計画数理学特論
～第10回：主問題における最適化法～

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

本日の講義について

- 一般的な制約付きの最適化問題

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x})$$
$$\text{subject to} \quad \boldsymbol{x}\in S$$

- S：実行可能領域(等式や不等式が混ざっていてもOK)
- 制約付き最適化問題の場合：KKT条件等を求める場合が多い
  → 複雑な連立方程式になる場合も多く，計算機で解きづらい場合も多い
- 制約なし最適化問題であれば，最急降下法やニュートン法，準ニュートン法などの手法を用いて，同じルーティンで解くことができる場合が多い
- 「制約付き」を「制約なし」にするにはどうすればよいか？
  → 制約条件を何らかの形で目的関数に組み込む

ペナルティ関数を用いた定式化

・制約式を目的関数に組み込んで，制約なし最適化問題として扱う

・制約式をどの程度違反しているかで「ペナルティ」を加えることで，
　・制約条件を違反すると目的関数の値は大きくなる
　→ 目的関数の最小化を考えると，違反はしない方が良いので，最適化を行うと制約条件を違反しないような方向へと進みやすい

Q1：違反した場合，どの程度ペナルティを課すのか？

Q2：(勾配法などを用いる場合)　制約条件をどのように連続かつ滑らかに目的関数と統合させるのか？

ペナルティ関数

- S：最適化問題の実行可能領域
- 次の条件を満たす関数$P(\boldsymbol{x})$をSのペナルティ関数とする
  - 関数$P: \mathbb{R}^n \to \mathbb{R}$は非負の値をとる連続関数
  - $\boldsymbol{x} \in S$に対して，$P(\boldsymbol{x}) = 0$
  - $\boldsymbol{x} \notin S$に対して，$P(\boldsymbol{x}) > 0$
- ペナルティ関数の例
  - 等式制約$S = \{\boldsymbol{x} \in \mathbb{R}^n | g_j(\boldsymbol{x}) = 0, j = 1,2, \dots\}$のとき，$P(\boldsymbol{x}) = \frac{1}{2}\sum_j g_j(\boldsymbol{x})^2$
  - 不等式制約$S = \{\boldsymbol{x} \in \mathbb{R}^n | g_j(\boldsymbol{x}) \le 0, j = 1,2, \dots\}$のとき，

$$P(\boldsymbol{x}) = \frac{1}{2}\sum_j \left(max\{g_j(\boldsymbol{x}), 0\}\right)^2$$

ペナルティ関数法を用いた変換

- 制約付き最適化問題の変換(ペナルティ関数と正の実数cの導入)

- 上記の問題のイメージ図

ペナルティ関数法の例題

・制約付き最適化問題

$$\min \quad e^x$$
$$\text{subject to} \quad x \geq 1$$

に対して，ペナルティ関数として$P(\boldsymbol{x}) = \frac{1}{2}\sum_j max\left\{g_j(\boldsymbol{x}), 0\right\}^2$ を用いた場合

正の実数cの値を10, 20, 30と変化させたときのペナルティ関数法の最適解は？

(解答)

ペナルティ関数法のアルゴリズム

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

STEP2：$c^{(k)}P\left(\boldsymbol{x}^{(k)}\right)$が十分に小さければ終了する.

STEP3：$\boldsymbol{x}^{(k)}$を初期点として，$q\left(c^{(k)},\boldsymbol{x}\right)$を最小化する制約なし最適化問題を解き，新たな点を$\boldsymbol{x}^{(k+1)}$とする.

STEP4：$c^{(k+1)}>c^{(k)}$を満たすパラメータの値$c^{(k+1)}$を定め，$k=k+1$として，STEP2へ戻る.

ペナルティ関数法の収束性・単調性

・無限大に発散する単調な正の数列：$\{c_k\}$ $(c_k \to \infty\ (k \to \infty))$

・$\min_{\boldsymbol{x} \in \mathbb{R}^n} q(c_k, \boldsymbol{x})$の最適解：$\boldsymbol{x}_k$

(ペナルティ関数法の収束性)

・数列$\{\boldsymbol{x}_k\}$の任意の集積点は元々の制約付き最適化問題の大域的最適解

(ペナルティ関数法の単調性)

・$c_k < c_{k+1}$ならば，$f(\boldsymbol{x}_k) < f(\boldsymbol{x}_{k+1})$

正確なペナルティ関数法

・実行可能領域が等式制約$S = \left\{ \boldsymbol{x} \in \mathbb{R}^n \middle| g_j(\boldsymbol{x}) = 0, j = 1,2, \ldots \right\}$のとき

$\rightarrow$ ペナルティ関数$P(\boldsymbol{x}) = \sum \left| g_j(\boldsymbol{x}) \right|$, $q(c, \boldsymbol{x}) = f(\boldsymbol{x}) + c \sum \left| g_j(\boldsymbol{x}) \right|$ と定義

(c：有限の正の値)

・$f(\boldsymbol{x}), g_j(\boldsymbol{x})$は連続微分可能

・点$\boldsymbol{x}^*$：制約付き最適化問題の2次の十分条件を満たす

・このとき，対応するラグランジュ乗数λ_j ($j = 1,2, \ldots$)において，

$$c > max_{j=1,2,\ldots} \lambda_j$$

を満たすcを設定すると，$\boldsymbol{x} = \boldsymbol{x}^*$は$q(c, \boldsymbol{x})$の局所最適解である.

バリア関数法

・制約式を目的関数に組み込んで，制約なし最適化問題として扱う
(この部分はペナルティ関数と同様)

・制約条件の境界付近になると途端に目的関数が悪くなる(大きくなる)ように，
「バリア」を設定する

(イメージ図)

バリア関数

- S：最適化問題の実行可能領域
- 次の条件を満たす関数$B(\boldsymbol{x})$をSのバリア関数とする
  - $B(\boldsymbol{x})$は$int(S)$上で定義された非負の値をとる連続関数
  - 点列$\{\boldsymbol{x}_k\} \subset int(S)$がSの境界の点に収束するとき，$\lim_{k\to\infty} B(\boldsymbol{x}_k) = \infty$が成立
- ペナルティ関数の例
  - 不等式制約$S = \left\{\boldsymbol{x} \in \mathbb{R}^n \middle| g_j(\boldsymbol{x}) \le 0, j = 1,2, \dots\right\}$のとき，

$$B(\boldsymbol{x}) = -\sum_j \frac{1}{g_j(\boldsymbol{x})}$$

$$B(\boldsymbol{x}) = -\sum_j \log\left(-g_j(\boldsymbol{x})\right) \quad \text{(対数バリア関数)}$$

バリア関数法を用いた変換

・バリア関数$B(\boldsymbol{x})$と目的関数$f(\boldsymbol{x})$の統合

$$r(c, \boldsymbol{x}) = f(\boldsymbol{x}) + \frac{1}{c} B(\boldsymbol{x})$$

・無限大に発散する単調な正の数例：$\{c_k\}\ \ (c_k \to \infty\ (k \to \infty))$に対して

$$\min_{\boldsymbol{x} \in \mathbb{R}^n}\ \ r(c_k, \boldsymbol{x}) \quad\quad \mathrm{s.t.}\ \ \ \boldsymbol{x} \in \mathrm{int}(S)$$

バリア関数法を用いた変換

- バリア関数$B(\boldsymbol{x})$と目的関数$f(\boldsymbol{x})$の統合

$$r(c, \boldsymbol{x}) = f(\boldsymbol{x}) + \frac{1}{c} B(\boldsymbol{x})$$

- 無限大に発散する単調な正の数例：$\{c_k\}$ $(c_k \to \infty\ (k \to \infty))$に対して

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \quad r(c_k, \boldsymbol{x}) \qquad \mathrm{s.t.} \quad \boldsymbol{x} \in \mathrm{int}(S)$$

- 直線探索を注意深く実行すれば，制約なし最適化問題と考えられる.
- 初期点は内点から選ぶ必要あり．(ただし，得られる点列$\{\boldsymbol{x}_k\}$は実行可能解)

バリア関数法のアルゴリズム

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

STEP2：$\frac{1}{c^{(k)}}B\left(\boldsymbol{x}^{(k)}\right)$が十分に小さければ終了する.

STEP3：$\boldsymbol{x}^{(k)}$を初期点として，$r\left(c^{(k)},\boldsymbol{x}\right)$を最小化する制約なし最適化問題を解き，新たな点を$\boldsymbol{x}^{(k+1)}$とする.

STEP4：$c^{(k+1)}>c^{(k)}$を満たすパラメータの値$c^{(k+1)}$を定め，$k=k+1$として，STEP2へ戻る.

バリア関数法の収束性・単調性

・無限大に発散する単調な正の数列：$\{c_k\}$ $(c_k \to \infty\ (k \to \infty))$

・$\min_{\boldsymbol{x} \in \mathbb{R}^n} r(c_k, \boldsymbol{x})$の最適解：$\boldsymbol{x}_k$

(バリア関数法の収束性)

・数列$\{\boldsymbol{x}_k\}$の任意の集積点は元々の制約付き最適化問題の大域的最適解

(バリア関数法の単調性)

・$c_k < c_{k+1}$ならば，$f(\boldsymbol{x}_k) \geq f(\boldsymbol{x}_{k+1})$

今回のまとめ

・制約付き最適化問題を制約なし最適化問題として解くための方法として,
  ・ペナルティ関数法
  ・バリア関数法

・ペナルティ関数法
  メリット：制約なし最適化の手法をダイレクトに適用可能
  デメリット：解の点列$\{\boldsymbol{x}_k\}$は基本的に実行可能領域外

・バリア関数法
  メリット：解の点列$\{\boldsymbol{x}_k\}$は基本的に実行可能領域内
  デメリット：c_kが大きくなるにつれ，境界付近で解が不安定