一个使用乘子法的最优化FORTRAN通用程序OPTIMA*

史明仁

（应用数学系）

摘　要

本文简略介绍了使用乘子法解有约束极小问题的一个算法。叙述了通用程序*OPTIMA*的各子程序的功能与所使用的各计算公式，最后给出了对几个标准试验问题的试算结果。

A FORTRAN General Purpose Routine by Multiplier Method for Minimization

Shi Ming-ren

(Department of Applied Mathematics)

Abstract

In this paper, an algorithm of multiplier method for the constrained minimization problems is introduced. The function of each subroutine in the 'OPTIMA' and the various formulas used are described. The numerical results of several standard problems are given at the end of this paper.

一、导　言

乘子法是把惩罚函数和Lagrange函数结合起来，构造增广Lagrange函数，然后通过

本文于1984年4月16日收到。

* 本程序是在邓乃扬先生指导下调试成功，并曾得到唐虎老师、研究生王志伟的协助以及计算机房陈志伟老师大力支持，特在此致谢。

求一系列的无约束极小点来获得原约束问题的解。它克服了惩罚函数法需要把惩罚因子取得很大而带来的病态性质，从而使算法更加有效。

我们调试成功的通用程序 $OPTIMA$ 通用性较好，既可用于解有约束问题（使用乘子法），又可用于解无约束问题。程序用 $FORTRAN$ 语言编成，在 PDP—11/23 微型计算机上调试通过，适用于国内绝大多数大、中、小型计算机。程序中提供了两种形式的增广 Lagrange 函数，以及四种迭代矩阵产生搜索方向的方法（并各带重置策略），用戶可以灵活选择。使用本程序时，用戶只须提供最低限度的三个子程序：①主程序，只有说明各数组的维数语句以及调用语句；②计算问题函数值及其非常数梯度分量值的表达式的两个子程序 $FXNS$ 与 $GRAD$.

一般有约束极小问题在本程序中的标准格式为

$$
\begin{aligned}
&\min f(x), \quad x \in R^n \\
&s.t. \quad p_i(x)=0, \qquad i=1, 2, \cdots, n_E \\
&\qquad\quad q_j(x) \leqslant 0, \qquad j=1, 2, \cdots, n_I \\
&\qquad\quad c_k \leqslant x_k \leqslant d_k, \qquad k=1, 2, \cdots, n \qquad (1\cdot1)
\end{aligned}
$$

自变量上下有界的约束，在理论上是不等约束函数为线性函数 c_k-x_k 或 x_k-d_k 的特殊情况。因实际问题中很多约束属于这一类，所以在程序中把它分立，它的函数及其梯度表达式不必由用戶提供，而只需输入 x_k 是否有上下界约束的标志以及数据 c_k 和 d_k。

使用乘子法求局部极小的算法步骤为：

(1) 算法开始置初值；形成增广 Lagrange 函数 $L_a(x)$[若为无约束问题，则 $L_a(x)=f(x)$]；置 $k=0$，$k_s=0$ [k 为累计一维搜索总次数，k_s 为累计对同一个 $L_a(x)$进行一维搜索的次数]。

(2) 检验收敛准则或停止准则是否满足；若满足，转（10）；否则转（3）。

(3) 计算下降方向 $r^{(k)}$。

(4) 一维搜索求 $x^{(k+1)}=\min\limits_{\rho\geqslant 0} L_a\left(x^{(k)}+\rho r^{(k)}\right)$。

(5) 计算新点 $x^{(k+1)}$处的信息。

(6) 检验收敛准则或停止准则是否满足；若满足，转（10）；否则转（7）[若为无约束问题直接转（9）]。

(7) 检验更新乘子与惩罚因子的条件是否满足；若满足，转（8）；否则置$k_s=k_s+1$，转（9）。

(8) 更新乘子与惩罚因子，形成新的 $L_a(x)$，置 $k_s=0$。

(9) 置 $k=k+1$，转（2）。

(10) 输出最后结果，停止算法。

通用程序 $OPTIMA$ 由 13 个子程序组成，其中

① 子程序 $OPTIMA$ 是控制、编排用的子程序，它组织、编排其它各子程序。

② 子程序 $INITL$ 从数据卡片读入初始数据，置算法初值及各种操作方式，输出初始信息，其中调用子程序 $CONGR$ 输出各问题函数的非零常数梯度分量值。

③　子程序 $DELTA$ 是在算法从 $x^{(k)}$ 出发求出新点 $x^{(k+1)}$ 后，计算梯度增量、自变量增量，及它们的模长、梯度的模长，并更新信息，以便下一次迭代用。

④子程序 $OUTPUT$ 可在用户需要时或停止准则满足以后，输出当前点的信息或最后结果。

其它子程序的功能及所使用的计算公式，分别简单介绍如下。

二、增广 Lagrange 函数及其梯度

1. 增广 Lagrange 函数La(x)

子程序 $AUGLAG$ 提供两种形式不同的增广 Lagrange 函数[1][2]，由用户在读入初值时，置 $l=0$ 或 $l=1$ 来选择（程序中以标识符 LAG 表示 l）。

两种形式的 $L_a(x)$ 可统一为

$$L_a(x)=f(x)+f_L(x)+\omega_1 f_E(x)+\omega_2 f_{I2}(x)+\omega_3 f_{I3}(x)-\frac{1}{4\omega_2}T(x) \qquad (2\cdot1)$$

其中，ω_1 为等式约束惩罚因子，ω_2、ω_3 分别是不等约束相应于乘子为正还是为0的惩罚因子[$l=1$ 时，$\omega_2=\omega_3$]，（2·1）式右端各函数为

$$f_L(x)=\sum_i \alpha_i p_i(x)+\sum_{j\in J_2^{(l)}}\beta_j q_j(x)+\sum_{k\in K_L^{(l)}}\beta_k(c_k-x_k)+$$
$$+\sum_{k\in K_U^{(l)}}\beta_k(x_k-d_k)$$

$$f_E(x)=\sum_i p_i^2(x);$$

$$f_{I2}(x)=\sum_{j\in J_2^{(l)}}q_j^2(x)+\sum_{k\in K_L^{(l)}}(c_k-x_k)^2+\sum_{k\in K^{(l)}}(x_k-d_k)^2$$

$$f_{I3}(x)=\begin{cases}0, & l=1;\\ \sum\limits_{j\in J_3}q_j^2(x)+\sum\limits_{k\in K_{L0}}(c_k-x_k)^2+\sum\limits_{k\in K_{U0}}(x_k-d_k)^2, & l=0\end{cases}$$

$$T(x)=\begin{cases}0, & l=0;\\ \sum\limits_{j\in J_2^*}\beta_j^2+\sum\limits_{k\in K_L^*}\beta_k^2+\sum\limits_{k\in K_U^*}\beta_k^2, & l=1.\end{cases} \qquad (2\cdot2)$$

其中，α_i 与 β_j、β_k 分别为等式约束、不等约束及自变量上下有界约束的乘子。所涉及的下标集合为

$$L^*=\{\,k\,|\,x_k \text{有下界约束}\,\}$$
$$U^*=\{\,k\,|\,x_k \text{有上界约束}\,\} \qquad (2.3)$$
$$J_2^{(1)}=\{\,j\,|\,\beta_j+2\omega_2 q_j(x)>0\,\};\quad J_2^*=\{\,j\,|\,\beta_j+2\omega_2 q_j(x)\leqslant 0\,\}$$
$$K_L^{(1)}=\{\,k\,|\,k\in L^*,\ \beta_k+2\omega_2(c_k-x_k)>0\,\}$$
$$K_L^*=\{\,k\,|\,k\in L^*,\ \beta_k+2\omega_2(c_k-x_k)\leqslant 0\,\}$$
$$K_U^{(1)}=\{\,k\,|\,k\in U^*,\ \beta_k+2\omega_2(x_k-d_k)>0\,\}$$

$$K_U^* = \{ k \mid k \in U^*, \ \beta_k + 2\omega_2(x_k - d_k) \leqslant 0 \} \tag{2.4}$$

$$J_2^{(0)} = \{ j \mid \beta_j > 0 \}; \quad J_3 = \{ j \mid \beta_j = 0, \ q_j(x) > 0 \}$$

$$K_L^{(0)} = \{ k \mid k \in L^*, \ \beta_k > 0 \}; \quad K_U^{(0)} = \{ k \mid k \in U^*, \ \beta_k > 0 \}$$

$$K_{L0} = \{ k \mid k \in L^*, \ c_k - x_k > 0, \ \beta_k = 0 \}$$

$$K_{U0} = \{ k \mid k \in U^*, \ x_k - d_k > 0, \ \beta_k = 0 \} \tag{2.5}$$

可以证明，两种增广 Lagrange 函数仅在 $\beta_j + 2\omega_2 q_j(x) < 0$ 这一种情况下不等价，我们上机试算结果也证实了这一点。

2. 增广 Lagrange 函数的梯度

子程序 $GALAG$ 是用来计算已知点 x 处 $L_a(x)$ 的梯度 $\nabla L_a(x)$ 的各分量值。

由（2.1）式对 x_k 求偏导，可得

$$\frac{\partial L_a(x)}{\partial x_k} = \frac{\partial f(x)}{\partial x_k} + g_L(k) + 2\{ \omega_1 g_E(k) + \omega_2 g_{I2}(k) + \omega_3 g_{I3}(k) \}$$

$$k = (1, 2, \cdots, n) \tag{2.6}$$

其中

$$g_L(k) \triangleq \frac{\partial f_L(x)}{\partial x_k} = \sum_i \alpha_i \frac{\partial p_i(x)}{\partial x_k} + \sum_{j \in J_2^{(l)}} \beta_j \frac{\partial q_j(x)}{\partial x_k} - \sigma_k \beta_k + \tau_k \beta_k;$$

$$g_E(k) \triangleq \frac{1}{2} \frac{\partial f_E(x)}{\partial x_k} = \sum_i p_i(x) \frac{\partial p_i(x)}{\partial x_k};$$

$$g_{I2}(k) \triangleq \frac{1}{2} \frac{\partial f_{I2}(x)}{\partial x_k} = \sum_{j \in J_2^{(l)}} q_j(x) \frac{\partial q_j(x)}{\partial x_k} - \sigma_k(c_k - x_k) + \tau_k(x_k - d_k);$$

$$g_{I3}(k) = \begin{cases} 0, & l = 1; \\ \sum\limits_{j \in J_3} q_j(x) \dfrac{\partial q_j(x)}{\partial x_k} - \sigma_{k0}(c_k - x_k) + \tau_{k0}(x_k - d_k), & l = 0 \end{cases} \tag{2.7}$$

σ_k、τ_k、σ_{k0}、τ_{k0} 分别仅当 $k \in K_L^{(l)}$、$k \in K_U^{(l)}$、$k \in K_{L0}$、$k \in K_{U0}$ 时取值 1，其它情况取值 0。其中涉及的函数与下标集合见（2.2）—（2.5）式。

三、乘子与惩罚因子迭代公式，不满足约束和

子程序 $UPDATE$ 是用来更新乘子与惩罚因子，子程序 SUM 是用来计算不满足约束和。

所用上式对两种 Lagrange 函数是一致的。

1. 乘子迭代公式

等式约束乘子迭代公式为

$$\alpha_i^+ = \alpha_i + 2\omega_1 p_i(x), \quad i = 1, 2, \cdots n_E; \tag{3.1}$$

不等约束乘子迭代公式为

$$\beta_j^+=\begin{cases}\beta_j+2\omega_2 q_j(x), & j\in J_2^{(1)}\cap J_2^{(0)};\\ 2\omega_3 q_j(x), & j\in J_3;\\ 0, & \text{其它情况。}\end{cases}\tag{3.2}$$

由（3.2）式可推出自变量上下有界约束的乘子迭代公式为

$$\beta_k^+=\begin{cases}\beta_k+2\omega_2(c_k-x_k), & k\in K_L^{(0)}\cap K_L^{(1)};\\ \beta_k+2\omega_2(x_k-d_k), & k\in K_U^{(0)}\cap K_U^{(1)};\\ 2\omega_3(c_k-x_k), & k\in K_{L0}\\ 2\omega_3(x_k-d_k), & k\in K_{U0}\\ 0, & \text{其它情况。}\end{cases}\tag{3.3}$$

2 惩罚因子迭代公式

$$\omega_i^+=\begin{cases}\omega_f\cdot\omega_i, & \text{当 } \omega_f\cdot\omega_i\leqslant\omega_{i\max}\\ \omega_{i\max}, & \text{当 } \omega_f\cdot\omega_i>\omega_{i\max}\end{cases}\qquad i=1,\ 2,\ 3\tag{3.4}$$

三个惩罚因子初值 ω_i 及其上界 $\omega_{i\max}$（$i=1,\ 2,\ 3$），倍因子 ω_f 均从数据卡片读入。$l=1$ 时，应使 $\omega_3=\omega_2$，$\omega_{3\max}=\omega_{2\max}$。

3 不满足约束和

不满足约束和 σ 可由下式确定：

$$\sigma^2=\sum_i p_i^2(x)+\sum_j\left[\max\left\{q_j(x),\ -\frac{\beta_j}{2\omega_2}\right\}\right]^2+$$
$$+\sum_{k\in L^*}\left[\max\left\{c_k-x_k,\ -\frac{\beta_k}{2\omega_2}\right\}\right]^2+$$
$$+\sum_{k\in U^*}\left[\max\left\{x_k-d_k,-\frac{\beta_k}{2\omega_2}\right\}\right]^2\tag{3.5}$$

σ 的大小是衡量 x 偏离可行域的程度。$\sigma<\varepsilon_1$，且梯度模长 $\|g\|<\varepsilon_1$，自变量增量的模长 $\|\Delta x\|<\varepsilon_8$（$\varepsilon_1$、$\varepsilon_8$ 为预给精度）这三个条件满足时，迭代收敛。

四、拟牛顿法迭代矩阵与计算搜索方向

子程序 RV 是根据由 $x^{(k)}$ 沿方向 $r^{(k)}$（初始值取负梯度方向 $-g^{(0)}$）搜索得 $x^{(k+1)}$ 时的下列信息：①对称正定矩阵 $H^{(k)}$（$H^{(0)}=I$）；②梯度增量 $y^{(k)}=\nabla L_a(x^{(k+1)})-\nabla L_a(x^{(k)})=g^{(k+1)}-g^{(k)}$；③自变量增量 $s^{(k)}=x^{(k+1)}-x^{(k)}$，用 Broyden 类[8]拟牛顿公式来迭代矩阵，然后取拟牛顿方向为下一个搜索方向。此时，矩阵的迭代公式为（省去上标 k）

$$H^+=H-\frac{Hy(Hy)^T}{y^THy}+\theta vv^T+\frac{ss^T}{s^Ty},\ (0\leqslant\theta\leqslant1)$$

$$v=(y^{T}Hy)^{\frac{1}{2}}\left(\frac{s}{s^{T}y}-\frac{Hy}{y^{T}Hy}\right) \tag{4.1}$$

本子程序中提供了两种公式：DFD 公式（$\theta=0$）与 $BFGS$（$\theta=1$）公式。

若在（4.1）中以 γH（$\gamma>0$）代替 H 所得公式为自选因子（self－scaling）的拟牛顿公式：[4][5][6]：

$$H^{+}=\left(H-\frac{Hy(Hy)^{T}}{y^{T}Hy}+\theta vv^{T}\right)\gamma+\frac{ss^{T}}{s^{T}y} \tag{4.2}$$

子程序 RV 中使用了两种自选因子策略：

$$DFP/SS：\theta=0.5,\ \gamma=\frac{g^{T}s}{g^{T}Hy}$$

$$BFGS/SS：\theta=1.0,\gamma=\frac{s^{T}y}{y^{T}Hy} \tag{4.3}$$

子程序 RV 提供了两种使用自选因子的方式：标识符 $ISS=1$，表示仅仅在第1次迭代后，即由 $H^{(0)}$ 求 $H^{(1)}$ 时采用[5]，而 $ISS=2$ 则是每次迭代矩阵时均采用(4.2)式。$ISS=0$ 表示不使用自选因子，而用（4.1）式。

迭代矩阵以后，取拟牛顿方向为下一个搜索方向，即

$$r^{(k+1)}=-H^{(k+1)}g^{(k+1)} \tag{4.4}$$

子程序RV对各种迭代矩阵的方法均设置了重置（$RESET$）策略。对二次正定函数来说，用拟牛顿方法最多迭代 n 次，应得到它的极小点。对一个非二次正定的函数 $f(x)$来说，若迭代 n 次以后仍未得到极小点，意味着还未进入 $f(x)$的二次区域，因此可认为继续使用已得到的信息迭代矩阵不见得好，可考虑把矩阵重置为单位阵 I，重取负梯度方向为搜索方向，这就是重置策略。

五、单向一维搜索

三个子程序 $DMIN$，$VALUE$，$SEARCH$ 互相配合完成从点 $x^{(k)}$ 出发，沿下降方向 $r^{(k)}$ 的单向一维搜索，即求最优步长 ρ_s，使

$$\overline{y}(\rho_s)=\min_{\rho\geq 0}\overline{y}(\rho)=\Delta\min_{\rho\geq 0}L_a(x^{(k)}+\rho r^{(k)}) \tag{5.1}$$

我们用下列二次函数来局部近似$\overline{y}(\rho)$：

$$y(\rho)=a(\rho-d_1)^2+b(\rho-d_1)+c \tag{5.2}$$

其中，$d_1\geq 0$ 是从 $x^{(k)}$ 出发沿 $r^{(k)}$ 方向的一个已知步长，而 a、b、c 三个常数则是根据三个已知数据拟合得到。子程序 $DMIN$ 用到两种拟合公式：点斜点拟合与三点拟合，什么情况下使用哪一种拟合由子程序$SEARCH$加以控制。确定 a、b、c 以后，若 $a>0$（$y''(\rho)=2a>0$），（5.2）式确定的二次函数有唯一极小点 ρ^*，它是 $y'(\rho)=0$ 的解，即

$$\rho^*=d_1-\frac{b}{2a} \tag{5.3}$$

1. **点斜点拟合**

已知点$d_1=0$（即出发点$x^{(k)}$）处函数值$y_1=\overline{y}(0)$，导数值$y'(0)=\nabla La(x^{(k)})^T r^{(k)}$，以及点$d_2$处函数值$y_2=\overline{y}(d_2)$，即可由（5.2）式得

$$\begin{aligned} & y_1=c, \\ & y_2=ad_2^2+bd_2+c, \\ & y'(0)=b \end{aligned} \tag{5.4}$$

由上定出a、b、c，代入（5.3）式，得极小点处：

$$\rho^*=-\frac{b}{2a}=0.5\cdot\frac{[-y'(0)d_2]\cdot d_2}{[-y'(0)d_2]+(y_2-y_1)} \tag{5.5}$$

2. **三点拟合**

若已知三点d_i及其函数値$y_i=\overline{y}(d_i)$（$i=1, 2, 3$），记

$$d_{31}=d_3-d_1,\quad d_{21}=d_2-d_1 \tag{5.6}$$

则由（5.2）式可得

$$\begin{aligned} & y_1=c \\ & y_2=ad_{21}^2+bd_{21}+c \\ & y_3=ad_{31}^2+bd_{31}+c \end{aligned} \tag{5.7}$$

解出a、b、c，代入（5.3）式，知极小点处：

$$\rho^*=d_1-0.5\frac{d_{21}\tau_2-d_{31}\tau_1}{\tau_1-\tau_2},$$

$$\tau_1=d_{31}(y_2-y_1),\quad \tau_2=d_{21}(y_3-y_1)。 \tag{5.8}$$

六、试 算 结 果

我们在PDP—11/23微型机上试算了四个无约束试验问题与七个有约束试验问题。限于篇幅，只列出一些主要结果。

T1：Banana function

$$\min f(x)=\sum_{k=1}^{N-1}[100(x_{k+1}-x_k^2)^2+(1-x_k)^2],\ N=2, 6, 10, 16, 30, 50,$$

100；初始点$x^{(0)}=(-1.2, 1, -1.2, 1, \cdots)^T$，$\varepsilon_1=\varepsilon_2=\varepsilon_3=10^{-7}$；极小点$x^*=(1, 1, 1, 1, \cdots, 1)^T$，$f(x^*)=0$。

T2：Biggs' function

$$\min f(x)=\sum_{i=1}^{10}(e^{-x_1 z_i}-x_3 e^{-x_2 z_i}-y_i)^2;$$

$$y_i=e^{-z_i}-5e^{-10z_i};\ z_i=(0.1)i,\ (i=1, 2\cdots, 10);$$

$x^{(0)}=(1, 2, 1)^T$，$\varepsilon_1=\varepsilon_2=\varepsilon_3=10^{-4}$；$x^*=(1, 10, 5)^T$，$f(x^*)=0$。

T3：Powell's function

$$\min f(x)=(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-x_4)^4$$

$x^{(0)}=(3,\ -1,\ 0,\ 1)^T$，$\varepsilon_1=\varepsilon_2=10^{-6}$，$\varepsilon_3=10^{-5}$，$x^*=(0,\ 0,\ 0,\ 0)^T$，$f(x^*)=0$。

T4：

$$\min f(x)=\sum_{i=1}^{N} i x_i^2,\quad N=10,\ 30,\ 50。$$

$x^{(0)}=(1,\ 1,\ 1,\ 1\cdots)^T$，$\varepsilon_1=\varepsilon_2=10^{-7}$，$\varepsilon_3=10^{-6}$，$x^*=(0,\ 0,\ 0,\ \cdots,\ 0)^T$，$f(x^*)=0.$

这四个无约束问题的结果见表6—1，其中k表示迭代次数，k_f、k_g分别表示调用子程序$FXNS$或$GRAD$的次数。

表 6.1

问　题	方　　法	k	k_f	k_g
T1，$N=2$	DFP，$BFGS$，$DFP/RESET$	35	99	36
	$BFGS/RESET$	34	118	35
	DFP/SS，$BFGS/SS(ISS=1,\ 2)$	31	100	32
$N=6$	DFP/SS，$BFGS/SS(ISS=1,\ 2)$	55	135	56
$N=10$	DFP/SS，$BFGS/SS(ISS=1,\ 2)$	71	166	72
$N=16$	DFP，$BFGS$，$DFP/RESET$	164	428	165
	DFP/SS	84	183	85
T2	$BFGS$	160	385	161
	$BFGS/SS\ (ISS=2)$	148	344	149
	$BFGS/RESET$	190	451	191
T3	DFP	321	675	322
	$DFP/RESET$	330	698	331
	$BFGS/RESET$	112	307	113
	$BFGS/SS\ (ISS=2)$	180	386	181
T4　$N=10$	DFP，$DFP/RESET$，$BFGS$，$BFGS/RESET$	11	22	12
	DFP/SS，$BFGS/SS(ISS=2)$	14	24	15
$N=30$	DFP，$DFP/RESET$，$BFGS$ $BFGS/RESET$	31	63	32
	DFP/SS，$BFGS/SS(ISS=1)$	30	37	31

T5：Beale's problem

$$\min f(x)=-8x_1-6x_2-4x_3+2x_1^2+2x_2^2+x_3^2+2x_1x_2+2x_1x_3+9;$$

$$s.t.\quad g_1(x)=x_1+x_2+2x_3-3\leqslant 0;$$

$$x_k\geqslant 0,\ k=1,\ 2,\ 3。$$

$x^{(0)}=\left(\frac{1}{2},\ \frac{1}{2},\ \frac{1}{2}\right)^T$，$\omega_1=\omega_{1\max}=0$，$\omega_2=\omega_3=1$，$\omega_{2\max}=\omega_{3\max}=16$，$\omega_f=4$；

$x^*=\left(\frac{4}{3},\ \frac{7}{9},\ \frac{4}{9}\right)^T$，$f(x^*)=\frac{1}{9}$，$\beta^*=\left(\frac{2}{9},\ 0,\ 0,\ 0,\ \right)^T$（注：$x^*$ 若以分数表示，为准确值，以小数表示，则为计算值，α^*，β^* 亦同）

T6：(Powell)

$\min f(x)=x_1x_2x_3x_4x_5$;

$$s.t.\quad p_1(x)=\sum_{i=1}^{5}x_i^2-10=0;$$

$$p_2(x)=x_2x_3-5x_4x_5=0;$$

$$p_3(x)=x_1^3+x_2^3+1=0$$

$x^{(0)}=(-2,\ 1.5,\ 2,\ -1,\ -1)^T$，$\omega_{2,3}=\omega_{2,3\max}=0$，$\omega_1=0.5$，$\omega_{1\max}=8$，$\omega_f=2$。

$x^*=(-1.71719,\ 1.59575,\ 1.82721,\ -0.763621,\ -0.763622)^T$，$f(x^*)=-2.91963$，$\alpha^*=(0.743738,\ -0.705133,\ 0.0969968)^T$。

T7：“around the world” (Pierre)

$\min f(x)=-x_2$

$$s.t.\quad p_1(x)=x_1^2+x_2^2+x_3^2-1=0$$

$$q_1(x)=2x_2-x_1-1\leqslant 0$$

$x^{(0)}=(-0.1,\ -1,\ 0.1)^T$，$\omega_i=0.25$，$\omega_{i\max}=1$，$(i=1,\ 2,\ 3)$，$\omega_f=2$;

$x^*=\left(\frac{3}{5},\ \frac{4}{5},\ 0\right)^T$，$f(x^*)=-\frac{4}{5}$，$\alpha^*=\frac{1}{4}$，$\beta^*=\frac{3}{10}$。

T8：a linear ploblem (Pierre)

$\min f(x)=-0.5x_1-x_2-0.5x_3-x_4$

$$s.t.\quad p_1(x)=x_1+x_2+x_3-2x_4-6=0$$

$$q_1(x)=\sum_{i=1}^{4}x_i-10\leqslant 0$$

$$q_2(x)=0.2x_1+0.5x_2+x_3+2x_4-10\leqslant 0$$

$$q_3(x)=2x_1+x_2+0.5x_3+0.2x_4-10\leqslant 0$$

$$x_k\geqslant 0,\ k=1,\ 2,\ 3,\ 4$$

$x^{(0)}=\left(0,\ 0,\ 0,\ 0\right)^T$，$\omega_i=1$，$\omega_{i\max}=10(i=1,\ 2,\ 3)$，$\omega_f=2$；$x^*=\left(0,\ \frac{26}{3},\ 0,\ \frac{4}{3}\right)^T$，

$f(x^*)=-10$，$\alpha^*=0$，$\beta^*=\left(1,\ 0,\ 0,\ \frac{1}{2},\ 0,\ \frac{1}{2},\ 0\right)^T$

T9：seven variables problem (Pierre)

$$\min f(x)=-5x_1-5x_2-4x_3-x_1x_3-6x_4-\frac{5x_5}{1+x_5}-\frac{8x_6}{1+x_6}-10\left(1-2e^{-x_7}+e^{-2x_7}\right);$$

s.t. $p_1(x)=2x_4+x_5+0.8x_6+x_7-5=0;$

$p_2(x)=x_2^2+x_3^2+x_5^2+x_6^2-5=0;$

$q_1(x)=\sum_{i=1}^{7} x_i-10\leqslant 0;$

$q_2(x)=\sum_{i=1}^{4} x_i-5\leqslant 0;$

$q_3(x)=x_1+x_3+x_5+x_6^2-x_7^2-5\leqslant 0;$

$x_k\geqslant 0,\ k=1,\ 2,\ \cdots,\ 7$。

$x^{(0)}=(0.1,\ 0.1,\ \cdots,\ 0.1)^T$; $\omega_i=1,\ \omega_{i\max}=32\ (i=1,\ 2,\ 3)$; $\omega_f=4$;

$x^*=(3.24190,\ 0,\ 1.63415,\ 0.123980,\ 0.889636,\ 1.24020,\ 2.87021)^T$

$f(x^*)=-44.4688$; $\alpha^*=(-0.315066,\ 0.185237)^T$, $\beta^*=(1.38482,\ 5.24895,\ 0,\ 0,\ 1.63058,\ 0,\ 0,\ 0,\ 0,\ 0)^T$;

T10：(Fiacco and McCormick)

$\min f(x)=\frac{1}{3}(x_1+1)^3+x_2$

s.t. $x_1\geqslant 1,\ x_2\geqslant 0$

$x^{(0)}=(1.125,\ 0.125)^T$, $x^*=(1,\ 0)^T$, $f(x^*)=\frac{8}{3}$, $\omega_1=\omega_{1\max}=0$, $\omega_{2,3}=1$, $\omega_{2,3\max}=64$, $\omega_f=4$。

T11：(Rosen and Suzuki's problem)

$\min f(x)=-5(x_1+x_2)-7(3x_3-x_4)+x_1^2+x_2^2+2x_3^2+x_4^2$

s.t. $q_1(x)=\sum_{i=1}^{4} x_i^2+x_1-x_2+x_3-x_4-8\leqslant 0;$

$q_2(x)=x_1^2+2x_2^2+x_3^2+2x_4^2-x_1-x_4-10\leqslant 0;$

$q_3(x)=2x_1^2+x_2^2+x_3^2+2x_1-x_2-x_4-5\leqslant 0$

$x^{(0)}=(0,\ 0,\ 0,\ 0)^T$, $\omega_1=\omega_{1\max}=0$, $\omega_{2,3}=1$, $\omega_{2,3\max}=16$, $\omega_f=4$;

$x^*=(0,\ 1,\ 2,\ -1)^T$, $f(x^*)=-44$, $\beta^*=(1,\ 0,\ 2)^T$。

以上七个有约束试验问题的主要试算结果见下表。（所有问题的任一个初始乘子均取0）

表 6.2

问题	精度	方法	k	k_f	k_g
T5	$\varepsilon_1=3\times10^{-5}$, $\varepsilon_2=\varepsilon_3=10^{-1}$	$DFP,\ l=0$	9	21	10
	$\varepsilon_1=2\times10^{-4}$, $\varepsilon_2=\varepsilon_3=10^{-1}$	$BFGS,\ l=1$	15	46	16
T6	$\varepsilon_1=3\times10^{-4}$, $\varepsilon_2=10^{-1}$, $\varepsilon_3=10^{-2}$	$DFP,\ l=0$	15	40	16

续上表

T7	$\varepsilon_1=10^{-5}$，$\varepsilon_2=\varepsilon_2=10^{-3}$	DFP ($l=0$, 1)	19	51	20
	$\varepsilon_1=10^{-5}$，$\varepsilon_2=\varepsilon_2=10^{-3}$	$BFGS$ ($l=0$, 1)	19	47	20
T8	$\varepsilon_1=2\times10^{-5}$，$\varepsilon_2=\varepsilon_3=10^{-3}$	DFP, $l=1$	22	53	23
	$\varepsilon_1=2\times10^{-5}$，$\varepsilon_2=\varepsilon_3=10^{-3}$	DFP, $l=0$	21	51	22
	$\varepsilon_1=10^{-4}$，$\varepsilon_2=\varepsilon_3=10^{-2}$	DFP, $l=0$	22	52	23
	$\varepsilon_1=2\times10^{-4}$，$\varepsilon_2=\varepsilon_3=10^{-1}$	$BFGS$, $l=1$	21	50	22
T9	$\varepsilon_1=2\times10^{-4}$，$\varepsilon_2=\varepsilon_3=10^{-2}$	DFP, $l=0$	40	96	41
T10	$\varepsilon_1=10^{-6}$，$\varepsilon_2=\varepsilon_3=10^{-3}$	$BFGS$, $l=1$	17	35	18
	$\varepsilon_1=5\times10^{-6}$，$\varepsilon_2=\varepsilon_3=10^{-2}$	DFP, $l=0$	19	47	20
T11	$\varepsilon_1=2\times10^{-4}$，$\varepsilon_2=\varepsilon_3=10^{-2}$	DFP, $l=1$	20	49	21
	$\varepsilon_1=3\times10^{-4}$，$\varepsilon_2=\varepsilon_3=10^{-2}$	$BFGS$, $l=1$	19	46	20

参 考 文 献

[1] 邓乃扬、诸梅芳，最优化计算方法，第九章，内部资料，1983。

[2] Lowe, M. J., Nonlinear programming: Augmented Lagrangian techniques for constrained minimization, Ph. D. Thesis, Montana State University, Bozeman, Montana, March, 1974.

[3] 邓乃扬等，无约束最优化方法，科学出版社，1982。

[4] Luenberger, D. G., Convergence rate of a penalty-function scheme, Journal of Optimization Theory and Applications, Vol. 7, pp. 39—51, Jan. 1971.

[5] Oren, S. S., Self-scaling variable metric algorithm without linear search for unconstrained minimization, Mathmatics of Computation, Vol. 27, pp. 873—885, Oct. 1973.

[6] Oren, S. S., Self-scaling variable metric(SSVM) algorithm, Management Science, Vol. 20, pp. 845—874, Jan. 1974.