が原点に対して凸であることは, 限界代替率減とよばれる性質によって導かれる. すなわち, CD を 1 枚しかもっていない場合と, CD をすでに 1000 枚もっている場合のそれぞれに関して, 追加の一枚の CD を得ることと引き替

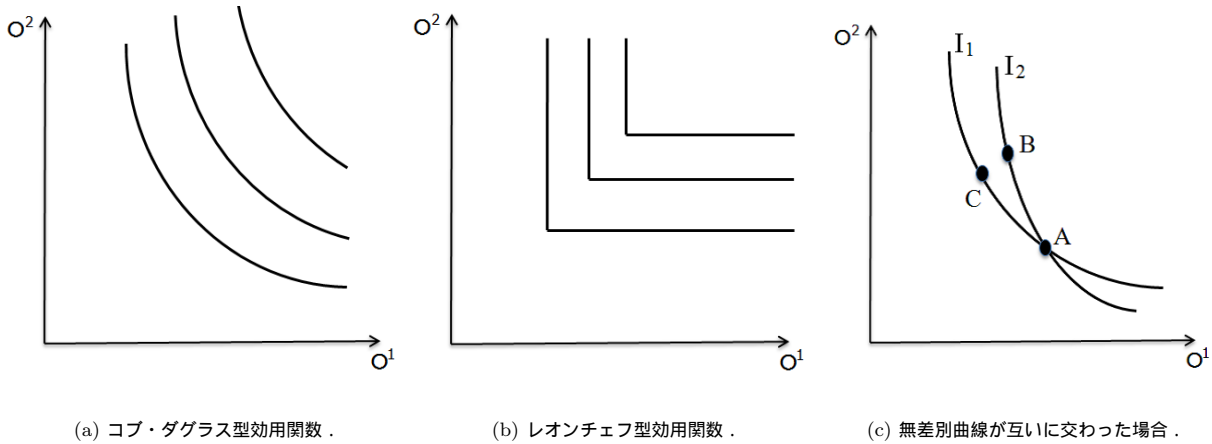


図 3 2 目的制約最適化問題における、レオンチェフおよびコブ・ダグラス型効用関数の無差別曲線。

えに、保有を諦めてもよいキャンディの本数を考えると、持っている CD の数が多い場合の方が、諦めてもよいキャンディの本数は少なくなると考えることが自然である。すなわち、キャンディに対する CD の相対的な価値の割合 (限界代替率) は、保有する CD の数が増加するにつれて遞減する。

個々の無差別曲線は互いに交わらないことを示す。たとえば、図 3(c) のように、無差別曲線 I_1 と I_2 が点 A で交差すると仮定する。点 A と点 B は無差別曲線 I_2 上にあるため無差別になる。また点 A と点 C は無差別曲線 I_1 上にあるため無差別になる。したがって、点 B と点 C も無差別になる。しかし、点 B の各目的の利得は点 C より大きいので、ユーザは明らかに点 C よりも点 B を選好する。このことは、点 B と点 C が無差別であることに反する。よって、個々の無差別曲線は互いに交わらない。また、各目的の利得が大きいほど効用は高いので、原点から遠い無差別曲線ほど効用水準は高い。

3 対話型解法

本章では、多目的制約最適化問題における対話型解法 (Interactive Algorithm for MO-COP, MO-IA) を提案する。また本解法の計算 / 空間複雑度をそれぞれ示す。本解法は、パレート解が存在しうる領域を、ユーザとのインタラクションにより段階的に狭めていく対話型解法である。まず、線形化加重和法 (AOF) を用いて、複数の重み付き目的関数の最適解を求めることにより、あるパレート解および、その他のパレート解が存在しうる領域を求める。ユーザは、既に求められたパレート解で満足 / 不満足を選択する。ユーザが満足した場合は、本解法を終了する。ユーザが満足しなかった場合は、パレート解が存在しうる領域中で、最も望ましい点を選択する。次に、本解法で定義する距離を用いて、ユーザが選択した望ましい点に最も近いパレート解を求め、その他のパレート解が存在しうる領域を更新し、新しい情報としてユーザに提示する。以上の操作を、ユーザが満足するまで繰り返す。

3.1 解法の概要

本解法は以下の 3 つの段階から構成される。

- 段階 1 : 各目的関数の最適解をそれぞれ独立に求める。
- 段階 2 : すべての重みが等しい単一の重み付き目的関数の最適解を求める。
- 段階 3 : ユーザが選択した、パレート解が存在しうる領域中の点に最も近いパレート解を求める。

段階 1 では、AOF を用いて、 m 個の重み付き目的関数の最適解を独立に求める。具体的には、目的関数 (o^1, \dots, o^m) に対して、 m 個の重み $(1, 0, \dots, 0), (0, 1, 0, \dots, 0), \dots, (0, \dots, 0, 1)$ をそれぞれ与え、 m 個の重み付き目的関数 o^1, \dots, o^m (m 個の制約最適化問題) の最適解をそれぞれ独立に求める。このとき、得られる m 個の最適値を $R_{max}^1, \dots, R_{max}^m$ と記述する。

段階 2 では、AOF を用いて、すべての重みが等しい単一の重み付き目的関数を作り、その最適解を求める。具体的には、目的関数 (o^1, \dots, o^m) に対して、重み $\alpha_1 = \frac{1}{m}, \dots, \alpha_m = \frac{1}{m}$ をそれぞれ与え、単一の重み付き目的関数 $\pi: \frac{1}{m}o^1 + \dots + \frac{1}{m}o^m$ を作り、その最適解を求める。 A^* および $R(A^*)$ を重み付き目的関数 π の最適解および A^* によって得られる元の問題の利得ベクトルとする。定理 1 より、 A^* は元の問題のパレート解である。以降では、このパレート解を候補解と呼ぶ。各目的関数の最適解および候補解に関して以下が成立する。

定理 2. 多目的制約最適化問題に関して、割当 A を候補解 A^* 以外のパレート解とする。このとき、 $R(A^*)$ および $R(A)$ に関して、以下が成立する。

- (1) $\sum_{l=1}^m R^l(A) \leq \sum_{l=1}^m R^l(A^*)$.
- (2) $\exists l: R^l(A^*) < R^l(A) \leq R_{max}^l$.

証明. A^* は単一の重み付き目的関数 π の最適解である。明らかに、 π に関して、

$$\frac{1}{m}R^1(A) + \dots + \frac{1}{m}R^m(A) \leq \frac{1}{m}R^1(A^*) + \dots + \frac{1}{m}R^m(A^*)$$

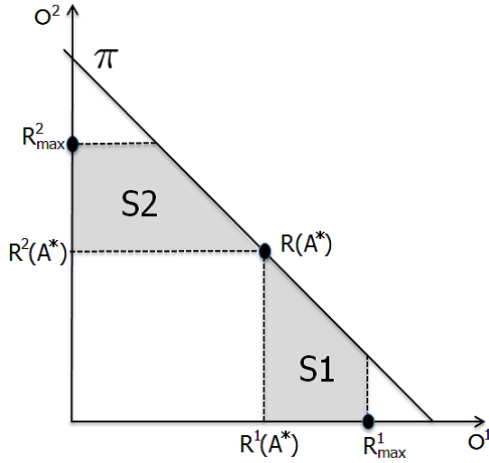


図 4 2 目的制約最適化問題のパレート解によって得られる利得ベクトルが存在しうる領域。

が成立し, (1) が成立する. このことは, π 上の利得ベクトルを支配するような割当 A が存在しないことを意味する.

R_{max}^l は目的関数 o^l を独立に求めたときの最適値である. したがって, 明らかに $R^l(A) \leq R_{max}^l$ が成立する. $\exists l : R^l(A^*) < R^l(A)$ が成立することを証明する. $\forall l : R^l(A^*) \geq R^l(A)$ が成立すると仮定する. $R(A)$ は $R(A^*)$ 以外の利得ベクトルであるため, 少なくとも 1 つの目的 l に関して, $\exists l : R^l(A^*) > R^l(A)$ が成立する. このとき, 定義 1 より, $R(A) \prec R(A^*)$ が成立する. すなわち, $R(A^*)$ が $R(A)$ を支配する. これは, 定義 2 より, A がパレート解であることに反する. よって, $\exists l : R^l(A^*) < R^l(A)$ が成立する. \square

例 3. 図 4 は 2 目的制約最適化問題のパレート解によって得られる利得ベクトルが存在しうる領域を表す. X 軸は目的 1 における利得を表し, Y 軸は目的 2 における利得を表す. 図中 π は, 段階 2 における単一の重み付き目的関数を表し, $R(A^*)$ は候補解 A^* によって得られる利得ベクトルを表す. 割当 A を A^* 以外の任意のパレート解とし, $R(A)$ を $R(A^*)$ 以外の利得ベクトルとする. 定理 2 (1) より, 利得ベクトル $R(A)$ は直線 π より下の領域に存在する. 定理 2 (2) より, $R^1(A^*) < R^1(A)$ または $R^2(A^*) < R^2(A)$ が成立する. また, 明らかに, $R^1(A) \leq R_{max}^1$ および $R^2(A) \leq R_{max}^2$ が成立するため, $R(A)$ が存在しうる領域は $S1$ または $S2$ となる.

ユーザは段階 2 で求めた候補解で満足 / 不満足を選択する. ユーザが候補解で満足した場合は, 本解法を終了する. ユーザが候補解で満足しなかった場合は, パレート解が存在しうる領域中で, 最も望ましい点を選択する. 以降では, この点をユーザの選好点と呼び, $R_u (= (R_u^1, \dots, R_u^m))$ と記述する.

段階 3 では, 以下で定義する距離を用いて, ユーザの選好点からの距離が最小となるような割当を求める. $R(A)$ を割当 A によって得られる利得ベクトルとして, $R(A)$ と R_u の距離を以下で定義する. R_u からの距離が同値の場合の処理として,

Algorithm 1 MO-IA (Phase 3)

MO-IA(X, D, O)

- 1 Given : R_u // user's preference point on π
- 2 $JOIN_1 = \text{null}, \dots, JOIN_n = \text{null}$
- 3 for each $i = n, \dots, 1$ // bottom-up processing
- 4 if i is a leaf then
- 5 $JOIN_i = R_i^{P_i} \oplus (\bigoplus_{h \in PP_i} R_i^h)$ // join all reward tables
- 6 Compute $\underset{a}{\text{argmin}} \text{dis}(R(a), R_u)$ for each a in combination of assignments of p_i and PP_i
- 7 $JOIN_i = JOIN_i \perp_{x_i}$ // use projection to eliminate x_i
- 8 else
- 9 $JOIN_i = R_i^{P_i} \oplus (\bigoplus_{h \in PP_i} R_i^h) \oplus (\bigoplus_{j \in C_i} JOIN_j)$
- 10 Compute $\underset{a}{\text{argmin}} \text{dis}(R(a), R_u)$
- 11 $JOIN_i = JOIN_i \perp_{x_i}$
- 12 end if
- 13 end for

パラメータ ϵ を導入する. ϵ は 1 より小さい正の実数とする.

$$\text{dis}(R(A), R_u) = f(R^1(A), R_u^1) + \dots + f(R^m(A), R_u^m),$$

$$\forall l : f(R^l(A), R_u^l) = \begin{cases} R_u^l - R^l(A) & (R_u^l \geq R^l(A)) \\ -\epsilon_l(R^l(A) - R_u^l) & (R_u^l < R^l(A)) \end{cases}$$

Algorithm 1 に段階 3 の擬似コードを示す. 本解法は擬似木を用いて解探索を行う. $JOIN_i$ は, ノード x_i が管理する利得表を表す. $R_i^{P_i}$ および R_i^h は, x_i と親ノード P_i および x_i と祖先ノード $h \in PP_i$ 間の利得表を表す. またオペレーション \oplus では, 2 つの利得表が 1 つに統合 (マージ) される. 本解法は, 擬似木内の葉ノードからルートノードに向け木辺に従ってボトムアップに処理を行う. ノード x_i が葉ノードの場合は, $R_i^{P_i}$ および R_i^h をマージする (line 5). 親および祖先ノードが取る値の各組合せに対して, ユーザの選好点 R_u からの距離が最小となるような割当 a を, それぞれ求める (line 6). これらの割当から自身の値を取り除いた利得表を $JOIN_i$ に保存する (line 7). ノード x_i が葉ノードでない場合は, x_i の子ノード $j \in C_i$ が管理する利得表 $JOIN_{j \in C_i}$ にアクセスし, $R_i^{P_i}, R_i^h$ および $JOIN_{j \in C_i}$ をマージする (line 9). 以下, ノード x_i が葉ノードのときと同様の処理を行う (line 10,11). 本解法では, 各変数 x_i はユーザの選好点 R_u からの距離が最小となるような値を選択する. したがって, すべての変数への割当 A に関して, 利得ベクトル $R(A)$ とユーザの選好点 R_u の距離は最小となる. 紙面の都合上, 証明は割愛する. 本解法 (段階 3) によって得られる解に関して以下が成立する.

定理 3. 多目的制約最適化問題に関して, 本解法 (段階 3) によって得られる割当はパレート解である.

証明. A^* を本解法によって得られる割当とする. A^* がパレート解であることを示すには, 定義 2 より, $R(A^*) \prec R(A)$ が成

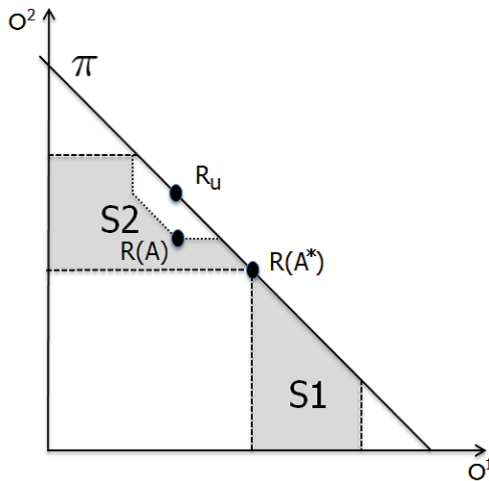


図 5 S1 および S2 は A および A* 以外のパレート解によって得られる利得ベクトルが存在しうる新しい領域を表す。元の領域 (図 4) と比べ、S2 が狭くなっている。

立するような割当 A が存在しないことを示せばよい。以下、背理法を用いてこのことを証明する。 $\exists A : R(A^*) \prec R(A)$ が成立すると仮定する。定義 1 より、(i) $\forall l : R^l(A^*) \leq R^l(A)$ かつ、(ii) $\exists l' : R^{l'}(A^*) < R^{l'}(A)$ が成立する。 R_u をユーザの選好点とする。(i) より、すべての目的 l に関して、 $R_u^l \geq R^l(A)$ の場合は、

$$f(R^l(A), R_u^l) = R_u^l - R^l(A) \leq R_u^l - R^l(A^*) = f(R^l(A^*), R_u^l)$$

が成立する。 $R_u^l < R^l(A)$ の場合は、

$$\begin{aligned} f(R^l(A), R_u^l) &= -\epsilon_l(R^l(A) - R_u^l) \\ &\leq -\epsilon_l(R^l(A^*) - R_u^l) = f(R^l(A^*), R_u^l) \end{aligned}$$

が成立する。また、(ii) より、少なくとも 1 つの目的関数 l' に対して、 $R_u^{l'} \geq R^{l'}(A)$ の場合は、

$$f(R^{l'}(A), R_u^{l'}) < f(R^{l'}(A^*), R_u^{l'})$$

が成立し、 $R_u^{l'} < R^{l'}(A)$ の場合は

$$f(R^{l'}(A), R_u^{l'}) < f(R^{l'}(A^*), R_u^{l'})$$

が成立する。以上より、 $dis(R(A), R_u) < dis(R(A^*), R_u)$ が成立し、 $dis(R(A^*), R_u)$ が最小であることに反する。したがって、 $R(A^*) \prec R(A)$ が成立するような割当 A は存在しない。すなわち、定義 2 より、割当 A* はパレート解である。□

段階 3 で得られる割当 A は、ユーザの選好点 R_u に最も近いパレート解である。このことは、ユーザの選好点から距離 $dis(R(A), R_u)$ の範囲にパレート解が存在しないことを意味する。段階 3 によって更新されるパレート解が存在しうる領域は、元の領域からこの範囲を取り除いた残りの領域となる。

例 4. 図 5 は本解法 (段階 3) によって得られるパレート解および、その他のパレート解によって得られる利得ベクトルが存在しうる領域を表す。 $R(A)$ は段階 3 によって得られる利得ベクトルを表す。 A および A* 以外のパレート解によって得られる利得ベクトルが存在しうる領域は、図 4 からユーザの選好点 R_u から距離 $dis(R(A), R_u)$ の範囲 (直線 π と S2 で囲まれた領域) を取り除いた残りの領域となる。

段階 3 によって得られるパレート解を新しい候補解として、ユーザは本解法で得られたパレート解 (既に提示された候補解、もしくは新しい候補解) で満足 / 不満足を選択する。ユーザが満足した場合は、本解法を終了する。ユーザが満足しなかった場合は、パレート解が存在しうる (新しい) 領域中で、ユーザの選好点を決定し、段階 3 の操作を再度行う。以下、ユーザが満足するまで、候補解の集合および、その他のパレート解が存在しうる領域を更新し、ユーザに提示する。

3.2 計算量

本解法 (段階 3) の計算 / 空間複雑度は問題の誘導幅 [3] で抑えられる。誘導幅とは制約最適化問題の解法の複雑度を決定する指標である。計算複雑度に関しては、 k を目的数、 $|D|$ をドメインサイズ、 w^* を問題の誘導幅、 e を制約数として $O(e \times k \times |D|^{w^*+1})$ で与えられる。空間複雑度に関しては、 n を変数の数として $O(n \times k \times |D|^{w^*})$ で与えられる。特に、問題が最も簡単な木構造の場合、計算量は定数のオーダーとなる。

4 評価実験

本章では、ユーザの選好に関する既存のモデルを用いて本解法をシミュレーションで評価する。具体的には、異なる設定の多目的制約最適化問題において、以下に定義する 4 人の (仮想的な) ユーザの選好を満たすまでに必要としたイテレーションの数を調べる。

- ユーザ 1: 線形の効用関数。
- ユーザ 2: パラメータ p が 0.5 のときの CES 型効用関数。
- ユーザ 3: コブ・ダグラス型効用関数。
- ユーザ 4: レオンチェフ型効用関数。

本解法の評価方法について述べる。まず候補解および、その他のパレート解が存在しうる領域中からユーザの選好点 (ユーザの効用関数と単一の重み付き目的関数 π の接点) を決定する。もし、候補解がユーザの選好を既に満たしている場合は、イテレーションの数を 1 とし、本解法を終了する。ユーザの選好を満たしていない場合は、ユーザの選好点に最も近いパレート解を求め、(新しい) 候補解とする。次に、既に求めた候補解 (はじめの候補解および新しい候補解) 以外のパレート解が存在しうる領域中から、ユーザの新しい選好点を無差別曲線を用いて決定する。既に求めた候補解の内の 1 つがユーザの選好を満たしている場合は、本解法を終了する。以上をユーザの選好を満たすまで繰り返し、終了時まで必要としたイテレーションの数を調べる。

表 1 2 目的制約最適化問題における実験結果 .

ノード数	ユーザ 1	ユーザ 2	ユーザ 3	ユーザ 4
10	2.5	1.6	2.1	25.3
20	2.5	2.0	1.9	17.4
30	2.6	2.2	1.8	12.8
40	2.8	2.3	1.6	13.4
50	2.8	2.5	1.7	12.9
60	2.9	2.3	1.5	11.1
70	2.9	2.3	1.5	11.9
80	2.9	2.5	1.4	11.5
90	2.8	2.5	1.2	12.5
100	2.8	2.6	1.3	10.4

実験における終了条件について述べる．多目的制約最適化問題に関して $R_u = (R_u^1, \dots, R_u^m)$ をユーザの選好点 $R(A)$ を候補解の 1 つ A によって得られる利得ベクトルとして，

$$u(R_u^1, \dots, R_u^m) \leq u(R^1(A), \dots, R^m(A))$$

が成立するとき，すなわち，候補解における効用が，ユーザの選好点における効用以上であるとき，ユーザの選好は満たされたとする．したがって，ユーザ 1, 2 および 3 の終了条件は，ある割当 A が存在し，以下が成立するときとなる，

$$\text{ユーザ 1 の終了条件 } \sum_{l=1}^m \alpha_l R_u^l \leq \sum_{l=1}^m \alpha_l R^l(A)$$

$$\text{ユーザ 2 の終了条件 } \sum_{l=1}^m \alpha_l \sqrt{R_u^l} \leq \sum_{l=1}^m \alpha_l \sqrt{R^l(A)}$$

$$\text{ユーザ 3 の終了条件 } \prod_{l=1}^m (R_u^l)^{\alpha_l} \leq \prod_{l=1}^m (R^l(A))^{\alpha_l}$$

またユーザ 4 に関しては，ある候補解によって得られる利得ベクトルが，すべての目的においてユーザの選好解を改善しているとき，ユーザ 4 の選好は満たされたとする．

$$\text{ユーザ 4 の終了条件 } \forall l : R_u^l \leq R^l(A)$$

実験における各変数のドメインサイズは 2 とし，各目的における各制約 (二項制約) の利得は 0 から 10 の整数値を一様分布の乱数により選択した．問題のインスタンスはランダムに生成した．CES 型効用関数のパラメータ $\alpha = (\alpha_1, \dots, \alpha_m)$ は，問題のインスタンスごとにランダムに決定した．実験結果は 50 インスタンスの平均値を表す．また，本解法 (段階 3) で定義した距離におけるパラメータ ϵ の値は 0.001 とした．

本解法は，ユーザ 1, 2 および 3 に対して有効な解法であることを示す．表 1 に 2 目的制約最適化問題における，異なるノード数での実験結果を示す．表 1 より，ノード数 10 におけるイテレーションの数は，ユーザ 1, 2 および 3 に対して，それぞれ 2.5, 1.6 および 2.1 だった．この結果はノード数が増えてもほとんど変わらなかった．実際，ノード数 100 におけるイテレーションの数は，ユーザ 1, 2 および 3 に対して，2.8, 2.6 および 1.3 だった．このことは，本解法が定義する“距離が近い”解と，ユーザ 1, 2 および 3 が考える“距離が近い”解とが一致しているためだと考える．一方，ユーザ 4 に対

表 2 3 目的制約最適化問題における実験結果 .

ノード数	ユーザ 1	ユーザ 2	ユーザ 3	ユーザ 4
10	2.4	2.3	2.3	44.4
20	2.2	2.1	2.4	33.0
30	2.2	2.2	2.3	46.8
40	2.2	2.7	2.2	36.7
50	2.4	2.3	2.0	43.6
60	2.2	2.6	2.1	38.8
70	2.3	2.6	2.0	40.7
80	2.4	2.5	2.0	30.5
90	2.3	2.3	2.0	41.8
100	2.3	2.4	2.0	42.9

しては，その他のユーザと比べ，多くのイテレーションを必要とした．表 1 より，イテレーションの数は，ノード数 10 では 25.3，ノード数 100 では 10.4 だった．このことは，本解法が定義する“距離が近い”解と，ユーザが考える“距離が近い”解との間に大きなずれがあるためだと考える．またノード数が増えるにつれイテレーションの数は減少した．このことは，ノード数が増えるにつれ，解空間が密になるためだと考える．

目的数を 3 に増やしても本解法の有効性は変わらなかった．表 2 に 3 目的制約最適化問題における，異なるノード数での実験結果を示す．表 2 より，ノード数 10 におけるイテレーションの数は，ユーザ 1, 2 および 3 に対して，2.4, 2.3 および 2.3 だった．この結果は，ノード数が増えても変わらなかった．ノード数 100 ではイテレーションの数は，ユーザ 1, 2 および 3 に対して，2.3, 2.4 および 2.0 だった．ユーザ 4 に対しては，目的数が 2 のときと比べ，さらに多くのイテレーションを必要とした．実際，ノード数 10 では 44.4，ノード数 100 では 42.9 だった．このことは，3 目的制約最適化問題の解空間が，2 目的のときと比べ，疎であるためだと考える．また表 1 では，ユーザ 4 に対して，ノード数が増えるにつれ，イテレーションの数は減少したが，表 2 では，ノード数とイテレーションの数の間に特徴的な関連性はみられなかった．このことは，2 目的制約最適化問題では，(2 次元の) 解空間がノード数 100 で密になるのに対して，3 目的制約最適化問題では，(3 次元の) 解空間がノード数 100 では疎であるためだと考える．

以上の結果から，本解法は，ユーザ 1, 2 および 3，すなわち，線形やコブ・ダグラスなど，パラメータ p が 0 から 1 の間の CES 型効用関数をもつユーザに対して，有効な解法であることがわかった．この結果は変数の数を増やしても変わらなかった．しかし，レオンチェフ型効用関数をもつユーザ 4 に対しては，他のユーザと比べ，多くのイテレーションを必要とした．実験における解空間内のパレートフロントの数やノード数および目的数との関連性等の詳細な解析は今後の課題とする．また本論文では，CES 型効用関数を用いたが，より複雑な効用関数をもつユーザにおける評価は今後の課題とする．

5 考察

レオンチェフ型効用関数におけるイテレーションの数を減らす方法を示す。実験より、レオンチェフ型効用関数をもつユーザでは、他のユーザと比べ、多くのイテレーションを必要とした。そこで、イテレーションの数を減らす方法として以下を提案する。まず、ユーザの選好点より、効用関数の係数 α を推定する。すなわち、推定した α を用いた仮想のユーザを考える。次に、イテレーションがある一定の回数で終了しなければ、推定した α を用いて、レオンチェフだと仮定して、推定したレオンチェフ型効用関数でのイテレーションが収束するまで (実際にはユーザには聞かずに、あくまで推定した関数を用いて) 繰り返しパレートフロントを求める。レオンチェフ型効用関数におけるイテレーションの数を再度調べた。実験では、イテレーションが 3 回で終了しなかった場合に提案手法を用いたところ、4 回目にはユーザの選好を満足した。

本解法の拡張として、複数のパレート解を求めることにより、パレート解が存在しうる領域をさらに狭くし、ユーザに提示する解法を提案する。実世界におけるインタラクションを考えた場合、ユーザの選好に対し、いくつかの候補を提示することは自然である。また、候補解が存在しうる範囲の情報はより詳細な方が望ましい。そこで、本解法の拡張として以下を提案する。具体的には、本解法の段階 3 で以下を行う。

段階 3' : あらかじめユーザの選好点以外に、いくつかの仮想的な選好点を決定し、各選好点に最も近いパレート解をそれぞれ求める。

従来の解法では、ある 1 つの候補解および、その他のパレート解が存在しうる領域を求め、ユーザに提示していた。拡張された解法では、いくつかの候補解および、各選好点と候補解の距離の範囲を元の領域から取り除くことで、より狭められた領域をユーザに提示可能である。仮想的な選好点は、たとえば、図 5 の S_2 から削られた領域と S_2 の境界上の点の内、直線 π と交わる 2 点を選ぶ。

6 結言

本論文では、多目的制約最適化問題においてパレート解が存在しうる領域を、ユーザとのインタラクションにより段階的に狭めていく対話型解法を提案した。また本解法の計算/空間複雑度を示した。実験では、ユーザの選好に関する既存のモデルを用いて本解法を評価し、本解法が線形やコブ・ダグラス型効用関数をもつユーザに対して有効な解法であることを示した。さらに、レオンチェフ型効用関数のイテレーションの数を減らす方法を示した。また本解法の拡張として、いくつかのパレート解を求めることにより、パレート解が存在しうる領域をさらに狭くし、ユーザに提示する解法を提案した。

今後の課題として、本解法は対話型解法であるため、実問題を対象に実験を行う必要がある。また、本論文とは異なる距離を定義し実験を行う。さらに本解法を分散型へ拡張する。

参考文献

- [1] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner. Approximation-guided evolutionary multi-objective optimization. In *IJCAI*, pages 1198–1203, 2011.
- [2] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.
- [3] R. Dechter. *Constraint Processing*. Morgan Kaufmann Publishers, 2003.
- [4] T. Erfani and V. Utyuzhnikov, Sergei. Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization. *Engineering Optimization*, 43(5):467–484, 2010.
- [5] U. Junker. Preference-based inconsistency proving: When the failure of the best is sufficient. In *ECAI*, pages 118–122, 2006.
- [6] R. Marinescu. Exploiting problem decomposition in multi-objective constraint optimization. In *CP*, pages 592–607, 2009.
- [7] R. Marinescu. Best-first vs. depth-first and/or search for multi-objective constraint optimization. In *ICTAI*, pages 439–446, 2010.
- [8] A. Messac and C. Mattson. Generating well-distributed sets of pareto points for engineering design using physical programming. *Optimization and Engineering*, 3(4):431–450, 2002.
- [9] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [10] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *FOCS*, pages 86–92, 2000.
- [11] P. Perny and O. Spanjaard. Near admissible algorithms for multiobjective search. In *ECAI*, pages 490–494, 2008.
- [12] E. Rollon and J. Larrosa. Bucket elimination for multiobjective optimization problems. *Journal of Heuristics*, 12(4-5):307–328, 2006.
- [13] E. Rollon and J. Larrosa. Multi-objective russian doll search. In *AAAI*, pages 249–254, 2007.
- [14] T. Schiex, H. Fargier, and G. Verfaillie. Valued constraint satisfaction problems: Hard and easy problems. In *IJCAI*, pages 631–639, 1995.
- [15] ジョセフ・E・スティグリッツ. スティグリッツ ミクロ経済学. 東洋経済新報社, 1998.
- [16] 奥野正寛, 鈴村興太郎. ミクロ経済学 1. 岩波書店, 1998.