最优化方法及其 Matlab 程序设计

马昌凤

2010 年 3 月 22 日

目　录

第十一章　二次规划

二次规划是非线性优化中的一种特殊情形, 它的目标函数是二次实函数, 约束函数都是线性函数. 由于二次规划比较简单, 便于求解 (仅次于线性规划), 并且一些非线性优化问题可以转化为求解一系列的二次规划问题 (即本书第十二章所介绍的“序列二次规划法”), 因此二次规划的求解方法较早引起人们的重视, 成为求解非线性优化的一个重要途径. 二次规划的算法较多, 本章仅介绍求解等式约束凸二次规划的零空间方法和拉格朗日方法以及求解一般约束凸二次规划的有效集方法.

11.1　等式约束凸二次规划的解法

我们考虑如下的二次规划问题

$$\begin{cases} \min \dfrac{1}{2}x^T Hx + c^T x, \\ \text{s.t. } Ax = b, \end{cases} \tag{11.1}$$

其中 $H \in \mathbb{R}^{n\times n}$ 对称正定, $A \in \mathbb{R}^{m\times n}$ 行满秩, $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$. 本节我们介绍两种求解问题 (11.1) 的数值方法, 即零空间方法和值空间方法（通常称为拉格朗日方法）.

11.1.1　零空间方法

设 x_0 满足 $Ax_0 = b$. 记 A 的零空间为

$$\mathcal{N}(A) = \left\{z \in \mathbb{R}^n \,|\, Az = 0\right\},$$

则问题 (11.1) 的任一可行点 x 可表示成 $x = x_0 + z$, $z \in \mathcal{N}(A)$. 这样, 问题

(11.1) 可等价变形为

$$\begin{cases} \min\ \dfrac{1}{2}z^THz + z^T(c+Hx_0), \\ \text{s.t.}\ \ Az = 0. \end{cases} \tag{11.2}$$

令 $Z \in \mathbb{R}^{n\times(n-m)}$ 是 $\mathcal{N}(A)$ 的一组基组成的矩阵, 那么, 对任意的 $d \in \mathbb{R}^{n-m}$, 有 $z = Zd \in \mathcal{N}(A)$. 于是问题 (11.2) 变为无约束优化问题

$$\min \frac{1}{2}d^T(Z^THZ)d + d^T[Z^T(c+Hx_0)]. \tag{11.3}$$

容易发现, 当 H 半正定时, Z^THZ 也是半正定的. 此时, 若 d^* 是 (11.3) 的稳定点, d^* 也是 (11.3) 的全局极小点, 同时 $x^* = x_0 + Zd^*$ 是 (11.1) 的全局极小点, $\lambda^* = A^+(Hx^*+c)$ 是相应的拉格朗日乘子, 其中 A^+ 是矩阵 A 的 Penrose 广义逆. 由于这种方法是基于约束函数的系数矩阵的零空间, 因此把它称之为零空间方法.

余下的问题就是如何确定可行点 x_0 和零空间 $\mathcal{N}(A)$ 的基矩阵 Z. 有多种方法来确定这样的 x_0 和 Z. 我们在此介绍 1974 年 Gill 和 Murry 所提

出的一种方法, 即先对 A^T 作 QR 分解

$$A^T = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1,\ Q_2]\begin{bmatrix} R \\ 0 \end{bmatrix}, \tag{11.4}$$

其中, Q 是一个 n 阶正交阵, R 是一个 m 上三角阵, $Q_1 \in \mathbb{R}^{n\times m}$, $Q_2 \in \mathbb{R}^{n\times(n-m)}$. 那么确立 x_0 和 Z 为

$$x_0 = Q_1 R^{-T} b, \quad Z = Q_2, \tag{11.5}$$

同时有

$$A^+ = Q_1 R^{-T}. \tag{11.6}$$

下面写出零空间方法的算法步骤:

算法 11.1 (零空间方法)

步 0　数据准备. 确定矩阵 H, A 和向量 c, b.

步 1　由 (11.4) 对 A^T 进行 QR 分解, 得 Q_1, Q_2 和 R.

步 2　按 (11.5) 计算可行点 x_0 和零空间 $\mathcal{N}(A)$ 的基矩阵 Z。

步 3　求解无约束优化子问题 (11.3) 得解 d^*.

步 4　计算全局极小点 $x^* = x_0 + Zd^*$ 和相应的拉格朗日乘子 $\lambda^* = A^+(Hx^* + c)$, 其中 A^+ 由 (11.6) 确定.

11.1.2　拉格朗日方法及其 Matlab 程序

下面我们来推导用拉格朗日乘子法解问题 (11.1) 的求解公式.

首先写出拉格朗日函数:

$$L(x,\lambda) = \frac{1}{2}x^T Hx + c^T x - \lambda^T (Ax - b), \tag{11.7}$$

令

$$\nabla_x L(x,\lambda) = 0, \quad \nabla_\lambda L(x,\lambda) = 0,$$

得到方程组

$$\begin{aligned} Hx - A^T\lambda &= -c, \\ -Ax \qquad &= -b. \end{aligned}$$

将上述方程组写成分块矩阵形式:

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ -b \end{bmatrix}. \tag{11.8}$$

我们称上述方程组的系数矩阵

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix}$$

为拉格朗日矩阵.

下面的定理给出了线性方程组 (11.8) 有唯一解的充分条件.

定理 11.1 设 $H \in \mathbb{R}^{n\times n}$ 对称正定, $A \in \mathbb{R}^{m\times n}$ 行满秩. 若在问题 (11.1) 的解 x^* 处满足二阶充分条件, 即

$$d^T H d > 0,\ \ \forall\, d \in \mathbb{R}^n,\ d \neq 0,\ Ad = 0,$$

则线性方程组 (11.9) 的系数矩阵非奇异, 即方程组 (11.9) 有唯一解.

证 设 (d,ν) 是下面的齐次线性方程组的解:

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} d \\ \nu \end{bmatrix} = 0, \tag{11.9}$$

即

$$Hd - A^T\nu = 0, \quad Ad = 0.$$

故

$$d^THd = d^TA^T\nu = 0, \quad Ad = 0.$$

于是由二阶充分性条件必有 $d=0$. 从而

$$A^T\nu = Hd = 0.$$

注意到 A 行满秩, 故必有 $\nu = 0$. 由此可知, 齐次线性方程组 (11.9) 只有零解, 因此其系数矩阵必然非奇异. 证毕. □

下面我们来导出方程 (11.8) 的求解公式. 根据定理 11.1, 拉格朗日矩

阵必然是非奇异的, 故可设其逆为

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix}^{-1} = \begin{bmatrix} G & -B^T \\ -B & C \end{bmatrix}.$$

由恒等式

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} G & -B^T \\ -B & C \end{bmatrix} = \begin{bmatrix} I_n & 0_{n\times m} \\ 0_{m\times n} & I_m \end{bmatrix}$$

可得

$$HG + A^T B = I_n, \quad -HB^T - A^T C = 0_{n\times m},$$

$$-AG = 0_{m\times n}, \qquad AB^T = I_m.$$

于是由上述 4 个等式得到矩阵 G, B, C 的表达式

$$G = H^{-1} - H^{-1}A^T(AH^{-1}A^T)^{-1}AH^{-1}, \tag{11.10}$$

$$B = (AH^{-1}A^T)^{-1}AH^{-1}, \tag{11.11}$$

$$C = -(AH^{-1}A^T)^{-1}. \tag{11.12}$$

因此, 由 (11.8) 可得解的表达式

$$\begin{bmatrix} \bar{x} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} G & -B^T \\ -B & C \end{bmatrix} \begin{bmatrix} -c \\ -b \end{bmatrix} = \begin{bmatrix} -Gc + B^T b \\ Bc - Cb \end{bmatrix}, \tag{11.13}$$

其中 G, B, C 分别由 (11.10), (11.11) 和 (11.12) 给出.

下面给出 $\bar{x}$ 和 $\bar{\lambda}$ 的另一种等价表达式. 设 x_k 是问题 (11.1) 的任一可行点, 即 x_k 满足 $Ax_k = b$. 而在此点处目标函数的梯度为 $g_k = \nabla f(x_k) =$

$Hx_k + c$. 利用 x_k 和 g_k, 可将 (11.13) 改写为

$$\begin{bmatrix} \bar{x} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} x_k - Gg_k \\ Bg_k \end{bmatrix}. \tag{11.14}$$

下面我们给出求解等式约束二次规划拉格朗日方法的 Matlab 程序.

程序 11.1 本程序用拉格朗日方法求解等式约束条件的二次规划问题.

```
function [x,lam,fval]=qlag(H,A,b,c)
% 功能: 用拉格朗日方法求解等式约束二次规划:
%             min f(x)=0.5*x'Hx+c'x, s.t. Ax=b
%输入: H,c分别是目标函数的矩阵和向量, A,b分别是
%           约束条件中的矩阵和向量
%输出: (x, lam) 是 KT 点, fval 是最优值.
IH=inv(H);
```

```
AHA=A*IH*A';
IAHA=inv(AHA);
AIH=A*IH;
G=IH-AIH'*IAHA*AIH;
B=IAHA*AIH;
C=-IAHA;
x=B'*b-G*c;
lam=B*c-C*b;
fval=0.5*x'*H*x+c'*x;
```

我们利用上述程序求解一个二次规划问题.

例 11.1 利用程序 11.1 求解下列问题

$$\min x_1^2+2x_2^2+x_3^2-2x_1x_2+x_3,$$

$$\text{s.t.}\ \ x_1+x_2+x_3=4,$$

$$2x_1-x_2+x_3=2.$$

解 容易写出

$$H=\begin{bmatrix}2 & -2 & 0\\ -2 & 4 & 0\\ 0 & 0 & 2\end{bmatrix},\ c=\begin{bmatrix}0\\0\\1\end{bmatrix},\ A=\begin{bmatrix}1 & 1 & 1\\ 2 & -1 & 1\end{bmatrix},\ b=\begin{bmatrix}4\\2\end{bmatrix}.$$

在 Matlab 命令窗口依次输入:

```
H=[2 -2 0;-2 4 0; 0 0 2];
c=[0 0 1]';
A=[1 1 1;2 -1 1];
b=[4 2]';
[x,lam]=qlag(H,A,b,c)
```

得到

```
x =

    1.9091
```

```
        1.9545
        0.1364
lam =
         2.6364
        -1.3636
fval =
         3.9773
```

11.2　一般凸二次规划的有效集方法

考虑一般二次规划

$$\begin{cases} \min \ \dfrac{1}{2}x^T Hx + c^T x, \\ \text{s.t.} \ \ a_i^T x - b_i = 0, \ i \in E = \{1, \cdots, l\}, \\ \qquad a_i^T x - b_i \geq 0, \ i \in I = \{l+1, \cdots, m\}, \end{cases} \tag{11.15}$$

其中 H 是 n 阶对称阵. 记 $I(x^*) = \{i \mid a_i^T x^* - b_i = 0,\ i \in I\}$, 下面的定理给出了问题 (11.15) 的一个最优性充要条件, 其证明可参见文献 [1].

定理 11.2 x^* 是二次规划问题 (11.15) 的局部极小点当且仅当

(1) 存在 $\lambda^* \in \mathbb{R}^m$, 使得

$$\begin{cases} Hx^* + c - \sum\limits_{i\in E} \lambda_i^* a_i - \sum\limits_{i\in I} \lambda_i^* a_i = 0, \\ a_i^T x^* - b_i = 0,\ i \in E, \\ a_i^T x^* - b_i \ge 0,\ i \in I, \\ \lambda_i^* \ge 0,\ i \in I;\ \lambda_i^* = 0,\ i \in I \backslash I(x^*). \end{cases}$$

(2) 记

$$\begin{aligned} \mathcal{S} \ = \ \{d \in \mathbb{R}^n \backslash \{0\} \mid & d^T a_i = 0, i \in E;\ d^T a_i \ge 0, i \in I(x^*); \\ & d^T a_i = 0, i \in I(x^*) \text{ 且 } \lambda_i^* > 0\}. \end{aligned}$$

则对于任意的 $d \in \mathcal{S}$, 均有 $d^T H d \ge 0$.

容易发现, 问题 (11.15) 是凸二次规划的充要条件是 H 半正定. 此时, 定理 11.2 的第二部分自然满足. 注意到凸优化问题的局部极小点也是全局极小点的性质, 我们有下面的定理:

定理 11.3 x^* 是凸二次规划的全局极小点的充要条件是 x^* 满足 KT 条件, 即存在 $\lambda^* \in \mathbb{R}^m$, 使得

$$\begin{cases} Hx^* + c - \sum\limits_{i\in E} \lambda_i^* a_i - \sum\limits_{i\in I} \lambda_i^* a_i = 0, \\ a_i^T x^* - b_i = 0,\ i \in E, \\ a_i^T x^* - b_i \geq 0,\ i \in I, \\ \lambda_i^* \geq 0,\ i \in I;\ \lambda_i^* = 0,\ i \in I \backslash I(x^*). \end{cases}$$

下面我们介绍求解一般凸二次规划问题的有效集方法及其 Matlab 实现.

11.2.1　有效集方法的理论推导

首先引入下面的定理, 它是有效集方法理论基础, 其证明可参见文献 [2].

定理 11.4 设 x^* 是一般凸二次规划问题 (11.15) 的全局极小点, 且在 x^* 处的有效集为 $S(x^*) = E \cup I(x^*)$, 则 x^* 也是下列等式约束凸二次规划

$$\begin{cases} \min\ \dfrac{1}{2}x^T H x + c^T x, \\ \text{s.t.}\ \ a_i^T x - b_i = 0,\ i \in S(x^*) \end{cases} \tag{11.16}$$

的全局极小点.

从上述定理可以发现, 有效集方法的最大难点是事先一般不知道有效集 $S(x^*)$, 因此只有想办法构造一个集合序列去逼近它. 即, 从初始点 x_0 出发, 计算有效集 $S(x_0)$, 解对应的等式约束子问题. 重复这一做法, 得到有效集序列 $\{S(x_k)\}$, $k = 0, 1, \cdots$, 使之 $S(x_k) \to S(x^*)$, 以获得原问题的最优解.

基于上述定理, 我们分 4 步来介绍有效集方法的算法原理和实施步骤.

第 1 步. 形成子问题并求出搜索方向 d_k. 设 x_k 是问题 (11.15) 的一个可行点, 据此确定相应的有效集 $S_k = E \cup I(x_k)$, 其中 $I(x_k) = \{i | a_i^T x_k - b_i = 0, i \in I\}$. 求解相应的子问题

$$\begin{cases} \min \ \dfrac{1}{2} x^T H x + c^T x, \\ \text{s.t.} \ \ a_i^T x - b_i = 0, \ i \in S_k. \end{cases} \tag{11.17}$$

上述问题等价于

$$\begin{cases} \min \ q_k(d) = \dfrac{1}{2} d^T H d + g_k^T d, \\ \text{s.t.} \ \ a_i^T d = 0, \ i \in S_k, \end{cases} \tag{11.18}$$

其中 $x = x_k + d$, $g_k = Gx_k + c$. 设求出问题 (11.18) 的全局极小点为 d_k, λ_k 是对应的拉格朗日乘子.

第 2 步. 进行线搜索确定步长因子 α_k. 假设 $d_k \neq 0$, 分两种情形讨论.

(1) 若 x_k+d_k 是问题 (11.15) 的可行点, 即

$$a_i^T(x_k+d_k)-b_i=0,\ i\in E \text{ 及 } a_i^T(x_k+d_k)-b_i\geq 0,\ i\in I.$$

则令 $\alpha_k=1$, $x_{k+1}=x_k+d_k$.

(2) 若 x_k+d_k 不是问题 (11.15) 的可行点, 则通过线搜索求出下降最好的可行点. 注意到目标函数是凸二次函数, 那么这一点应该在可行域的边界上达到. 因此只要求出满足可行条件的最大步长 α_k 即可.

当 $i\in S_k$ 时, 对于任意的 $\alpha_k\geq 0$, 都有 $a_i^Td_k=0$ 和 $a_i^T(x_k+\alpha_kd_k)=a_i^Tx_k=b_i$, 此时, $\alpha_k\geq 0$ 不受限制. 当 $i\notin S_k$ 时, 即第 i 个约束是严格的不等式约束, 此时要求 α_k 满足 $a_i^T(x_k+\alpha_kd_k)\geq b_i$, 即

$$\alpha_ka_i^Td_k\geq b_i-a_i^Tx_k,\ \ i\notin S_k.$$

注意到上式右端非正, 故当 $a_i^Td_k\geq 0$ 时, 上式恒成立. 而当 $a_i^Td_k<0$ 时, 由上式可解得

$$\alpha_k\leq\frac{b_i-a_i^Tx_k}{a_i^Td_k}.$$

故有

$$\alpha_k = \bar{\alpha}_k = \min\left\{\frac{b_i - a_i^T x_k}{a_i^T d_k} \,\middle|\, a_i^T d_k < 0\right\}.$$

合并 (1) 和 (2) 可得

$$\alpha_k = \min\{1, \bar{\alpha}_k\}. \tag{11.19}$$

第 3 步. 修正 S_k. 当 $\alpha_k = 1$, 有效集不变, 即 $S_{k+1} := S_k$. 而当 $\alpha_k < 1$ 时,

$$\alpha_k = \bar{\alpha}_k = \frac{b_{i_k} - a_{i_k}^T x_k}{a_{i_k}^T d_k},$$

故 $a_{i_k}^T(x_k + \alpha_k d_k) = b_{i_k}$, 因此在 x_{k+1} 处增加了一个有效约束, 即 $S_{k+1} := S_k \cup \{i_k\}$.

第 4 步. 考虑 $d_k = 0$ 的情形. 此时 x_k 是问题 (11.17) 的全局极小点. 若这时对应的不等式约束的拉格朗日乘子均为非负, 则 x_k 也是问题 (11.15) 的全局极小点, 迭代终止. 否则, 如果对应的不等式约束的拉格朗日乘子有负的分量, 那么需要重新寻找一个下降可行方向.

设 $\lambda_{j_k} < 0,\ j_k \in I(x_k)$. 现在要求一个下降可行方向 d_k, 满足 $g_k^T d_k < 0$ 且 $a_j^T d_k = 0,\ \forall j \in E;\ a_j^T d_k \geq 0,\ \forall j \in I(x_k)$. 为简便计, 按下述方式选取 d_k:

$$a_{j_k}^T(x_k + d_k) > b_{j_k},$$

$$a_j^T(x_k + d_k) = b_j,\ \forall j \in S_k, j \neq j_k,$$

即

$$\begin{cases} a_{j_k}^T d_k > 0, \\ a_j^T d_k = 0,\ \forall j \in S_k, j \neq j_k, \end{cases} \tag{11.20}$$

另一方面, 注意到 x_k 是子问题 (11.17) 的全局极小点, 故有

$$Hx_k + c - \sum_{i \in S_k} \lambda_i^k a_i = 0,$$

即

$$g_k = A_k \lambda_k,$$

其中

$$A_k = (a_i)_{i\in S_k}, \quad \lambda_k = (\lambda_i^k)_{i\in S_k}.$$

从而, $g_k^T d_k = \lambda_k^T A_k^T d_k$. 由 (11.20) 知

$$A_k^T d_k = \sum_{j\in S_k} (a_j^T d_k)\, e_j = (a_{j_k}^T d_k)\, e_{j_k},$$

于是有

$$g_k^T d_k = \lambda_k^T (a_{j_k}^T d_k)\, e_{j_k} = \lambda_{j_k}^k (a_{j_k}^T d_k) < 0.$$

上式表明, 由 (11.20) 确定的 d_k 是一个下降可行方向. 因此, 令 $S_k' = S_k\backslash\{j_k\}$, 则修正后的子问题

$$\begin{cases} \min\ \ q_k(d) = \dfrac{1}{2} d^T H d + g_k^T d, \\ \text{s.t.}\ \ a_i^T d = 0,\ i\in S_k' \end{cases}$$

的全局极小点必然是原问题的一个下降可行方向.

11.2.2 有效集方法的算法步骤

经过上面的分析和推导, 我们现在可以写出有效集方法的算法步骤.

算法 11.2 (**有效集方法**)

步 0 选取初值. 给定初始可行点 $x_0 \in \mathbb{R}^n$, 令 $k := 0$.

步 1 解子问题. 确定相应的有效集 $S_k = E \cup I(x_k)$. 求解求解子问题

$$\begin{cases} \min \ q_k(d) = \frac{1}{2}d^T H d + g_k^T d, \\ \text{s.t.} \ a_i^T d = 0, \ i \in S_k, \end{cases}$$

得极小点 d_k 和拉格朗日乘子向量 λ_k. 若 $d_k \neq 0$ 转步 3; 否则, 转步 2.

步 2 检验终止准则. 计算拉格朗日乘子

$$\lambda_k = B_k g_k,$$

其中

$$g_k = Hx_k + c, \quad B_k = (A_k H^{-1} A_k^T)^{-1} A_k H^{-1}, \quad A_k = (a_i)_{i \in S_k}.$$

令

$$(\lambda_k)_t = \min_{i \in I(x_k)} \{(\lambda_k)_i\}.$$

若 $(\lambda_k)_t \geq 0$，则 x_k 是全局极小点，停算. 否则，若 $(\lambda_k)_t < 0$，则令 $S_k := S_k \backslash \{t\}$，转步 1.

步 3 确定步长 α_k. 令 $\alpha_k = \min\{1, \bar{\alpha}_k\}$，其中

$$\bar{\alpha}_k = \min_{i \notin S_k} \left\{ \frac{b_i - a_i^T x_k}{a_i^T d_k} \,\middle|\, a_i^T d_k < 0 \right\}.$$

令 $x_{k+1} := x_k + \alpha_k d_k$.

步 4 若 $\alpha_k = 1$，则令 $S_{k+1} := S_k$；否则，若 $\alpha_k < 1$，则令 $S_{k+1} = S_k \cup \{j_k\}$，其中 j_k 满足

$$\bar{\alpha}_k = \frac{b_{j_k} - a_{j_k}^T x_k}{a_{j_k}^T d_k}.$$

步 5 令 $k := k + 1$，转步 1.

下面给出算法 11.2 的收敛性定理.

定理 11.5 假设问题 (11.15) 中的矩阵 H 对称正定. 若在算法 11.2 每步迭代中的矩阵

$$A_k = (a_i)_{i \in S_k}$$

列满秩, 且 $\alpha_k \neq 0$, 则算法 11.2 在有限步之内得到问题 (11.15) 的全局极小点.

证 注意到, 若 $d_k = 0$, 则 x_k 是子问题 (11.17) 的 KT 点和全局极小点. 若 $d_k \neq 0$ 且 $\alpha_k = 1$, 则 $S_{k+1} = S_k$, 这时关于 x_{k+1} 的子问题仍为 (11.17), 所以, x_{k+1} 是 (11.17) 的全局极小点. 只有当 $\alpha_k < 1$ 时, x_{k+1} 才不是 (11.17) 的全局极小点, 但这时要加进一个约束, 形成新的子问题. 这样的过程最多连续 n 次, 因为这时子问题至少有 n 个等式约束, 从而 x_k 为唯一的可行点, 因而是对应的子问题 (11.17) 的全局极小点.

另一方面, 子问题 (11.17) 虽然约束条件在变化, 但其目标函数与原问题 (11.15) 是一致的. 由于 $\alpha_k \neq 0$, 故每次迭代目标函数值减少. 再注意到 H 正定, 因此子问题的全局极小点是唯一的, 从而不会出现子问题的全局极小点被两个不同的 x_k 达到的情形. 而约束个数的有限性保证了有效集

S_k 不同个数的有限性. 因此, 不失一般性, 设有效集 S_k 不同个数为 N_0, 那么不同子问题的个数也为 N_0. 故而, 迭代至多在 $N_0 n$ 步之后, x_k 遍历所有子问题的全局极小点. 由定理 11.4 知, 算法 11.2 在有限步之内达到问题 (11.15) 的全局极小点. $\square$

例 11.2 用有效集方法求解下列二次规划问题

$$
\begin{cases}
\min\ \ f(x) = x_1^2 - x_1x_2 + 2x_2^2 - x_1 - 10x_2, \\
\text{s.t.}\ \ -3x_1 - 2x_2 \geq -6, \\
\qquad\quad x_1 \geq 0,\ x_2 \geq 0.
\end{cases}
$$

解 首先确定矩阵 H 和向量 c:

$$
H = \begin{bmatrix} 2 & -1 \\ -1 & 4 \end{bmatrix}, \quad c = \begin{bmatrix} -1 \\ -10 \end{bmatrix}.
$$

取初始可行点 $x_0 = (0,0)^T$. 在 x_0, 有效集 $S_0 = \{2,3\}$. 求解相应的子

问题

$$\begin{cases} \min\ d_1^2 - d_1 d_2 + 2d_2^2 - d_1 - 10d_2, \\ \text{s.t.}\ d_1 = 0,\ \ d_2 = 0. \end{cases}$$

得解 $d_0 = (0,0)^T$. 因此, x_0 是相应的子问题 (11.17) 的最优解. 计算拉格朗日乘子. 由 $S_0 = \{2,3\}$ 知

$$A_0 = (a_i)_{i \in S_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad g_0 = Hx_0 + c = \begin{bmatrix} -1 \\ -10 \end{bmatrix}.$$

故

$$\begin{bmatrix} (\lambda_0)_2 \\ (\lambda_0)_3 \end{bmatrix} = [(A_0 H^{-1} A_0^T)^{-1} A_0 H^{-1}] g_0 = \begin{bmatrix} -1 \\ -10 \end{bmatrix}.$$

由此可知, x_0 不是所求问题的最优解.

将 $(\lambda_0)_3 = -10$ 对应的约束, 即原问题的第 3 个约束从有效集 S_0 中

去掉, 置 $S_0=\{2\}$, 再解相应的子问题:

$$\begin{cases} \min\ d_1^2-d_1d_2+2d_2^2-d_1-10d_2, \\ \text{s.t.}\ d_1=0. \end{cases}$$

得解 $d_0=\left(0,\dfrac{5}{2}\right)^T$.

由于 $d_0\neq 0$, 需要计算步长 α_0. 注意到

$$\begin{aligned}\bar{\alpha} &= \min\left\{\frac{b_i-a_i^Tx_0}{a_i^Td_0}\middle| i\notin S_0, a_i^Td_0<0\right\} \\ &= \frac{-6}{(-3,-2)\cdot(0,\ 2.5)^T}=\frac{-6}{-5}=1.2.\end{aligned}$$

故 $\alpha_0=\min\{1,\bar{\alpha}\}=1$. 令

$$x_1=x_0+\alpha_0d_0=(0,0)^T+1\cdot\left(0,\frac{5}{2}\right)^T=\left(0,\frac{5}{2}\right)^T.$$

计算出 $g_1=\nabla f(x_1)=\left(-\dfrac{7}{2},\ 0\right)^T$.

因 $\alpha_0 = 1$, 置 $S_1 = \{2\} (= S_0)$. 在 x_1 处, 计算相应的拉格朗日乘子. 此时 $A_1 = (1, 0)$, 那么,

$$(\lambda_1)_2 = [(A_1 H^{-1} A_1^T)^{-1} A_1 H^{-1}] g_1 = -\frac{7}{2}.$$

由于 $(\lambda_1)_2 < 0$, 故 x_1 不是问题的最优解. 于是将指标 2 从 S_1 中剔除掉, 则更新后的 $S_1 = \emptyset$. 再解相应的子问题

$$\min\ d_1^2 - d_1 d_2 + 2d_2^2 - \frac{7}{2} d_1,$$

得解向量 $d_1 = \left(2, \dfrac{1}{2}\right)^T$.

由于 $d_1 \neq 0$, 需要计算步长 α_1. 注意到

$$\begin{aligned} \bar{\alpha} &= \min \left\{ \frac{b_i - a_i^T x_1}{a_i^T d_1} \middle| i \notin S_1, a_i^T d_1 < 0 \right\} \\ &= \frac{-6 - (-3, -2) \cdot (0,\ 2.5)^T}{(-3, -2) \cdot (2,\ 0.5)^T} = \frac{1}{7}. \end{aligned}$$

故

$$\alpha_1 = \min\{1, \bar{\alpha}\} = \frac{1}{7}.$$

令

$$x_2 = x_1 + \alpha_1 d_1 = \left(0,\ \frac{5}{2}\right)^T + \frac{1}{7}\left(2,\ \frac{1}{2}\right)^T = \left(\frac{2}{7},\ \frac{18}{7}\right)^T.$$

计算出 $g_2 = \nabla f(x_2) = (-3,\ 0)^T$.

在 x_2 处, 第一个约束是有效约束, 即 $S_2 = \{1\}$, 解相应的子问题:

$$\begin{cases} \min\ \ d_1^2 - d_1 d_2 + 2d_2^2 - 3d_1, \\ \text{s.t.}\ \ 3d_1 - 2d_2 = 0. \end{cases}$$

得解向量 $d_2 = \left(\frac{3}{14}, -\frac{9}{28}\right)^T$. 计算步长 α_2. 因

$$\begin{aligned}\bar{\alpha} &= \min\left\{\frac{b_i - a_i^T x_2}{a_i^T d_2}\Big| i \notin S_2, a_i^T d_2 < 0\right\} \\ &= \frac{0-(0,1)\cdot\left(\frac{2}{7}, \frac{18}{7}\right)^T}{(0,1)\cdot\left(\frac{3}{14}, -\frac{9}{28}\right)^T} = \frac{-\frac{18}{7}}{-\frac{9}{28}} = 8.\end{aligned}$$

故

$$\alpha_2 = \min\{1, \bar{\alpha}\} = 1.$$

令

$$x_3 = x_2 + \alpha_2 d_2 = \left(\frac{2}{7}, \frac{18}{7}\right)^T + 1\cdot\left(\frac{3}{14}, -\frac{9}{28}\right)^T = \left(\frac{1}{2}, \frac{9}{4}\right)^T.$$

计算出 $g_3 = \nabla f(x_3) = \left(-\frac{9}{4}, -\frac{3}{2}\right)^T$.

在点 x_3 处, 计算相应的拉格朗日乘子. 此时 $S_3 = \{1\}$, $A_3 = (-3, -2)$,

那么,

$$\lambda_1^3 = [(A_3 H^{-1} A_3^T)^{-1} A_3 H^{-1}] g_3 = \frac{3}{4} > 0.$$

因此 $x_3 = \left(\frac{1}{2}, \frac{9}{4}\right)^T$ 是所求的最优解.

11.3　有效集方法的 Matlab 程序

由于有效集方法是求解凸二次规划问题的一种值得推荐的方法, 本节给出有效集方法的 Matlab 程序. 在实际使用有效集方法求解凸二次规划时, 一般用渐进有效集约束指标集代替有效集约束指标集, 即取 $\{i \in I \mid a_i^T x - b_i \le \varepsilon\}$ 近似代替 $I(x^*)$, 其中 $\varepsilon > 0$ 是比较小的常数. 这样做的好处是使得迭代步长不至于太短. 此外, 算法 11.2 还需要确立一个初始可行点. 可采用下述方法: 给出一个初始估计点 $\bar{x} \in \mathbb{R}^n$, 定义下列线性

规划

$$\begin{cases} \min\ e^T z, \\ \text{s.t.}\ a_i^T x + \tau_i z_i - b_i = 0,\ i \in E = \{1, \cdots, l\}, \\ \qquad a_i^T x + z_i - b_i \geq 0,\ i \in I = \{l+1, \cdots, m\}, \\ \qquad z_1 \geq 0, \cdots, z_m \geq 0, \end{cases} \tag{11.21}$$

其中, $e = (1, \cdots, 1)^T$, $\tau_i = -\text{sign}(a_i^T \bar{x} - b_i)$, $i \in E$. 问题 (11.21) 的一个初始可行点为

$$x = \bar{x},\ \ z_i = |a_i^T \bar{x} - b_i|\,(i \in E);\ z_i = \max\{b_i - a_i^T \bar{x},\ 0\}\,(i \in I).$$

不难证明, 如果 $\tilde{x}$ 是问题 (11.15) 的可行点, 那么 $(\tilde{x}, 0)$ 是子问题 (11.21) 的最优解. 反之, 如果问题 (11.15) 有可行点, 则问题 (11.21)的最优值为 0, 从而子问题 (11.21) 的任何一个解产生问题 (11.15) 的一个可行点.

下面给出用有效集方法求解一般凸二次规划问题的 Matlab 程序, 在某种意义下, 该程序是通用的.

程序 11.2 本程序主要适用于求解一般约束条件下的凸二次规划问题.

```
function [x,lamk,exitflag,output]=qpact(H,c,Ae,be,Ai,bi,x0)
%功能: 用有效集方法解一般约束二次规划问题:
%    min f(x)=0.5*x'*H*x+c'*x,
%    s.t.  a'_i*x-b_i=0,(i=1,...,l),
%          a'_i*x-b_i>=0,(i=l+1,...,m)
%输入: x0是初始点, H, c分别是目标函数二次型矩阵和向量;
%    Ae=(a_1,...,a_l)',  be=(b_1,...,b_l)';
%    Ai=(a_{l+1},...,a_m), bi=(b_{l+1},...,b_m)'.
%输出: x是最优解, lambda是对应的乘子向量;
%       output是结构变量, 输出极小值f(x), 迭代次数k等
%       信息, exitflag是算法终止类型
%%%%%%%%%%%%%%%%% 主程序开始  %%%%%%%%%%%%%%%%%
% 初始化
```

```
epsilon=1.0e-9; err=1.0e-6;
k=0; x=x0;  n=length(x);  kmax=1.0e3;
ne=length(be); ni=length(bi); lamk=zeros(ne+ni,1);
index=ones(ni,1);
for (i=1:ni)
    if(Ai(i,:)*x>bi(i)+epsilon), index(i)=0; end
end
%算法主程序
while (k<=kmax)
    %求解子问题
    Aee=[];
    if(ne>0), Aee=Ae; end
    for(j=1:ni)
        if(index(j)>0), Aee=[Aee; Ai(j,:)]; end
    end
```

```
gk=H*x+c;
[m1,n1] = size(Aee);
[dk,lamk]=qsubp(H,gk,Aee,zeros(m1,1));
if(norm(dk)<=err)
    y=0.0;
    if(length(lamk)>ne)
        [y,jk]=min(lamk(ne+1:length(lamk)));
    end
    if(y>=0)
        exitflag=0;
    else
        exitflag=1;
        for(i=1:ni)
            if(index(i)&(ne+sum(index(1:i)))==jk)
                index(i)=0; break;
```

```
                end
            end
        end
        k=k+1;
    else
        exitflag=1;
        %求步长
        alpha=1.0; tm=1.0;
        for(i=1:ni)
            if((index(i)==0)&(Ai(i,:)*dk<0))
                tm1=(bi(i)-Ai(i,:)*x)/(Ai(i,:)*dk);
                if(tm1<tm)
                    tm=tm1; ti=i;
                end
            end
```

```
            end
            alpha=min(alpha,tm);
            x=x+alpha*dk;
            %修正有效集
            if(tm<1), index(ti)=1; end
        end
        if(exitflag==0), break; end
        k=k+1;
    end
    output.fval=0.5*x'*H*x+c'*x;
    output.iter=k;
    %%%%%%%% 求解子问题 %%%%%%%%%%%%%%%%
    function [x,lambda]=qsubp(H,c,Ae,be)
    ginvH=pinv(H);
    [m,n]=size(Ae);
```

```
if(m>0)
    rb=Ae*ginvH*c + be;
    lambda=pinv(Ae*ginvH*Ae')*rb;
    x=ginvH*(Ae'*lambda-c);
else
    x=-ginvH*c;
    lambda=0;
end
```

注 (1) 关于上述程序, 在子函数 qsubp 中, 使用的是广义逆 (pinv 是 Matlab 软件内置的求广义逆的函数), 这样不仅可计算 H 是奇异阵的情形, 同时保证了计算的数值稳定性. 另外, 该子函数还包含了子问题为无约束二次规划时的解. (2) 使用程序 11.2 时, 需要用户提供所有问题的目标函数和约束函数的有关数据, 可通过编制一个 m 文件来解决.

例 11.3 利用程序 11.2 重新求解例 11.2, 即

$$\begin{cases} \min\ \ f(x) = x_1^2 - x_1x_2 + 2x_2^2 - x_1 - 10x_2, \\ \text{s.t.}\ \ -3x_1 - 2x_2 \geq -6, \\ \qquad\quad x_1 \geq 0,\ x_2 \geq 0. \end{cases}$$

解 首先确定有关数据:

$$H = \begin{bmatrix} 2 & -1 \\ -1 & 4 \end{bmatrix},\ \ c = \begin{bmatrix} -1 \\ -10 \end{bmatrix},\ \ \text{Ae} = [\,],$$

$$\text{be} = [\,],\ \ \text{Ai} = \begin{bmatrix} -3 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},\ \ \text{bi} = \begin{bmatrix} -6 \\ 0 \\ 0 \end{bmatrix}.$$

编制一个利用上述数据调用程序 11.2 的函数文件 callqpact.m

```
function callqpact
H=[2 -1; -1 4];
c=[-1 -10]';
Ae=[ ];  be=[ ];
Ai=[-3 -2; 1 0; 0 1];
bi=[-6 0 0]';
x0=[0 0]';
[x,lambda,exitflag,output]=qpact(H,c,Ae,be,Ai,bi,x0)
```

然后在 Matlab 命令窗口键入 callqpact, 回车即得结果

```
x =
    0.5000
    2.2500
lambda =
    0.7500
exitflag =
```

```
        0
output =
        fval: -13.7500
        iter: 8
```

可以看出, 上述结果跟例 11.2 用手算是一致的.

例 11.4 利用程序 11.2 求解下列二次规划问题

$$\begin{cases} \min\ \ f(x)=\dfrac{1}{2}x_1^2-x_1x_2+x_2^2-6x_1-2x_2, \\ \text{s.t.}\ \ -2x_1-x_2\ge -3, \\ \qquad\quad x_1-x_2\ge -1, \\ \qquad\quad -x_1-2x_2\ge -2, \\ \qquad\quad x_1\ge 0,\ x_2\ge 0. \end{cases}$$

该问题有精确解 $x^*=\left(\dfrac{4}{3},\dfrac{1}{3}\right)^T$, 最优值 $f(x^*)=-8\dfrac{1}{9}$.

解 在 Matlab 命令窗口输入下列命令:

```
H=[1 -1;-1 2];
c=[-6 -2]';
Ai=[-2 -1;1 -1;-1 -2; 1 0;0 1];
bi=[-3 -1 -2 0 0]';
x=[0 0]';
[x,lambda,exitflag,output]=qpact(H,c,[],[],Ai,bi,x0)
```

得计算结果为

```
x =
    1.3333
    0.3333
lambda =
    2.4444
    0.1111
```

```
exitflag =

    0

output =

    fval: -8.1111

    iter: 7
```