クレジット:
UTokyo Online Education 数理手法Ⅲ 2018 寒野善博

東京大学 THE UNIVERSITY OF TOKYO

非線形計画問題の解法の概要

☆この資料は，期末試験の範囲外☆

一般の制約付き最適化問題

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) && \text{(1a)} \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \le 0, \quad i = 1, \dots, m, && \text{(1b)} \\ & h_l(\boldsymbol{x}) = 0, \quad l = 1, \dots, r && \text{(1c)} \end{aligned}$$

を考える [*1]．ここで，目的関数 $f : \mathbb{R}^n \to \mathbb{R}$ と制約関数 $g_1, \dots, g_m, h_1, \dots, h_r : \mathbb{R}^n \to \mathbb{R}$ はすべて微分可能であるとする．このような最適化問題の解法には，逐次 2 次計画法，内点法，乗数法，勾配射影法，有効制約法，信頼領域法，フィルタ法などがある．以下では，逐次 2 次計画法と内点法の概要を説明する．なお，Python のパッケージ pyOpt や Matlab の Optimization Toolbox などでは複数の解法が利用可能であり，どの解法を用いるかは容易に選択できるため，実問題を扱う際にはいくつかの解法を試してみるのもよい．

1. 逐次 2 次計画法

逐次 2 次計画法 (SQP = sequential quadratic programming) は制約付き最適化問題に対する最も有力な手法の一つであり，多くの最適化ソルバーに組み込まれている．逐次 2 次計画法は，問題 (1) を近似する 2 次計画問題を繰り返し解くことで，問題 (1) の局所最適解を得る．

より具体的には，逐次 2 次計画法の各反復では，$\Delta\boldsymbol{x} \in \mathbb{R}^n$ を変数とする次の最適化問題を解く：

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\Delta\boldsymbol{x}^\top B_k \, \Delta\boldsymbol{x} + \nabla f(\boldsymbol{x}_k)^\top \Delta\boldsymbol{x} && \text{(2a)} \\ \text{subject to} \quad & g_i(\boldsymbol{x}_k) + \nabla g_i(\boldsymbol{x}_k)^\top \Delta\boldsymbol{x} \le 0, \quad i = 1, \dots, m, && \text{(2b)} \\ & h_l(\boldsymbol{x}_k) + \nabla h_l(\boldsymbol{x}_k)^\top \Delta\boldsymbol{x} = 0, \quad l = 1, \dots, r. && \text{(2c)} \end{aligned}$$

ここで，目的関数に現れる B_k は正定値対称行列とする [*2]．また，制約関数はすべて $\boldsymbol{x}_k$ において 1 次近似されている．問題 (2) は，2 次計画問題とよばれ，効率よく解くことができる問題である．

問題 (2) の最適解を $\Delta\boldsymbol{x}^*$ とすると，$\boldsymbol{x}_k$ は

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta\boldsymbol{x}^*$$

と更新される．このとき，$\Delta\boldsymbol{x}^* = \boldsymbol{0}$ が成り立てば，$\boldsymbol{x}_k$ が元の問題 (1) の KKT 条件を満たすことを示すことができる．

Python のライブラリ pyOpt では，SNOPT という逐次 2 次計画法のソルバーが利用できる．また，Matlab の Optimization Toolbx の関数 `fmincon` にも逐次 2 次計画法が組み込まれている．

[*1] 以下の解説では，問題 (1) は凸計画問題ではないことを前提とする．

[*2] 行列 B_{k+1} は，問題 (1) のラグランジュ関数のヘッセ行列を近似するように，準ニュートン法に準じた方法で生成する．詳細は，文献 [2, 5.4.3 節], [3, 9.2 節] などを参照されたい．

2. 内点法

内点法は，もとは線形計画問題や 2 次計画問題に対する解法として開発されたものである．その有効性から，他の問題への適用も有望視されて，非線形計画問題への拡張が進んだ．

内点法では，パラメータ $\rho > 0$ を導入して，問題 (1) を近似する次の問題を考える：

$$\text{Minimize} \quad f(\boldsymbol{x}) - \rho \sum_{i=1}^{m} \log(-g_i(\boldsymbol{x})) \tag{3a}$$

$$\text{subject to} \quad h_l(\boldsymbol{x}) = 0, \quad l = 1, \dots, r. \tag{3b}$$

ここで，$-\log(-g_i(\boldsymbol{x}))$ は $g_i(\boldsymbol{x}) < 0$ が 0 に近づくにつれて大きな値をとる関数であり，これが目的関数に加えられることで制約 $g_i(\boldsymbol{x}) \le 0$ の代わりの役割を果たしている．このような関数を，障壁関数（または，バリア関数）とよび，$\rho > 0$ を障壁パラメータとよぶ．ここで，$\rho \to 0$ とすると，問題 (3) の最適解は元の問題 (1) の近似解とみなすことができる．そこで，$\rho > 0$ を徐々に小さくしながら問題 (3) を近似的に解くことを繰り返すというのが，内点法の基本的な考え方である [*3]．

IPOPT や KNITRO という内点法のソフトウェアには，Python のインターフェースがある．また，Matlab の関数 `fmincon` には，内点法も組み込まれている．

参考文献

[1] 小島 政和，土谷 隆，水野 眞治，矢部 博：『内点法』，朝倉書店 (2001).

[2] 矢部 博：『工学基礎・最適化とその応用』，数理工学社 (2006).

[3] 山下 信雄：『非線形計画法』，朝倉書店 (2015).

（以上）

*3 詳細は，文献 [1, 第 8 章], [3, 9.3 節], [2, 5.4.4 節] などを参照されたい．