最优化方法及其 Matlab 程序设计

第九章 罚函数法

从本章开始, 我们讨论约束优化问题的求解方法. 首先介绍求解约束优化问题的经典算法—罚函数法. 其基本思想是: 根据约束条件的特点将其转化为某种惩罚函数加到目标函数中去, 从而将约束优化问题转化为一系列的无约束优化问题来求解. 本章主要介绍外罚函数法、内点法和乘子法.

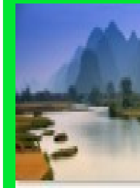
§9.1 外罚函数法

我们首先通过一个简单的例子来说明罚函数的构造.

例 9.1 *求解约束优化问题:*

$$\begin{aligned} \min\ & f(x) = (x_1-1)^2 + (x_2-1)^2, \\ \text{s.t.}\ & x_1 + x_2 = 1. \end{aligned}$$

解 由等式约束得 $x_2 = 1 - x_1$, 代入目标函数得到一个无约束的



单变量极小化问题

$$\min \ \phi(x_1) = (x_1 - 1)^2 + x_1^2,$$

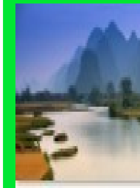
其全局极小点为 $x_1^* = 0.5$, 从而得到原问题的全局极小点为 $x^* = (0.5, 0.5)^T$. 现在要使构造的罚函数 $\bar{P}(x)$ 满足

$$\bar{P}(x) \begin{cases} = 0, & x_1 + x_2 - 1 = 0, \\ > 0, & x_1 + x_2 - 1 \neq 0, \end{cases}$$

只要令 $\bar{P}(x) = (x_1 + x_2 - 1)^2$ 即可. 现在考察目标函数和上述罚函数的组合

$$\begin{aligned} P(x, \sigma) &= f(x) + \sigma \bar{P}(x) \\ &= (x_1 - 1)^2 + (x_2 - 1)^2 + \sigma (x_1 + x_2 - 1)^2, \end{aligned}$$

其中 $\sigma > 0$ 是充分大的正数, 称为罚参数或罚因子. 求这个组合函数



Back
Close

的极小点. 由

$$\frac{\partial P(x,\sigma)}{\partial x_1} = \frac{\partial P(x,\sigma)}{\partial x_2} = 0,$$

得

$$\begin{cases} (1+\sigma)x_1 + \sigma x_2 = 1+\sigma, \\ \sigma x_1 + (1+\sigma)x_2 = 1+\sigma. \end{cases}$$

求解上述方程组得

$$x_1(\sigma) = x_2(\sigma) = \frac{\sigma+1}{2\sigma+1}.$$

令 $\sigma \to +\infty$, 有

$$\big(x_1(\sigma), x_2(\sigma)\big)^T \to \left(\frac{1}{2}, \frac{1}{2}\right)^T = x^*.$$

这样, 我们就从无约束优化问题极小点的极限得到了原问题的极小点.

下面我们将这种思想方法推广到一般约束的优化问题. 考虑

$$
\begin{cases}
\min \ f(x), & x \in \mathbb{R}^n, \\
\text{s.t.} \ \ h_i(x) = 0, & i \in E = \{1, \cdots, l\}, \\
\qquad\ g_i(x) \geq 0, & i \in I = \{1, \cdots, m\}.
\end{cases}
\tag{9.1}
$$

记可行域为 $\mathcal{D} = \{x \in \mathbb{R}^n \,|\, h_i(x) = 0 \,(i \in E),\ g_i(x) \geq 0 \,(i \in I)\}$. 构造罚函数

$$
\bar{P}(x) = \sum_{i=1}^{l} h_i^2(x) + \sum_{i=1}^{m} [\min\{0, g_i(x)\}]^2 \tag{9.2}
$$

和增广目标函数

$$
P(x, \sigma) = f(x) + \sigma \bar{P}(x), \tag{9.3}
$$

其中 $\sigma > 0$ 是罚参数或罚因子. 不难发现, 当 $x \in \mathcal{D}$ 时, 即 x 为可行点时, $P(x, \sigma) = f(x)$, 此时, 目标函数没有受到额外惩罚; 而当 $x \notin \mathcal{D}$ 时, 即 x 为不可行点时, $P(x, \sigma) > f(x)$, 此时, 目标函数受到了额外的

惩罚. $\sigma > 0$ 越大, 受到的惩罚越重. 当 $\sigma > 0$ 充分大时, 要使 $P(x,\sigma)$ 达到极小, 罚函数 $\bar{P}(x)$ 应充分小才可以. 从而 $P(x,\sigma)$ 的极小点充分逼近可行域 $\mathcal{D}$, 而其极小值自然充分逼近 $f(x)$ 在 $\mathcal{D}$ 上的极小值. 这样求解一般约束优化问题 (9.1) 就可以化为求解一系列无约束的优化问题

$$\min \ P(x, \sigma_k),$$

其中 $\{\sigma_k\}$ 是正数序列, 且 $\sigma_k \to +\infty$.

从例 9.1 可以看出, 当 $\sigma \to \infty$ 时, $P(x,\sigma)$ 的极小点 $x(\sigma) \to x^*$, 但

$$x_1(\sigma) + x_2(\sigma) - 1 = \frac{2\sigma + 2}{2\sigma + 1} - 1 = \frac{1}{2\sigma + 1} \neq 0,$$

即 $x(\sigma) \notin \mathcal{D}$, 也就是说 $x(\sigma)$ 是从可行域的外部趋于 x^* 的. 因此上述的罚函数法也称为外罚函数法 (或外点法).

下面给出外罚函数法的详细算法步骤.

Back
Close

(外罚函数法)

步 0 给定初始点 $x_0 \in \mathbb{R}^n$, 终止误差 $0 \leq \varepsilon \ll 1$. $\sigma_1 > 0$, $\gamma > 1$. 令 $k := 1$.

步 1 以 x_{k-1} 为初始点求解子问题

$$\min_{x \in \mathbb{R}^n} P(x, \sigma_k) = f(x) + \sigma_k \bar{P}(x). \tag{9.4}$$

令其极小点为 x_k.

步 2 若 $\sigma_k \bar{P}(x_k) \leq \varepsilon$, 停算, 输出 $x^* \approx x_k$ 作为近似极小点.

步 3 令 $\sigma_{k+1} := \gamma \sigma_k$, $k := k + 1$, 转步 1.

注 由上述算法可知, 外罚函数法结构简单, 可以直接调用无约束优化算法的通用程序, 因而容易编程实现. 缺点是: (1) x_k 往往不是可行点, 这对于某些实际问题是难以接受的; (2) 罚参数 σ_k 的选取比较困难, 取的过小, 可能起不到“惩罚”的作用, 而取得过大则可能造成

$P(x, \sigma_k)$ 的 Hesse 阵的条件数很大, 从而带来数值技术上的困难; (3) 注意到 $\bar{P}(x)$ 一般是不可微的, 因而难以直接使用利用导数的优化算法, 从而收敛速度缓慢.

下面讨论算法 9.1 的收敛性. 我们首先证明下面的引理.

引理 9.1 设 $\{x_k\}$ 是由算法 9.1 产生的迭代序列. 若 x_k 是子问题 (9.4) 的全局极小点, 则有下述结论成立:

$$P(x_{k+1}, \sigma_{k+1}) \geq P(x_k, \sigma_k), \tag{9.5}$$

$$\bar{P}(x_{k+1}) \leq \bar{P}(x_k), \tag{9.6}$$

$$f(x_{k+1}) \geq f(x_k). \tag{9.7}$$

证 (1) 注意到 $\sigma_{k+1} \geq \sigma_k > 0$, 因此有

$$\begin{aligned} P(x_{k+1}, \sigma_{k+1}) &= f(x_{k+1}) + \sigma_{k+1}\bar{P}(x_{k+1}), \\ &\geq f(x_{k+1}) + \sigma_k\bar{P}(x_{k+1}), \\ &= P(x_{k+1}, \sigma_k) \geq P(x_k, \sigma_k), \end{aligned}$$

即 (9.5) 成立. 由题设 x_k, x_{k+1} 分别是 $P(x, \sigma_k)$ 和 $P(x, \sigma_{k+1})$ 的全局极小点, 故有

$$f(x_{k+1}) + \sigma_k\bar{P}(x_{k+1}) \geq f(x_k) + \sigma_k\bar{P}(x_k), \tag{9.8}$$

$$f(x_k) + \sigma_{k+1}\bar{P}(x_k) \geq f(x_{k+1}) + \sigma_{k+1}\bar{P}(x_{k+1}). \tag{9.9}$$

上面两式相加并整理可得

$$(\sigma_{k+1} - \sigma_k)\bar{P}(x_k) \geq (\sigma_{k+1} - \sigma_k)\bar{P}(x_{k+1}),$$

Back
Close

即

$$(\sigma_{k+1} - \sigma_k)[\bar{P}(x_k) - \bar{P}(x_{k+1})] \geq 0,$$

从而必有 $\bar{P}(x_k) - \bar{P}(x_{k+1}) \geq 0$, 即 (9.6) 成立. 最后, 由 (9.8) 立即可得

$$f(x_{k+1}) - f(x_k) \geq \sigma_k[\bar{P}(x_k) - \bar{P}(x_{k+1}) \geq 0.$$

证毕. $\square$

下面我们给出算法 9.1 的收敛性定理.

定理 9.1 设 $\{x_k\}$ 和 $\{\sigma_k\}$ 是由算法 9.1 产生的序列, x^* 是约束优化问题 (9.1) 的全局极小点. 若 x_k 为无约束子问题 (9.4) 的全局极小点, 且罚参数 $\sigma_k \to +\infty$, 则 $\{x_k\}$ 的任一聚点 $\bar{x}$ 都是 (9.1) 的全局极小点.

证 设 x^∞ 是序列 $\{x_k\}$ 的一个聚点, 不失一般性, 设 $x_k \to x^\infty$, $k \to +\infty$. 由题设 x^* 是原问题的全局极小点, 因而必为可行点, 故有 $\bar{P}(x^*) =$

Back
Close

0. 下面分两步证明本定理的结论.

(1) 先证 x^∞ 是原问题的可行点, 亦即 $\bar{P}(x^\infty)=0$. 事实上, 由引理 9.1 知 $\{P(x_k,\sigma_k)\}$ 是单调递增有上界的序列, 因此极限存在, 设为 P^∞. 此外, 注意到 $\{f(x_k)\}$) 也是单调递增的, 且

$$f(x_k)\le P(x_k,\sigma_k)\le f(x^*),$$

即序列 $\{f(x_k)\}$ 也收敛, 记其极限为 f^∞. 于是有

$$\lim_{k\to\infty}\sigma_k\bar{P}(x_k)=\lim_{k\to\infty}\left[P(x_k,\sigma_k)-f(x_k)\right]=P^\infty-f^\infty.$$

因 $\sigma_k\to+\infty$, 故必有 $\lim\limits_{k\to\infty}\bar{P}(x_k)=0$. 由 $\bar{P}$ 的连续性知 $\bar{P}(x^\infty)=0$, 即 x^∞ 是可行点.

(2) 再证 x^∞ 是全局极小点, 亦即 $f(x^\infty)=f(x^*)$. 由 $f(x)$ 的连续性及 $x_k\to x^\infty$ 可知

$$f(x^\infty)=\lim_{k\to\infty}f(x_k)\le f(x^*).$$

注意到 x^* 是问题的全局极小点, 故显然有 $f(x^*) \leq f(x^\infty)$. 从而 $f(x^\infty) = f(x^*)$, 从而 x^∞ 为原问题的全局极小点. 证毕. $\square$

注 上述定理要求算法的每一迭代步求解子问题得到的 x_k 必须是无约束问题 $\min P(x, \sigma_k)$ 的全局极小点. 这一点在实际计算中是很难操作的, 因为求无约束优化问题全局极小点至今仍然是一个很困难的问题. 故算法 9.1 (外罚函数法) 经常遇到迭代失败的情形是很正常的. 此外, 算法 9.1 之所以选用 $\sigma_k \bar{P}(x_k) \leq \varepsilon$ 作为终止条件, 是因为

$$\lim_{k\to\infty} \sigma_k \bar{P}(x_k) = \lim_{k\to\infty} [P(x_k, \sigma_k) - f(x_k)] = P^\infty - f^\infty = 0$$

的缘故.

§9.2 内点法

1. 不等式约束问题的内点法

内点法一般只适用于不等式约束的优化问题:

$$\begin{cases} \min\ \ f(x), & x \in \mathbb{R}^n, \\ \text{s.t.}\ \ g_i(x) \geq 0, & i = 1, \cdots, m. \end{cases} \tag{9.10}$$

记可行域 $\mathcal{D} = \{x \in \mathbb{R}^n : g_i(x) \geq 0,\ i = 1, \cdots, m\}$. 内点法也属于罚方法的范畴, 其基本思想是保持每一个迭代点 x_k 是可行域 $\mathcal{D}$ 的内点, 可行域的边界被筑起一道很高的“围墙”作为障碍, 当迭代点靠近边界时, 增广目标函数值骤然增大, 以示“惩罚”, 并阻止迭代点穿越边界. 因此, 内点法也称为内罚函数法或障碍函数法, 它只是用于可行域的内点集非空的情形, 即

$$\mathcal{D}_0 = \{x \in \mathbb{R}^n : g_i(x) > 0,\ i = 1, \cdots, m\} \neq \emptyset.$$

类似于外罚函数法, 我们需要构造如下的增广目标函数

$$H(x, \tau) = f(x) + \tau \bar{H}(x),$$

其中 $\bar{H}(x)$ 是障碍函数, 它需要满足如下性质: 当 x 在 $\mathcal{D}_0$ 趋向于边界时, 至少有一个 $g_i(x)$ 趋向于 0, 而 $\bar{H}(x)$ 要趋向于无穷大. 因此可以取约束函数的倒数之和为障碍函数可满足要求, 即

$$\bar{H}(x)=\sum_{i=1}^{m}\frac{1}{g_i(x)}, \tag{9.11}$$

或者取对数障碍函数

$$\bar{H}(x)=-\sum_{i=1}^{m}\ln[g_i(x)]. \tag{9.12}$$

参数 $\tau>0$ 称为罚因子或罚参数. 这样, 当 x 在 $\mathcal{D}_0$ 中时, $\bar{H}(x)>0$ 是有限的; 当 x 接近边界时, $\bar{H}(x)\to+\infty$, 从而增广目标函数的值也趋向于无穷大, 因此得到了严重的“惩罚”.

由于约束优化问题的极小点一般在可行域的边界上达到, 因此与外罚函数法中的罚因子 $\sigma_k\to+\infty$ 相反, 内点法中的罚因子则要求

$\tau_k \to 0$. 于是, 求解问题 (9.10) 就可以转化为求解序列无约束优化子问题:

$$\min \;\; H(x, \tau_k) = f(x) + \tau_k \bar{H}(x). \tag{9.13}$$

现在我们来看一个简单的例子.

例 9.2 用内点法求解下面的优化问题:

$$\begin{cases} \min \;\; 2x_1 + 3x_2, \\ \text{s.t.} \;\;\; 1 - 2x_1^2 - x_2^2 \geq 0. \end{cases}$$

解 给出增广目标函数为

$$H(x, \tau) = 2x_1 + 3x_2 - \tau \ln(1 - 2x_1^2 - x_2^2).$$

令

$$\frac{\partial H}{\partial x_1} = 2 + \frac{4\tau x_1}{1 - 2x_1^2 - x_2^2} = 0, \tag{9.14}$$

$$\frac{\partial H}{\partial x_2} = 3 + \frac{2\tau x_2}{1 - 2x_1^2 - x_2^2} = 0. \tag{9.15}$$

由 (9.14)-(9.15) 可得 $x_2 = 3x_1$. 代入 (9.14) 得

$$0 = 1 + \frac{2\tau x_1}{1 - 2x_1^2 - (3x_1)^2} = 1 + \frac{2\tau x_1}{1 - 11x_1^2},$$

即

$$11x_1^2 - 2\tau x_1 - 1 = 0.$$

解得

$$x_1(\tau) = \frac{\tau \pm \sqrt{\tau^2 + 11}}{11} \ \to \ \pm\frac{1}{\sqrt{11}}, \ (\tau \to 0).$$

自然有

$$x_2(\tau) = 3x_1(\tau) \ \to \ \pm\frac{3}{\sqrt{11}}, \ (\tau \to 0).$$

注意到目标函数的表达式, 可得所求问题的全局极小点和全局极小值分别为

$$x^* = \left(-\frac{1}{\sqrt{11}},\ -\frac{3}{\sqrt{11}}\right)^T, \quad f(x^*) = 2x_1^* + 3x_2^* = -\sqrt{11}.$$

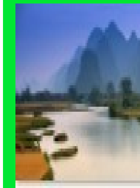
上面的问题比较简单, 可以将子问题 $\min H(x,\tau)$ 最优解的解析表达式求出来, 然后对罚参数 $\tau \to 0$ 取极限而得到原问题的极小点. 一般来说, 对于较复杂的问题, 只能用数值方法来求子问题的近似全局极小点.

下面给出内点法的详细算法步骤.

算法 9.2 (内点法)

步 0　给定初始点 $x_0 \in \mathcal{D}_0$, 终止误差 $0 \le \varepsilon \ll 1$. $\tau_1 > 0$, $\varrho \in (0,1)$. 令 $k := 1$.

步 1　以 x_{k-1} 为初始点求解无约束子问题 (9.13), 得极小点 x_k.



步 2　若 $\tau_k \bar{H}(x_k) \le \varepsilon$, 停算, 输出 $x^* \approx x_k$ 作为近似极小点.

步 3　令 $\tau_{k+1} := \varrho\tau_k$,　$k := k+1$, 转步 1.

注　由上述算法可以看出, 内点法的优点是结构简单, 适应性强. 但是随着迭代过程的进行, 罚参数 τ_k 将变得越来越小, 趋向于零, 使得增广目标函数的病态性越来越严重, 这给无约束子问题的求解带来了数值实现上的困难, 以致导致迭代的失败. 此外, 内点法的初始点 x_0 要求是一个严格的可行点, 一般来说这也是比较麻烦甚至困难的.

下面我们来考虑内点法的收敛性. 先看下面的引理.

引理 9.2　设序列 $\{x_k\}$ 由算法 9.2 产生, 且每个 x_k 都是无约束子问题 (9.13) 的全局极小点. 那么增广目标函数序列 $\{H(x_k, \tau_k)\}$ 是单调下降的, 即

$$H(x_{k+1}, \tau_{k+1}) \le H(x_k, \tau_k).$$

证 注意到 x_{k+1} 是 $H(x,\tau_{k+1})$ 的全局极小点, 且由算法有 $\tau_{k+1}\le\tau_k$, 故

$$
\begin{aligned}
H(x_{k+1},\tau_{k+1}) &= f(x_{k+1})+\tau_{k+1}\bar{H}(x_{k+1}) \\
&\le f(x_k)+\tau_{k+1}\bar{H}(x_k) \\
&\le f(x_k)+\tau_k\bar{H}(x_k) \\
&= H(x_k,\tau_k).
\end{aligned}
$$

证毕. □

下面的定理给出了算法 9.2 的收敛性.

定理 9.2 设 $f(x)$ 在 $\mathcal{D}$ 上存在全局极小点 x^* 且内点集 $\mathcal{D}_0\neq\emptyset$. $\{(x_k,\tau_k)\}$ 是由算法 9.2 产生的序列. 若 x_k 是 $H(x,\tau_k)$ 的全局极小点且 $\{\tau_k\}\downarrow 0$, 那么 $\{x_k\}$ 的任一聚点 $\bar{x}$ 都是问题 (9.10) 的全局极小点.

证 由定理的条件, 有 $x_k\in\mathcal{D}_0\subset\mathcal{D}$ 且 x^* 是 $f(x)$ 在 $\mathcal{D}$ 上的全局

极小点. 故有 $f(x^*) \le f(x_k) \le H(x_k,\tau_k)$, 即序列 $\{H(x_k,\tau_k)\}$ 有下界, 于是由引理 9.2 知 $\lim\limits_{k\to\infty} H(x_k,\tau_k)$ 存在, 不妨记为 H^*.

下面证明 $H^* = f(x^*)$. 事实上, 显然有 $H^* \ge f(x^*)$. 以下只需证明 $H^* \le f(x^*)$. 由 $f(x)$ 的连续性可知, 对于任意的 $\varepsilon > 0$, 存在 $\delta > 0$, 使得满足 $\|\bar{x} - x^*\| \le \delta$ 的 $\bar{x} \in \mathcal{D}_0$, 有

$$f(\bar{x}) - f(x^*) < \varepsilon.$$

又因 $\{\tau_k\} \downarrow 0$, 故对于上述的 ε 和 $\bar{x}$, 存在正数 k_0, 使得当 $k \ge k_0$ 时, 有

$$\tau_k \bar{H}(x_k) \le \varepsilon.$$

注意到 x_k 是 $H(x,\tau_k)$ 的全局极小点, 即有

$$H(x_k,\tau_k) \le H(\bar{x},\tau_k).$$

这样, 我们有

$$
\begin{aligned}
H(x_k,\tau_k)-f(x^*) &\le H(\bar{x},\tau_k)-f(x^*)\\
&= [f(\bar{x})-f(x^*)] + \tau_k\bar{H}(\bar{x})\\
&< \varepsilon + \varepsilon = 2\varepsilon.
\end{aligned}
$$

令 $\varepsilon \to 0^+$, 并对上式取极限, 即得 $H^* \le f(x^*)$, 从而 $H^* = f(x^*)$. 最后, 对不等式

$$
f(x^*) \le f(x_k) \le H(x_k,\tau_k)
$$

取极限得 $\lim\limits_{k\to\infty} f(x_k) = f(x^*)$. □

2. 一般约束问题的内点法

Back
Close

现在, 我们考虑一般约束优化问题

$$\begin{cases} \min \ f(x), & x \in \mathbb{R}^n, \\ \text{s.t.} \ \ h_i(x) = 0, & i = 1, \cdots, l, \\ \qquad g_i(x) \geq 0, & i = 1, \cdots, m \end{cases} \tag{9.16}$$

内点法特征的罚函数方法. 途径之一是对于等式约束利用“外罚函数”的思想, 而对于不等式约束则利用“障碍函数”的思想, 构造出所谓混合增广目标函数

$$H(x, \mu) = f(x) + \frac{1}{2\mu}\sum_{i=1}^{l} h_i^2(x) + \mu \sum_{i=1}^{m} \frac{1}{g_i(x)}, \tag{9.17}$$

或

$$H(x, \mu) = f(x) + \frac{1}{2\mu}\sum_{i=1}^{l} h_i^2(x) - \mu \sum_{i=1}^{m} \ln[g_i(x)]. \tag{9.18}$$

于是可以类似于内点法或外罚函数法的算法框架建立起相应的算法. 但由此建立的算法的初始点的选取仍然是个困难的问题.

另一种途径是, 引入松弛变量 y_i, $i=1,\cdots,m$, 将问题等价地转化为

$$
\begin{cases}
\min \ \ f(x), & x \in \mathbb{R}^n, \\
\text{s.t.} \ \ h_i(x) = 0, & i = 1, \cdots, l, \\
\qquad\ \ g_i(x) - y_i = 0, & i = 1, \cdots, m, \\
\qquad\ \ y_i \geq 0, & i = 1, \cdots, m.
\end{cases} \tag{9.19}
$$

然后构造等价问题 (9.19) 的混合增广目标函数:

$$
\psi(x, y, \mu) = f(x) + \frac{1}{2\mu} \sum_{i=1}^{l} h_i^2(x) + \frac{1}{2\mu} \sum_{i=1}^{m} \left[g_i(x) - y_i\right]^2 + \mu \sum_{i=1}^{m} \frac{1}{y_i}, \tag{9.20}
$$

Back
Close

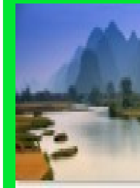
或

$$\psi(x,y,\mu)=f(x)+\frac{1}{2\mu}\sum_{i=1}^{l}h_i^2(x)+\frac{1}{2\mu}\sum_{i=1}^{m}\left[g_i(x)-y_i\right]^2-\mu\sum_{i=1}^{m}\ln y_i. \tag{9.21}$$

在此基础上, 类似于前面的外罚函数法与内点法的算法框架, 可以建立起相应的求解算法. 值得说明的是, 此时, 任意的 (x,y), $y>0$ 均可作为一个合适的初始点来启动相应的迭代算法.

§9.3 乘子法

乘子法是 Powell 和 Hestenes 于 1969 年针对等式约束优化问题同时独立提出的一种优化算法, 后于 1973 年经 Rockfellar 推广到求解不等式约束优化问题. 其基本思想是从原问题的拉格朗日函数出发, 再加上适当的罚函数, 从而将原问题转化为求解一系列的无约束优化子



问题. 由于外罚函数法中的罚参数 $\sigma_k \to +\infty$, 因此增广目标函数变得“越来越病态”. 增广目标函数的这种病态性质是外罚函数法的主要缺点, 而这种缺陷在乘子法中由于引入拉格朗日函数及加上适当的罚函数而得以有效的克服.

1. 等式约束问题的乘子法

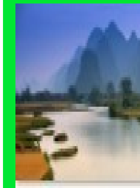
考虑等式约束优化问题

$$\begin{cases} \min \ \ f(x), \\ \text{s.t.} \ \ h(x) = 0, \end{cases} \tag{9.22}$$

其中 $h(x) = (h_1(x), \cdots, h_l(x))^T$. 记可行域 $\mathcal{D} = \{x \in \mathbb{R}^n \,|\, h(x) = 0\}$, 则上述问题的拉格朗日函数为

$$L(x, \lambda) = f(x) - \lambda^T h(x),$$

其中 $\lambda = (\lambda_1, \cdots, \lambda_l)^T$ 为乘子向量. 设 (x^*, λ^*) 是问题 (9.22) 的 KT



点, 则由最优性条件有

$$\nabla_x L(x^*,\lambda^*)=0,\quad \nabla_\lambda L(x^*,\lambda^*)=-h(x^*)=0.$$

此外, 不难发现, 对于任意的 $x\in\mathcal{D}$, 有

$$L(x^*,\lambda^*)=f(x^*)\le f(x)=f(x)-(\lambda^*)^T h(x)=L(x,\lambda^*).$$

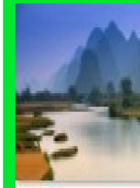
上式表明, 若已知乘子向量 λ^*, 则问题 (9.22) 可等价地转化为

$$\begin{cases}\min\ \ L(x,\lambda^*),\\ \text{s.t.}\ \ h(x)=0.\end{cases}\tag{9.23}$$

可以考虑用外罚函数法求解问题 (9.23), 其增广目标函数为

$$\psi(x,\lambda^*,\sigma)=L(x,\lambda^*)+\frac{\sigma}{2}\|h(x)\|^2.$$

可以证明, 当 $\sigma>0$ 适当大时, x^* 是 $\psi(x,\lambda^*,\sigma)$ 的极小点. 由于乘子



λ^* 事先并不知道, 故可考虑下面的增广目标函数

$$\begin{aligned} \psi(x,\lambda,\sigma) &= L(x,\lambda) + \frac{\sigma}{2}\|h(x)\|^2 \\ &= f(x) - \lambda^T h(x) + \frac{\sigma}{2}\|h(x)\|^2. \end{aligned} \tag{9.24}$$

可以这样操作: 首先固定一个 $\lambda = \bar{\lambda}$, 求 $\psi(x, \bar{\lambda}, \sigma)$ 的极小点 $\bar{x}$; 然后再适当改变 λ 的值, 再求新的 $\bar{x}$, 直到求得满足要求的 x^* 和 λ^* 为止. 具体地说, 我们在第 k 次迭代求无约束子问题 $\min\ \psi(x, \lambda_k, \sigma)$ 的极小点 x_k. 则由取极值的必要条件, 有

$$\nabla_x \psi(x_k, \lambda_k, \sigma) = \nabla f(x_k) - \nabla h(x_k)[\lambda_k - \sigma h(x_k)] = 0.$$

而在原问题的 KT 点 (x^*, λ^*) 处有

$$\nabla f(x^*) - \nabla h(x^*)\lambda^* = 0, \quad h(x^*) = 0.$$

我们的目的自然希望 $\{x_k\} \to x^*$, 且 $\{\lambda_k\} \to \lambda^*$. 于是将上面两式相

Back
Close

比较后, 可取乘子序列 $\{\lambda_k\}$ 的更新公式为

$$\lambda_{k+1} = \lambda_k - \sigma h(x_k). \tag{9.25}$$

由上式可以看出 $\{\lambda_k\}$ 收敛的充要条件是 $\{h(x_k)\} \to 0$. 下面我们证明 $h(x_k) = 0$ 也是判别 (x_k, λ_k) 是 KT 对的充要条件.

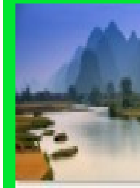
定理 9.3 设无约束优化问题

$$\min \psi(x, \lambda_k, \sigma) = L(x, \lambda_k) + \frac{\sigma}{2}\|h(x)\|^2 \tag{9.26}$$

的极小点为 x_k. 则 (x_k, λ_k) 是 (9.22) 的 KT 对的充要条件是 $h(x_k) = 0$.

证 必要性显然. 我们证明充分性. 因为 x_k 是 (9.26) 的极小点, 且 $h(x_k) = 0$. 则对任意的可行点 $x \in \mathcal{D} = \{x \in \mathbb{R}^n \mid h(x) = 0\}$, 有

$$f(x) = \psi(x, \lambda_k, \sigma) \geq \psi(x_k, \lambda_k, \sigma) = f(x_k),$$



Back
Close

即 x_k 是 (9.22) 的极小点. 另一方面, 注意到 x_k 也是 (9.26) 的稳定点, 故有

$$\nabla_x \psi(x_k, \lambda_k, \sigma) = \nabla f(x_k) - \nabla h(x_k)\lambda_k = 0.$$

上式表明 λ_k 是相应于 x_k 的拉格朗日乘子向量, 即 (x_k, λ_k) 是问题 (9.22) 的 KT 对. 证毕. $\square$

基于上述讨论, 我们给出求解等式约束问题 (9.22) 的乘子法的详细步骤如下 (由于该算法是由 Powell 和 Hestenes 首先独立提出来的, 因此也称之为 PH 算法):

算法 9.3 PH 算法

步 0 选取初始值. 给定 $x_0 \in \mathbb{R}^n$, $\lambda_1 \in \mathbb{R}^l$, $\sigma_1 > 0$, $0 \le \varepsilon \ll 1$, $\vartheta \in (0,1)$, $\eta > 1$. 令 $k := 1$.

步 1 求解子问题. 以 x_{k-1} 为初始点求解无约束子问题的极小点

x_k :

$$\min \psi(x, \lambda_k, \sigma_k) = f(x) - \lambda_k^T h(x) + \frac{\sigma_k}{2} \|h(x)\|^2. \tag{9.27}$$

步 2 检验终止条件. 若 $\|h(x_k)\| \le \varepsilon$, 停算, 输出 x_k 作为原问题的近似极小点; 否则, 转步 3.

步 3 更新罚参数. 若 $\|h(x_k)\| \ge \vartheta \|h(x_{k-1})\|$, 令 $\sigma_{k+1} := \eta \sigma_k$; 否则, $\sigma_{k+1} := \sigma_k$.

步 4 更新乘子向量. 令 $\lambda_{k+1} = \lambda_k - \sigma_k h(x_k)$.

步 5 令 $k := k+1$, 转步 1.

下面我们来看一个简单的例子.

例 9.3 用乘子法求解约束优化问题

$$\begin{cases} \min \quad f(x) = x_1^2 - x_2^2, \\ \text{s.t.} \quad x_2 = -1. \end{cases}$$

Back
Close

解 首先写出所求问题相应于乘子法的增广目标函数:

$$\begin{aligned}\psi(x,\lambda,\sigma) &= x_1^2 - x_2^2 - \lambda(x_2+1) + \frac{\sigma}{2}(x_2+1)^2 \\ &= x_1^2 + \left(\frac{\sigma}{2} - 1\right)x_2^2 + (\sigma-\lambda)x_2 - \lambda - \frac{\sigma}{2}.\end{aligned}$$

令

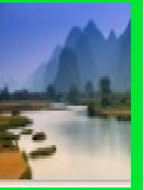
$$\frac{\partial\psi}{\partial x_1} = 2x_1 = 0, \quad \frac{\partial\psi}{\partial x_2} = (\sigma-2)x_2 - (\lambda-\sigma) = 0.$$

对于 $\sigma > 2$, 解上述关于 x_1 和 x_2 的二元一次方程组得稳定点

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \dfrac{\lambda-\sigma}{\sigma-2} \end{pmatrix}.$$

注意到约束条件 $x_2 = -1$, 可得 $\lambda = 2$. 从而 $\bar{x} = (0,-1)^T = x^*$.

从上面的例子可以发现, 乘子法并不要求罚参数 σ 趋于无穷大, 只要求它大于某个正数即可. 下面我们从理论上来证明这一事实.



引理 9.3 已知矩阵 $U \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{n\times m}$. 则对任意满足 $S^Tx=0$ 的非零向量 x 都有 $x^TUx>0$ 的充要条件是存在常数 $\sigma^*>0$, 使得对任意的 $\sigma \ge \sigma^*$ 有,

$$x^T(U+\sigma SS^T)x>0, \quad \forall\, 0 \ne x \in \mathbb{R}^n.$$

证 充分性是显然的. 注意到 $S^Tx=0$, 立即有

$$x^TUx = x^TUx + \sigma x^TSS^Tx = x^T(U+\sigma SS^T)x > 0.$$

必要性. 注意到, 若存在常数 $\sigma^*>0$, 使得

$$x^T(U+\sigma^* SS^T)x>0, \quad \forall\, 0 \ne x \in \mathbb{R}^n,$$

则任取 $\sigma \ge \sigma^*$, 恒有

$$x^T(U+\sigma SS^T)x \ge x^T(U+\sigma^* SS^T)x > 0, \quad \forall\, 0 \ne x \in \mathbb{R}^n.$$

以下只需证明上述 σ^* 的存在性. 事实上, 若这样的 σ^* 不存在, 则对任意的正整数 k, 必存在向量 x_k 且 $\|x_k\| = 1$, 使得

$$x_k^T(U + kSS^T)x_k \le 0.$$

因 $\{x_k\}$ 是有界序列, 故必有收敛的子序列 (仍记为它本身), 其极限为 $\hat{x}$, 切 $\|\hat{x}\| = 1$. 那么, 对上式取极限, $k \to \infty$, 得

$$\hat{x}^T U \hat{x} + \lim_{k\to\infty} k\|S^T x_k\|^2 \le 0.$$

由此, 必有 $\|S^T x_k\| \to \|S^T \hat{x}\| = 0$ (否则, $k\|S^T x_k\|^2 \to \infty$), 同时 $\hat{x}^T U \hat{x} \le 0$, 这与必要性的条件矛盾. 证毕. $\square$

定理 9.4 设等式约束优化问题 (9.22) 的 KT 点 (x^*, λ^*) 满足二阶充分性条件 (见定理 8.2), 则存在一个 $\sigma^* > 0$, 对所有的 $\sigma \ge \sigma^*$, x^* 是增广目标函数 $\psi(x, \lambda^*, \sigma)$ (由 (9.24) 所定义) 的严格局部极小点. 进

Back
Close

一步, 若 $h(\bar{x}) = 0$, 且 $\bar{x}$ 对某个 $\bar{\lambda}$ 是 $\psi(x, \bar{\lambda}, \sigma)$ 的局部极小点, 则 $\bar{x}$ 也是问题 (9.22) 的局部极小点.

证 注意到 $\psi(x, \lambda^*, \sigma) = L(x, \lambda^*) + \frac{\sigma}{2}\|h(x)\|^2$, 求其梯度和二阶导数得

$$\begin{aligned}
\nabla_x \psi(x, \lambda^*, \sigma) &= \nabla_x L(x, \lambda^*) + \sigma \nabla h(x)\, h(x), \\
\nabla_x^2 \psi(x^*, \lambda^*, \sigma) &= \nabla_x^2 L(x^*, \lambda^*) + \sigma \nabla h(x^*) \nabla h(x^*)^T.
\end{aligned}$$

由二阶充分性条件知, 对每个满足 $d^T \nabla h(x^*) = 0$ 的向量 $d \neq 0$, 有

$$d^T \nabla_x^2 L(x^*, \lambda^*) d > 0.$$

于是由引理 9.3 知, 存在 $\sigma^* > 0$ 使得当 $\sigma \geq \sigma^*$ 且 $d \neq 0$ 时, 有

$$d^T \nabla_x^2 \psi(x^*, \lambda^*) d = d^T [\nabla_x^2 L(x^*, \lambda^*) + \sigma \nabla h(x^*) \nabla h(x^*)^T] d > 0.$$

再由二阶充分性条件知, $\nabla_x L(x^*, \lambda^*) = 0$, $h(x^*) = 0$. 故而有

$$\nabla_x \psi(x^*, \lambda^*, \sigma) = \nabla_x L(x^*, \lambda^*, \sigma) = 0,$$

这表明增广目标函数 $\psi(x, \lambda^*, \sigma)\,(\sigma \geq \sigma^*)$ 在 x^* 处满足二阶充分性条件, 故由定理 1.8 可知, x^* 是 $\psi(x, \lambda^*, \sigma)$ 的严格局部极小点. 这就证明了定理的第一个结论.

下面我们来证明第二个结论. 若 $h(\bar{x}) = 0$, 且 $\bar{x}$ 对某个 $\bar{\lambda}$ 是 $\psi(x, \bar{\lambda}, \sigma)$ 的局部极小点. 则对任意与 $\bar{x}$ 充分靠近的 $\hat{x}$ (即 $h(\hat{x}) = 0$), 有

$$\psi(\bar{x}, \bar{\lambda}, \sigma) \leq \psi(\hat{x}, \bar{\lambda}, \sigma).$$

因 $h(\bar{x}) = h(\hat{x}) = 0$, 故有

$$\psi(\bar{x}, \bar{\lambda}, \sigma) = f(\bar{x}), \quad \psi(\hat{x}, \bar{\lambda}, \sigma) = f(\hat{x}).$$

上式表明, 对于 $\bar{x}$ 的某个邻域中的任意可行点 $\hat{x}$, 均有 $f(\bar{x}) \leq f(\hat{x})$,

即 $\bar{x}$ 是问题 (9.22) 的局部极小点. 证毕. $\square$

2. 一般约束问题的乘子法

下面我们考虑同时带有等式和不等式约束的优化问题的乘子法:

$$\begin{cases} \min \ f(x), \\ \text{s.t.} \ \ h_i(x) = 0, \ i = 1, \cdots, l, \\ \qquad g_i(x) \geq 0, \ i = 1, \cdots, m. \end{cases} \tag{9.28}$$

其基本思想是把解等式约束优化问题的乘子法推广到不等式约束优化问题, 即先引进辅助变量把不等式约束化为等式约束, 然后再利用最优性条件消去辅助变量. 为叙述的方便计, 我们先考虑如下只带有不等式约束的最优化问题

$$\begin{cases} \min \ f(x), \\ \text{s.t.} \ \ g_i(x) \geq 0, \ i = 1, \cdots, m. \end{cases}$$

引进辅助变量 $y_i\,(i=1,\cdots,m)$, 可以将上面的优化问题化为等价的等式约束优化问题:

$$\begin{cases} \min \;\; f(x), \\ \text{s.t.} \;\; g_i(x)-y_i^2=0, \;\; i=1,\cdots,m. \end{cases}$$

利用算法 9.1 求解, 此时增广拉格朗日函数为

$$\tilde{\psi}(x,y,\lambda,\sigma)=f(x)-\sum_{i=1}^{m}\lambda_i[g_i(x)-y_i^2]+\frac{\sigma}{2}\sum_{i=1}^{m}[g_i(x)-y_i^2]^2.$$

为了消去辅助变量 y, 可考虑 $\tilde{\psi}$ 关于变量 y 的极小化. 由一阶必要条件, 令 $\nabla_y\tilde{\psi}(x,y,\lambda,\sigma)=0$ 可得

$$2y_i\lambda_i-2\sigma y_i[g_i(x)-y_i^2]=0, \quad i=1,\cdots,m,$$

即

$$y_i[\sigma y_i^2-(\sigma g_i(x)-\lambda_i)]=0, \quad i=1,\cdots,m.$$

Back
Close

故当 $\sigma g_i(x) - \lambda_i > 0$ 时, 有

$$y_i^2 = \frac{1}{\sigma}[\sigma g_i(x) - \lambda_i] = g_i(x) - \frac{1}{\sigma}\lambda_i.$$

否则, 由 $\sigma y_i^2 + [\lambda_i - \sigma g_i(x)] \geq 0$ 可推得 $y_i = 0$. 综合起来, 有

$$y_i^2 = \begin{cases} \dfrac{1}{\sigma}[\sigma g_i(x) - \lambda_i], & \sigma g_i(x) - \lambda_i > 0, \\ 0, & \sigma g_i(x) - \lambda_i \leq 0, \end{cases} \quad i = 1, \cdots, m.$$

即有

$$g_i(x) - y_i^2 = \begin{cases} \dfrac{\lambda_i}{\sigma}, & \sigma g_i(x) - \lambda_i > 0, \\ g_i(x), & \sigma g_i(x) - \lambda_i \leq 0, \end{cases} \quad i = 1, \cdots, m. \tag{9.29}$$

因此, 当 $\sigma g_i(x) - \lambda_i \le 0$ 时, 我们有

$$
\begin{aligned}
& -\lambda_i[g_i(x) - y_i^2] + \frac{\sigma}{2}[g_i(x) - y_i^2]^2 \\
&= -\lambda_i g_i(x) + \frac{\sigma}{2}[g_i(x)]^2 \\
&= \frac{1}{2\sigma}\big[(\sigma g_i(x) - \lambda_i)^2 - \lambda_i^2\big].
\end{aligned}
$$

而当 $\sigma g_i(x) - \lambda_i > 0$ 时, 有

$$
\begin{aligned}
& -\lambda_i[g_i(x) - y_i^2] + \frac{\sigma}{2}[g_i(x) - y_i^2]^2 \\
&= -\frac{1}{\sigma}\lambda_i^2 + \frac{1}{2\sigma}\lambda_i^2 = -\frac{1}{2\sigma}\lambda_i^2.
\end{aligned}
$$

综合上述两种情形, 有

$$
-\lambda_i[g_i(x) - y_i^2] + \frac{\sigma}{2}[g_i(x) - y_i^2]^2 = \frac{1}{2\sigma}\Big([\min\{0, \sigma g_i(x) - \lambda_i\}]^2 - \lambda_i^2\Big).
$$

Back
Close

将其代回到 $\tilde{\psi}(x,y,\lambda,\sigma)$ 中去得

$$\begin{aligned}\tilde{\phi}(x,\lambda,\sigma) &= \min_y \tilde{\psi}(x,y,\lambda,\sigma)\\ &= f(x)+\frac{1}{2\sigma}\sum_{i=1}^{m}\Big([\min\{0,\sigma g_i(x)-\lambda_i\}]^2-\lambda_i^2\Big).\end{aligned}\tag{9.30}$$

于是, 将 (9.29) 代入乘子迭代公式 $(\lambda_{k+1})_i=(\lambda_k)_i-\sigma[g_i(x_k)-(y_k)_i^2]$ 得

$$(\lambda_{k+1})_i=\begin{cases}0, & \sigma g_i(x_k)-(\lambda_k)_i>0,\\ (\lambda_k)_i-\sigma g_i(x_k), & \sigma g_i(x_k)-(\lambda_k)_i\le 0,\end{cases}$$

即

$$(\lambda_{k+1})_i=\max\{0,(\lambda_k)_i-\sigma g_i(x_k)\}\ge 0,\quad i=1,\cdots,m.\tag{9.31}$$

同样, 在终止准则

$$\Big(\sum_{i=1}^{m}[g_i(x_k)-(y_k)_i^2]^2\Big)^{1/2}\le\varepsilon$$

中代入 (9.29) 得

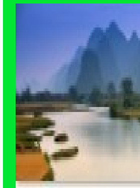
$$\left(\sum_{i=1}^{m}\left[\min\left\{g_i(x_k),\ \frac{(\lambda_k)_i}{\sigma}\right\}\right]^2\right)^{1/2}\le\varepsilon. \tag{9.32}$$

现在我们回到一般约束优化问题 (9.28), 我们来构造求解 (9.28) 的乘子法. 此时, 增广拉格朗日函数为

$$\begin{aligned}\psi(x,\mu,\lambda,\sigma) &= f(x)-\sum_{i=1}^{l}\mu_i h_i(x)+\frac{\sigma}{2}\sum_{i=1}^{l}h_i^2(x)\\ &\quad+\frac{1}{2\sigma}\sum_{i=1}^{m}\left([\min\{0,\sigma g_i(x)-\lambda_i\}]^2-\lambda_i^2\right).\end{aligned} \tag{9.33}$$

乘子迭代的公式为

$$\begin{aligned}(\mu_{k+1})_i &= (\mu_k)_i-\sigma h_i(x_k),\quad i=1,\cdots,l,\\ (\lambda_{k+1})_i &= \max\{0,(\lambda_k)_i-\sigma g_i(x_k)\},\quad i=1,\cdots,m.\end{aligned}$$



令

$$\beta_k = \left(\sum_{i=1}^{l} h_i^2(x_k) + \sum_{i=1}^{m} \left[\min \left\{ g_i(x_k), \frac{(\lambda_k)_i}{\sigma} \right\} \right]^2 \right)^{1/2}, \tag{9.34}$$

则终止准则为

$$\beta_k \le \varepsilon.$$

下面给出求解一般约束优化问题 (9.28) 乘子法的详细步骤. 由于这一算法是由 Rockfellar 在 PH 算法的基础上提出的, 因此简称为 PHR 算法.

算法 9.4 PHR 算法

步 0 选取初始值. 给定 $x_0 \in \mathbb{R}^n$, $\mu_1 \in \mathbb{R}^l$, $\lambda_1 \in \mathbb{R}^m$, $\sigma_1 > 0$, $0 \le \varepsilon \ll 1$, $\vartheta \in (0,1)$, $\eta > 1$. 令 $k := 1$.

步 1 求解子问题. 以 x_{k-1} 为初始点求解无约束子问题

$$\min \psi(x, \mu_k, \lambda_k, \sigma_k),$$

得极小点 x_k, 其中 $\psi(x,\mu_k,\lambda_k,\sigma_k)$ 由 (9.33) 定义.

步 2 检验终止条件. 若 $\beta_k \le \varepsilon$, 其中 β_k 由 (9.34) 定义, 则停止迭代, 输出 x_k 作为原问题的近似极小点; 否则, 转步 3.

步 3 更新罚参数. 若 $\beta_k \ge \vartheta\beta_{k-1}$, 令 $\sigma_{k+1} := \eta\sigma_k$; 否则, $\sigma_{k+1} := \sigma_k$.

步 4 更新乘子向量. 计算

$$\begin{aligned}(\mu_{k+1})_i &= (\mu_k)_i - \sigma h_i(x_k), \quad i = 1,\cdots,l,\\ (\lambda_{k+1})_i &= \max\{0, (\lambda_k)_i - \sigma g_i(x_k)\}, \quad i = 1,\cdots,m.\end{aligned}$$

步 5 令 $k := k+1$, 转步 1.

§9.4 乘子法的 Matlab 实现

本小节我们给出算法 9.4 (PHR 乘子法) 的 Matlab 通用程序, 然

后利用该程序求解一个简单的约束优化问题.

程序 9.1 (PHR 算法程序)

```
function [x,mu,lambda,output]=multphr(fun,hf,gf,dfun,dhf,dgf,x0)
%功能: 用乘子法解一般约束问题:  min f(x), s.t. h(x)=0, g(x)>=0
%输入:  x0是初始点, fun, dfun分别是目标函数及其梯度;
%   hf, dhf分别是等式约束 (向量) 函数及其Jacobi矩阵的转置;
%   gf, dgf分别是不等式约束 (向量) 函数及其Jacobi矩阵的转置;
%输出:  x是近似最优点, mu, lambda分别是相应于等式约束和不
%   等式约束的乘子向量; output是结构变量, 输出近似极小值f, 迭
%   代次数, 内迭代次数等
maxk=500;   %最大迭代次数
sigma=2.0;  %罚因子
eta=2.0;  theta=0.8;  %PHR算法中的实参数
k=0; ink=0;  %k, ink分别是外迭代和内迭代次数
epsilon=1e-5;  %终止误差值
x=x0;  he=feval(hf,x); gi=feval(gf,x);
n=length(x); l=length(he); m=length(gi);
%选取乘子向量的初始值
mu=0.1*ones(l,1);  lambda=0.1*ones(m,1);
```

```
btak=10;  btaold=10;  %用来检验终止条件的两个值
while(btak>epsilon & k<maxk)
    %调用BFGS算法程序求解无约束子问题
    [x,ival,ik]=bfgs('mpsi','dmpsi',x0,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma);
    ink=ink+ik;
    he=feval(hf,x); gi=feval(gf,x);
    btak=0.0;
    for(i=1:l), btak=btak+he(i)^2;   end
    for(i=1:m)
        temp=min(gi(i),lambda(i)/sigma);
        btak=btak+temp^2;
    end
    btak=sqrt(btak);
    if btak>epsilon
        if(k>=2 & btak > theta*btaold)
            sigma=eta*sigma;
        end
        %更新乘子向量
        for(i=1:l),  mu(i)=mu(i)-sigma*he(i); end
        for(i=1:m)
```

```
            lambda(i)=max(0.0,lambda(i)-sigma*gi(i));
        end
    end
    k=k+1;
    btaold=btak;
    x0=x;
end
f=feval(fun,x);
output.fval=f;
output.iter=k;
output.inner_iter=ink;
output.bta=btak;
%%%%%%%%%%%%%%%%%%  增广拉格朗日函数 %%%%%%%%%%%%%%%%%%%%%
function psi=mpsi(x,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma)
f=feval(fun,x);  he=feval(hf,x);  gi=feval(gf,x);
l=length(he); m=length(gi);
psi=f;  s1=0.0;
for(i=1:l)
    psi=psi-he(i)*mu(i);
    s1=s1+he(i)^2;
```

Back
Close

```
end
psi=psi+0.5*sigma*s1;
s2=0.0;
for(i=1:m)
    s3=max(0.0, lambda(i) - sigma*gi(i));
    s2=s2+s3^2-lambda(i)^2;
end
psi=psi+s2/(2.0*sigma);
%%%%%%%%%%%%%%%%%%  增广拉格朗日函数的梯度%%%%%%%%%%%%%%%%%%%%
function dpsi=dmpsi(x,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma)
dpsi=feval(dfun,x);
he=feval(hf,x);  gi=feval(gf,x);
dhe=feval(dhf,x);  dgi=feval(dgf,x);
l=length(he); m=length(gi);
for(i=1:l)
    dpsi=dpsi+(sigma*he(i)-mu(i))*dhe(:,i);
end
for(i=1:m)
    dpsi=dpsi+(sigma*gi(i)-lambda(i))*dgi(:,i);
end
```

例 9.4 利用程序 9.1 求解约束优化问题

$$\begin{aligned} \min\ \ & f(x)=(x_1-2)^2+(x_2-1)^2, \\ \text{s.t.}\ \ & x_1-2x_2+1=0, \\ & 0.25x_1^2+x_2^2\le 1. \end{aligned}$$

取初始点为 $x_0=(3,3)^T$, 该问题有精确解 $x^*=\Big(\frac{1}{2}(\sqrt{7}-1),\frac{1}{4}(\sqrt{7}+1)\Big)^T\approx(0.82288,0.91144)^T$, $f(x^*)=\frac{1}{4}(\sqrt{7}-5)^2+\frac{1}{16}(\sqrt{7}-3)^2\approx 1.39346$.

解 先编制目标函数文件 f1.m

```
function f=f1(x)
f=(x(1)-2.0)^2+(x(2)-1.0)^2;
```

等式约束函数文件 h1.m

```
function he=h1(x)
he=x(1)-2.0*x(2)+1.0;
```

不等式约束函数文件 g1.m

```
function gi=g1(x)
gi=-0.25*x(1)^2-x(2)^2+1;
```

目标函数的梯度文件 df1.m

```
function g=df1(x)
g=[2.0*(x(1)-2.0), 2.0*(x(2)-1.0)]';
```

等式约束（向量）函数的Jacobi矩阵（转置）文件 dh1.m

```
function dhe=dh1(x)
dhe=[1.0, -2.0]';
```

不等式约束（向量）函数的Jacobi矩阵（转置）文件 dg1.m

```
function dgi=dg1(x)
dgi=[-0.5*x(1), -2.0*x(2)]';
```

然后在 Matlab 命令窗口输入如下命令:

```
x0=[3,3]';
[x,mu,lambda,output]=multphr('f1','h1','g1','df1','dh1','dg1',x0);
```

得到如下输出:

```
x =

    0.8229

    0.9114

mu =

  -1.5945

lambda =

   1.8465

output =

          fval: 1.3934

          iter: 23

    inner_iter: 82

           bta: 9.5419e-006
```