二阶锥面约束优化问题及其求解方法

目录

1 原优化问题

考虑如下带有二阶锥面约束的优化问题：

$$
\begin{aligned}
\min_{x\in\mathbb{R}^n} \quad & c^T x \\
\text{s.t.} \quad & Ax = b \\
& \|x_i\|_2 = t_i, \quad i = 1, 2, \ldots, m
\end{aligned} \tag{1}
$$

其中：

- $x = (x_1^T, x_2^T, \ldots, x_m^T, t_1, t_2, \ldots, t_m)^T \in \mathbb{R}^n$ 为决策变量
- $c \in \mathbb{R}^n$ 为目标函数系数向量
- $A \in \mathbb{R}^{p\times n}$，$b \in \mathbb{R}^p$ 定义线性等式约束
- $x_i \in \mathbb{R}^{n_i}$ 为第 i 个锥变量分量
- $t_i \in \mathbb{R}$ 为第 i 个锥的高度变量
- $\|x_i\|_2 = t_i$ 定义二阶锥的面约束（boundary constraint）

注记 1. 二阶锥面约束 $\|x_i\|_2 = t_i$ 要求点 (x_i, t_i) 必须位于二阶锥的表面，而不是内部。这与标准的二阶锥约束 $\|x_i\|_2 \le t_i$ 不同。

定义 1 (二阶锥面)**.** 对于给定的 $x_i \in \mathbb{R}^{n_i}$ 和 $t_i \in \mathbb{R}$，二阶锥面定义为：

$$
\mathcal{S}^{n_i} = \{(x_i, t_i) \in \mathbb{R}^{n_i+1} : \|x_i\|_2 = t_i,\ t_i \ge 0\}
$$

问题 (1) 是非凸的，因为二阶锥面约束是非凸约束。直接求解该问题在计算上具有挑战性。

2 松弛后优化问题

为了求解原问题，我们首先考虑其凸松弛形式。将二阶锥面约束 $\|x_i\|_2 = t_i$ 松弛为二阶锥约束 $\|x_i\|_2 \le t_i$，得到：

$$
\begin{aligned}
\min_{x\in\mathbb{R}^n} \quad & c^T x \\
\text{s.t.} \quad & Ax = b \\
& \|x_i\|_2 \le t_i, \quad i = 1, 2, \ldots, m
\end{aligned} \tag{2}
$$

定理 1. 问题 (2) 是一个标准的二阶锥规划（*Second-Order Cone Programming, SOCP*）问题，是凸优化问题，可以通过内点法等高效算法求解。

定义 2 (二阶锥). 对于给定的 $x_i \in \mathbb{R}^{n_i}$ 和 $t_i \in \mathbb{R}$，二阶锥（*Lorentz cone*）定义为：

$$\mathcal{K}^{n_i} = \{(x_i, t_i) \in \mathbb{R}^{n_i+1} : \|x_i\|_2 \leq t_i\}$$

注记 2. 松弛问题 (2) 的可行域包含原问题 (1) 的可行域，即原问题的可行域是松弛问题可行域的子集。因此，松弛问题的最优值提供了原问题最优值的下界。

设 x^*_{relax} 为松弛问题 (2) 的最优解。如果 x^*_{relax} 满足所有的面约束 $\|x^*_i\|_2 = t^*_i$，则 x^*_{relax} 也是原问题的最优解。否则，需要进一步处理。

3 基于方向的线性割平面方法

当松弛问题的解不满足面约束 $\|x_i\|_2 = t_i$ 时，通常有 $\|x_i\|_2 < t_i$（点在锥内部）。本节介绍一种基于方向的纯线性割平面方法，通过在不同方向上添加线性割约束逐步收紧可行域。

3.1 Cauchy-Schwarz 不等式

Cauchy-Schwarz 不等式是线性代数中的基本不等式，在二阶锥规划中具有重要应用。对于任意向量 $u, v \in \mathbb{R}^n$，有：

$$|u^T v| \leq \|u\|_2 \|v\|_2 \tag{3}$$

等号成立当且仅当 u 和 v 线性相关（即存在实数 λ 使得 $u = \lambda v$ 或 $v = \lambda u$）。

不等式 (3) 可以等价地写作 $(u^T v)^2 \leq (u^T u)(v^T v)$。在二阶锥约束 $\|x_i\|_2 \leq t_i$ 中，Cauchy-Schwarz 不等式提供了方向投影的上界。

引理 1 (方向投影的上界). 设 (x_i, t_i) 满足二阶锥约束 $\|x_i\|_2 \leq t_i$。对于任意方向向量 $d \in \mathbb{R}^{n_i}$，由 *Cauchy-Schwarz* 不等式可得：

$$\frac{d^T x_i}{\|d\|_2} \leq \|x_i\|_2 \leq t_i \tag{4}$$

证明. 由 Cauchy-Schwarz 不等式：$|d^T x_i| \leq \|d\|_2 \|x_i\|_2$，因此 $d^T x_i \leq \|d\|_2 \|x_i\|_2$（注意 $d^T x_i$ 可能为负，但上界仍成立）。除以 $\|d\|_2 > 0$（若 $d = 0$，不等式平凡成立），得 $\frac{d^T x_i}{\|d\|_2} \leq \|x_i\|_2$。结合锥约束 $\|x_i\|_2 \leq t_i$，即得结论。 □

该引理表明，对于锥内部的点，沿任意方向 d 的投影均不超过 t_i。这为构造方向性线性割约束提供了基础。

3.2 带收缩因子的方向割

为了更有效地收紧可行域，我们引入收缩因子 $r \in [0,1]$，构造如下线性割：

$$\frac{d^T x_i}{\|d\|_2} \geq r \cdot t_i \tag{5}$$

定理 2 (收缩割的有效性). 当 $r = 1$ 时，约束 (5) 要求 $\frac{d^T x_i}{\|d\|_2} \geq t_i$。由于 *Cauchy-Schwarz* 不等式给出 $\frac{d^T x_i}{\|d\|_2} \leq \|x_i\|_2 \leq t_i$，当 $r = 1$ 时，仅有沿方向 d 的锥面点 $x_i = \frac{t_i}{\|d\|_2} d$ 满足约束（此时等号成立）。当 $r < 1$ 时，约束放松，保留更多的锥面和锥内部点。

注记 3. 约束 (5) 的几何意义：

- 在超球面 $x_i^T x_i = t_i^2$ 上，该约束定义了一个半空间的下界
- 当 $r = 1$ 时，约束最紧：只有在方向 d 上的锥面点满足（$d^T x_i = t_i \|d\|$）
- 当 $r < 1$ 时，约束放松：保留方向 d 附近一个范围内的锥面点
- r 越小，保留的角度范围越大；r 越大（接近 *1*），保留的范围越小
- 通过选择不同的 r 值，可以控制割的紧度

3.3 收缩因子 r 的几何分析

我们分析收缩因子 r 与实际被割掉区域的关系。

3.3.1 二维情况分析

考虑二维情况 $x_i = (x_1, x_2)^T \in \mathbb{R}^2$，方向 $d = (d_1, d_2)^T$。约束 (5) 变为：

$$\frac{d_1 x_1 + d_2 x_2}{\sqrt{d_1^2 + d_2^2}} \geq r \cdot t \tag{6}$$

锥面上的点：设 $x_1^2 + x_2^2 = t^2$（锥面），参数化为 $x_1 = t\cos\theta$，$x_2 = t\sin\theta$。
设方向 d 对应角度 α（即 $d_1 = \|d\|\cos\alpha$，$d_2 = \|d\|\sin\alpha$），则：

$$\frac{d^T x}{\|d\|_2} = t(\cos\alpha\cos\theta + \sin\alpha\sin\theta) = t\cos(\theta - \alpha) \tag{7}$$

约束 (5) 要求：

$$\cos(\theta - \alpha) \geq r \tag{8}$$

定理 3 (二维锥面上的可行角度). 对于二维锥面上的点，约束 (5) 保留的角度范围为：

$$\theta \in [\alpha - \arccos(r), \alpha + \arccos(r)] \tag{9}$$

当 $r = 1$ 时，仅保留方向 d 本身（$\arccos(1) = 0$，退化为一点）；当 $r \to 0$ 时，保留以方向 d 为中心的 $180°$ 范围（$\arccos(0) = \pi/2$，即半圆）；当 $r = -1$ 时，保留整个圆周（$\arccos(-1) = \pi$）。

证明. 由 $\cos(\theta-\alpha) \geq r$，得 $|\theta-\alpha| \leq \arccos(r)$。因此可行角度为 $\theta \in [\alpha-\arccos(r), \alpha+\arccos(r)]$。 □

几何解释（二维）：

- $r=1$：约束 $d^T x \geq t\|d\|$，仅保留方向 d 本身（$\arccos(1)=0°$）
- $r=1/\sqrt{2} \approx 0.707$：$\arccos(r)=45°$，保留 $90°$ 角度范围
- $r=1/2$：$\arccos(r)=60°$，保留 $120°$ 角度范围
- $r=0$：$\arccos(r)=90°$，保留 $180°$ 角度范围（以方向 d 为中心的半圆）

3.3.2 三维情况分析

考虑三维情况 $x_i=(x_1,x_2,x_3)^T \in \mathbb{R}^3$，方向 $d \in \mathbb{R}^3$。锥面为球面 $x_1^2+x_2^2+x_3^2=t^2$。

约束 (5) 在球面上定义了一个球冠（spherical cap）。设 γ 为 x_i 与 d 之间的夹角，则：

$$\cos\gamma = \frac{d^T x_i}{\|d\|_2\|x_i\|_2} = \frac{d^T x_i}{\|d\|_2 t} \tag{10}$$

约束变为：$\cos\gamma \geq r$，即：

$$\gamma \leq \arccos(r) \tag{11}$$

定理 4 (三维球冠体积比)**.** *在三维球面上，约束 (5) 保留的球冠占球面面积的比例为：*

$$\text{比例} = \frac{1-r}{2} \tag{12}$$

证明. 球冠的面积比例为 $\frac{1}{2}(1-\cos\theta_0)$，其中 $\theta_0 = \arccos(r)$ 是球冠的半顶角。约束 $\cos\gamma \geq r$ 即 $\gamma \leq \arccos(r)$ 保留了与方向 d 夹角不超过 θ_0 的球冠，其面积比例为 $\frac{1-r}{2}$。 □

几何解释（三维）：

- $r=1$：保留 0% 的球面（仅方向 d 本身，$\arccos(1)=0$）
- $r=0.5$：保留 25% 的球面（一个球冠）
- $r=0$：保留 50% 的球面（以方向 d 为中心的半球）
- $r=-1$：保留 100% 的球面（整个球面）

注记 4. *从上述分析可见：*

- *较大的 r 值（接近 1）产生更紧的割，仅保留方向 d 附近的锥面点*
- *较小的 r 值产生较松的割，保留更多方向的锥面点*
- *实际应用中，r 的选择需要平衡收敛速度和可行性保证*
- *层次化策略：先用较小的 r 值覆盖较大范围，再逐步增大 r 值精确逼近锥面*

3.4 层次化方向细分与收缩因子策略

为了系统地逼近锥面，我们采用层次化方向细分策略，并在每层使用适当的收缩因子 r。

3.4.1 第一层：象限等分线方向（$r = r_{min}$）

在第一层，我们使用象限的等分线方向，收缩因子 $r = r_{min}$（较小值，如 0.5 或 0.6），以覆盖较大的角度范围。对于二维情况：

$$\begin{aligned} d^{(1)} = (1,1)^T: &\quad \frac{x_{i1} + x_{i2}}{\sqrt{2}} \geq r_{min} \cdot t_i \\ d^{(2)} = (-1,1)^T: &\quad \frac{-x_{i1} + x_{i2}}{\sqrt{2}} \geq r_{min} \cdot t_i \\ d^{(3)} = (-1,-1)^T: &\quad \frac{-x_{i1} - x_{i2}}{\sqrt{2}} \geq r_{min} \cdot t_i \\ d^{(4)} = (1,-1)^T: &\quad \frac{x_{i1} - x_{i2}}{\sqrt{2}} \geq r_{min} \cdot t_i \end{aligned} \tag{13}$$

3.4.2 第二层：1/2 象限细分（r 适度增大）

在第二层，我们在相邻方向之间插入新方向，并使用适度增大的收缩因子。对于第一象限：

$$\begin{aligned} d^{(1,1)} = (2,1)^T: &\quad \frac{2x_{i1} + x_{i2}}{\sqrt{5}} \geq r_2 \cdot t_i \\ d^{(1,2)} = (1,2)^T: &\quad \frac{x_{i1} + 2x_{i2}}{\sqrt{5}} \geq r_2 \cdot t_i \end{aligned} \tag{14}$$

其中 $r_2 > r_{min}$，例如当 $r_{min} = 0.5$ 时，可取 $r_2 \approx 0.7$。

3.4.3 第 k 层：一般化策略

定义 3 (层次化收缩因子). *在第 k 层，我们使用收缩因子：*

$$r_k = r_{min} + \frac{k-1}{K_{max} - 1} \cdot (1 - r_{min}) \tag{15}$$

其中 K_{max} 是最大层数，$r_{min} \in [0.5, 0.7]$ 是初始收缩因子。这使得收缩因子从 r_{min} 逐渐递增到 1。

3.5 自适应添加割的算法

基于上述分析，我们设计一个自适应添加割平面的算法。

Algorithm 1 自适应方向割平面方法

输入：问题 (1) 的数据；容差 $\epsilon > 0$；最大层数 L_{max}；$r_{min} = 0.5$

输出：近似最优解 x^*

初始化：$k \leftarrow 0$，$\ell \leftarrow 1$，线性割集合 $\mathcal{L} \leftarrow \emptyset$

生成第一层方向集 $\mathcal{D}_1$（所有象限的等分线）

计算第一层收缩因子：$r_1 = r_{min}$

for 每个锥 i 和每个方向 $d \in \mathcal{D}_1$ **do**

  添加割 $\frac{d^T x_i}{\|d\|_2} \geq r_1 \cdot t_i$ 到 $\mathcal{L}$

end for

repeat

  求解带割的松弛问题：

$$
\begin{aligned}
x^{k+1} = \arg\min \quad & c^T x \\
\text{s.t.} \quad & Ax = b \\
& \|x_i\|_2 \leq t_i, \quad i = 1, \ldots, m \\
& \text{所有}\mathcal{L}\text{ 中的线性割}
\end{aligned}
$$

  计算残差：$\Delta_i^{k+1} = t_i^{k+1} - \|x_i^{k+1}\|_2$，$\forall i$

  if $\max_i \Delta_i^{k+1} < \epsilon$ **then**

    break // 收敛到面约束

  end if

  // 自适应添加新割

  for 每个锥 $i = 1, \ldots, m$ **do**

    if $\Delta_i^{k+1} > \epsilon$ **then**

      识别当前点 x_i^{k+1} 所在方向：$\hat{d}_i = x_i^{k+1} / \|x_i^{k+1}\|_2$

      在方向 $\hat{d}_i$ 附近查找已有割的违反程度

      计算收缩因子：$r = r_{min} + \frac{\ell-1}{L_{max}-1} \cdot (1 - r_{min})$

      在 $\hat{d}_i$ 附近生成新方向 d^{new}（通过 Farey 序列或角度插值）

      添加新割：$\frac{(d^{new})^T x_i}{\|d^{new}\|_2} \geq r \cdot t_i$ 到 $\mathcal{L}$

    end if

  end for

  if 连续多次迭代无改进且 $\ell < L_{max}$ **then**

    $\ell \leftarrow \ell + 1$

    生成第 ℓ 层全局方向集 $\mathcal{D}_\ell$

    计算对应的收缩因子：$r_\ell = r_{min} + \frac{\ell-1}{L_{max}-1} \cdot (1 - r_{min})$

    for 每个锥 i 和每个方向 $d \in \mathcal{D}_\ell$ **do**

      添加割 $\frac{d^T x_i}{\|d\|_2} \geq r_\ell \cdot t_i$ 到 $\mathcal{L}$

    end for

  end if

  $k \leftarrow k + 1$

until $k \geq K_{max}$

注记 5. 算法的关键特点：

- **自适应性**：根据当前解的位置动态添加割
- **层次化**：从粗到细逐步细分方向
- **收缩因子递增**：随层数增加逐步增大 r（从 r_{min} 到 1），使割从较松逐步变紧，精确逼近锥面
- **局部与全局结合**：既在违反点附近添加局部割，也定期添加全局方向割

3.6 方法特点与复杂度分析

注记 6. 基于方向的线性割平面方法具有以下特点：

- **纯线性**：所有割都是形如 $d^T x \geq r \cdot t$ 的线性约束
- **可调节紧度**：通过收缩因子 r 控制割的紧度
- **几何直观**：每个割对应一个方向上的投影约束
- **自适应细分**：根据求解进展决定是否进入下一层
- **低维高效**：对于二维和三维问题特别有效

注记 7. 复杂度分析：

- **方向数量**：第 ℓ 层有 $O(2^{n_i \cdot \ell})$ 个方向
- **约束总数**：经过 L 层后，每个锥有 $O(2^{n_i \cdot L})$ 个线性割
- **逼近精度**：由收缩因子和方向细分共同决定
- **计算成本**：每次迭代求解一个 *SOCP*，规模随割数线性增长

定理 5 (收敛性)**.** *对于给定的容差 $\epsilon > 0$，存在有限层数 L 和适当的收缩因子序列 $\{r_k\}$，使得算法 1 返回的解满足：*

$$\max_i |\|x_i\|_2 - t_i| \leq \epsilon$$

证明. 通过层次化细分，方向集覆盖所有可能的方向。对于每个方向，收缩因子的递减保证了逐步逼近锥面。由于每次添加割都收紧可行域，且 SOCP 求解保证全局最优，因此算法最终收敛。 □

4 差凸规划迭代方法

差凸规划（Difference of Convex Programming, DC Programming）是处理非凸优化问题的有效方法。我们可以将二阶锥面约束改写为差凸形式，并利用松弛问题的解作为初始值进行迭代。

4.1 差凸分解

二阶锥面约束具有多种等价形式，可以从不同角度进行差凸分解。本节介绍三种常用的分解方法。

4.1.1 基本范数形式的差凸分解

二阶锥面约束 $\|x_i\|_2 = t_i$ 可以等价地表示为：

$$\|x_i\|_2 \leq t_i \quad \text{且} \quad t_i \leq \|x_i\|_2 \tag{16}$$

注意到：

- 约束 $\|x_i\|_2 \leq t_i$ 是凸的
- 约束 $t_i \leq \|x_i\|_2$ 等价于 $-\|x_i\|_2 \leq -t_i$，可以改写为 $t_i - \|x_i\|_2 \leq 0$

函数 $g_i(x_i, t_i) = t_i - \|x_i\|_2$ 是凹函数（两个凸函数的差）。因此约束 $g_i(x_i, t_i) \leq 0$ 是非凸的。

4.1.2 平方形式的差凸分解

二阶锥面约束 $\|x_i\|_2 = t_i$ 两边平方后等价于：

$$\|x_i\|_2^2 = t_i^2 \quad \Leftrightarrow \quad x_i^T x_i = t_i^2 \tag{17}$$

这可以改写为：

$$x_i^T x_i \leq t_i^2 \quad \text{且} \quad t_i^2 \leq x_i^T x_i \tag{18}$$

注意到：

- 约束 $x_i^T x_i \leq t_i^2$ 等价于 $\|(x_i, t_i)\|_2 \leq \sqrt{2} t_i$（旋转二阶锥，见下一小节），但更常用的等价形式是：

$$\begin{bmatrix} 2t_i \\ x_i \end{bmatrix} \in \mathcal{Q}^{n_i+1} \quad \text{当且仅当} \quad x_i^T x_i \leq t_i^2 \tag{19}$$

  实际上，这等价于 $\|x_i\|_2 \leq t_i$（在 $t_i \geq 0$ 时）

- 约束 $t_i^2 \leq x_i^T x_i$ 等价于 $x_i^T x_i - t_i^2 \geq 0$

函数 $h_i(x_i,t_i)=x_i^Tx_i-t_i^2$ 是凸函数减凸函数（$x_i^Tx_i$ 是凸的，t_i^2 也是凸的）。约束 $h_i(x_i,t_i)\geq 0$ 等价于 $-h_i(x_i,t_i)\leq 0$，即：

$$t_i^2-x_i^Tx_i\leq 0 \tag{20}$$

函数 $\tilde{g}_i(x_i,t_i)=t_i^2-x_i^Tx_i$ 是凸函数（t_i^2 是凸的）减去凸函数（$x_i^Tx_i$ 是凸的），因此是 DC 函数（凸函数的差）。

注记 8. 平方形式的优点：

- 线性化后得到二次约束，可能更精确
- 在某些应用中（如椭球约束），平方形式更自然
- 避免了范数计算中的不可微问题（当 $\|x_i\|_2=0$ 时）

4.1.3 旋转二阶锥的差凸分解

旋转二阶锥（Rotated Second-Order Cone, RSOC）定义为：

$$\mathcal{Q}_r^{n+2}=\left\{(x,y,z)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}:\|x\|_2^2\leq yz,\, y\geq 0,\, z\geq 0\right\} \tag{21}$$

对于二阶锥面约束 $\|x_i\|_2=t_i$，我们可以引入辅助变量 s_i，将其改写为旋转锥形式：

$$\|x_i\|_2^2=t_i^2 \quad\Leftrightarrow\quad \|x_i\|_2^2\leq t_i\cdot t_i \quad 且 \quad t_i\cdot t_i\leq\|x_i\|_2^2 \tag{22}$$

更一般地，引入辅助变量 $s_i=t_i$，约束变为：

$$\begin{aligned}&x_i^Tx_i\leq t_is_i,\quad t_i\geq 0,\quad s_i\geq 0\\&t_is_i\leq x_i^Tx_i\\&s_i=t_i\end{aligned} \tag{23}$$

注意到：

- 约束 $(x_i,t_i,s_i)\in\mathcal{Q}_r^{n_i+2}$ 即 $x_i^Tx_i\leq t_is_i,t_i\geq 0,s_i\geq 0$ 是凸的
- 约束 $t_is_i\leq x_i^Tx_i$ 等价于 $x_i^Tx_i-t_is_i\geq 0$
- 约束 $s_i=t_i$ 是线性的

函数 $\phi_i(x_i,t_i,s_i)=x_i^Tx_i-t_is_i$ 中：

- $x_i^Tx_i$ 是 x_i 的凸函数
- t_is_i 是凹函数（双线性项在适当约束下）

约束 $\phi_i(x_i, t_i, s_i) \geq 0$ 等价于 $-\phi_i(x_i, t_i, s_i) \leq 0$：

$$t_i s_i - x_i^T x_i \leq 0 \tag{24}$$

注记 9. 旋转二阶锥形式的优点：

- 某些 *SOCP* 求解器对旋转锥有专门优化
- 双线性项 $t_i s_i$ 的线性化可能比 t_i^2 更简单
- 在某些应用中（如几何规划），旋转锥形式更自然
- 当 $s_i = t_i$ 时，退化为标准二阶锥面约束

4.2 线性化与迭代

在差凸规划框架下，我们对凹部分进行线性化。以下分别给出三种分解方法的线性化形式。

4.2.1 基本范数形式的线性化

在当前迭代点 (x_i^k, t_i^k) 处对凹函数 $-\|x_i\|_2$ 进行线性化：

$$-\|x_i\|_2 \approx -\|x_i^k\|_2 - \frac{(x_i^k)^T}{\|x_i^k\|_2}(x_i - x_i^k) \tag{25}$$

因此，约束 $t_i - \|x_i\|_2 \leq 0$ 线性化为：

$$t_i + \frac{(x_i^k)^T}{\|x_i^k\|_2}(x_i - x_i^k) \leq \|x_i^k\|_2 \tag{26}$$

整理后：

$$\frac{(x_i^k)^T}{\|x_i^k\|_2} x_i - t_i \leq \frac{(x_i^k)^T x_i^k}{\|x_i^k\|_2} - t_i^k \tag{27}$$

4.2.2 平方形式的线性化

对于平方形式的约束 $t_i^2 - x_i^T x_i \leq 0$，在点 (x_i^k, t_i^k) 处线性化：
对 t_i^2 在 t_i^k 处线性化：

$$t_i^2 \approx (t_i^k)^2 + 2t_i^k(t_i - t_i^k) \tag{28}$$

对 $x_i^T x_i$ 在 x_i^k 处线性化：

$$x_i^T x_i \approx (x_i^k)^T x_i^k + 2(x_i^k)^T(x_i - x_i^k) \tag{29}$$

因此，约束 $t_i^2 - x_i^T x_i \leq 0$ 线性化为：

$$(t_i^k)^2 + 2t_i^k(t_i - t_i^k) - (x_i^k)^T x_i^k - 2(x_i^k)^T(x_i - x_i^k) \leq 0 \tag{30}$$

简化后：

$$2t_i^k t_i - 2(x_i^k)^T x_i \leq (x_i^k)^T x_i^k - (t_i^k)^2 \tag{31}$$

4.2.3 旋转二阶锥形式的线性化

对于旋转锥形式的约束 $t_i s_i - x_i^T x_i \le 0$（其中 $s_i = t_i$），在点 $(x_i^k, t_i^k, s_i^k = t_i^k)$ 处线性化：

双线性项 $t_i s_i$ 在 (t_i^k, s_i^k) 处线性化：

$$t_i s_i \approx t_i^k s_i^k + s_i^k(t_i - t_i^k) + t_i^k(s_i - s_i^k) \tag{32}$$

由于 $s_i = t_i$，上式变为：

$$t_i^2 \approx (t_i^k)^2 + 2t_i^k(t_i - t_i^k) \tag{33}$$

对 $x_i^T x_i$ 的线性化同平方形式。因此，约束 $t_i s_i - x_i^T x_i \le 0$ 线性化为：

$$(t_i^k)^2 + 2t_i^k(t_i - t_i^k) - (x_i^k)^T x_i^k - 2(x_i^k)^T(x_i - x_i^k) \le 0 \tag{34}$$

这与平方形式的线性化一致（当 $s_i = t_i$ 时）。

注记 10. 三种线性化形式的比较：

- **基本范数形式**：需要计算 $\|x_i^k\|_2$ 和梯度 $(x_i^k)^T/\|x_i^k\|_2$，当 $\|x_i^k\|_2 \approx 0$ 时需要特殊处理
- **平方形式**：避免了范数计算，但引入了二次项的线性化
- **旋转锥形式**：当 $s_i = t_i$ 时与平方形式等价，但提供了更灵活的建模方式

实际应用中，基本范数形式最常用；平方形式在 x_i 接近零点时更稳定。

4.3 算法流程

Algorithm 2 基于差凸规划的迭代方法（基本范数形式）

输入：问题 (1) 的数据 c, A, b；容差 $\epsilon > 0$；最大迭代次数 K_{max}

输出：可行解 x^*

求解松弛问题 (2)，得到初始解 x^0

初始化：$k \leftarrow 0$

repeat

  求解线性化子问题：

$$
\begin{aligned}
x^{k+1} = \arg\min \quad & c^T x \\
\text{s.t.} \quad & Ax = b \\
& \|x_i\|_2 \le t_i, \quad i = 1, \dots, m \\
& \frac{(x_i^k)^T}{\|x_i^k\|_2} x_i - t_i \le \frac{(x_i^k)^T x_i^k}{\|x_i^k\|_2} - t_i^k, \quad i = 1, \dots, m
\end{aligned}
$$

  计算残差：$r^{k+1} = \max_i |\|x_i^{k+1}\|_2 - t_i^{k+1}|$

  if $r^{k+1} < \epsilon$ **then**

    break // 收敛到可行解

  end if

  $k \leftarrow k + 1$

until $k \ge K_{max}$

return x^k

Algorithm 3 基于差凸规划的迭代方法（平方形式）

输入：问题 (1) 的数据 c, A, b；容差 $\epsilon > 0$；最大迭代次数 K_{max}
输出：可行解 x^*
求解松弛问题 (2)，得到初始解 x^0
初始化：$k \leftarrow 0$
repeat
  求解线性化子问题：

$$\begin{aligned} x^{k+1} = \arg\min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & \|x_i\|_2 \le t_i, \quad i = 1, \ldots, m \\ & 2t_i^k t_i - 2(x_i^k)^T x_i \le (x_i^k)^T x_i^k - (t_i^k)^2, \quad i = 1, \ldots, m \end{aligned}$$

  计算残差：$r^{k+1} = \max_i |\|x_i^{k+1}\|_2 - t_i^{k+1}|$
  if $r^{k+1} < \epsilon$ **then**
    break // 收敛到可行解
  end if
  $k \leftarrow k + 1$
until $k \ge K_{max}$
return x^k

定理 6 (收敛性). 如果算法 *2* 或算法 *3* 产生的序列 $\{x^k\}$ 有界，则其任意收敛子列的极限点是问题 (1) 的稳定点。

注记 11. 差凸规划方法的优点包括：

- 每次迭代求解的子问题仍然是凸优化问题（*SOCP*）
- 利用松弛问题的解作为初始值，通常能快速收敛
- 目标函数值单调递减（当可行时）
- 对于某些问题，能够收敛到二阶锥面约束的可行解

注记 12. 三种 *DC* 分解方法的选择建议：

- **基本范数形式**：最常用，几何意义清晰，适用于大多数问题
- **平方形式**：当初始点接近原点时更稳定（避免 $\|x_i\|_2 = 0$ 的奇异性）
- **旋转锥形式**：当使用支持旋转锥的求解器时，可能有计算优势；也可用于更一般的双线性约束

实践中，可以根据问题特点选择最合适的分解方法，或尝试多种方法取最优解。

4.4 投影步骤（可选）

如果迭代后仍未精确满足面约束，可以添加投影步骤：
对每个锥 i，将 (x_i^k, t_i^k) 投影到二阶锥面：

$$\begin{cases} \bar{x}_i = x_i^k \\ \bar{t}_i = \|x_i^k\|_2 \end{cases} \quad \text{或} \quad \begin{cases} \bar{x}_i = \frac{t_i^k}{\|x_i^k\|_2} x_i^k \\ \bar{t}_i = t_i^k \end{cases} \tag{35}$$

根据目标函数值选择更优的投影点。

5 总结

本文研究了带有二阶锥面约束的优化问题及其求解方法：

- **原问题**：带有线性目标函数、线性等式约束和二阶锥面约束的非凸优化问题
- **凸松弛**：将面约束松弛为锥约束，得到可高效求解的 SOCP 问题
- **基于方向的纯线性割平面方法**：利用方向投影添加纯线性约束，通过层次化细分和收缩因子逐步逼近锥面
- **差凸规划方法**：利用 DC 分解和迭代线性化，从松弛解出发求得面约束的可行解

两种方法各有优劣：

- 纯线性割平面方法易于理解和实现，保持问题的凸性，但对高维问题效率较低
- 差凸规划方法收敛快，但求解的是非凸问题，可能收敛到局部解

在实际应用中，可以结合两种方法的优点，设计混合算法以获得更好的性能。