

# 二阶锥松弛交流潮流优化的锥还原方法 (SOCP-AC-OPF-Relaxation-Recovery-Cut-Adding- Method)

## 目录

<b>1</b>	<b>符号表</b>	<b>2</b>
1.1	索引与集合	2
1.2	决策变量	2
1.2.1	支路变量	2
1.2.2	节点变量	2
1.2.3	发电机变量	3
1.2.4	可调负荷变量	3
1.3	网络参数	3
1.4	锥还原相关符号	3
<b>2</b>	<b>优化模型</b>	<b>5</b>
<b>3</b>	<b>基于方向割的锥还原方法</b>	<b>6</b>
3.1	问题背景	6
3.2	锥松弛间隙检测	6
3.3	基于方向的锥还原方法	6
3.3.1	Cauchy-Schwarz 不等式	6
3.3.2	方向割约束	7
3.4	算法总体流程	8
3.5	算法伪代码	8
3.6	算法复杂度分析	10

# 1 符号表

## 1.1 索引与集合

符号	描述	说明
$t$	时段索引	$t \in \{0, 1, \dots, T-1\}$
$i, j, k$	节点索引	$i, j, k \in N$
$(i, j)$	支路索引	支路 $(i, j) \in E$ 表示连接节点 $i$ 和 $j$ 的支路
$g$	发电机索引	$g \in G$
$dr$	可调负荷索引	$dr \in DR$
$N$	节点集合	配电网所有节点的集合, $ N  =$ 节点总数
$E$	支路集合	配电网所有支路的集合, 每条支路表示为 $(i, j)$
$G$	发电机集合	所有发电机的集合, $ G  =$ 发电机总数
$DR$	可调负荷集合	所有可调负荷的集合, $ DR  =$ 可调负荷总数
$G_j$	节点 $j$ 的发电机集合	连接在节点 $j$ 的所有发电机
$DR_j$	节点 $j$ 的可调负荷集合	连接在节点 $j$ 的所有可调负荷

## 1.2 决策变量

### 1.2.1 支路变量

符号	描述	单位
$P_{t,i,j}$	支路 $(i, j)$ 在时段 $t$ 的有功功率	p.u.
$Q_{t,i,j}$	支路 $(i, j)$ 在时段 $t$ 的无功功率	p.u.
$l_{t,i,j}$	支路 $(i, j)$ 在时段 $t$ 的电流幅值平方 $ I_{ij} ^2$	p.u.

### 1.2.2 节点变量

符号	描述	单位
----	----	----

$v_{t,i}$	节点 $i$ 在时段 $t$ 的电压幅值平方 $ V_i ^2$	p.u.
$p_{t,i}$	节点 $i$ 在时段 $t$ 的有功注入功率	p.u.
$q_{t,i}$	节点 $i$ 在时段 $t$ 的无功注入功率	p.u.

### 1.2.3 发电机变量

符号	描述	单位
$P_{g,t,g}$	发电机 $g$ 在时段 $t$ 的有功出力	p.u.
$Q_{g,t,g}$	发电机 $g$ 在时段 $t$ 的无功出力	p.u.

### 1.2.4 可调负荷变量

符号	描述	单位
$P_{DR,t,dr}$	可调负荷 $dr$ 在时段 $t$ 的削减量 (正数表示削减)	p.u.

## 1.3 网络参数

符号	描述	单位
$r_{ij}$	支路 $(i, j)$ 的电阻	p.u.
$x_{ij}$	支路 $(i, j)$ 的电抗	p.u.
$V_{\min}$	节点电压幅值下限	p.u. (默认 0.9)
$V_{\max}$	节点电压幅值上限	p.u. (默认 1.1)

## 1.4 锥还原相关符号

符号	描述	单位
$\Delta_{t,i,j}$	支路 $(i, j)$ 在时段 $t$ 的锥松弛间隙	p.u. <sup>2</sup>
$\mathbf{x}_{ij}$	支路功率向量, $\mathbf{x}_{ij} = (P_{t,i,j}, Q_{t,i,j})^T$	p.u.
$\mathbf{d}$	方向向量, $\mathbf{d} = (d_P, d_Q)^T$	无量纲
$r$	收缩因子, $r \in [0, 1]$	无量纲
$r_{\min}$	初始收缩因子	无量纲
$r_k$	第 $k$ 层的收缩因子	无量纲

$\mathcal{L}$	线性割约束集合	-
$\mathcal{D}_\ell$	第 $\ell$ 层的方向集合	-
$L_{\max}$	最大层数	正整数 (默认 3-5)
$K_{\max}$	最大迭代次数	正整数 (默认 50)
$\epsilon$	收敛容差	p.u. <sup>2</sup> (默认 $10^{-6}$ )

---

## 2 优化模型

$$\min_{P_{t,i,j}, Q_{t,i,j}, l_{t,i,j}, v_{t,i}, p_{t,i}, q_{t,i}, P_{g,t,g}, Q_{g,t,g}, P_{DR,t,dr}} \sum_{t=0}^{T-1} \sum_{dr \in DR_j} -P_{DR,t,dr} \quad (1)$$

s.t.

$$v_{t,0} = 1.0, \quad \forall t \in \{0, 1, \dots, T-1\} \quad (1)$$

$$p_{t,j} = \sum_{g \in G_j} P_{g,t,g} + \sum_{dr \in DR_j} P_{DR,t,dr} + P_{d,t,j}, \quad \forall t, j \in N \quad (2a)$$

$$q_{t,j} = \sum_{g \in G_j} Q_{g,t,g} + Q_{d,t,j}, \quad \forall t, j \in N \quad (2b)$$

$$p_{t,j} = \sum_{(j,k) \in E} P_{t,j,k} - \sum_{(i,j) \in E} (P_{t,i,j} - r_{ij} \cdot l_{t,i,j}), \quad \forall t, i, j, k \in N \quad (3a)$$

$$q_{t,j} = \sum_{(j,k) \in E} Q_{t,j,k} - \sum_{(i,j) \in E} (Q_{t,i,j} - x_{ij} \cdot l_{t,i,j}), \quad \forall t, i, j, k \in N \quad (3b)$$

$$v_{t,j} = v_{t,i} - 2(r_{ij}P_{t,i,j} + x_{ij}Q_{t,i,j}) + (r_{ij}^2 + x_{ij}^2)l_{t,i,j}, \quad \forall t, (i, j) \in E \quad (4)$$

$$(2P_{t,i,j})^2 + (2Q_{t,i,j})^2 + (l_{t,i,j} - v_{t,i})^2 \leq (l_{t,i,j} + v_{t,i})^2, \quad \forall t, (i, j) \in E \quad (5)$$

$$P_{g,\min}(t) \leq P_{g,t,g} \leq P_{g,\max}(t), \quad \forall t, g \in G \quad (6)$$

$$P_{DR,\min}(t) \leq P_{DR,t,dr} \leq P_{DR,\max}(t), \quad \forall t, dr \in DR \quad (7)$$

$$V_{\min}^2 \leq v_{t,i} \leq V_{\max}^2, \quad \forall t, i \in N \quad (8)$$

$$-2.5 \leq P_{t,i,j} \leq 2.5, \quad \forall t, (i, j) \in E \quad (9a)$$

$$-2.5 \leq Q_{t,i,j} \leq 2.5, \quad \forall t, (i, j) \in E \quad (9b)$$

$$0 \leq l_{t,i,j} \leq 2.5, \quad \forall t, (i, j) \in E \quad (9c)$$

### 3 基于方向割的锥还原方法

#### 3.1 问题背景

在二阶锥交流潮流优化模型中，二阶锥松弛约束 (5) 对原始非凸约束进行了凸松弛：

$$l_{t,i,j} \cdot v_{t,i} = P_{t,i,j}^2 + Q_{t,i,j}^2 \Rightarrow \left\| \begin{bmatrix} 2P_{t,i,j} \\ 2Q_{t,i,j} \\ l_{t,i,j} - v_{t,i} \end{bmatrix} \right\|_2 \leq l_{t,i,j} + v_{t,i} \quad (2)$$

松弛的几何意义：

- 原始约束要求点  $(P_{t,i,j}, Q_{t,i,j}, l_{t,i,j}, v_{t,i})$  必须位于**二阶锥的表面**（锥面约束）
- 松弛后的约束允许点位于**二阶锥的内部或表面**（锥约束）

在辐射状配电网中，该松弛通常是紧的（tight），即最优解自然满足等式约束。但在某些情况下，松弛解可能落在锥的内部，此时需要进行**锥还原**，将解投影到锥面上以获得原问题的可行解。

#### 3.2 锥松弛间隙检测

设 SOCP 求解得到的最优解为  $(P_{t,i,j}^*, Q_{t,i,j}^*, l_{t,i,j}^*, v_{t,i}^*)$ ，定义**锥松弛间隙**为：

$$\Delta_{t,i,j} = l_{t,i,j}^* \cdot v_{t,i}^* - [(P_{t,i,j}^*)^2 + (Q_{t,i,j}^*)^2] \quad (3)$$

间隙判定：

- 若  $\Delta_{t,i,j} \approx 0$ （在数值误差范围内），则松弛是紧的，无需还原
- 若  $\Delta_{t,i,j} > \epsilon$ （给定容差），则需要进行锥还原

#### 3.3 基于方向的锥还原方法

当检测到松弛间隙时，采用基于方向的迭代方法将解还原到锥面上。该方法基于**Cauchy-Schwarz 不等式**，导出**方向割约束**。

##### 3.3.1 Cauchy-Schwarz 不等式

对于支路  $(i, j)$ ，记  $\mathbf{x}_{ij} = (P_{t,i,j}, Q_{t,i,j})^T$ ，则二阶锥约束为：

$$\|\mathbf{x}_{ij}\|_2 \leq \sqrt{l_{t,i,j} \cdot v_{t,i}} \quad (4)$$

对于任意方向向量  $\mathbf{d} = (d_P, d_Q)^T$ ，由 Cauchy-Schwarz 不等式可得：

$$\frac{\mathbf{d}^T \mathbf{x}_{ij}}{\|\mathbf{d}\|_2} \leq \|\mathbf{x}_{ij}\|_2 \leq \sqrt{l_{t,i,j} \cdot v_{t,i}} \quad (5)$$

这表明：沿任意方向  $\mathbf{d}$  的投影不超过  $\sqrt{l_{t,i,j} \cdot v_{t,i}}$ 。

### 3.3.2 方向割约束

为了收紧可行域，在方向  $\mathbf{d}$  上添加带收缩因子的方向割：

$$\frac{\mathbf{d}^T \mathbf{x}_{ij}}{\|\mathbf{d}\|_2} \geq r \cdot \sqrt{l_{t,i,j} \cdot v_{t,i}} \quad (6)$$

其中  $r \in [0, 1]$  为收缩因子：

- $r = 1$ ：最紧的割，仅保留方向  $\mathbf{d}$  上的锥面点
- $r < 1$ ：较松的割，保留方向  $\mathbf{d}$  附近一定角度范围内的锥面点

### 3.4 算法总体流程

层次化方向割锥还原算法由三个阶段组成：初始化、主迭代循环和终止处理。算法以 SOCP 松弛解作为起点，通过逐步添加方向割约束收紧可行域，驱动松弛解向锥面靠近，直至锥松弛间隙低于收敛容差  $\epsilon$  或达到最大迭代次数  $K_{\max}$ 。

**阶段一：初始化** 以初始 SOCP 解  $(P^*, Q^*, l^*, v^*)$  为出发点，对每条支路  $(i, j)$  提取当前功率方向  $\mathbf{d} = (P_{t,i,j}^*, Q_{t,i,j}^*)^T$ ，计算初始收缩因子  $r_{\min} = \|\mathbf{d}\| / \sqrt{l_{t,i,j}^* v_{t,i}^*}$ （即当前解在锥面方向上的投影比），并将对应的方向割约束加入约束集合  $\mathcal{L}$ ，同时置层计数  $\ell = 1$ 、迭代计数  $k = 0$ 、间隙历史集合  $\mathcal{H} = \emptyset$ 。

**阶段二：主迭代循环** 每轮迭代包含以下步骤：

**步骤 1 (SOCP 求解)**: 在当前割约束集合  $\mathcal{L}$  下求解 SOCP，得到新解  $(P^k, Q^k, l^k, v^k)$ 。

**步骤 2 (间隙计算)**: 对所有支路和时段计算锥松弛间隙  $\Delta_{t,i,j}^k = l_{t,i,j}^k v_{t,i}^k - [(P_{t,i,j}^k)^2 + (Q_{t,i,j}^k)^2]$ ，并取最大间隙  $\Delta_{\max}^k$  记录至历史集合  $\mathcal{H}$ 。

**步骤 3 (全局收敛检验)**: 若  $\Delta_{\max}^k \leq \epsilon$ ，则所有支路的松弛均已收紧，返回当前解并标记为收敛。

**步骤 4 (层内自适应方向更新)**: 对仍有间隙 ( $\Delta_{t,i,j}^k > \epsilon$ ) 的支路，以当前解方向  $\hat{\mathbf{d}}$  和当前层收缩因子  $r_\ell$  生成新的方向割，补充进  $\mathcal{L}$ 。收缩因子  $r_\ell$  在各层间线性递增（从  $r_{\min}$  到 1），层号越高割越紧。

**步骤 5–6 (停滞检测与层次提升)**: 若近  $W_{\text{stag}}$  次迭代的最大间隙改善量均低于  $\epsilon_{\text{stag}}$ ，判定当前层已停滞。此时若  $\ell < L_{\max}$ ，则提升至下一层：增大收缩因子  $r_\ell$ ，生成新层全局方向集  $\mathcal{D}_\ell$  并添加对应的割约束，清空间隙历史重新监测；否则退出循环。

**阶段三：终止处理** 若总迭代次数达到  $K_{\max}$  仍未收敛，或层数达到最大层数  $L_{\max}$  仍未收敛，返回当前最优解并标记为未完全收敛，供后续处理使用。

### 3.5 算法伪代码

---

**算法 1** 层次化方向割锥还原算法

---

**输入**: SOCP 松弛解  $(P^*, Q^*, l^*, v^*)$ ，参数  $\epsilon, \epsilon_{\text{stag}}, W_{\text{stag}}, L_{\max}, K_{\max}$

**输出**: 还原解  $(\bar{P}, \bar{Q}, \bar{l}, \bar{v})$ ，收敛状态

1: // 阶段一：初始化

2:  $k \leftarrow 0, \ell \leftarrow 1$

3:  $\mathcal{L} \leftarrow \emptyset$

▷ 方向割约束集合

4:  $\mathcal{H} \leftarrow \emptyset$

▷ 间隙历史记录

5:  $\mathbf{d} \leftarrow (P_{t,i,j}^*, Q_{t,i,j}^*)^T, \forall (i, j) \in E, t \in [0, T-1]$

▷ 初始解的方向集合

6:  $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{d}^T \mathbf{x}_{ij} / \|\mathbf{d}\| \geq r_1 \sqrt{l_{t,i,j}^* v_{t,i}^*}\}$

▷ 初始层的方向割约束

7:  $r_{\min} \leftarrow \|\mathbf{d}\| / \sqrt{l_{t,i,j}^* v_{t,i}^*}$

▷ 初始收缩因子

8: // 阶段二：主迭代循环

```

9: while  $k < K_{\max}$  do
10:    $k \leftarrow k + 1$ 
11:   // 步骤 1: 求解带约束的 SOCP 问题
12:    $(P^k, Q^k, l^k, v^k) \leftarrow \text{SOLVE\_SOCP}(\mathcal{L})$ 
13:   // 步骤 2: 计算锥松弛间隙
14:   for all  $(i, j) \in E, t \in [0, T - 1]$  do
15:      $\Delta_{t,i,j}^k \leftarrow l_{t,i,j}^k \cdot v_{t,i}^k - [(P_{t,i,j}^k)^2 + (Q_{t,i,j}^k)^2]$ 
16:   end for
17:    $\Delta_{\max}^k \leftarrow \max_{t,i,j} \Delta_{t,i,j}^k; \mathcal{H} \leftarrow \mathcal{H} \cup \{\Delta_{\max}^k\}$ 
18:   // 步骤 3: 检查全局收敛
19:   if  $\Delta_{\max}^k \leq \epsilon$  then
20:     return  $(P^k, Q^k, l^k, v^k)$ , CONVERGED
21:   end if
22:   // 步骤 4: 层内自适应方向搜索 (固定  $r_\ell$ )
23:    $r_\ell \leftarrow r_{\min} + (\ell - 1)/(L_{\max} - 1) \cdot (1 - r_{\min})$ 
24:   for all  $(i, j) \in E, t \in [0, T - 1]$  且  $\Delta_{t,i,j}^k > \epsilon$  do
25:      $\hat{\mathbf{d}} \leftarrow (P_{t,i,j}^k, Q_{t,i,j}^k)^T / \|(P_{t,i,j}^k, Q_{t,i,j}^k)\|$ 
26:      $\mathcal{L} \leftarrow \mathcal{L} \cup \{\hat{\mathbf{d}}^T \mathbf{x}_{ij} / \|\hat{\mathbf{d}}\| \geq r_\ell \sqrt{l_{t,i,j} v_{t,i}}\}$ 
27:   end for
28:   // 步骤 5: 判断层内停滞
29:   if  $|\mathcal{H}| \geq W_{\text{stag}}$  then
30:      $H_{\text{recent}} \leftarrow \text{LAST\_N\_ELEMENTS}(\mathcal{H}, W_{\text{stag}})$ 
31:      $\text{improvements} \leftarrow \{H_{\text{recent}}[i] - H_{\text{recent}}[i + 1] \mid i = 1, \dots, W_{\text{stag}} - 1\}$ 
32:     if  $\max(\text{improvements}) < \epsilon_{\text{stag}}$  then
33:       // 步骤 6: 层次提升
34:       if  $\ell < L_{\max}$  then
35:          $\ell \leftarrow \ell + 1; r_\ell \leftarrow r_{\min} + (\ell - 1)/(L_{\max} - 1) \cdot (1 - r_{\min})$ 
36:          $\mathcal{D}_\ell \leftarrow \text{GENERATE\_GLOBAL\_DIRECTIONS}(\ell)$ 
37:         for all  $(i, j) \in E, t \in [0, T - 1], \mathbf{d} \in \mathcal{D}_\ell$  do
38:            $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{d}^T \mathbf{x}_{ij} / \|\mathbf{d}\| \geq r_\ell \sqrt{l_{t,i,j} v_{t,i}}\}$ 
39:         end for
40:          $\mathcal{H} \leftarrow \emptyset$  ▷ 清空历史, 开始新层监测
41:       else
42:         break
43:       end if
44:     end if
45:   end if
46: end while
47: // 阶段三: 终止处理

```

48: **return**  $(P^k, Q^k, l^k, v^k)$ , NOTFULLYCONVERGED

---

### 3.6 算法复杂度分析

时间复杂度（单次迭代）:

操作	复杂度	说明
SOCP 求解	$O(n^3)$	$n$ 为决策变量数
间隙计算	$O(T \cdot  E )$	遍历所有支路和时段
方向割添加	$O(T \cdot  E )$	最多添加 $T \cdot  E $ 个约束
<b>总计</b>	$O(n^3)$	由 SOCP 求解主导

空间复杂度:

数据结构	复杂度	说明
决策变量	$O(T \cdot  E )$	支路变量和节点变量
方向割约束集合 $\mathcal{L}$	$O(K_{\max} \cdot T \cdot  E )$	最坏情况
<b>总计</b>	$O(K_{\max} \cdot T \cdot  E )$	-

迭代次数:

- 典型情况: 10-30 次迭代
- 最坏情况:  $K_{\max} = 50$  次迭代
- 层数: 通常 2-3 层