二次优化问题的 SDP 松弛求解算法

范丽君，艾文宝
北京邮电大学运筹学与控制论系，北京（100876）
Email：fanlijun985@gmail.com

摘　要：信赖域方法是求解非线性优化问题的一种十分有效的方法，而信赖域方法在每一步迭代中的核心问题是求解一个信赖域子问题，信赖域子问题可以归结为一类二次优化问题。Sturm 和 Zhang 通过一个秩一分解的方法最先将信赖域子问题和半正定规划联系了起来，证明了无约束信赖域子问题的半正定松弛是紧的，原二次优化问题的最优解可以由松弛问题的最优解得到。Sturm 和 Zhang 的证明虽然是构造性的，但是如何给出一个简单的分解并没有涉及。本文的主要工作则是在 Sturm 和 Zhang 的工作基础上，给出了一个简单的矩阵分解方法来获得原问题的精确解或近似解，并利用 Matlab 软件编程实现。我们给出了此方法的有效性证明，初步的数值结果也表明该方法是有效的。进一步，我们利用同样的方法来求解等式约束优化问题的信赖域子问题（常称为两球问题），初步的数值结果表明，该方法对 95%以上的两球问题都是有效的。

关键词：信赖域方法；信赖域子问题；SDP 松弛；二次优化

1　引言

信赖域方法是非线性优化的一类重要的数值计算方法，它在近二十年受到非线性优化的高度重视，一直是非线性优化的研究热点，关于信赖域方法的第一本专著由 Cohn，Gould 和 Toint[12] 给出。目前，信赖域方法已经和传统的线搜索方法并列为非线性规划的两类主要的数值方法。

信赖域子问题的求解直接影响了信赖域方法的有效性，许多学者对信赖域子问题进行了深入的研究。考虑到对于大规模问题，每次迭代精确求解信赖域子问题的计算量和不必要性，许多学者对如何有效地近似求解信赖域子问题做了深入的研究，主要有 Powell[13]的折线法及在此基础上 Dennis 和 Mei[14]的双折线方法等等。

Strum 和 Zhang 通过一个秩一分解的方法最先将信赖域子问题和半正定规划联系了起来[15]。Strum 和 Zhang 的方法是基于特征值分解的，但我们知道特征值分解并不是一件容易的事情。Strum 和 Zhang 的证明虽然是构造性的，即在理论上证明了通过分解可以得到原问题的精确解，但是如何给出一个简单的分解并没有涉及。本文的主要目的就在于应用半正定规划及其算法，构造一种基于无约束信赖域子问题的二次优化问题的改进型松弛求解方法，并利用相关软件如 SeDuMi 来解答半正定规划问题，然后在此基础上利用求出的半正定的最优解通过矩阵分解的方法来获得原问题的精确解或近似解，并编程实现，得到原二次优化问题的最优解。数值结果表明此方法是有效的。同时作为本文的拓展我们还将处理等式约束优化问题的信赖域子问题（常称为两球问题）的半正定松弛模型，用 SeDuMi 来编程求解，并验证松弛最优解的第一列元素是否为原问题的最优解。

在本文第二部分，我们将给出具体的二次优化问题的 SDP 求解算法；第三部分将给出运行结果数值分析，阐述算法合理性和实用性，加以总结给出论文不足的地方以及还可以改进的地方；第四部分将对文章做出总结和下一步可以做的工作。

2　二次优化问题的 SDP 松弛求解算法

2.1 半正定规划问题介绍

半正定规划问题是一大类圆锥凸规划问题。

定义 2.1 给定 $C \in R^n, A_i \in R^n, i = 1, \cdots, m$ 和 $b \in R^m$，半正定规划问题是为优化问题

$$(SDP)\quad \begin{aligned} &\min C \bullet X \\ &s.t. A_i \bullet X = b_i, i = 1, 2, \cdots, m \\ &X \succeq 0. \end{aligned}$$

找到一个矩阵解 $X \in R^n$.

其中运算$\bullet$是矩阵的内积：$A \bullet B = trA^T B$，A_i 是一个 $m \times m$ 的对称矩阵，$X \succeq 0$ 表示 X 是一个半正定矩阵，$X \succ 0$ 表示 X 是一个正定矩阵。如果一个点 $X \succ 0$ 并且满足 (SDP) 中所有的方程，那么我们称它为一个（原始）严格可行解或内点可行解。

(SDP) 的对偶问题可以写成如下形式：

$$(SDD)\quad \begin{aligned} &\max b^T y \\ &s.t. \textstyle\sum_i^m y_i A_i + S = C \\ &S \succeq 0 \end{aligned}$$

它类似于线性规划 (LP) 的对偶问题 (LD)。其中 $y \in R^m, S \in R^n$。如果一个点 $(y, S \succ 0)$ 满足 (SDD) 中的所有方程，那么它称为一个对偶内点可行解。

由于半正定矩阵的子集在 $n \times n$ 中构成一个锥,所以半正定规划可作为一种特殊的凸规划求解,线性规划的内点法被成功地用来求解半定规划问题, *Strum* 等人在文献[16]中给出了一个长步长路径跟随算法,该算法的迭代复杂性为 $O(n \log \frac{1}{\varepsilon})$,其中, ε 为算法精度, *Jiang* 给出了一个长步长原始对偶路径跟随算法,该算法在至多 $O(n|\ln \varepsilon|)$ 次迭代内可以找到一精度为 ε 的最优解。其他参见相关文献[17-20],在此篇中用的是 *JosF.Strum* 所提出的 *SeDuMi* 算法。

SeDuMi 是 *Matlab* 的一个附加工具包，可以用来解线性规划，二次规划和半正定规划问题。*SeDuMi* 能够处理复值数据和变量。并且，大规模的优化问题都能够通过矩阵分解的方法得到有效的解决。

SeDuMi 代表 *Self − Dual − Minimization* :它在自对偶齐次锥上执行优化问题的自对偶插入技术。这个技术是由 *Ye,Yodd* 和 *Mizuno* 提出的，本质上它能够在一个单一的方向上解决某一优化问题：或者得到一个最优解，或者证明原问题的不可行性。自对偶齐次锥上的优化问题，或更准确的说，对称锥上的优化问题，最初是由 *Nesterov* 和 *Todd* 研究的，现在已经成为一个非常活跃的研究领域。半正定规划是对称锥上的规划的一类特殊情形。*Vandenberghe* 和 *Boyd* 提出的著名的软件包 SP 是求解半正定规划的第一个软件工具。

2.2 无约束优化问题的信赖域子问题的松弛问题及算法实现

Strum, *Zhang*, *Ye* 和 *Ai* 等已给出了二次优化问题的半正定松弛规划模型,详细内容请参考[15]。本文中我们主要考虑基于无约束信赖域子问题的二次优化问题和等式约束信赖域子问题的二次优化问题的半正定松弛规划模型。

2.2.1 松弛问题模型

我们考虑无约束优化问题的信赖域子问题：

$$(Q) \quad \text{Minimize} \quad g_k^T x + \tfrac{1}{2} x^T B_k x = \varphi_k(x), x \in R^n \tag{2-1}$$

$$\text{Subject to } \|x\| \le \Delta_k . \tag{2-2}$$

我们应用如下记号：

$$M(q_0) := \begin{bmatrix} 0 & g_k \\ g_k^T & B_k \end{bmatrix}, M(q_1) := \begin{bmatrix} -\Delta_k & 0 \\ 0 & I \end{bmatrix}$$

然后，(Q) 可以写成如下等价形式：

$$\text{(Q)} \quad \text{Minimize} \quad M(q_0) \bullet \begin{bmatrix} t \\ x \end{bmatrix} \begin{bmatrix} t \\ x \end{bmatrix}^T = g_k^T d + \tfrac{1}{2} d^T B_k d$$

$$\text{Subject to} \quad M(q_1) \bullet \begin{bmatrix} t \\ x \end{bmatrix} \begin{bmatrix} t \\ x \end{bmatrix}^T = x^T x - \Delta_k \le 0,$$

$$t^2 = 1$$

(Q) 等价于如下问题： Minimize $M(q_0) \bullet X$

$$\text{Subject to} \quad \begin{array}{l} M(q_1) \bullet X \le 0 \\ I_{00} \bullet X = 1 \\ X \succeq 0, \\ rank(X) = 1. \end{array}$$

所以我们写出 (Q) 的松弛问题为：

$$\text{(SP)} \quad \text{Minimize} \quad M(q_0) \bullet X$$

$$\text{Subject to} \quad \begin{array}{l} M(q_1) \bullet X \le 0 \\ I_{00} \bullet X = 1 \\ X \succeq 0, \end{array}$$

其中 $I_{00} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \in S^{(n+1)}$。

不失一般性，我们可以假定 $\Delta_k = 1$，即：$M(q_1) = \begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix}$，

所以 (Q) 的松弛问题为： (SP) Minimize $M(q_0) \bullet X$

$$\text{Subject to} \quad \begin{array}{l} \begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X \le 0 \\ I_{00} \bullet X = 1 \\ X \ge 0 \end{array}$$

通过求解此半正定规划 $SeDuMi$ 得到的解记为 $X^* = (x_0, x_1, \cdots x_n)$。

2.2.2 算法实现

求解问题(SP)的程序实现过程如下：

输入目标函数的系数矩阵的维数 n，通过以下程序语句：

```
A = randn(n);
for i=1:n
    for j=i+1:n
        A(j,i)=A(i,j);
    end
end
```

将产生一个随机对称矩阵 A，然后通过调用我们自己用 SeDuMi 编写的函数 SDP_sedumi(n)就会得到(SP)的一个最优解 X^*。

2.3 由松弛问题的最优解求原问题的解的方法及算法实现

在这一小节中我们先介绍Strum和Zhang给出的由松弛问题的最优解求原问题的最优解的方法（简称 SZ 方法），然后给出我们构造的由松弛问题的最优解求原问题的最优解的方法，我们认为此方法是 SZ 方法的一个更简便的改进型算法。

2.3.1 SZ 方法

为了了解 SZ 方法我们首先给出如下定理(证明见[21])：

定理 2.3.1. 如果一个矩阵方程 $A \bullet X = b$ 存在一个半正定的解 $X \succeq 0$，则存在 X 的一个秩一分解：$X = x_1x_1^T + x_2x_2^T + \cdots + x_rx_r^T$，使得 $A \bullet x_ix_i^T = \dfrac{b}{r}$，其中 $r = rank(X)$。

我们考虑如下问题：(SP_1) Minimize $M(q_0) \bullet X$

$$\text{Subject to}\quad \begin{aligned} &\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X \le 0, \\ &I_{00} \bullet X = 1, \\ &X \ge 0, \end{aligned}$$

设 (y_0^*, y_1^*, S) 为 (SP_1) 的对偶最优解，即拉格朗日函数相应的拉格朗日乘子，SZ 方法分如下两种情形求原问题的最优解：

情形 1：若 $y_1^* = 0$，

则 $M(q_0) \bullet X^* = M(q_0) \bullet (x_1x_1^T + x_2x_2^T + \cdots + x_rx_r^T) \le 0,$ 无 妨 设 $M(q_0) \bullet x_1x_1^T \le 0,$

则 $x_1x_1^T$ 是 (SP_1) 的一个最优解，x_1 是原问题的一个最优解。

证明参见文献[21]。

情形 2：若 $y_1^* > 0$，由定理 2.2.1 可知存在 X^* 的一个秩一分解：

$X^* = x_1^*x_1^{*T} + x_2^*x_2^{*T} + \cdots + x_r^*x_r^{*T}$，

使得 $M(q_0) \bullet x_i^*x_i^{*T} = 0$ 成立，记 $x_i^* = \left(x_{i1}^*, x_{i2}^*, \cdots, x_{in+1}^*\right)^T$，令 $x_i^* := \dfrac{x_i^*}{x_{i1}^*}$，则 $x_i^*x_i^{*T}$ 是 (SP_1) 的一个最优解，x_i^* 是原问题的一个最优解。

证明参见文献[21]。

2.3.2 我们构造的改进型算法及程序实现

以上为 *Sturm* 和 *Zhang* 给出的由松弛问题的最优解求原问题的最优解的方法（SZ 方法），但是在定理 2.3.1 中 X 的秩一分解是基于特征值分解的方法，我们都知道特征值分解是一个比较困难的事情。*Sturm* 和 *Zhang* 的证明虽然是构造性的，即 *Sturm* 和 *Zhang* 在理论上证明了通过分解的方法可以得到原问题的精确解，但是如何给出一个简单的分解并没有涉及。基于此种想法，我们构造了如下的由松弛问题的最优解求原问题的最优解的方法，我们的算法实际上是一个更简便的改进型算法。

为方便后文讨论，我们首先给出如下引理。

引理 2.3.1. $X^* - x_0 x_0^T$ 为半正定矩阵。

证明：由于 $X^* \geq 0$，所以对于任意的 $Y \in R^n$， $Y^T XY \geq 0$，

$$Y^T XY = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} y_i y_j = x_{11} y_1^2 + 2\sum_{j=2}^{n} x_{1i} y_1 y_j + \sum_{i=2}^{n}\sum_{j=2}^{n} x_{ij} y_i y_j \, (x_{11} = 1)$$

$$= (y_1 + \sum_{j=2}^{n} x_{1i} y_j)^2 + (\sum_{i=2}^{n}\sum_{j=2}^{n} x_{ij} y_i y_j - (\sum_{j=2}^{n} x_{1i} y_j)^2)$$

$$= (y_1, y_2, \cdots, y_m)\begin{pmatrix} 1 \\ x_{12} \\ \vdots \\ x_{1m} \end{pmatrix}(1, x_{12}, \cdots, x_{1m})\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} + (\sum_{i=2}^{n}\sum_{j=2}^{n} x_{ij} y_i y_j - (\sum_{j=2}^{n} x_{1i} y_j)^2) \geq 0$$

由于 $\sum_{i=2}^{n}\sum_{j=2}^{n} x_{ij} y_i y_j - (\sum_{j=1}^{n} x_{1i} y_j)^2$ 与 y_1 无关，所以 $\sum_{i=2}^{n}\sum_{j=2}^{n} x_{ij} y_i y_j - (\sum_{j=1}^{n} x_{1i} y_j)^2 \geq 0$，即

$$Y^T XY - (y_1, y_2, \cdots, y_m)\begin{pmatrix} 1 \\ x_{12} \\ \vdots \\ x_{1m} \end{pmatrix}(1, x_{12}, \cdots, x_{1m})\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = Y^T (X - \begin{pmatrix} 1 \\ x_{12} \\ \vdots \\ x_{1m} \end{pmatrix}(1, x_{12}, \cdots, x_{1m}))Y \geq 0.$$

因此 $X^* - x_0 x_0^T$ 为半正定矩阵。

记： $X^* - x_0 x_0^T = B$。

我们分两种情形给出求原问题的最优解的方法：

情形 1： $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* < 0$，则 $x = x_0$ 就是原问题的解。

情形 2： $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* = 0$

记： x_m 为 B 的对角线元素最大的列

方法如下：(1)令 $x = x_0$，若 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet x_0 x_0^T > 0$，则原问题无可行解， $x = \phi$，

转(3)；若 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet x_0 x_0^T < 0$ ，则转(2)，否则转(3)；

(2)令 $x = x_0 + \alpha x_m$，求解 α 满足方程：

$$\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet xx^T = 0 \tag{2-3}$$

(3)输出 x。

则此方法求出的解 x 为原问题的最优解。

综上所述，我们得到算法 2.3.1：

算法 2.3.1 (由松弛问题的最优解得到原问题的最优解的算法)

步 1　给定 X^*，记 x_0 是 X^* 的第一列元素，令 $x = x_0$，若 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* < 0$ ，则转步 5；若 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* = 0$ ，则转步 2，否则转步 4。

步 2　若 $y_1^* \begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet xx^T < \varepsilon$ ，则转步 5；否则，转步 3；

步 3　计算 $B = X^* - x_0 x_0^T$，搜索矩阵 B 的对角线元素最大的列，记为 x_m，令 $x = x_0 + \alpha x_m$，求解 α 满足方程：$\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet xx^T = 0$,转步 5；

步 4　$x = \phi$；

步 5　输出 x。

算法 2.3.1 的程序实现过程如下：

取程序一得到的半正定矩阵 X^* 的第一列元素为 x_0，令 $B = X^* - x_0 x_0^T$，记 B 的对角线元素最大的列为 x_m，我们通过调用我们自己编写的 *Matlab* 函数 solution(A, X)即可得到原问题的一个近似最优解 x,obj_value_2，返回值 t 是算法 2.3.1 步 3 中所需求解的方程的根。

2.4 等式约束优化的信赖域子问题的松弛问题及算法实现

考虑 CDT 子问题：

$$\begin{aligned} &\min_{x \in R^n} \phi(d) = \frac{1}{2} d^T B d + g^T d \\ &s.t. \quad \|d\| \le \Delta \\ &\qquad \|A^T d + c\| \le \xi \end{aligned} \tag{2-4}$$

其中 $B \in R^{n \times n}$ 是对称矩阵，$g \in R^n$，$A \in R^{n \times m}$，$c \in R^m$，$\Delta > 0$，$\xi \ge 0$。CDT 子问题的提出是为了克服信赖域方法求解等式约束优化问题时产生的约束不相容性。问题(2.4.1)也通常称为两球问题。与一球问题情形类似，我们得到式(2-4)的松弛问题为：

(SP_2) minimize $M(q_0) \bullet X$

$$\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X \le 0$$

Subject to $M(q_1) \bullet X \le 0$

$$I_{00} \bullet X = 1$$

$$X \ge 0$$

求解此半正定松弛问题的算法实现过程如下：

输入松弛问题中矩阵的维数 n，程序将通过如下语句随机产生对称矩阵 A,B 作为目标函数的系数矩阵和约束函数的系数矩阵，

```
A = randn(n);
for i=1:n
    for j=i+1:n
        A(j,i)=A(i,j);
    end
end
B = randn(n);
for i=1:n
    for j=i+1:n
        B(j,i)=B(i,j);
    end
end
```

然后通过调用我们自己用 SeDuMi 编写的函数 SDP_sedumi2(n)，将会输出两球问题的半正定松弛近似最优解 X 和最优值 obj_value_1，我们模仿一球问题的方法，选取 X 的第一列作为原问题的最优解，得到最优值 obj_value_2,我们比较 obj_value_1 和 obj_value_2 可以发现它们之间有时会存在间隙，也就是说对于两球问题，松弛问题的最优解的第一列不一定是原问题的最优解。

3 实验结果及改进方法

3.1 程序结果及可行性分析

我们对算法(2.3.1)进行了数值实验，我们取误差精度 $\xi = 10^{-5}$，将目标函数的系数矩阵的维数 n 由 3 取到 50，系数矩阵 $M(q_0)$ 均是随机选取的，在目标函数的系数矩阵维数 n 固定的情况下，对每个 n 在程序中都运行了 50 次，也就是说我们得到的最少值是在运行 50 次的情况下求得的平均值，我们可以这么认为这样求得的最少值误差很少，精度很大这是由于目标函数的系数矩阵的随机选取性，每次求得的最少值并不是一样的，于是选取次数越多，得到的均值更稳定，在运行 50 次的情况下形成的误差就可以忽略不计了。

现在将实验结果以列表的形式给出，如表 3-1：

首先我们给出表格中符号的含义：

n：矩阵 $M(q_0)$ 的维数；

x：原问题的精确或近似最优解；

α： 方程(2.2.10)中的系数；

obj_value_1：半正定松弛问题 (SP_1) 的最优值；

obj_value_2：由算法(2.2.1)得到的原问题的最优值；

$A-error$：绝对误差，定义如下：

$A-error=\left|obj\_value\_1-obj\_value\_2\right|$

$R-error$：相对误差，定义如下：

$$R-error=\left|\frac{obj\_value\_1-obj\_value\_2}{obj\_value\_1}\right|$$

表 3-1　算法(2.2.1)的计算结果

n	α	obj_value1	obj_value2	$A-error$	$R-error$ (%)
3	0	-1.37804	-1.37789	0.00001	0.0022
4	0	-1.06879	-1.06876	0.00011	0.0115
5	0	-0.64399	-0.64403	0.00004	0.0062
6	0	-1.42610	-1.42600	0.00010	0.0070
7	0	-1.70200	-1.70190	0.00010	0.0059
8	0	-3.01760	-3.01770	0.00010	0.0033
9	0	-7.15905	-7.15905	0	0
10	0	-7.90609	-7.90610	0.00001	0.0001
11	0	-2.72563	-2.72563	0	0
12	0	-2.12842	-2.12842	0	0
13	0	-2.83829	-2.83829	0	0
14	0	-2.74798	-2.74798	0	0
15	0	-8.82595	-8.82595	0	0
16	0	-9.54804	-9.54804	0	0
17	0	-8.53155	-8.53155	0	0
18	0	-11.47457	-11.47457	0	0
19	0	-11.40672	-11.40672	0	0
20	0	-10.67842	-10.67842	0	0
21	0	-10.76604	-10.76604	0	0
22	0	-11.18850	-11.18850	0	0
23	0	-12.86765	-12.86765	0	0
24	0	-13.47377	-13.47377	0	0
25	0	-13.85712	-13.85712	0	0
26	0	-13.53292	-13.53292	0	0
27	0	-10.82861	-10.82861	0	0
28	0	-14.62733	-14.62733	0	0
29	0	-14.07048	-14.07048	0	0
30	0	-13.76482	-13.76482	0	0
31	0	-17.99273	-17.99273	0	0
32	0	-12.84932	-12.84932	0	0
33	0	-16.47754	-16.47754	0	0
34	0	-14.10405	-14.10405	0	0
35	0	-14.71126	-14.71126	0	0
36	0	-15.56217	-15.56217	0	0
37	0	-18.70129	-18.70129	0	0
38	0	-18.77235	-18.77235	0	0
39	0	-13.32069	-13.32069	0	0
40	0	-12.87292	-12.87292	0	0
41	0	-16.15789	-16.15789	0	0
42	0	-20.84618	-20.84618	0	0
43	0	-17.88472	-17.88472	0	0
44	0	-16.58939	-16.58939	0	0
45	0	-16.47033	-16.47033	0	0
46	0	-19.11430	-19.11430	0	0
47	0	-17.05413	-17.05413	0	0
48	0	-19.05056	-19.05056	0	0
49	0	-17.41869	-17.41869	0	0
50	0	-22.40385	-22.40385	0	0

表 3-1 中的结果我们可以从图 3-1 中得到更直观的体现：

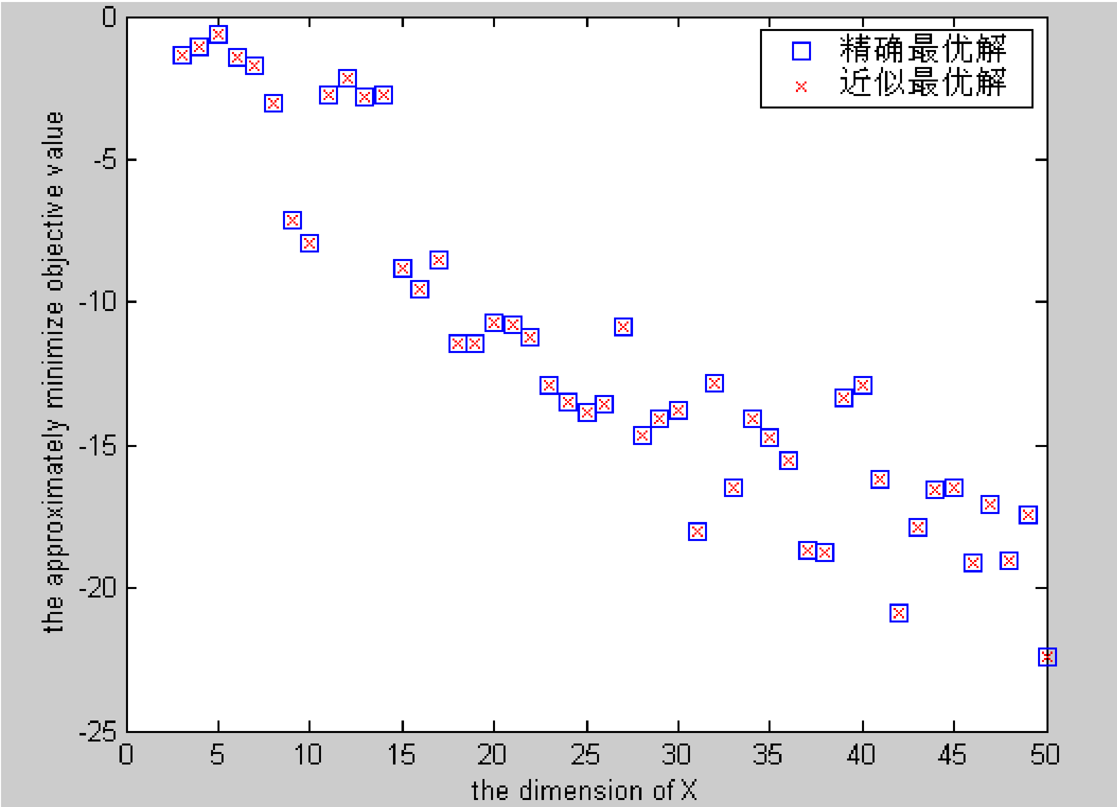

图 3-1　近似最优值与精确最优值的比较

下面我们从绝对误差角度来观察算法 2.3.1 得到的解与原问题的最优解的近似程度(如图 3-2)，图中折线是以目标函数的系数矩阵的维数为横坐标，以算法 2.3.1 得到的解与原问题的最优解的比值为竖坐标得到的曲线，直线是算法 2.3.1 的绝对误差的平均值，由图可知算法 2.3.1 的绝对误差小于 1.0001，绝对误差的平均值为 1.00027，并且绝对误差随着目标函数的系数矩阵的维数的增加而减小。由以上实验结果我们可以发现由算法 2.3.1 得到的原二次优化问题的最优解的相对误差非常小，达到几乎可以忽略的程度，因此我们认为此算法是有效的。同时我们在大量的实验当中，发现α 以非常大的概率等于零，至少我现在还没有找出一个例子使得$\alpha \neq 0$，我认为这并不是一个巧合，所以在绝大多数情况下原问题的最优解就是半正定松弛问题的最优解的第一列元素，因此用此算法求原二次优化问题的最优解具有很高的效率。

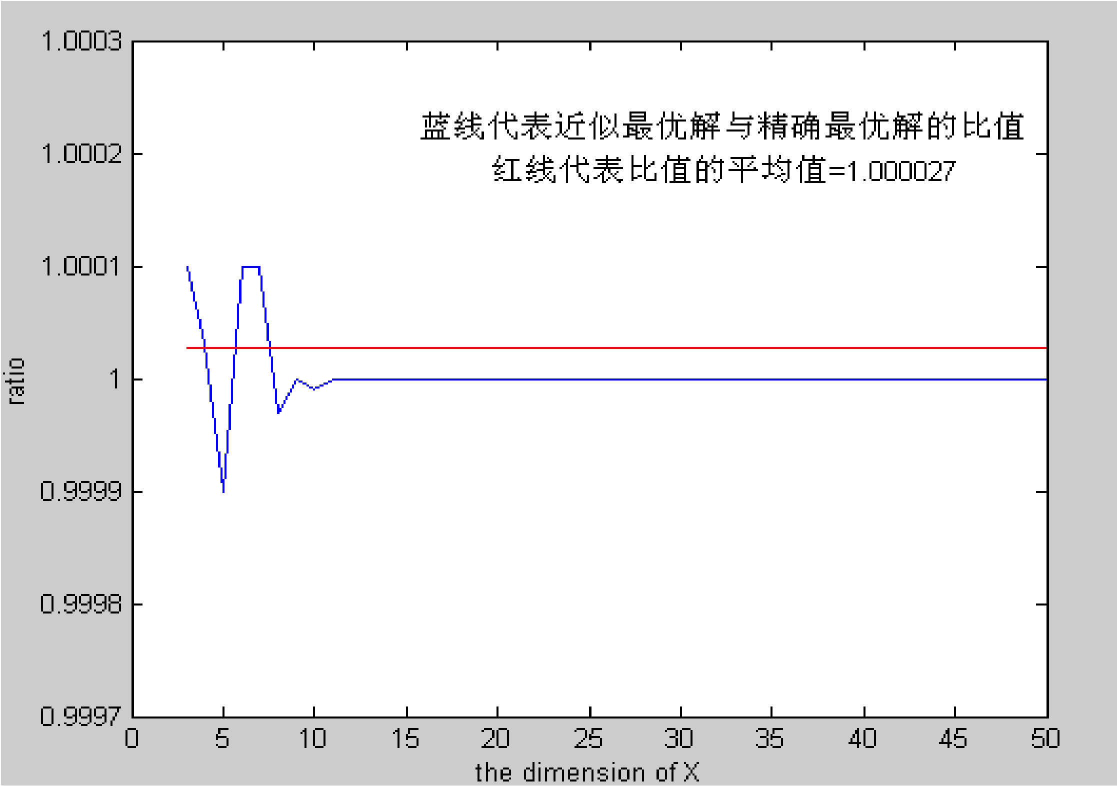

图 3-2 近似最优解的误差

对于两球问题，我们同样选取误差精度$\xi = 10^{-5}$，我们应用程序三求解出半正定松弛问题的最优解 X 和最优值obj_value_1，由无约束信赖域子问题情形的启发，我们选取 X 的第一列元素作为原两球问题的近似最优解，得出相应的最优值为obj_value_2，我们将要考察obj_value_1和obj_value_2之间是否存在间隙，即对于两球问题，是否也满足在绝大多数情况下，松弛问题的半正定解的第一列元素即为原问题的最优解？我们随机选取 50 个实例，调用程序三得出结果，见表 3-2：

表 3-2 两球问题的半正定松弛的紧性

n	obj_value1	obj_value2	$A-error$	$R-error$ (%)
3	-2.4824	-2.4824	0	0
4	**-1.8342**	**-1.4021**	**0.4321**	**23.56%**
5	-4.4781	-4.4781	0	0
6	-6.1231	-6.1231	0	0
7	**-5.8816**	**-1.6208**	**4.2608**	**72.44%**
8	-8.3032	-8.3032	0	0
9	-7.1825	-7.1825	0	0
10	-6.7392	-6.7392	0	0
11	-7.6866	-7.6866	0	0
12	-7.0112	-7.0112	0	0
13	-10.4580	-10.4580	0	0
14	-8.8765	-8.8765	0	0
15	-6.3248	-6.3248	0	0
16	-8.3328	-8.3328	0	0
17	-9.6358	-9.6358	0	0
18	-14.7458	-14.7458	0	0
19	-10.0677	-10.0677	0	0
20	-10.3814	-10.3814	0	0
21	-12.6114	-12.6114	0	0
22	-11.8449	-11.8449	0	0
23	-11.2662	-11.2662	0	0
24	-11.3620	-11.3620	0	0
25	-12.1161	-12.1161	0	0
26	-12.2222	-12.2222	0	0
27	-13.3907	-13.3907	0	0
28	-13.1583	-13.1583	0	0
29	-17.2629	-17.2629	0	0
30	-14.9473	-14.9473	0	0
31	**-13.1550**	**-9.2009**	**3.9541**	**30.06%**
32	-16.0706	-16.0706	0	0
33	-16.4704	-16.4704	0	0
34	-18.9023	-18.9023	0	0
35	-16.5130	-16.5130	0	0
36	-13.3098	-13.3098	0	0
37	-19.0420	-19.0420	0	0
38	-16.5440	-16.5440	0	0
39	-17.1839	-17.1839	0	0
40	**-17.4017**	**-16.3393**	**1.0624**	**6.11%**
41	-15.8990	-15.8990	0	0
42	18.7659	18.7659	0	0
43	-17.2345	-17.2345	0	0
44	-18.7331	-18.7331	0	0
45	-19.1202	-19.1202	0	0
46	-15.9977	-15.9977	0	0
47	-20.2692	-20.2692	0	0
48	-18.1447	-18.1447	0	0
49	-19.7878	-19.7878	0	0
50	-19.8653	-19.8653	0	0

为了更直观的体现，我们将表 3-2 中的实验数据结果以图表的形式给出，见图 3-3：

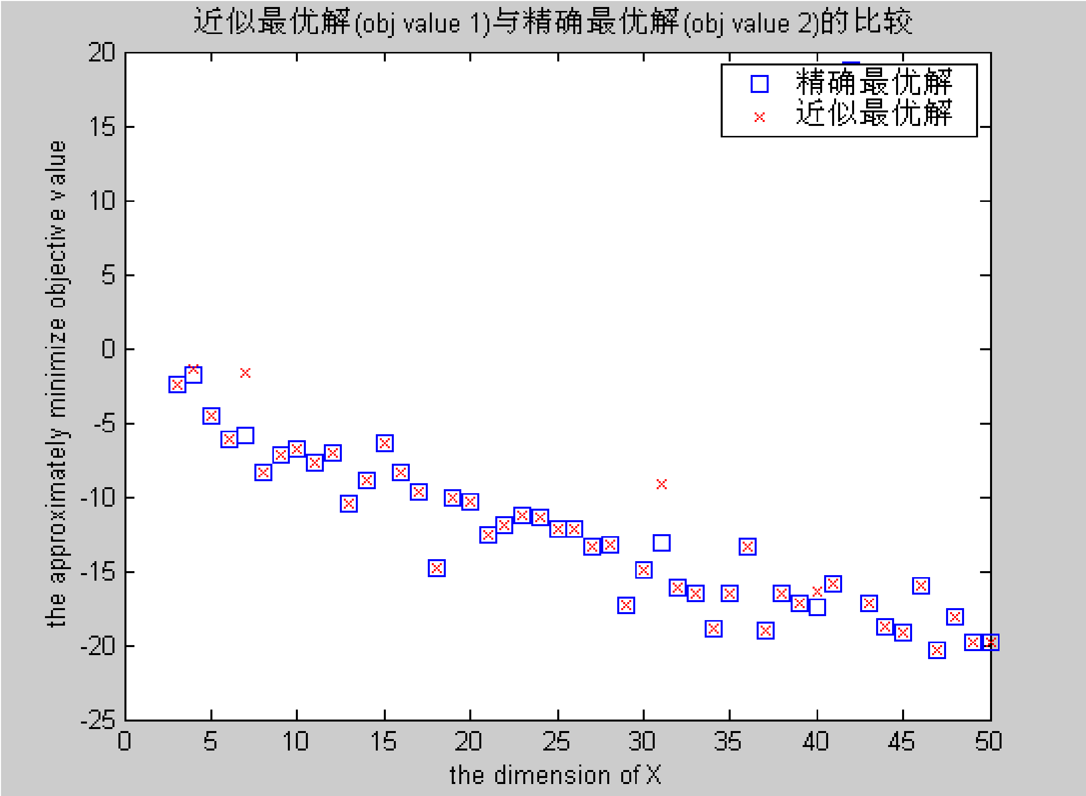

图 3-3　两球问题的半正定松弛最优解和近似最优解的比较

由图 3.1.5，我们可以看出，选取两球问题的半正定松弛解的第一列元素作为原问题的最优解是不可以的，他们之间会存在间隙，也就是说半正定松弛不一定总是紧的。因此为了利用(SDP)松弛方法求解两球问题，我们还需要构造由松弛问题的最优解 X 求原问题的近似最优解的方法，这将是我们后续的工作。

3.2 算法的有效性证明

通过上一小节的实验结果，我们可以发现用我们的算法求解二次优化问题是非常有效的。在这一小节中，我们将要在理论上证明此方法的有效性。

Sturm 和 *Zhang* [23]中已经证明了(Q_1)的松弛问题(SP_1)是紧的，即(Q_1)的最优值和(SP_1)的最优值相等，因此为证明此算法的有效性，只需证明由算法（2.3.1）得到的原二次优化问题的最优值与(SP_1)的最优值相差很小。

考虑松弛问题： (SP_1)　Minimize　$M(q_0)\bullet X$

$$\text{Subject to}\quad \begin{aligned} &\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X \le 0, \\ &I_{00} \bullet X = 1, \\ &X \ge 0, \end{aligned}$$

其拉格朗日函数为：

$$L(X,y) = M(q_0)\bullet X + y_1\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix}\bullet X - y_0(I_{00}\bullet X - 1) - S\bullet X$$

设 X^* 为(SP_1)的最优解，相应的拉格朗日乘子为 y_0^*, y_1^*, S^*。

由(KKT)一阶必要条件可知，X^* 和 y_0^*, y_1^*, S^* 需满足如下方程组：

$$S^* = M(q_0) + y_1\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} - y_0 I_{00} \ge 0, \tag{3-1}$$

$$y_1^* \ge 0, \quad y_1^*\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* = 0, \tag{3-2}$$

$$S^* \bullet X^* = 0, \tag{3-3}$$

$$I_{00} \bullet X^* = 0, \tag{3-4}$$

$$\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* \le 0. \tag{3-5}$$

设 x 是由算法(2.3.1)得到的结果，$X^* = (x_0^*, x_1^*, \cdots, x_n^*)$，下面我们只需验证 xx^T 满足条件式(3-1)-式(3-5)成立即可，我们分如下两种情况讨论：

情形 1：$\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* < 0$，由算法 2.2.1 可知 $x = x_0^*$。

证明：(1) 式(3-1)与 x 无关，因此式(3-1)成立。

由互补松紧条件可知 $y_1^* = 0$，因此式(3-2)成立。

由 $S^* \bullet X^* = S^* \bullet (x_0^*, x_1^*, \cdots, x_n^*) = 0$, 可知 $x_i^*, i = 0, \cdots, n$ 是 S^* 的零子空间，即 $S^* \bullet x_i^* x_i^{*T} = 0$，因此式(3-3)成立。

(4) $X = xx^T = x_0 x_0^T$,满足 $X_{00} = 1$，因此式(3-4)成立。

(5) 由于 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* \le 0$，所以 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet (x_0 x_0^T + B) \le 0$

由于 $B \ge 0$,且 B 的第一行与第一列元素均为零，所以 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet B \ge 0$，

所以 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet x_0 x_0^T \le 0$，因此式(3-5)成立。

由矩阵分析知识我们可以知道 $X = xx^T = x_0 x_0^T$ 的秩为一，因此 x 是原二次规划问题的最优解。

情形 2：$\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet X^* = 0$，式(3-1)和式(3-3)可用同样方法验证，式(3-4)和式(3-5)也很容易得到验证，我们只需验证在此种情形下，由算法 2.3.1 得出的结果满足式(3-2)即可。

由算法 2.3.1 可知当 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet x_0^* x_0^{*T} = 0$ 时，$x = x_0^*$，此时 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet xx^T = 0$ 成立，因此式(3-2)成立；

当 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet x_0^* x_0^{*T} < 0$ 时，我们构造 x 满足方程 $\begin{bmatrix} -1 & 0 \\ 0 & I \end{bmatrix} \bullet xx^T = 0$，因此(3-2)自然

成立。

综上所述，算法 2.3.1 得到的结果 x 满足式(3-1)-(3-5)，因此可证明此算法是有效的。

3.3 程序的可塑性

该程序只有一个输入变量，只要 n 做相应的修改，就可以得出相应的结果，操作简便，而且由于目标函数的系数矩阵是随机选取的，因此更具有普遍性和说服力。

3.4 程序的意义

本程序提供了一种从半正定规划问题的角度求解基于无约束信赖域子问题和等式约束信赖域子问题的二次优化问题的方法，并且具有很好的有效性，由此程序求出的原问题的近似最优值和精确最优值误差非常的小，达到几乎可以忽略不计的程度。

4 总结与展望

本文实现了基于无约束优化信赖域子问题的二次优化问题的半正定松弛求解方法，重点描述了算法的实现过程，程序可行性和有效性，主要从矩阵分解方面来设计源程序，在矩阵分解的支持下，使得算法对实际有着一定的指导意义。*Sturm* 和 *Zhang* 采用秩一分解的方法最先建立了信赖域子问题和半正定规划之间的联系，提出了通过半正定规划求解信赖域子问题的思想，*Sturm* 和 *Zhang* 的证明虽然是构造性的，即从理论上证明了通过分解可以得到原问题的精确解，但是如何给出一个简单的分解并没有涉及，基于此种想法，我们构造了一种简单的改进型基于信赖域子问题的二次优化问题的半正定松弛求解算法，我们能够能够在理论上证明此方法是有效的，并且数值结果也表明此算法是有效的。

但本文只彻底给出了无约束优化信赖域子问题的半正定松弛求解方法，对于两球问题如何由松弛问题的解求原问题的最优解还并没有得出结果，这将是我接下来的工作。

参考文献

[1]袁亚湘，孙文瑜著《最优化理论与方法》科学出版社，1997.01:559-561.
[2]D.M. Gay, Computing optimal local constrained step, SIAM J. Sci. Stat. Comp.,2(1981),186-197.
[3]J.J. Mo e' , and D.C. Sorensen, Computing a trust region step, SIAM J. Sci. Stat. Comp.,4(1983),553-572.(1983b)
[4]M.D.Hebden, An algorithm for minimization using exact second derivatives,Tech.Rep.T.P.515, A.E.R.E., Theoretical physics division, Harwell, Berkshire, England,1973
[5]M.J.D. Powell, A new algorithm for unconstrained optimization, In: J.B. Rosen, O.L. Mangassarian and K. Ritter, eds., Nonlinear Programming, Academic press, New York, 1970, 31-66.
[6]J.E. Dennis and H.H.W. Mei, Two new unconstrained optimization algorithms which use function and Gradient values, J. Opt. Theory and Appl.,28(1979),453-482.
[7]F. Rendl and H. Wolkowicz, A semidefinite framework for trust region subproblem with applications to large scale minimization, Math. Prog., 77(1997),273-299
[8]T. Steihaug, The conjugate gradient method and trust regions in large scale optimization, SIAM. J Numer.Anal.,7 (1997),141-161.
[9]Ph. L. Toint, Towards an efficient sparsity exploiting Newton method for minimization, In: I.S. Duff, ed., Sparse matrices and their uses (Academic Press, London, 1981), 57-88.
[10]Y. Yuan, on the truncated conjugate gradient method, Math. Prog., 87(2000),561-571.(2000b) .
[11]D.C. Sorensen, Minimization of a large-scale quadratic function subject to a sphere constraint, SIAM K.Opt.,7(1997),141-161.
[12]A.R. Conn, N. Gould and P.L. Toint, Trust-region Methods, SIAM, Phiadelphia, USA,2000.
[13]M.J.D. Powell, A new algorithm for unconstrained optimization, In: J.B. Rosen, O.L. Mangassarian and K. Ritter, eds., Nonliear Programming, Academic press, New York, 1970, 31-66.
[14]J.E. Dennis and H.H.W. Mei, Two new unconstrained optimization algorithms which use function and

Gradient values, J. Opt. Theory and Appl.,28(1979),453-482.
[15]J.F. Sturm and S. Zhang, On cones of nonnegative quadratic functions, Mathematics of Operations Research 28(2003), 246-267.
[16]Sturm Jos F, Zhang Shuzhong. On the long-step path-following method for semidefinite Programming [J]. Operations Research Letters, 1998, 22: 1452150.
[17]Faybusovich L. Semidefinite programming: a path-following algorithm for a linear quadratic functional
[J]. S IAM J. Optimization, 1996, 6 (4) : 100721024.
[18]N esterov Yu, N emirovakii A S. Interior point polynomial algorithms in convex programming [A].In:
SIAM Stud. App l. Math. , Vo l. 13[C]. S IAM , Philadephia, U SA , 1994.
[19]Vandenberghe L ,Boyd S. Semidefinite programming[J]. S IAM Review, 1996, 38: 49295.
[20]A lizadeh F. Interior point methods in semidefinite programming with applications to combinatorial optimization [J]. S IAM J. Optm.1995, 5: 13251.
[21]J.F. Strum and S. Zhang, On cones of nonnegative quadratic functions, Mathematics of Operations Research 28(2003), 246-267.

COMPUTING QUADRATIC OPTIMIZATION PROBLEMS BY SDP RELAXATION

Fan Lijun，Ai Wenbao
Department of Optimization and Control, Beijing University of Post and Telecommunications, Beijing, PRC, (100876)

Abstract
Trust region method is a kind of the most important and effective numerical methods for nonlinear programming. As we know, the key in each iteration for the trust region method is to solve a trust region subproblem, which is a kind of quadratic optimization problem. Strum and Zhang proved that we can obtain a solution of a trust region problem using Semidefinite Programming(SDP) relaxation and matrix rank-one decomposition technique. Although the proof presented by Strum and Zhang is constructive, how to give a simple and practical decomposition is yet a problem. In this paper, we present a simple and practical matrix decomposition, based on the Sturm and Zhang's work. We prove that this decomposition is correct and effective in theory. The preliminary numerical results show that the method is effective. Furthermore, we use the similar method to solve constrained trust region subproblems (usually named as two-ball problem). The preliminary numerical results show that the method is effective for 95 % of instances.
Keywords: trust region method; trust region subproblem; SDP relaxation; quadratic optimization