

線形最大リグレット最小化問題の解法について

大阪大学大学院工学研究科	乾口 雅弘 (Masahiro Inuiguchi)
大阪大学大学院工学研究科	谷野 哲三 (Tetsuzo Tanino)
大阪大学大学院工学研究科	東谷 英貴 (Hidetaka Higashitani)

概要

本研究では、目的関数の係数が凸多面体で制限された線形計画問題の最大リグレット最小解の計算方法について考察する。従来に提案されているいくつかのアプローチを述べるとともに、新たに外部近似法に基づくアプローチを示す。いずれのアプローチが効率的かを調べるため、計算機実験により計算時間が比較される。その結果、凸多面体が矩形の場合は、問題の変形によるアプローチが極めて早いこと、矩形でない場合は、外部近似法に基づくアプローチが有効であることが示される。

Key Words: 凸多面体パラメータ, 最大リグレット最小化, 線形計画法, 緩和法, 外部近似法, 分枝限定法

1 はじめに

目的関数の係数が不明確でその取りうる範囲が凸多面体で表現された線形計画問題、

$$\begin{aligned} & \text{maximize } \gamma^T x \\ & \text{sub. to } x \in X = \{x \mid Ax \leq b, x \geq 0\} \end{aligned} \quad (1)$$

を考える。ただし、 $x = (x_1, x_2, \dots, x_n)^T$ が決定変数で、 A は $m \times n$ 行列、 $b = (b_1, b_2, \dots, b_m)^T$ である。また、 X は有界であり、 γ の取りうる範囲 Γ は、次式で定められる。

$$\Gamma = \{c = (c_1, \dots, c_n)^T \mid Dc \leq g\} \quad (2)$$

D は $p \times n$ 行列であり、 $g = (g_1, g_2, \dots, g_p)^T$ である。 Γ は有界であると仮定する。特に、 Γ が

$$\Gamma = \{c = (c_1, \dots, c_n)^T \mid c_i^L \leq c_i \leq c_i^R, i = 1, \dots, n\} \quad (3)$$

と表される場合には、問題 (1) は区間線形計画問題となり、比較的良好に研究されている。この区間線形計画問題は、従来、区間目的関数の下限値と上限値とを同時に最大化する 2 目的の問題と解釈され、そのパレート最適解が合理的な解と考えられていた [1, 2, 3]。著者ら [4, 5] は、この 2 目的線形計画問題に完全最適解が存在しても最も合理的な解であるとは必ずしもいえないことを示し、最も合理的な解として必然的最適解を、最低限の合理性を満たす解として可能的最適解を定義した。また、必然的最適解が存在する場合にはそれに一致し、存在しない場合には必然的最適性からの乖離をできる限り小さくする可能的最適解として、最大リグレット最小解を提案している。必然的最適解、可能的最適解、最大リグレット最小解の概念は、 Γ が式 (2) の凸多面体の場合にも導入されている [6]。一方、各証券の収益率がファジィ数で与えられたポートフォリオ選択問題に従来のファジィ線形計画法

を適用しても、安全な分散投資解が得られないことを示すとともに、最大リグレット最小化を導入することにより、分散投資解が得られることを示している [7]。このように、最大リグレット最小解は、ロバスト性の面のみでなく、リスク回避の面でも有用である。

問題 (1) に対する最大リグレット最小化問題は、

$$\underset{x \in X}{\text{minimize}} \quad \max_{c \in \Gamma, y \in X} (c^T y - c^T x) \quad (4)$$

と定式化される。問題 (4) の解法として、いくつかの方法が提案されている。いずれも次の緩和法に基づいたアルゴリズムであるが、Step 3 のリグレット最大化問題の解法が異なる。ただし、 $\varepsilon > 0$ は許容誤差である。

アルゴリズム 1

Step 1. $c^0 \in \Gamma$ について、 $\max_{x \in X} c^{0T} x$ の最適解 $z^0 \in X$ を求める。

Step 2. $r^0 = 0, k = 1, x^0 = z^0$ と設定する。

Step 3. x^0 に対するリグレット最大化問題、

$$\underset{c \in \Gamma, y \in X}{\text{maximize}} (c^T y - c^T x^0) \quad (5)$$

を解き、最適解を (c^k, z^k) とし、最適値を r^k とする。

Step 4. $r^k \leq r^0 + \varepsilon$ ならば、終了する。このとき、最大リグレット最小解は x^0 である。

Step 5. 緩和問題、

$$\begin{aligned} &\underset{r}{\text{minimize}} \quad r \\ &\text{sub. to} \quad Ax \leq b \\ &\quad \quad c^j T z^j - c^j T x \leq r, \quad j = 0, 1, \dots, k \end{aligned} \quad (6)$$

を解き、 (x^0, r^0) をその最適解により更新する。 $k = k + 1$ として、Step 3 に戻る。

Step 3 のリグレット最大化問題 (5) は、非凹計画問題となり、従来、種々の解法が提案されている。乾口ら [4, 6] は、問題 (1) の可能的最適端点集合を予め求めることにより、リグレット最大化問題 (5) を解く 2 段階アプローチを提案している。また、Inuiguchi と Sakawa [6, 8] は、リグレット最大化問題 (5) が 2 レベル計画問題 [9] であることから、下位問題を最適性条件で置き換える Kuhn-Tucker アプローチ [9] を適用することにより、分枝限定法に基づいて解けることを示している。また、区間線形計画問題の場合にリグレット最大化問題 (5) の部分問題の解が容易にわかることを利用して、Mausser と Laguna [10] は、リグレット最大化問題を混合 0-1 計画問題に変形して、分枝限定法を用いて解く方法を提案するとともに、数値実験により、このアプローチが 2 レベル計画問題によるアプローチよりも効率的であることを示している。

本研究では、リグレット最大化問題 (5) が凸最大化問題であることを鑑み、外部近似法に基づいた最大リグレット最小解の計算方法を考察する。凸最大化問題に対して種々の方法が提案されている [11] が、制約集合が凸多面体となることやアルゴリズム 1 の Step 3 を通過するたびにリグレット最大化問題を解かなければならないことから、外部近似法の適用を考える。外部近似法を適用すれば、一度、リグレット最大化問題が解かれると、2 回目以

降のリグレット最大化問題を最初から解く必要はなく、先に解いた問題の最後の状態から継続して解くことができる。

いずれのアプローチが効率的か調べるため、各手法に基づいたプログラムを C 言語により作成し、計算機実験により各手法の効率性を比較検討する。これにより、従来十分に検討されていなかった凸多面体の場合の解法の効率性が議論できる。

2 従来のリグレット最大化法

2.1 2 段階アプローチ

制約集合 X の可能的最適解集合を $\Pi B(X)$ とすると、

$$\Pi B(X) = \left\{ \mathbf{y} \mid \exists \mathbf{c} \in \Gamma, \mathbf{c}^T \mathbf{y} = \max_{\mathbf{z} \in X} \mathbf{c}^T \mathbf{z}, \mathbf{y} \text{ は } X \text{ の端点} \right\} \quad (7)$$

となる。 $\Pi B(X)$ を用いると、リグレット最大化問題 (5) は、

$$\underset{\mathbf{c} \in \Gamma, \mathbf{y} \in \Pi B(X)}{\text{maximize}} \quad (\mathbf{c}^T \mathbf{y} - \mathbf{c}^T \mathbf{x}^0) \quad (8)$$

と書ける。 $\mathbf{y} \in X$ が固定されたとき最大リグレットを出力する関数を f とすると、 $f: X \rightarrow \mathbf{R}$ は次のように定められる。

$$f(\mathbf{y}) = \max_{\mathbf{c} \in \Gamma} (\mathbf{c}^T \mathbf{y} - \mathbf{c}^T \mathbf{x}^0) \quad (9)$$

X の有界性より、 $\Pi B(X)$ は有限集合となる。 $\Pi B(X)$ の各要素 $\mathbf{y}^j$ について、 $f(\mathbf{y}^j)$ を線形計画法により算出し、その最大値を与える要素 $\mathbf{y}^j$ とそれに対応する問題 (9) の解 $\mathbf{c}^j$ を求めると、 $\mathbf{y}^j, \mathbf{c}^j$ がリグレット最大化問題 (5) の最適解となる。

このことより、予め $\Pi B(X)$ を求めることにより、リグレット最大化問題を解く方法が考えられ、2 段階アプローチと呼ばれている [6]。特に、 Γ が式 (3) で表される区間線形計画問題の場合には、Steuer の方法 [12] により、 $\Pi B(X)$ が列挙できるとともに、 $f(\mathbf{y}^j) = \mathbf{c}^T \mathbf{y}^j$ 、 $\mathbf{c} \in \Gamma$ なる $\mathbf{c} = (c_1, c_2, \dots, c_n)^T$ が次式で求められ、 $f(\mathbf{y}^j)$ が容易に評価できるので、2 段階アプローチは当初から提案されている [4]。

$$c_i = \begin{cases} c_i^L; & y_i^j \leq x_i^0 \text{ のとき} \\ c_i^R; & y_i^j > x_i^0 \text{ のとき} \end{cases} \quad (10)$$

ただし、 $\mathbf{y}^j = (y_1^j, y_2^j, \dots, y_n^j)^T$ 、 $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ である。

Steuer の方法 [12] は、 Γ が式 (3) で表される場合に、ある可能的最適端点から隣接する端点の可能的最適性を調べることにより、 $\Pi B(X)$ の要素を列挙する方法であるが、 Γ が一般の凸多面体の場合にも拡張することができる。すなわち、 k 番目の非基底変数を基底に入れてできる端点の可能的最適性テストは、次の線形計画問題の最適値が 0 となることを確認すればよい [13]。

$$\begin{aligned} & \text{minimize} \quad \mathbf{g}^T \mathbf{v} \\ & \text{subject to} \quad \mathbf{Y} \mathbf{u} - \mathbf{y}_k \mathbf{w} - \mathbf{D}^T \mathbf{v} = \mathbf{0} \\ & \quad \mathbf{u} \geq \mathbf{0}, \mathbf{w} \geq 0, \mathbf{v} \geq \mathbf{0} \end{aligned} \quad (11)$$

ただし、問題 (1) の制約条件にスラック変数を導入した標準形の制約条件に対する現在の可能的最適端点の基底行列を B 、非基底行列を A^N とすると、 $\mathbf{Y} = ((B^{-1}A^N)^T, -I)^T$ と定められる。また、 $\mathbf{y}_k$ は $\mathbf{Y}$ の第 k 列である。

2.2 2 レベル計画問題によるアプローチ

大規模な問題においては、 $\Pi B(X)$ の要素の数が膨大なものとなり、メモリの制限から、すべて列挙することが困難となる可能性があることから、 $\Pi B(X)$ の列挙を伴わない最大リグレット問題の解法として、2 レベル計画問題による方法が考えられている [6, 8].

問題 (5) は次の 2 レベル計画問題に書き換えられる.

$$\begin{aligned} & \underset{c, y}{\text{maximize}} && (c^T y - c^T x^0) \\ & \text{sub. to} && c \in \Gamma, Ay \leq b, y \geq 0 \\ & && c^T y = \underset{z}{\text{maximum}} c^T z \\ & && \text{sub. to } Az \leq b, z \geq 0 \end{aligned} \quad (12)$$

下位問題を最適性条件で置き換え、冗長な制約条件、 $u^T b \leq c^R y$ を加えると、問題 (11) は、

$$\begin{aligned} & \underset{y, u}{\text{maximize}} && u^T x^0 \\ & \text{sub. to} && DA^T u - Du \leq g, u^T b \leq c^R y, Ay \leq b, y \geq 0, u \geq 0, u^T y = 0 \end{aligned} \quad (13)$$

となる. ただし、 $c^R = (c_1^R, c_2^R, \dots, c_n^R)^T$ で、各成分は

$$c_j^R = \max_{c \in \Gamma} c_j, \quad j = 1, 2, \dots, n \quad (14)$$

と定められる. 問題 (13) は、最後の制約条件 $u^T y = 0$ を除けば、線形計画問題となるので、この制約条件を除いた緩和問題を解き、 $u_i = 0$ あるいは $y_i = 0$ を加えていく分枝限定法により解くことができる. 各緩和問題の有界性は、追加した冗長な制約条件 $u^T b \leq c^R y$ により保証される [6, 8].

2.3 問題の変形によるアプローチ

Γ が式 (3) で表される区間線形計画問題の場合には、式 (9) の右辺の問題が式 (10) により容易に解けることから、最大リグレット最小化問題 (5) を、差異変数 z^+, z^- を用いて、

$$\begin{aligned} & \text{maximize} && c^R z^+ - c^L z^- \\ & \text{sub. to} && y - z^+ + z^- = x^0, Ay \leq b, y \geq 0, z^+ \geq 0, z^- \geq 0, z^{+T} z^- = 0 \end{aligned} \quad (15)$$

と書くことができる [10]. 2 レベル計画問題によるアプローチと同様、問題 (15) の最後の制約条件 $z^{+T} z^- = 0$ を除けば、線形計画問題になることより、この条件を除いた緩和問題を解き、 $z_i^+ = 0$ あるいは $z_i^- = 0$ を加えていく分枝限定法による解法が考えられる. ただし、緩和問題の有界性を保証するため、十分大きい正数 $M_j, j = 1, 2, \dots, n$ を用いた冗長な制約条件、

$$z_j^- \leq x_j^0, z_j^+ \leq (M_j - x_j^0), \quad j = 1, 2, \dots, n \quad (16)$$

を加えることが望ましい. Mausser と Laguna [10] は、式 (16) の代わりに、十分大きい正数 M と 0-1 変数 $w_j, j = 1, 2, \dots$ を用いて表される制約条件、

$$z_j^- \leq x_j^0 w_j, z_j^+ \leq (M - x_j^0)(1 - w_j), \quad w_j \in \{0, 1\}, \quad j = 1, 2, \dots, n \quad (17)$$

を追加して、混合 0-1 計画問題として解くことを提案し、2 レベル計画問題によるアプローチより高速に最大リグレット問題が解けることを示している.

3 外部近似法によるアプローチ

3.1 凸最大化問題への変形

式 (9) で定義される関数 $f: X \rightarrow \mathbf{R}$ は凸関数となることが次のように証明できる.

命題 1 式 (9) の関数 $f: X \rightarrow \mathbf{R}$ は凸関数である.

(証明) 任意の $y^i \in X, i = 1, 2, \lambda \in [0, 1]$ について,

$$\begin{aligned} f(\lambda y^1 + (1 - \lambda)y^2) &= \max_{c \in \Gamma} (c^T y(\lambda) - c^T x^0) \\ &= \max_{c \in \Gamma} \{ \lambda (c^T y^1 - c^T x^0) + (1 - \lambda) (c^T y^2 - c^T x^0) \} \\ &\leq \lambda \max_{c \in \Gamma} (c^T y^1 - c^T x^0) + (1 - \lambda) \max_{c \in \Gamma} (c^T y^2 - c^T x^0) \\ &\leq \lambda f(y^1) + (1 - \lambda) f(y^2) \end{aligned}$$

が成立する. また, 定義域 X は凸集合であるので, f は凸関数となる. (証明終)

リグレット最大化問題 (5) は,

$$\underset{y \in X}{\text{maximize}} \quad f(y) \quad (18)$$

と書け, 命題 1 より, 凸最大化問題 [11] になる. このことより, 凸最大化問題に対して提案されている種々の解法を適用することができる. ここでは, 制約集合 X が凸多面体であることから, 外部近似法 [11] に基づいた解法を考える.

3.2 外部近似法に基づいた解法アルゴリズム

問題 (8) に示すように, 問題 (18) の最適解は $\Pi B(X)$ 内にある. このこととより, 次のような外部近似法に基づいた問題 (18) の解法アルゴリズムが考えられる.

アルゴリズム 2

Step 1. $p = 0$ として, $X \subseteq Y_0$ なる集合 Y_0 を定める.

Step 2. $\Pi B(Y_p)$ を求める.

Step 3. すべての $y \in \Pi B(Y_p)$ について $f(y)$ を評価し, 最大の $f(y)$ を与える y を y^p とし, $f(y^p) = c^T y^p$ なる $c \in \Gamma$ を d^p とする.

Step 4. $f(y^p) \leq r^0$ ならば終了する. このとき, 最大リグレット値は r^0 以下となり, $r^k = r^0$ でアルゴリズム 1 に復帰する. また, $y^p \in X$ の場合も終了する. このとき, $c^k = d^p$, $z^k = y^p$, $r^k = f(y^p)$ として, アルゴリズム 1 に復帰する. ただし, r^0, r^k, c^k, z^k はアルゴリズム 1 の r^0, r^k, c^k, z^k である.

Step 5. 線形計画問題

$$\underset{y \in X}{\text{maximize}} \quad d^{pT} y \quad (19)$$

の最適解を w^p を求め, w^p でスラック変数が非基底変数となっている制約条件により定められる集合を Z とする.

Step 6. $Y_{p+1} = Y_p \cap Z, p = p + 1$ と更新して, Step 2 に戻る.

3.3 アルゴリズム 2 における各 Step での操作

3.3.1 Step 1 での Y_0 の定め方

アルゴリズム 2 はアルゴリズム 1 で Step 3 に反復するたびに呼び出される。アルゴリズム 2 で生成される Y_p は常に $X \subseteq Y_p$ を満たすので、2 回目以降のアルゴリズム 2 の呼び出しにおいては、その前に呼び出されたアルゴリズム 2 の最後の Y_p により、 Y_0 を定めればよい。ここでは、最初にアルゴリズム 2 が呼び出された場合の Y_0 の定め方を考察する。

最初にアルゴリズム 2 が呼び出される直前に、アルゴリズム 1 の Step 1 で z^0 が求められているので、この基底解を用いて、 Y_0 を定めることにする。すなわち、 z^0 においてスラック変数が非基底変数となっている n 本の制約条件と非負条件で Y_0 を定める。しかし、これだけでは、 Y_0 の有界性が保証できるとは限らないので、さらに次の制約条件を加える。

$$\mathbf{1}_n^T \mathbf{x} \leq \Sigma = \max_{\mathbf{y} \in X} \mathbf{1}_n^T \mathbf{y} \quad (20)$$

ただし、 $\mathbf{1}_n = (1, 1, \dots, 1)^T$ なる n 次元ベクトルである。右辺の最適値は、アルゴリズム 1 を開始する直前に計算し、アルゴリズム 1 の Step 1 を再最適化計算により行えば、 z^0 の X に関する基底形式より容易に Y_0 に関する基底形式が求められる。すなわち、 z^0 の $X = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ に関する基底逆行列 B^{-1} 、行列 $\mathbf{A}$ およびベクトル $\mathbf{b}$ から、 $Y_0 = \{\mathbf{x} \mid \hat{\mathbf{A}}\mathbf{x} \leq \hat{\mathbf{b}}\}$ に関する基底逆行列 $\hat{B}^{-1}$ 、行列 $\hat{\mathbf{A}}$ およびベクトル $\hat{\mathbf{b}}$ は次のアルゴリズムにより求められる。ただし、 i 番目の制約条件に対応するスラック変数は s_i であると仮定する。

アルゴリズム 3

Step 1 β を X に関する基底変数の集合で初期化するとともに、 $\hat{B}^{-1} = B^{-1}$ 、 $\hat{\mathbf{A}} = \mathbf{A}$ 、 $\hat{\mathbf{b}} = \mathbf{b}$ と初期化する。

Step 2 β 内のスラック変数の中で、最大の添え字を i とする。このとき、この基底変数 s_i は i 番目の制約条件のスラック変数になる。 β 内にスラック変数が存在しなければ、Step 6 へ行く。

Step 3 $\hat{\mathbf{A}}$ の i 番目の行、 $\hat{\mathbf{b}}$ の第 i 成分を削除する。

Step 4 $\hat{B}^{-1}$ の i 番目の列は、単位ベクトルになっているので、その成分が 1 になっている行および i 番目の列を $\hat{B}^{-1}$ から削除する。

Step 5 $\beta = \beta - \{s_i\}$ と更新し、Step 2 へ戻る。

Step 6 式 (20) の制約条件を加えるため、 $\hat{\mathbf{A}}$ 、 $\hat{\mathbf{b}}$ 、 $\hat{B}^{-1}$ を次のように拡張する。

$$\hat{\mathbf{A}} = \begin{pmatrix} \hat{\mathbf{A}} & \mathbf{0}_n \\ \mathbf{1}_n^T & \mathbf{0}_n^T & 1 \end{pmatrix}, \quad \hat{\mathbf{b}} = \begin{pmatrix} \hat{\mathbf{b}} \\ \Sigma \end{pmatrix}, \quad \hat{B}^{-1} = \begin{pmatrix} \hat{B}^{-1} & \mathbf{0} \\ \mathbf{1}_n^T \hat{B}^{-1} & 1 \end{pmatrix}$$

3.3.2 Step 2 の $\Pi B(Y_p)$ の求め方

2 回目以降に呼び出されたアルゴリズム 2 における Y_0 に対する $\Pi B(Y_0)$ については、先に呼び出されたアルゴリズム 2 において既に求められている。ここでは、最初に呼び出されたアルゴリズム 2 における $\Pi B(Y_0)$ の求め方と Step 6 での Y_p の更新に伴った $\Pi B(Y_p)$ の更新方法について述べる。

最初に呼び出されたアルゴリズム 2 における $\Pi B(Y_0)$ は, 2.1 節で述べた Steuer の列挙法 [12] を拡張した方法 (可能的最適性テストに式 (11) を用いる列挙法) により求めることができる. 先に述べたように, アルゴリズム 1 の Step 1 終了後に可能的最適端点 z^0 の Y_0 に関する基底形式が容易に求められるので, この端点を起点として可能的最適端点列挙を行えばよい.

アルゴリズム 2 の Step 6 での Y_p の更新に伴った $\Pi B(Y_p)$ の更新は, 次のアルゴリズムに従えばよい.

アルゴリズム 4

Step 1 $\Pi B(Y_p) = \Pi B(Y_{p-1})$ とする.

Step 2 アルゴリズム 2 の Step 5 で求められた w^{p-1} の Y_p に関する基底形式をアルゴリズム 5 により求め, 新たに加えられた制約条件のスラック変数のいずれかが非基底変数となる範囲で, 可能的最適性テストに式 (11) を用いる列挙法を適用し, 生成された可能的最適端点を $\Pi B(Y_p)$ に追加する.

アルゴリズム 5

Step 1 β を w^{p-1} の X に関する基底変数の集合で初期化し, $\hat{B}^{-1} = B^{-1}$, $\hat{A} = A$, $\hat{b} = b$ と初期化する. また, I_{p-1} を Y_{p-1} に加えられている制約条件の添え字集合とする.

Step 2 β 内のスラック変数の中で, 最大の添え字を i とする. この基底変数 s_i は i 番目の制約条件のスラック変数になる. β 内にスラック変数が存在しなければ, 終了する.

Step 3 $i \notin I_{p-1}$ ならば, $\hat{A}$ の i 番目の行, $\hat{b}$ の第 i 成分を削除する. そうでなければ, Step 5 へ行く.

Step 4 $\hat{B}^{-1}$ の i 番目の列は, 単位ベクトルになっているので, その成分が 1 になっている行および i 番目の列を $\hat{B}^{-1}$ から削除する.

Step 5 $\beta = \beta - \{s_i\}$ と更新して, Step 2 へ戻る.

4 数値実験

4.1 実験概要

以上の解法のいずれが効率的かを調べるため, パーソナルコンピュータ (pentium II 400MHz, 128MB) を用いて数値実験を行った. Free BSD ver.2.2.8 上の C 言語 (GNU project C and C++ Compiler ver 2.7) を用いて, 各解法のプログラムを作成した. Γ が式 (3) で表される矩形の場合と, 式 (2) の一般の場合に分けて, 各解法により最大リグレット最小解が求められるまでの計算時間 (CPU Time, 秒) を比較した. 決定変数の数 (x の次元 n), X の制約条件数 (A の行の数 m), Γ の制約条件数 (D の行の数 p) が異なるいくつかの場合について, 10 個の問題を発生させ, 計算時間の平均, 分散, 最大値, 最小値, 中央値を求めた.

4.2 Γ が矩形の場合

4.2.1 問題の生成方法

制約条件 $Ax \leq b$ を生成するため, 原点を中心とする超楕円体を用意し, 非負象限で m 個の接超平面を一樣乱数を用いて発生させ, 超楕円体を含むように, $Ax \leq b$ を定めた. す

表 1: Γ が矩形の場合の実験結果

n	m	アプローチ	平均	分散	最大値	最小値	中央値
5	10	2 段階	0.0395	0.0011	0.125	0.015	0.023
		2 レベル計画	0.7049	0.1157	1.257	0.273	0.6675
		問題の変形	0.0902	0.0018	0.164	0.039	0.074
		外部近似法	0.0362	0.0010	0.109	0.007	0.023
	15	2 段階	0.0760	0.0041	0.257	0.031	0.050
		2 レベル計画	2.5418	2.7635	6.343	1.078	1.8005
		問題の変形	0.1768	0.0049	0.312	0.101	0.144
		外部近似法	0.0682	0.0037	0.242	0.023	0.046
	25	2 段階	0.3010	0.0715	0.773	0.070	0.1635
		2 レベル計画	9.7441	41.7437	25.921	3.789	7.265
		問題の変形	0.3941	0.0321	0.742	0.156	0.3355
		外部近似法	0.1558	0.0166	0.406	0.039	0.093
10	15	2 段階	16.5871	397.8003	65.687	1.031	8.480
		2 レベル計画	71.9534	2036.8832	182.546	25.250	53.906
		問題の変形	1.4416	0.4979	2.882	0.609	1.253
		外部近似法	16.0341	295.9618	55.898	1.273	9.691
	20	2 段階	74.9309	7883.5408	293.648	0.203	49.1125
		2 レベル計画	161.6902	17724.2119	465.492	15.710	102.2925
		問題の変形	2.6237	2.7139	6.125	0.562	2.2765
		外部近似法	39.6534	2088.3297	133.671	0.187	16.6445
	35	2 段階	90.9410	13326.6253	345.718	0.789	29.6365
		2 レベル計画	757.4230	285838.4229	1562.070	82.859	740.7925
		問題の変形	4.9268	11.4718	11.476	0.718	3.6675
		外部近似法	40.6002	3446.5210	196.585	0.828	11.757

なわち, $[0, 1]$ 内の値をとる独立な一様乱数 r_1^j , $j = 1, 2, \dots, n$, $[1, 3]$ 内の値をとる独立な一様乱数 r_2^j , $i = 1, 2, \dots, n$ を発生させることにより, 1 本の制約条件を

$$\sum_{j=1}^n r_1^j r_2^j x_j \leq |r_1|$$

と生成した. ただし, $\mathbf{r}_1 = (r_1^1, r_1^2, \dots, r_1^n)^T$ で, $|\mathbf{r}_1|$ は $\mathbf{r}_1$ のノルムである. また, 式 (3) の Γ の生成は, $j = 1, 2, \dots, n$ について, c_j^L, c_j^R を一様乱数により発生させることにより行った. すなわち, $\{1, 2, 3\}$ 内の値をとる一様乱数 r_3^j , $\{2, 3, 4\}$ 内の値をとる一様乱数 r_4^j , $\{0, 1, \dots, 9\}$ 内の値をとる独立な一様乱数 r_5^j, r_6^j により, 次のように生成した.

$$c_j^L = r_3^j + 0.1r_5^j, \quad c_j^R = r_4^j + 0.1r_6^j, \quad j = 1, 2, \dots, n$$

4.2.2 実験結果と考察

いくつかのパラメータの組合せ (n, m) について数値実験を行い, 表 1 に示す結果が得られた. ただし, 問題の変形によるアプローチでは, 予備実験において最も良好な結果が得られた次の場合を用いた. すなわち,

$$M_j = \max_{y \in X} y_j, \quad j = 1, 2, \dots, n$$

により, M_j を評価し, 式 (16) の制約条件を加えた場合を考えた. なお, 計測した計算時間には, M_j の評価に必要な計算時間も含まれている.

表 1 より, n, m がともに十分小さい場合を除き, 問題の変形によるアプローチが極めて効率的であることがわかる. また, 分散も小さく安定しており, 端点を列挙しないので, 必要な計算機の記憶容量も少ない. したがって, 区間線形計画問題の場合は, 四つのアプローチのうち, 問題の変形によるアプローチが最も適しているといえる. 他のアプローチについては, 外部近似法によるアプローチの方が, 2 段階アプローチよりも効率的であることがわかる. 2 レベル計画問題によるアプローチはあまり効率的でないようである.

4.3 Γ が矩形でない場合

4.3.1 問題の生成方法

制約条件 X の生成方法は, 区間線形計画問題の場合と同様である. 凸多面体 Γ の生成は, 次の手順で行った.

1. X の発生と同様に, 原点を中心とする超楕円体を用意し, $(p - (n + 1))$ 個の接超平面を一樣乱数に発生させ, これらの接超平面を境界とする超楕円体を含む制約集合 Γ_1 を定める.
2. $[15, 22]$ 内の値をとる独立な一樣乱数 $r_7^j, j = 1, 2, \dots, n$ を発生させ, Γ_1 を $r_7 = (r_7^1, r_7^2, \dots, r_7^n)^T$ だけ平行移動した Γ_2 を求める.
3. Γ_2 は有界とは限らないので, $[0, 3]$ 内の値をとる独立な一樣乱数 $r_8^j, j = 1, 2, \dots, n+1$ を発生させ, Γ_2 に n 本の制約条件 $c_j \leq r_8^j + 9, j = 1, 2, \dots, n$ および 1 本の制約条件 $1_n^T c \leq 37 - r_8^{n+1}$ を加え, Γ_3 とする.
4. $[0, 1]$ 内の値をとる独立な一樣乱数 $r_9^{ij}, i = 1, 2, \dots, n, j = 1, 2, \dots, n$ を発生させ, r_9^{ij} を (i, j) 成分とする正則な行列 Q で Γ_3 を線形変換し, Γ とする.

4.3.2 実験結果と考察

いくつかのパラメータの組合せ (n, m, p) について数値実験を行い, 表 2 に示す結果が得られた. ただし, $n = 15, m = 30, p = 20$ の場合については, 生成した 10 問題の一つは, 可能的最適端点が多く, 本研究で作成した 2 段階アプローチでは, 最大リグレット最小解が計算できなかった. このため, 残りの 9 問題での各統計量を算出した. 表 2 に示すように, n, m が十分大きい場合には, 外部近似法によるアプローチが最も効率的である. また, 2 レベル計画問題によるアプローチは, Γ が矩形でない凸多面体の場合も, あまり効率的ではなかった. $n = 15, m = 30, p = 20$ の場合のように, 計算効率化のため, 未探索の可能的最適基底の基底逆行列をすべて保存する 2 段階アプローチでは, 記憶容量の制限から計算できない場合がある. 未探索の可能的最適基底の基底逆行列を保存する代わりに基底の組合せを保存すれば, より規模の大きい問題も 2 段階アプローチで解ける. ただし, 基底逆行列の計算にかなりの時間を要すると予想される.

表 2: Γ が矩形でない場合の実験結果

n	m	p	アプローチ	平均	分散	最大値	最小値	中央値
5	10	15	2 段階	0.0253	0.00036	0.070	0.007	0.019
			2 レベル計画	1.3004	0.59839	3.000	0.507	1.1405
			外部近似法	0.0409	0.00090	0.109	0.015	0.031
	15	15	2 段階	0.0284	0.00025	0.062	0.015	0.023
			2 レベル計画	4.8690	6.59400	10.460	2.242	3.8355
			外部近似法	0.0316	0.00015	0.062	0.023	0.027
	25	15	2 段階	0.0712	0.00046	0.093	0.023	0.0815
			2 レベル計画	17.4511	35.08627	25.101	3.320	16.679
			外部近似法	0.0909	0.00116	0.164	0.023	0.097
10	15	20	2 段階	1.1957	6.59	8.890	0.179	0.367
			2 レベル計画	74.2363	2159.25	192.640	26.515	54.816
			外部近似法	1.7104	11.99	12.054	0.179	0.5775
	20	20	2 段階	1.7761	7.63	9.820	0.273	0.519
			2 レベル計画	249.6801	39557.96	734.515	77.328	157.0305
			外部近似法	1.8286	4.44	7.703	0.414	0.859
	30	20	2 段階	16.0627	969.51	105.132	0.593	3.3585
			2 レベル計画	765.8563	185784.70	1514.460	310.710	701.0815
			外部近似法	7.0410	100.21	35.539	0.429	3.742
15	20	20	2 段階	58.1739	1310.36	107.539	3.820	67.6245
			2 レベル計画	1426.9410	342209.90	2547.125	766.359	1254.535
			外部近似法	67.7551	1946.86	125.500	5.445	79.8045
	30	20	2 段階	283.7852	217283.10	1580.109	10.023	115.648
			2 レベル計画	5777.5130	7239909.00	10519.080	1923.367	5579.570
			外部近似法	120.3581	8390.39	308.875	20.945	102.398

5 おわりに

目的関数の係数ベクトルの取りうる範囲が有界な凸多面体で与えられた線形計画問題の最大リグレット最小解の計算方法について議論した。部分問題が、双線形の凸最大化問題であることから、種々のアプローチが考えられる。比較したアプローチのうち、 Γ が矩形の場合については、問題の変形によるアプローチ、 Γ が矩形でない場合については、外部近似法によるアプローチが、それぞれ、有効であった。凸最大化問題や双線形計画問題、非凸 2 次計画問題、2 レベル計画問題に対して、種々の解法が考えられているので、これらの解法を基にしたアプローチも可能である。最大リグレット最小化問題の性質をどれだけ解法に反映できるかが、解法の高速度性を左右するであろう。他の解法に基づいたアプローチの考察が、今後の課題である。

参考文献

- [1] H. Rommelfanger, R. Hanuscheck and J. Wolf: Linear Programming with Fuzzy Objectives, *Fuzzy Sets and Systems*, **29** pp.31–48 (1989).
- [2] H. Ishibuchi and H. Tanaka: Mutiobjective Programming in Optimization of the Interval Objective Function, *European Journal of Operational Research*, **48** pp.219–225 (1990).

- [3] N. Furukawa: A Parametric Total Order on Fuzzy Numbers and a Fuzzy Shortest Route Problem, *Optimization*, **10** pp.367–377 (1994).
- [4] 乾口, 久米: 最大リグレット最小化に基づく区間目的関数をもつ線形計画問題の解の概念, *日本経営工学会誌*, **42**(3) pp.193–199 (1991).
- [5] M. Inuiguchi and Y. Kume: Minimax Regret in Linear Programming Problems with an Interval Objective Function, in: G. H. Tzeng, H. F. Wang, U. P. Wen and P. L. Yu (Eds.), *Multiple Criteria Decision Making*, Springer-Verlag, New York, pp.65–74 (1994).
- [6] M. Inuiguchi and M. Sakawa: Minimax Regret Solution Algorithms for Linear Program with Convex Polyhedral Objective Coefficients, *Proceedings of Second European Workshop on Fuzzy Decision Analysis and Neural Networks for Management, Planning and Optimization*, pp.116–125 (1997).
- [7] M. Inuiguchi and T. Tanino: Portfolio Selection under Independent Possibilistic Information, *Fuzzy Sets and Systems* (to appear).
- [8] M. Inuiguchi and M. Sakawa: Maximum Regret Analysis in Linear Programs with an Interval Objective Function, *Proceedings of International Workshop on Soft Computing in Industry '96*, pp.308–317 (1996).
- [9] K. Shimizu, Y. Ishizuka and J. F. Bard: *Nondifferentiable and Two-Level Mathematical Programming*, Kluwer Academic Publishers, Boston (1997).
- [10] H. E. Mausser and M. Laguna: A New Mixed Integer Formulation for the Maximum Regret Problem, *International Transactions in Operational Research*, **5** pp.389–403 (1998).
- [11] R. Horst and H. Tuy: *Global Optimization: Deterministic Approaches, Third, Revised and Enlarged Edition*, Springer-Verlag, Berlin (1995).
- [12] R. E. Steuer: Algorithms for linear programming problems with interval objective function coefficients, *Mathematics of Operations Research*, **6**(3) pp.333–348 (1981).
- [13] 東谷, 乾口, 谷野: 目的関数の係数ベクトルが凸多面体で表された線形計画問題における可能的最適端点の列挙, 第14回ファジィシステムシンポジウム講演論文集, pp.419–420 (1998).