2 次錐最適化の基本と応用

村松正和

電気通信大学
情報理工学研究科 情報・通信工学専攻
Email: MasakazuMuramatsu@uec.ac.jp

2013/5/17 at 構造計画研究所

Outline

2 次錐最適化とは

錐線形計画の歴史

2 次錐最適化の解き方

2 次錐最適化の応用

2 次錐最適化

この講演の目標：
2 次錐最適化とはどういうものか, その アイデア を知る

2 次錐最適化

この講演の目標：
2 次錐最適化とはどういうものか, その アイデア を知る

- 2 次「錘」最適化ではありません
- 2 次錐最適化問題 ＝ Second-Order Cone Programming — **SOCP** と一般に略される
- **SCOP** は

2 次錐最適化

この講演の目標：
2 次錐最適化とはどういうものか，その アイデア を知る

- 2 次「錘」最適化ではありません
- 2 次錐最適化問題 = Second-Order Cone Programming — **SOCP** と一般に略される
- **SCOP** は Solver for COnstraint Programming

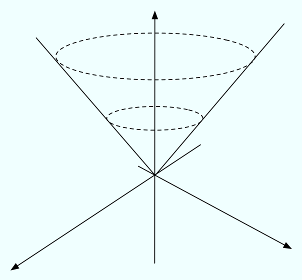
全然内容が異なるので注意してください

- 前者は最適化問題のカテゴリ
- 後者は最適化問題を解くソフトウェア

本講演では SOCP とは書かず、2 次錐最適化と書くことにします

2 次錐の例

3 次元空間の 2 次錐 $\{ (x_0, x_1, x_2)^T \mid x_0 \geq \sqrt{x_1^2 + x_2^2} \}$

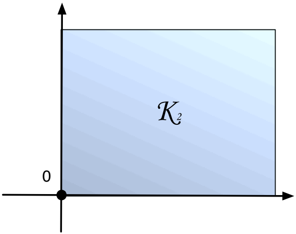


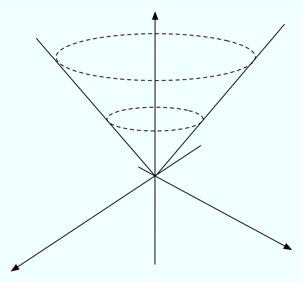
- x_0 軸に垂直な面で切ると、円 (とその内部)
- 凸集合かつ閉集合
- 錐: ベクトル $\mathbf{x}$ がその集合に入っていれば、それに正の数をかけたベクトルもその集合に入っている ($\forall \mathbf{x} \in K, \forall a > 0,\ a\mathbf{x} \in K$)

さまざまな次元の２次錐

r 次元の２次錐の定義

$$\mathcal{K}_r = \left\{ \mathbf{x} \in \mathbb{R}^r \;\middle|\; x_0 \geq \sqrt{\sum_{j=1}^{r-1} x_j^2} \right\}$$





$$x_0 \geq \sqrt{x_1^2} \Leftrightarrow x_0 \geq |x_1|$$

2 次錐の直積

$$\mathcal{K} = \mathcal{K}_{n_1} \times \mathcal{K}_{n_2} \times \cdots \times \mathcal{K}_{n_p}$$

- それぞれの $\mathcal{K}_{n_i}$ は n_i 次元の 2 次錐
- $\mathcal{K} \subseteq \mathbb{R}^N$
- $\mathcal{K}_{n_i} \subseteq \mathbb{R}^{n_i}$ $\quad \sum_{i=1}^{p} n_i = N$

2 次錐の直積

$$\mathcal{K} = \mathcal{K}_{n_1} \times \mathcal{K}_{n_2} \times \cdots \times \mathcal{K}_{n_p}$$

- それぞれの $\mathcal{K}_{n_i}$ は n_i 次元の 2 次錐
- $\mathcal{K} \subseteq \mathbb{R}^N$
- $\mathcal{K}_{n_i} \subseteq \mathbb{R}^{n_i} \quad \sum_{i=1}^{p} n_i = N$

定義: $\mathbf{x} \in \mathbb{R}^N$ が $\mathcal{K}$ に含まれる

$$\Leftrightarrow \mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_p \end{pmatrix}, \qquad \mathbf{x}_i \in \mathcal{K}_{n_i} \ (i = 1, \cdots, p).$$

2 次錐の直積

$$\mathcal{K} = \mathcal{K}_{n_1} \times \mathcal{K}_{n_2} \times \cdots \times \mathcal{K}_{n_p}$$

- それぞれの $\mathcal{K}_{n_i}$ は n_i 次元の 2 次錐
- $\mathcal{K} \subseteq \mathbb{R}^N$
- $\mathcal{K}_{n_i} \subseteq \mathbb{R}^{n_i} \qquad \sum_{i=1}^{p} n_i = N$

定義: $\mathbf{x} \in \mathbb{R}^N$ が $\mathcal{K}$ に含まれる

$$\Leftrightarrow \mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_p \end{pmatrix}, \qquad \mathbf{x}_i \in \mathcal{K}_{n_i} \ (i = 1, \cdots, p).$$

注: 非負変数は $\mathcal{K}_1$ の元とみなすことができる

2 次錐最適化問題

等式標準形の 2 次錐最適化問題

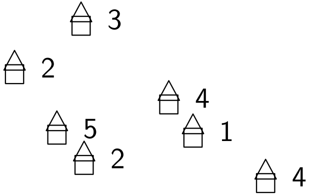
$$\text{最小化} \quad \mathbf{c}^T\mathbf{x} \quad \text{条件} \quad A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{K}$$

- $\mathcal{K}$ は 2 次錐の直積
- 線形制約 + 2 次錐制約
- 線形目的関数

注: もし $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_1 \times \cdots \times \mathcal{K}_1$ ならば、これは等式標準形の線形計画問題 (LP).

例: Weber 問題

砂漠に 7 組の家がある。共同でお金を出し合い井戸を掘るとき，どの位置に掘るのがもっとも公平だろうか？



家	x	y	人数
a	24	54	2
b	60	63	1
c	1	84	2
d	23	100	3
e	84	48	4
f	15	64	5
g	52	74	4

1 つの考え：**水の輸送量** × **距離の合計** を最小にする．
（各家の水の消費量は人数に比例 ）

Weber 問題の定式化

- 家の集合: H
- 家 i の位置: (X_i, Y_i) $(i \in H)$
- 家 i が 1 日に必要とする水の量: $w_i > 0$
- 井戸の位置を (x, y) とすると家 i から井戸までの距離は
$$\sqrt{(X_i - x)^2 + (Y_i - y)^2}$$
- 問題:
$$\sum_{i \in H} w_i \sqrt{(X_i - x)^2 + (Y_i - y)^2}$$
を最小にする (x, y) を見つけよ.

2 次錐最適化への定式化

$$
\begin{array}{ll}
\text{最小化} & \sum_{i \in H} w_i z_i \\
\text{条件} & \sqrt{(X_i - x)^2 + (Y_i - y)^2} \leq z_i \quad \forall i \in H
\end{array}
$$

(最適解では上記の $\leq$ が実は $=$ で成り立つことに注意)

2 次錐最適化への定式化

最小化	$\sum_{i \in H} w_i z_i$
条件	$\sqrt{(X_i - x)^2 + (Y_i - y)^2} \leq z_i \quad \forall i \in H$

(最適解では上記の $\leq$ が実は $=$ で成り立つことに注意)

$$\Downarrow$$

2 次錐最適化問題

最小化	$\sum_{i \in H} w_i z_i$
条件	$x_i = X_i - x, \;\; y_i = Y_i - y \; (i \in H)$
	$(z_i, x_i, y_i) \in \mathcal{K}_3 \; (i \in H)$

- 変数の数は $3|H| + 2$
- 等式制約 + 2 次錐制約
- 線形目的関数

Outline

線形最適化 (LP)

- 1947 年 George Dantzig により提案

線形最適化 (LP)

- 1947 年 George Dantzig により提案
- 線形制約, 線形目的関数
- 標準形 LP

$$\text{最小化} \quad \mathbf{c}^T\mathbf{x} \quad \text{条件} \quad A\mathbf{x} = \mathbf{b}, \ \ \mathbf{x} \geq \mathbf{0}$$

線形最適化 (LP)

- 1947 年 George Dantzig により提案
- 線形制約, 線形目的関数
- 標準形 LP

$$\text{最小化}\quad \mathbf{c}^T\mathbf{x}\quad \text{条件}\quad A\mathbf{x}=\mathbf{b},\ \ \mathbf{x}\geq \mathbf{0}$$

- 双対問題

$$\text{最大化}\quad \mathbf{b}^T\mathbf{y}\quad \text{条件}\quad \mathbf{s}+A^T\mathbf{y}=\mathbf{c},\ \ \mathbf{s}\geq \mathbf{0}$$

- 弱双対定理: $\mathbf{c}^T\mathbf{x}\geq \mathbf{b}^T\mathbf{y}$

線形最適化 (LP)

- 1947 年 George Dantzig により提案
- 線形制約, 線形目的関数
- 標準形 LP

$$\text{最小化} \quad \mathbf{c}^T\mathbf{x} \quad \text{条件} \quad A\mathbf{x} = \mathbf{b}, \ \ \mathbf{x} \geq \mathbf{0}$$

- 双対問題

$$\text{最大化} \quad \mathbf{b}^T\mathbf{y} \quad \text{条件} \quad \mathbf{s} + A^T\mathbf{y} = \mathbf{c}, \ \ \mathbf{s} \geq \mathbf{0}$$

- 弱双対定理: $\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}$
- 双対定理: 最適解 $\mathbf{x}^*$, $(\mathbf{y}^*, \mathbf{s}^*)$ は $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$ を満たす

LP に対するアルゴリズム

1. 単体法
   - 1947 年, George Dantzig. (LP の提案と同時)
   - 効率的に解くことができる
   - 理論的には「多項式時間」でない

LP に対するアルゴリズム

1. 単体法
   - 1947 年, George Dantzig. (LP の提案と同時)
   - 効率的に解くことができる
   - 理論的には「多項式時間」でない
2. 内点法
   - 1984 年, Narendra Karmarker.
   - 理論的に「多項式時間」
   - 実際にも効率よく解くことができる

注. 内点法そのものはそれより前に例えば Dikin (1964) により提案されている.
Karmarker は「ある」内点法の提案とその「多項式時間収束」の証明を与えた

錐線形計画とは

LP

$$\text{最小化}\quad \mathbf{c}^T\mathbf{x}\quad \text{条件}\quad A\mathbf{x}=\mathbf{b},\ \mathbf{x}\geq \mathbf{0}$$

$$\Updownarrow$$

$$\text{最小化}\quad \mathbf{c}^T\mathbf{x}\quad \text{条件}\quad A\mathbf{x}=\mathbf{b},\ \mathbf{x}\in \mathcal{K}$$

$\mathcal{K}=\{\,\mathbf{x}\in\mathbb{R}^n\,|\,x_1\geq 0, x_2\geq 0,\cdots,x_n\geq 0\,\}$: 第一象限

錐線形計画とは

LP

$$\text{最小化}\quad \mathbf{c}^T\mathbf{x}\quad \text{条件}\quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$$

$$\Updownarrow$$

$$\text{最小化}\quad \mathbf{c}^T\mathbf{x}\quad \text{条件}\quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathcal{K}$$

$\mathcal{K} = \{\, \mathbf{x} \in \mathbb{R}^n \,|\, x_1 \geq 0, x_2 \geq 0, \cdots, x_n \geq 0 \,\}$: 第一象限
$\mathcal{K}$ は閉凸錐, つまり

1. 閉 (closed) — 閉集合ということ
2. 凸 (convex) — 凸集合ということ
3. 錐 (cone) — $\mathbf{x} \in \mathcal{K}$ のとき, 任意の $\lambda \geq 0$ に対して $\lambda\mathbf{x} \in \mathcal{K}$.

錐線形計画とは

LP

$$\text{最小化} \quad \mathbf{c}^T\mathbf{x} \quad \text{条件} \quad A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}$$

$$\Updownarrow$$

$$\text{最小化} \quad \mathbf{c}^T\mathbf{x} \quad \text{条件} \quad A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{K}$$

$\mathcal{K} = \{ \mathbf{x} \in \mathbb{R}^n \,|\, x_1 \geq 0, x_2 \geq 0, \cdots, x_n \geq 0 \}$: 第一象限
$\mathcal{K}$ は閉凸錐, つまり

1. 閉 (closed) — 閉集合ということ
2. 凸 (convex) — 凸集合ということ
3. 錐 (cone) — $\mathbf{x} \in \mathcal{K}$ のとき, 任意の $\lambda \geq 0$ に対して $\lambda\mathbf{x} \in \mathcal{K}$.

- $\mathcal{K}$ として第一象限をとれば、LP
- 他の閉凸錐をとることもできる ⇒ 錐線形計画
- 特に 2 次錐を取れば、2 次錐最適化問題

さまざまな錐線形計画問題

錐 $\mathcal{K}$ と対応する最適化問題

1. 第一象限 $\{\, \mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{x} \geq \mathbf{0} \,\}$: 線形最適化 (LP)
2. 2 次錐とその直積 : 2 次錐最適化 (SOCP)
3. 半正定値行列錐 : 半正定値最適化 (SDP)

錐線形計画問題の性質

- 凸計画問題
- 双対問題が定義できる

$$\text{最小化} \quad \mathbf{c}^T\mathbf{x} \quad \text{条件} \quad A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{K}$$

$$\Updownarrow \text{双対}$$

$$\text{最大化} \quad \mathbf{b}^T\mathbf{y} \quad \text{条件} \quad \mathbf{s} + A^T\mathbf{y} = \mathbf{c}, \ \mathbf{s} \in \mathcal{K}^*$$

ただし $\mathcal{K}^* = \{\,\mathbf{s}\,|\,\mathbf{s}^T\mathbf{x} \geq 0 \ (\forall \mathbf{x} \in \mathcal{K})\,\}$: 双対錐

錐線形計画問題の性質

- 凸計画問題
- 双対問題が定義できる

$$\text{最小化} \quad \mathbf{c}^T\mathbf{x} \quad \text{条件} \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathcal{K}$$

$$\Updownarrow \text{双対}$$

$$\text{最大化} \quad \mathbf{b}^T\mathbf{y} \quad \text{条件} \quad \mathbf{s} + A^T\mathbf{y} = \mathbf{c},\ \mathbf{s} \in \mathcal{K}^*$$

ただし $\mathcal{K}^* = \{\, \mathbf{s} \,|\, \mathbf{s}^T\mathbf{x} \geq 0 \ (\forall \mathbf{x} \in \mathcal{K}) \,\}$: 双対錐

- 弱双対定理 ($\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}$) が成り立つ
- 双対定理は ($\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$ なる最適解が存在する) は LP とはやや違った形で成り立つ

錐線形計画問題の性質

- 凸計画問題
- 双対問題が定義できる

$$\text{最小化} \quad \mathbf{c}^T\mathbf{x} \quad \text{条件} \quad A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{K}$$

$$\Updownarrow \text{双対}$$

$$\text{最大化} \quad \mathbf{b}^T\mathbf{y} \quad \text{条件} \quad \mathbf{s} + A^T\mathbf{y} = \mathbf{c}, \ \mathbf{s} \in \mathcal{K}^*$$

  ただし $\mathcal{K}^* = \{\, \mathbf{s} \,|\, \mathbf{s}^T\mathbf{x} \geq 0 \ (\forall \mathbf{x} \in \mathcal{K}) \,\}$: 双対錐
- 弱双対定理 ($\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}$) が成り立つ
- 双対定理は ($\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$ なる最適解が存在する) は LP とはやや違った形で成り立つ
- 内点法で効率よく解かれる (“多項式時間”)

線形計画から錐線形計画へ

- 1956 年, Duffin.
  — 錐線形計画の双対理論
  — アルゴリズムの話はない
- 1989 年, Nesterov と Nemirovski.
  — 錐線形計画に対する主内点法
  — 半正定値計画 (SDP)
  — 2 次錐計画 (SOCP)
- 1991 年, Nesterov と Todd
  — 対称錐線形計画に対する主双対内点法
- 1990 年代
  — 対称錐線形計画に対する主双対内点法の実装に関する研究始まる
- 2012 年
  商用整数最適化ソルバーに 2 次錐最適化ソルバーが搭載される

Outline

2 次錐最適化とは

錐線形計画の歴史

2 次錐最適化の解き方

2 次錐最適化の応用

Gurobi で扱える 2 次錐制約の表現

1. $\mathbf{x}^T\mathbf{x} \le y^2$, y は非負変数: 2 次錐制約 (を 2 乗したもの)
2. $\mathbf{x}^T\mathbf{x} \le yz$, y, z は非負変数: 回転つき 2 次錐制約
3. $\mathbf{x}^T Q\mathbf{x} + \mathbf{q}^T\mathbf{x} \le b$: 凸 2 次制約

注 1: $\sqrt{\cdot}$ が直接現れる形でのサポートはない
注 2: これらの制約はみな 2 次錐制約に帰着される

2 次錐制約に帰着可能な制約

1. $\mathbf{x}^T\mathbf{x} \leq t^2$, t は非負変数
2. $\mathbf{x}^T\mathbf{x} \leq ts$, t, s は非負変数
3. $\mathbf{x}^T Q\mathbf{x} + \mathbf{q}^T\mathbf{x} \leq b$
4. $\mathbf{x}^T\mathbf{x} \leq t$.
5. $xy \geq 1$, x は非負変数.
6. $\frac{xy}{x+y} \geq 1$, x, y は非負変数.
7. x^2/y の最小化 (x, y は非負変数)
   ($\Leftrightarrow$ $x^2 \leq ty$ という制約の元での t の最小化)

注: 1,2,3 以外は自分で Gurobi に合うように変形する必要がある
例えば, 5 は 2 の形に帰着可能だが、どうやるかわかりますか？

凸 2 次制約に関する注意

凸 2 次制約

$$\mathbf{x}^T Q\mathbf{x} + \mathbf{q}^T\mathbf{x} \leq b$$

- Q は半正定値

凸2次制約に関する注意

凸2次制約

$$\mathbf{x}^T Q \mathbf{x} + \mathbf{q}^T \mathbf{x} \leq b$$

- Q は半正定値
- 集合 $\{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{x}^T Q \mathbf{x} + \mathbf{q}^T \mathbf{x} \leq b\}$ は凸集合
- Q が半正定値でないとき，上記集合は凸集合にならない．
- この場合、2次錐制約に帰着は不可能．

2次関数においては Q が半正定値かどうかが鍵

Gurobi/python における 2 次錐制約の書き方

$$x_i^2 + y_i^2 \leq z_i^2$$

```
model.addQConstr(x[i]*x[i]+y[i]*y[i]<=z[i]*z[i])
```

$$\sum_{i \in I} x_i^2 \leq z^2$$

```
model.addQConstr(quicksum(x[i]*x[i] for i in I)
   <=z[i]*z[i])
```

注：addQconstr() の代わりに addConstr() を使っても同じ結果が得られる

起きやすいエラー

Weber 問題の Gurobi python でのコーディング

```
model.addConstr((x[i]-X)*(x[i]-X)
        + (y[i]-Y)*(y[i]-Y) <= z[i]*z[i])
```

起きやすいエラー

Weber 問題の Gurobi python でのコーディング

```
model.addConstr((x[i]-X)*(x[i]-X)
        + (y[i]-Y)*(y[i]-Y) <= z[i]*z[i])
```

実行すると…

`gurobipy.GurobiError: Q matrix is not positive semi-definite (PSD)` 「Q が半正定値じゃないで〜」

起きやすいエラー

Weber 問題の Gurobi python でのコーディング

```
model.addConstr((x[i]-X)*(x[i]-X)
       + (y[i]-Y)*(y[i]-Y) <= z[i]*z[i])
```

実行すると…

```
 gurobipy.GurobiError:  Q matrix is not positive
semi-definite (PSD)
```

「Q が半正定値じゃないで〜」

- Q が半正定値でないと, Gurobi は扱えない
- でもここでの Q は実際には半正定値である
- Q が複雑な形をしているので、Gurobi は半正定値かどうか判定できない

起きやすいエラー

Weber 問題の Gurobi python でのコーディング

```
model.addConstr((x[i]-X)*(x[i]-X)
        + (y[i]-Y)*(y[i]-Y) <= z[i]*z[i])
```

実行すると…

```
 gurobipy.GurobiError:  Q matrix is not positive
semi-definite (PSD)
```
「Q が半正定値じゃないで〜」

- Q が半正定値でないと, Gurobi は扱えない
- でもここでの Q は実際には半正定値である
- Q が複雑な形をしているので、Gurobi は半正定値かどうか判定できない

解決策: 一旦他の変数で受ける

```
model.addConstr(xa[i]*xa[i]+ya[i]*ya[i]<=z[i]*z[i])
model.addConstr(xa[i]==(x[i] − X))
model.addConstr(ya[i]==(y[i] − Y))
```

Outline

「新しい数理最適化」で扱った例題

1. Weber 問題
2. 経済発注量問題 (分数最適化)
3. 混合問題に対するロバスト最適化
4. ポートフォリオ最適化
    4.1 マーコヴィッツのモデル
    4.2 損失確率をおさえるモデル
5. 凹費用関数を持つ単一ソース制約付き施設配置問題

以上は全て 2 次錐最適化問題に帰着させ、ソースコードも書かれている

「新しい数理最適化」で扱った例題

1. Weber 問題
2. 経済発注量問題 (分数最適化)
3. 混合問題に対するロバスト最適化
4. ポートフォリオ最適化
   4.1 マーコヴィッツのモデル
   4.2 損失確率をおさえるモデル
5. 凹費用関数を持つ単一ソース制約付き施設配置問題

以上は全て 2 次錐最適化問題に帰着させ、ソースコードも書かれている
以下では混合問題に対するロバスト最適化について説明する

混合問題の例

- 4 種類の原料を調達・混合して 1 種類の製品を製造している工場
- 原料には，3 種類の成分が含まれており，
成分 1 は 20% 以上に，成分 2 は 30% 以上に，成分 3 は 40% 以上になるように混合したい．
- 各原料の成分含有比率は，表の通り
- 原料の単価は 1 トンあたりそれぞれ 5, 6, 8, 20 万円
- 製品 1 トンを最小費用で製造する混合率を求めよ

成分	1	2	3
原料 1	25	15	30
原料 2	30	30	10
原料 3	15	65	0
原料 4	10	5	85

混合問題の定式化

$$
\begin{array}{rlll}
\text{最小化} & \displaystyle\sum_{i\in I} p_i x_i & & \\
\text{条件} & \displaystyle\sum_{i\in I} a_{ij} x_i \geq LB_j & j \in J \\
 & x_i \geq 0 & i \in I.
\end{array}
$$

- 原料 i, 単位当たりの値段 p_i
- 成分 j
- 原料 i の単位あたりの成分 j の含有量 a_{ij}
- 目標の製品の成分 j の最低含有量 LB_j

混合問題の定式化

$$
\begin{array}{lll}
\text{最小化} & \displaystyle\sum_{i\in I} p_i x_i & \\
\text{条件} & \displaystyle\sum_{i\in I} a_{ij} x_i \geq LB_j & j \in J \\
& x_i \geq 0 & i \in I.
\end{array}
$$

- 原料 i, 単位当たりの値段 p_i
- 成分 j
- 原料 i の単位あたりの成分 j の含有量 a_{ij}
- 目標の製品の成分 j の最低含有量 LB_j

これは通常の線形最適化問題

誤差に関する仮定

$\boxed{\text{原料 } i \text{ が成分 } j \text{ を含む割合 } a_{ij} \text{ が誤差 } e_{ij} \text{ を持つ}}$

- e_{ij} は a_{ij} より大きいのか小さいのかわからない
- それほど $|e_{ij}|$ は大きくない

$$(*) \qquad \sum_{i \in I} e_{ij}^2 \leq 1 \quad j \in J$$

誤差に関する仮定

$\boxed{\text{原料 } i \text{ が成分 } j \text{ を含む割合 } a_{ij} \text{ が誤差 } e_{ij} \text{ を持つ}}$

- e_{ij} は a_{ij} より大きいのか小さいのかわからない
- それほど $|e_{ij}|$ は大きくない

$$(*) \qquad \sum_{i \in I} e_{ij}^2 \leq 1 \quad j \in J$$

元々の制約

$$\sum_{i \in I} a_{ij} x_i \geq LB_j$$

誤差を考慮した制約

$$\sum_{i \in I} (a_{ij} + e_{ij}) x_i \geq LB_j$$

ただし e_{ij} は $(*)$ を満たすどの値を取るかわからない。

最悪の仮定

e_{ij} は次を満たす

$$(*) \qquad \sum_{i \in I} e_{ij}^2 \leq 1 \quad j \in J$$

$(*)$ を満たすどのような e_{ij} が来ても製品が十分な成分を持つようにする

$$\min \left\{ \sum_{i \in I} (a_{ij} + e_{ij}) x_i \ : \ \sum_{i \in I} e_{ij}^2 \leq 1 \right\} \geq LB_j$$

最悪の仮定

e_{ij} は次を満たす

$$(*) \qquad \sum_{i \in I} e_{ij}^2 \leq 1 \quad j \in J$$

(∗) を満たすどのような e_{ij} が来ても製品が十分な成分を持つようにする

$$\min \left\{ \sum_{i \in I} (a_{ij} + e_{ij}) x_i \ : \ \sum_{i \in I} e_{ij}^2 \leq 1 \right\} \geq LB_j$$

あるいは

$$\min \left\{ \sum_{i \in I} e_{ij} x_i \ : \ \sum_{i \in I} e_{ij}^2 \leq 1 \right\} \geq LB_j - \sum_{i \in I} a_{ij} x_i$$

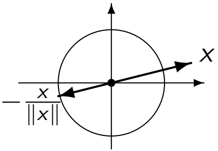
球上での線形関数の最小化

$$\min \left\{ \sum_{i \in I} e_{ij} x_i \ : \ \sum_{i \in I} e_{ij}^2 \leq 1 \right\}$$

- $e_{ij} (i \in I)$ が変数, x_i はデータとみなす (j は固定)
- 最適解は

$$e_{ij} = -\frac{x_i}{\|x\|}$$

- 最適値は $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$



2次錐制約へ

$$\min\left\{\sum_{i\in I} e_{ij}x_i \ :\ \sum_{i\in I} e_{ij}^2 \le 1\right\} \ge LB_j - \sum_{i\in I} a_{ij}x_i$$

$$\Downarrow e_{ij} = -x/\|x\|$$

$$-\sum_{i\in I}\frac{x_i^2}{\|x\|} = \underbrace{-\|x\| \ge LB_j - \sum_{i\in I} a_{ij}x_i}_{\text{2 次錐制約}}$$

Gurobi/python によるモデル記述

```
def robust(I, J, a, p, LB):
    model = Model("robust_mix")
    x = {}
    for i in I:
        x[i] = model.addVar()
    zaux = {}
    for k in J:
        zaux[j] = model.addVar()
    model.update()
    for j in J:
        model.addConstr(zaux[j] == - LB[j]
                + quicksum(a[i,j] * x[i] for i in I))
        model.addConstr(quicksum(x[i]*x[i] for i in I)
                <= zaux[j]*zaux[j])
    model.setObjective(quicksum(p[i]*x[i] for i in I),
                GRB.MINIMIZE)
    model.update()
    return model
```

例題の解

- ロバストバージョン
  最適値 9.3957427, 最適解 $(1.5521883, 0, 0.2043501, 0)$
- 通常バージョン
  最適値 9.038461538, 最適解 $(1.5, 0, 0.19230769, 0)$

（誤差に対する耐性を見込む分、お金がかかる）

例題の解

- ロバストバージョン
  最適値 9.3957427, 最適解 $(1.5521883, 0, 0.2043501, 0)$
- 通常バージョン
  最適値 9.038461538, 最適解 $(1.5, 0, 0.19230769, 0)$

（誤差に対する耐性を見込む分、お金がかかる）一般に、線形最適化問題に関して、係数行列に同様な形で誤差が導入されるモデルは 2 次錐最適化問題に定式化される

終わりに

- 2 次錐最適化問題が簡単に，高速に解かれるようになった

終わりに

- 2 次錐最適化問題が簡単に，高速に解かれるようになった
- Gurobi/python による 2 次錐制約のモデル記述

終わりに

- 2 次錐最適化問題が簡単に，高速に解かれるようになった
- Gurobi/python による 2 次錐制約のモデル記述
- さまざまな制約が 2 次錐制約に帰着可能
(ただし、凸制約でないときは帰着不可能)

終わりに

- 2 次錐最適化問題が簡単に，高速に解かれるようになった
- Gurobi/python による 2 次錐制約のモデル記述
- さまざまな制約が 2 次錐制約に帰着可能
  (ただし、凸制約でないときは帰着不可能)
- さまざまな応用例

終わりに

- 2 次錐最適化問題が簡単に, 高速に解かれるようになった
- Gurobi/python による 2 次錐制約のモデル記述
- さまざまな制約が 2 次錐制約に帰着可能
  (ただし、凸制約でないときは帰着不可能)
- さまざまな応用例
- Gurobi 5.5 では 整数制約 2 次最適化問題も求解可能

終わりに

- 2 次錐最適化問題が簡単に，高速に解かれるようになった
- Gurobi/python による 2 次錐制約のモデル記述
- さまざまな制約が 2 次錐制約に帰着可能
(ただし、凸制約でないときは帰着不可能)
- さまざまな応用例
- Gurobi 5.5 では 整数制約 2 次最適化問題も求解可能

もっと詳しく 2 次錐最適化、錐線形計画を知りたいときには…

- 「最適化法」田村明久／村松正和著　共立出版
- 「内点法」小島政和／水野真治／土谷隆ほか