計画数理学特論
～第12回：その他の最適化手法①：内点法～

担当：蓮池隆　(経営システム工学科)
e-mail: thasuike@waseda.jp

今回の講義：内点法

・不等式制約付きの最適化問題

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \\ &\text{subject to}\ \ g_i(\boldsymbol{x}) \le 0, i = 1,2,\dots,m \\ &\qquad\qquad\quad\ g_i(\boldsymbol{x}) = 0, i = m+1,\dots,p \end{aligned}$$

・この問題に対して，スラック変数$s_i \ge 0$を導入した等式最適化問題を考える

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \\ &\text{subject to}\ \ g_i(\boldsymbol{x}) + s_i = 0, i = 1,2,\dots,m \\ &\qquad\qquad\quad\ g_i(\boldsymbol{x}) = 0, i = m+1,\dots,p \\ &\qquad\qquad\quad\ s_i \ge 0, i = 1,2,\dots,m \end{aligned}$$

→ この問題の実行可能領域内の内点を通りながら最適化にたどり着く**内点法**をまず学習

今回の講義：内点法

・不等式制約付きの最適化問題

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \\ &\text{subject to}\ \ g_i(\boldsymbol{x}) \leq 0, i = 1,2,\dots,m \\ &\qquad\qquad\quad g_i(\boldsymbol{x}) = 0, i = m+1,\dots,p \end{aligned}$$

・この問題に対して，スラック変数$s_i \geq 0$を導入した等式最適化問題を考える

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \\ &\text{subject to}\ \ g_i(\boldsymbol{x}) + s_i = 0, i = 1,2,\dots,m \\ &\qquad\qquad\quad g_i(\boldsymbol{x}) = 0, i = m+1,\dots,p \\ &\qquad\qquad\quad s_i \geq 0, i = 1,2,\dots,m \end{aligned}$$

・**内点**：$s_i \geq 0$，つまり$g_i(\boldsymbol{x}) < 0$を満たす点

実行可能内点：$g_k(\boldsymbol{x}) = 0$も満たす内点

バリア関数の導入

・スラック変数s_iの非負制約に対して，**バリア関数を導入**($\rho > 0$)

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) - \rho \sum \log s_i$$

$$\begin{aligned} \text{subject to} \quad & g_i(\boldsymbol{x}) + s_i = 0, i = 1,2,\dots,m \\ & g_i(\boldsymbol{x}) = 0, i = m+1,\dots,p \\ & s_i \geq 0, i = 1,2,\dots,m \end{aligned}$$

・1次の必要条件を導入

1次の必要条件

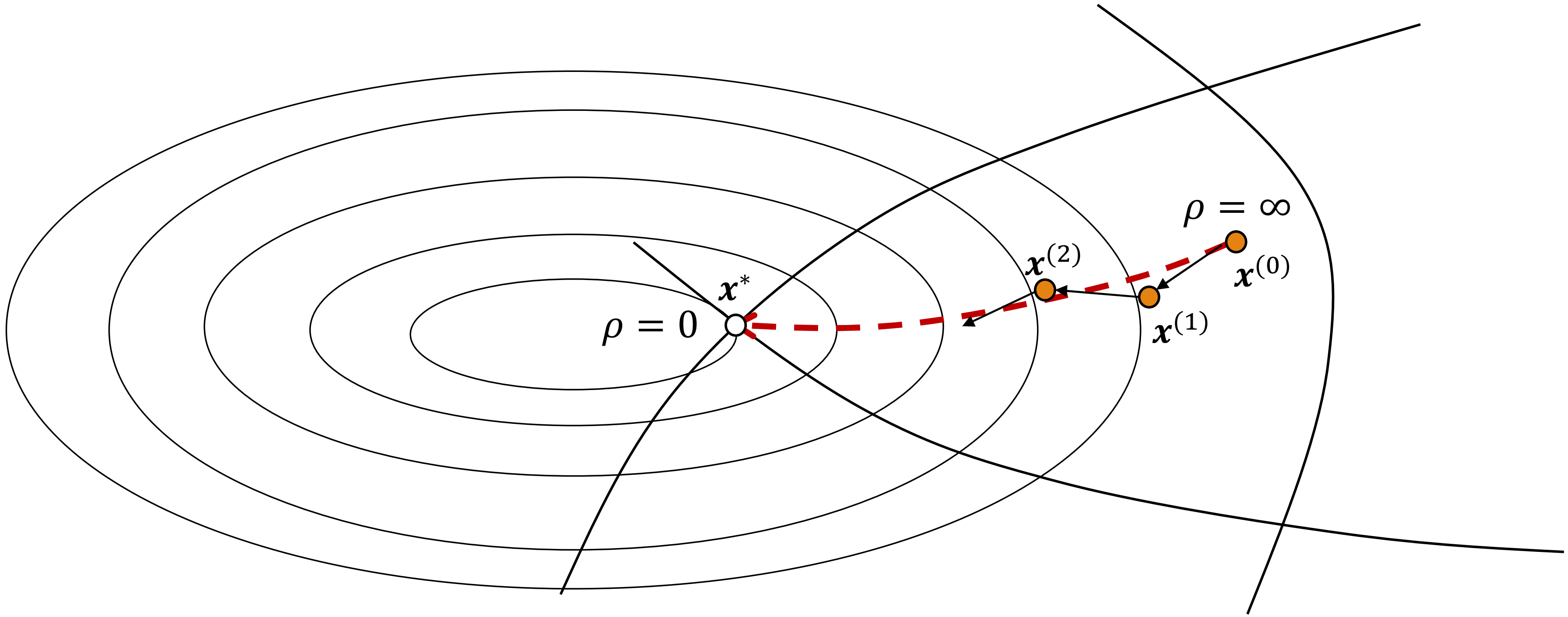
・先ほどの問題の局所最適解を$(\boldsymbol{x}^*, \boldsymbol{s}^*)$とするとき，1次の必要条件

$$\nabla f(\boldsymbol{x}^*) + \sum \lambda_i^* \nabla g_i(\boldsymbol{x}^*) = \mathbf{0}$$

$$\begin{gathered} \lambda_i^* s_i^* = \rho, i = 1,2, \dots, m \\ g_i(\boldsymbol{x}) + s_i^* = 0, i = 1,2, \dots, m \\ g_i(\boldsymbol{x}) = 0, i = m+1, \dots, p \\ s_i^* > 0, \lambda_i^* > 0, i = 1,2, \dots, m \end{gathered}$$

・$\rho \to 0$のとき，これらの条件を満たす$\big(\boldsymbol{x}^*(\rho), \boldsymbol{s}^*(\rho), \boldsymbol{\lambda}^*(\rho)\big)$がとる軌跡：**中心パス**

中心パスのイメージ



$\rho = \infty$
$\boldsymbol{x}^{(0)}$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}^{*}$
$\rho = 0$

反復法による点列生成

- 現在の点：$\left(\boldsymbol{x}^{(k)}, \boldsymbol{s}^{(k)}, \boldsymbol{\lambda}^{(k)}\right)$
- $\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{s}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}\right) = \left(\boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}, \boldsymbol{s}^{(k)} + \Delta\boldsymbol{s}, \boldsymbol{\lambda}^{(k)} + \Delta\boldsymbol{\lambda}\right)$

$$\rightarrow \begin{cases} \nabla f\left(\boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}\right) = \nabla f\left(\boldsymbol{x}^{(k)}\right) + \nabla^2 f\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x} \\ g_i\left(\boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}\right) = g_i\left(\boldsymbol{x}^{(k)}\right) + \nabla g_i\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x} \\ \nabla g_i\left(\boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}\right) = \nabla g_i\left(\boldsymbol{x}^{(k)}\right) + \nabla^2 g_i\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x} \end{cases}$$

- これらの式を先ほどの1次の必要条件に代入すると…

1次の必要条件

$$\nabla^2 f\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x} + \sum\left(\lambda_i^{(k)}\nabla^2 g_i\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x} + \Delta\lambda_i \nabla g_i\left(\boldsymbol{x}^{(k)}\right)\right)$$
$$= -\nabla f\left(\boldsymbol{x}^{(k)}\right) - \sum\left(\lambda_i^{(k)}\nabla g_i\left(\boldsymbol{x}^{(k)}\right)\right)$$

$$\lambda_i^{(k)}\Delta s_i + \Delta\lambda_i s_i^{(k)} = \rho - \lambda_i^{(k)} s_i^{(k)}, i = 1,2,\dots,m$$
$$\nabla g_i\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x} + \Delta s_i = -g_i\left(\boldsymbol{x}^{(k)}\right) - s_i^{(k)}, i = 1,2,\dots,m$$
$$\nabla g_i\left(\boldsymbol{x}^{(k)}\right)\Delta\boldsymbol{x} = -g_i\left(\boldsymbol{x}^{(k)}\right), i = m+1,\dots,p$$

・どの方程式も，$\left(\boldsymbol{x}^{(k)}, \boldsymbol{s}^{(k)}, \boldsymbol{\lambda}^{(k)}\right)$は与えられているので，右辺は定数
→ 左辺にある$\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{s}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}\right)$の連立方程式と見ればOK
ρの更新も合わせれば，アルゴリズムが構築できる.

ステップ幅の決定

- $(\Delta\boldsymbol{x}, \Delta\boldsymbol{s}, \Delta\boldsymbol{\lambda})$方向に進めば，最適解の方向には近づく
  $\rightarrow$ ステップ幅α_kを決める必要性あり
- 目的関数とペナルティ関数を足し合わせた**メリット関数を導入**

$$\phi_\eta(\boldsymbol{x}, \boldsymbol{s}) = f(\boldsymbol{x}) - \rho \sum \log s_i + \eta \sum |g_i(\boldsymbol{x}) + s_i| + \eta \sum |g_i(\boldsymbol{x})|$$

- 制約なし最適化問題：$\min\ \phi_\eta(\boldsymbol{x}, \boldsymbol{s})$で，直線探索によりステップ幅$\alpha_k$を求める
  $(\boldsymbol{s}^{(k)} + \alpha_k \Delta\boldsymbol{s} > \boldsymbol{0}, \boldsymbol{\lambda}^{(k)} + \alpha_k \Delta\boldsymbol{\lambda} > \boldsymbol{0}$ は満たすこと$)$

内点法のアルゴリズム

STEP1：初期点$\left(\boldsymbol{x}^{(0)}, \boldsymbol{s}^{(0)}, \boldsymbol{\lambda}^{(0)}\right)$およびパラメータの初期値$\rho_0$を定め，$k=0$とする.

STEP2：ρ_k が十分に小さければ，アルゴリズム終了

STEP3：1次の必要条件から得られる連立方程式を解いて，$(\Delta\boldsymbol{x}, \Delta\boldsymbol{s}, \Delta\boldsymbol{\lambda})$を求める.

$\boldsymbol{s}^{(k)}+\alpha_k\Delta\boldsymbol{s}>\boldsymbol{0}, \boldsymbol{\lambda}^{(k)}+\alpha_k\Delta\boldsymbol{\lambda}>\boldsymbol{0}$を満たすステップ幅$\alpha_k$を求める.

STEP4：$\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{s}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}\right)=\left(\boldsymbol{x}^{(k)}+\alpha_k\Delta\boldsymbol{x}, \boldsymbol{s}^{(k)}+\alpha_k\Delta\boldsymbol{s}, \boldsymbol{\lambda}^{(k)}+\alpha_k\Delta\boldsymbol{\lambda}\right)$とする.

また，$\rho_{k+1}<\rho_k$を満たすパラメータの値ρ_{k+1}を定め，$k=k+1$ として STEP2に戻る.

例を通してアルゴリズムを考える

(例題)

$$\min \ \ f(\boldsymbol{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$$
$$subject\ to \ \ g(\boldsymbol{x}) = x_1^2 - x_2 \leq 0$$

例を通してアルゴリズムを考える

(例題：スラック変数導入)

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\
subject\ to \quad & g(\boldsymbol{x}) = x_1^2 - x_2 + s = 0 \\
& s \geq 0
\end{aligned}
$$

(アルゴリズム)

STEP1：初期点$\boldsymbol{x}^{(0)} = (1, 2), s^{(0)} = 1, \lambda^{(0)} = 0$とする.

また，$\rho_0 = 1$，$\rho_{k+1} = 0.1 * \lambda^{(k+1)} s^{(k+1)}$とする.

例を通してアルゴリズムを考える

(例題：スラック変数導入)

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\ subject\ to \quad & g(\boldsymbol{x}) = x_1^2 - x_2 + s = 0 \\ & s \geq 0 \end{aligned}$$

(アルゴリズム)

STEP1：初期点$\boldsymbol{x}^{(0)} = (1, 2), s^{(0)} = 1, \lambda^{(0)} = 0$とする.
　　　また，$\rho_0 = 1$， $\rho_{k+1} = 0.1 * \lambda^{(k+1)} s^{(k+1)}$とする.
STEP2：$\rho_k = 1$　(まだ小さくないとして)

例を通してアルゴリズムを考える

(例題：スラック変数導入)

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\ \mathit{subject\ to} \quad & g(\boldsymbol{x}) = x_1^2 - x_2 + s = 0 \\ & s \geq 0 \end{aligned}$$

(アルゴリズム)

STEP3：次の必要条件から得られる連立方程式を解いて，$(\Delta\boldsymbol{x}, \Delta\boldsymbol{s}, \Delta\boldsymbol{\lambda})$を求める.

$$\Delta\boldsymbol{x} = (2.083, -1.271), \Delta s = -1.688, \Delta\lambda = 1.000$$

例を通してアルゴリズムを考える

(例題：スラック変数導入)

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\ subject\ to \quad & g(\boldsymbol{x}) = x_1^2 - x_2 + s = 0 \\ & s \geq 0 \end{aligned}$$

(アルゴリズム)

STEP3：次の必要条件から得られる連立方程式を解いて，$(\Delta\boldsymbol{x}, \Delta\boldsymbol{s}, \Delta\boldsymbol{\lambda})$を求める.

$$\Delta\boldsymbol{x} = (2.083, -1.271), \Delta s = -1.688, \Delta\lambda = 1.000$$

$s^{(k)} + \alpha_k \Delta s > 0, \lambda^{(k)} + \alpha_k \Delta\lambda > 0$より，$\alpha_k < 0.593$の範囲内で設定

$\rightarrow \alpha_k = 0.474$と設定.

例を通してアルゴリズムを考える

(例題：スラック変数導入)

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\ \mathit{subject\ to} \quad & g(\boldsymbol{x}) = x_1^2 - x_2 + s = 0 \\ & s \geq 0 \end{aligned}$$

(アルゴリズム)

STEP4：$\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{s}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}\right) = \left(\boldsymbol{x}^{(k)} + \alpha_k \Delta\boldsymbol{x}, \boldsymbol{s}^{(k)} + \alpha_k \Delta\boldsymbol{s}, \boldsymbol{\lambda}^{(k)} + \alpha_k \Delta\boldsymbol{\lambda}\right)$とする.

$$\boldsymbol{x}^{(1)} = (1.099, 1.398), s^{(1)} = 0.199, \lambda^{(1)} = 0.474$$

$\rho_1 = 0.1 * 0.199 * 0.474 = 0.0095$ として，STEP2へ戻る.

例を通してアルゴリズムを考える

(例題：スラック変数導入)

$$
\begin{aligned}
\min \quad & f(\boldsymbol{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\
subject\ to \quad & g(\boldsymbol{x}) = x_1^2 - x_2 + s = 0 \\
& s \geq 0
\end{aligned}
$$

(アルゴリズム)

STEP4：$\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{s}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}\right) = \left(\boldsymbol{x}^{(k)} + \alpha_k \Delta \boldsymbol{x}, \boldsymbol{s}^{(k)} + \alpha_k \Delta \boldsymbol{s}, \boldsymbol{\lambda}^{(k)} + \alpha_k \Delta \boldsymbol{\lambda}\right)$とする.

$$\boldsymbol{x}^{(1)} = (1.099, 1.398), s^{(1)} = 0.199, \lambda^{(1)} = 0.474$$

$\rho_1 = 0.1 * 0.199 * 0.474 = 0.0095$ として，STEP2へ戻る.

これを繰り返すと，$\boldsymbol{x}^{(8)} = (0.946, 0.894)$となり，最適解がほぼ得られる.

凸2次計画問題の場合

- 凸2次計画問題(Q：半正定値行列)

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} && \frac{1}{2}\boldsymbol{x}^t\mathrm{Q}\boldsymbol{x}+\boldsymbol{c}^t\boldsymbol{x} \\ &\text{subject to} && A\boldsymbol{x}=\boldsymbol{b}, \boldsymbol{x}\geq\boldsymbol{0} \end{aligned}$$

- 1次の必要条件

$$\mathrm{Q}\boldsymbol{x}^*+\boldsymbol{c}-A\boldsymbol{u}^*-\boldsymbol{v}^*=\boldsymbol{0}$$
$$v_j^*x_j^*=\rho,\quad j=1,2,\ldots,n$$
$$A\boldsymbol{x}^*=\boldsymbol{b}$$
$$x_j^*>0, v_j^*>0,\quad j=1,2,\ldots,n$$

凸2次計画問題の場合

・凸2次計画問題(Q：半正定値行列)

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^n} && \frac{1}{2}\boldsymbol{x}^t \mathrm{Q}\boldsymbol{x} + \boldsymbol{c}^t\boldsymbol{x} \\ &\text{subject to} && A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

・連立方程式(近似ではなく，正確な形で求められる)

$$\mathrm{Q}\Delta\boldsymbol{x} - A\Delta\boldsymbol{x} - \Delta\boldsymbol{v} = -\mathrm{Q}\boldsymbol{x}^{(k)} - \boldsymbol{c} + A\boldsymbol{u}^{(k)} + \boldsymbol{v}^{(k)}$$

$$v_j^{(k)}\Delta x_j + \Delta v_j x_j^{(k)} = \rho - v_j^{(k)} x_j^{(k)}, \quad j = 1,2,\dots,n$$

$$A\Delta\boldsymbol{x} = \boldsymbol{b} - A\boldsymbol{x}^{(k)}$$

・各反復において，連立方程式を解くことで，中心パスに十分近い点列が求められ，凸2次計画問題を効率的に解くことができる.

今回のまとめ

- 内点法について，アルゴリズムも含めて解説
- 内点法は，初期内点が得られれば，どのような制約付き非線形最適化問題でも適用できる汎用的な解法