

基于遗传算法的定日镜场优化设计

摘 要

塔式电站是利用太阳能光热进行发电的一项新型清洁技术,优化塔式电站布局是实现“碳达峰”“碳中和”的重要工作.定日镜是塔式电站的重要组成部分,通过建立数学模型优化出合理的定日镜布阵方式能大大提升塔式电站输出功率.

针对问题一:构建定日镜变换模型、光学效率计算模型与输出热功率模型.首先,遍历所有定日镜,确定镜面中心的太阳高度角与方位角,在吸收塔坐标系下利用根据**光线反射规律**确定定日镜法向的高度角与方位角,通过球坐标系与直角坐标系的转化规则得到镜面的单位法向量.其次,以正北为竖直方向建立镜面坐标系,通过旋转**欧拉角变换矩阵**与坐标平移,得到定日镜在吸收塔坐标系下的坐标.计算阴影遮挡效率时同时考虑吸收塔对入射光的遮挡与其余定日镜对入射、反射光线的遮挡.通过计算被遮挡采样点占比得到阴影遮挡效率.计算截断效率时视太阳为球形光源,反射光为光锥.当采样点不被遮挡时,在采样点光锥内计算均匀分布的 5 条采样光线的方向向量,统计光锥中射入集热器的光线占比得到截断效率.综合上述模型计算得到的年平均光学效率为 0.4881,年平均输出热功率为 29.3809MW,单位镜面面积年平均输出热功率为 0.4677 kW/m².对各月结果进行分析,发现各效率关于夏至日呈对称关系,春分日与秋分日各效率相近.对各定日镜的年平均光学效率进行分析,发现北侧定日镜拥有更高的光学效率.

针对问题二:构建问题二的**单目标优化模型**,视单位镜面面积年平均输出热功率为目标函数,将吸收塔位置、定日镜位置视为决策变量.通过预设布阵方式,将决策变量数量降至 6 个.设计 6 维**遗传算法**加快求解速度.求得的最大单位镜面面积年平均输出热功率为 0.6713 kW/m².分析各月相关效率,发现较问题一而言均有所提升.

针对问题三:在问题二的基础上构建问题三的**单目标优化模型**,仍以单位镜面面积年平均输出热功率为目标函数,将吸收塔位置、定日镜位置、安装高度、尺寸视为决策变量.针对各个方面的决策变量进行一系列简化,同时经多次试验后设定场内定日镜上限为 3500 面.针对 7005 个自变量设计 7005 维遗传算法求解.求得的最大单位镜面面积年平均输出热功率为 0.7456 kW/m².与问题二各月相关效率进行对比分析,发现问题三模型下平均光学效率在冬季有所提升,这是冬季平均截断效率提升所致.

考虑定日镜对光线的遮挡时,为简化问题,仅考虑了目标定日镜周围最近的 11 面定日镜.本文针对考虑的定日镜数量做了灵敏性分析.

关键词: 太阳能光热发电站; 欧拉角; 矩阵变换; 遗传算法; 单目标优化模型

一 问题重述

塔式电站是利用太阳能光热进行发电的一项新型清洁技术,优化塔式电站布局是实现“碳达峰”“碳中和”的重要工作.定日镜是塔式电站聚集太阳能的基本组件,其朝向可由两转轴实时控制,在工作时能确保使太阳光线经反射后指向集热器中心.现计划在 $39.4^{\circ}\text{N}, 98.5^{\circ}\text{E}$, 海拔高度 3000 m 处建设最大半径为 350 m 的圆形定日镜场.

规定吸热塔高度为 80 m , 集热器为一圆柱, 高 8 m , 半径为 3.5 m . 要求吸热塔周围半径 100 m 处不安装定日镜, 定日镜为平面矩形镜面, 其边长介于 $2\sim 8\text{ m}$ 之间, 中心离地高度在 $2\sim 6\text{ m}$ 之间, 且能保证镜面在转动时不触及地面. 相邻定日镜底座中心宽度多于镜面宽度 5 m 以上.

问题一 将吸收塔建设于镜场正中心, 定日镜尺寸为 $6\text{ m}\times 6\text{ m}$, 中心离地 4 m . 所有定日镜中心坐标由附件给出, 计算该镜场的年平均光学效率、年平均输出热功率与单位镜面面积的年平均输出热功率.

问题二 要求镜场额定年平均输出热功率为 60 MW . 当所有定日镜尺寸与中心离地高度相同时, 为使得镜场的年平均输出热功率尽可能大, 设计镜场的吸收塔坐标, 定日镜尺寸、数目、安装高度、位置坐标.

问题三 当各定日镜的尺寸与安装高度可以不同时, 其他条件与问题二相同, 求解问题二的相关参数.

二 问题分析

2.1 问题一的分析

问题一需要依据给定的定日镜位置坐标与定日镜尺寸参数, 计算定日镜场的年平均光学效率、年平均输出热功率与单位镜面面积的年平均输出热功率.

为计算上述结果, 需要先构建定日镜变换模型, 确定定日镜在吸收塔坐标系下的坐标. 为描述镜面方向, 需要根据镜面中心的条件计算镜面的高度角与方位角. 通过两次矩阵变换与一次平移变换进行参考系转换.

其次, 构建光学效率计算模型. 针对年平均光学效率, 需要分别计算阴影遮挡效率、余弦效率、大气透射率与集热器截断效率. 对镜面栅格化, 镜面上均匀分布采样点, 阴影遮挡效率分入射光线被吸收塔遮挡、入射光或反射光被其余定日镜遮挡两方面考虑; 集热器截断效率类似, 针对每个不被遮挡的采样点, 在其光锥上设计采样光线, 计算射入集热器的采样光线占总光线数量之比.

构建输出热功率模型, 计算得到光学效率后, 通过模型计算单位镜面平均输出热功率.

2.2 问题二的分析

问题二是问题一的反问题, 由于题目涉及约束条件, 在问题一的基础上建立含约束的单目标优化模型. 以同心圆方式布阵, 通过镜面所在圆的半径与镜面极角描述电站布阵规律. 运用问题一所得模型描述目标函数, 通过遗传算法搜索最优布阵方式, 从而求解出单位镜面面积年平均输出热功率的最大值.

2.3 问题三的分析

问题三在问题二的基础上解除了各定日镜尺寸、安装高度相同的限制. 因此,

额外视每块镜面的宽度、高度为决策变量. 通过预设布阵方式进一步减少决策变量数目, 改进问题二模型, 采用遗传算法进行寻优, 得出满足条件的最优解.

三 模型假设

1. 忽略闰年的情况, 仅考虑每年为 365 天, 2 月共 28 天;
2. 在问题一中, 仅考虑与目标定日镜距离最近的 11 面定日镜对光线的遮挡; 在问题二与问题三中, 假设定日镜的布阵方式能够忽略其余定日镜对光线的遮挡, 仅考虑吸收塔对入射光线的遮挡;
3. 在问题二与问题三中, 视太阳为点光源, 忽略光锥的影响.

四 符号说明

表 1 符号说明

符号	符号意义	单位
h_1	定日镜安装高度	m
h_2	集热器中心离地高度	m
h_3	集热器高度	m
R_0	集热器半径	m
R_1	太阳半径	m
R_m	最小布阵半径, 在此题中 $R_m = 100\text{ m}$	m
d	日地距离	m
α_s	太阳高度角	°
α_m	定日镜镜面高度角	°
γ_s	太阳方位角	°
γ_m	定日镜镜面方位角	°

五 模型的建立与求解

5.1 问题一模型的建立与求解

在确定定日镜朝向的基础上计算各光学效率, 以此计算定日镜场输出热功率. 为完成这一过程, 需要构建定日镜变换模型、光线效率计算模型与输出热功率模型。

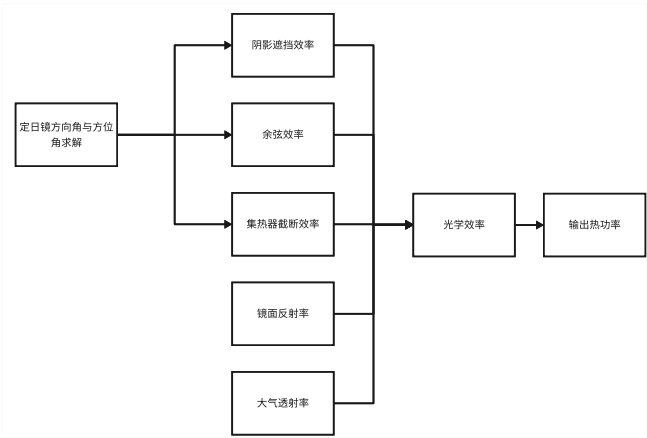


图 1 问题一求解流程

5.1.1 定日镜变换模型的建立

定日镜的姿态可通过定日镜高度角、方位角及中心坐标完全确定，其中各定日镜中心坐标已给出。

1. 定日镜高度角

太阳中心点发出的光线经定日镜中心反射后指向集热器中心，如图 2 所示。太阳高度角

$$\alpha_s = \arcsin(\cos \delta \cos \varphi \cos \omega + \sin \delta \sin \varphi),$$

式中 φ 为电站所处纬度； ω 为太阳时角，且有 $\omega = \frac{\pi}{12}(ST-12)$ ，其中 ST 为当地

时间； δ 为太阳赤纬角，且有 $\delta = \arcsin \left[\sin \frac{2\pi D}{365} \sin \left(\frac{2\pi}{360} 23.45 \right) \right]$ ，其中 D 为天数，且定义春分日为第 0 天。

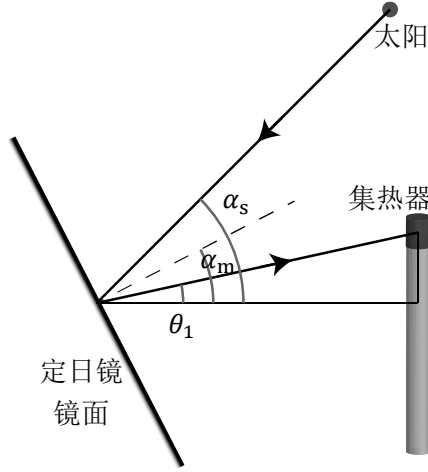


图 2 定日镜高度角求解示意图

由于 \sin 函数为周期函数，题示条件下各日期与 D 的关系参见附录 2。定日镜中心和集热器中心的连线与水平面的夹角为

$$\theta_1 = \arctan \frac{\Delta h}{r} = \arctan \frac{h_2 - h_1}{\sqrt{x_i^2 + y_i^2}},$$

因此，定日镜镜面法线高度角为

$$\alpha_m = \theta_1 + \frac{\alpha_s - \theta_1}{2}. \quad (1)$$

2. 定日镜方位角

以吸收塔中心 O 点为原点，正北方向为 y 轴正方向，正东方向为 x 轴正方向，建立平面直角坐标系。

每月 21 日的 9:00、10:30、12:00、13:30 与 15:00 时太阳方位角

$$\gamma_s = \arccos \frac{\sin \delta - \sin \alpha_s \sin \varphi}{\cos \alpha_s \cos \varphi}$$

处于区间 $(95^\circ, 127^\circ)$ 内，因此只考虑太阳 S 在第四象限的情况。

①当定日镜中心 M 处于第一、二象限的部分位置时，三者关系如图 3 所示，图

中 \overline{SM} 为入射光线， \overline{MO} 为出射光线， d 为日地距离，且应有 $d \gg r$ 。

图中 β （单位：角度）可由 M 点坐标 (x, y) 确定

$$\beta = \arctan \frac{y}{x},$$

在 $\triangle SOM$ 中, 运用余弦定理, 逐步求得

$$\gamma_m = 180^\circ - \left(\frac{\theta_3}{2} - (90^\circ - \beta) \right).$$

类似地, 当 M 位于第二象限且镜面法线位于点 S 之下时,

$$\gamma_m = - \left(\frac{\theta_3}{2} - (90^\circ - \beta) \right). \quad (2)$$

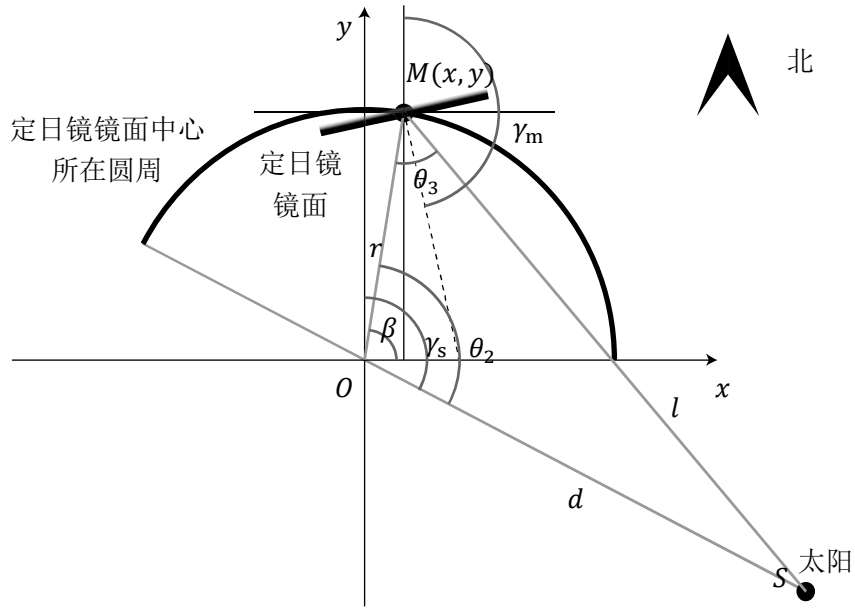


图3 定日镜中心处于第一、二象限部分区域时方位角计算示意图

②当 M 处于第三、四象限且镜面法线位于点 S 之上时, OSM 关系如图4所示. 可得

$$\gamma_m = \beta + \frac{1}{2} \arccos \frac{r^2 + l^2 - d^2}{2rl}. \quad (3)$$

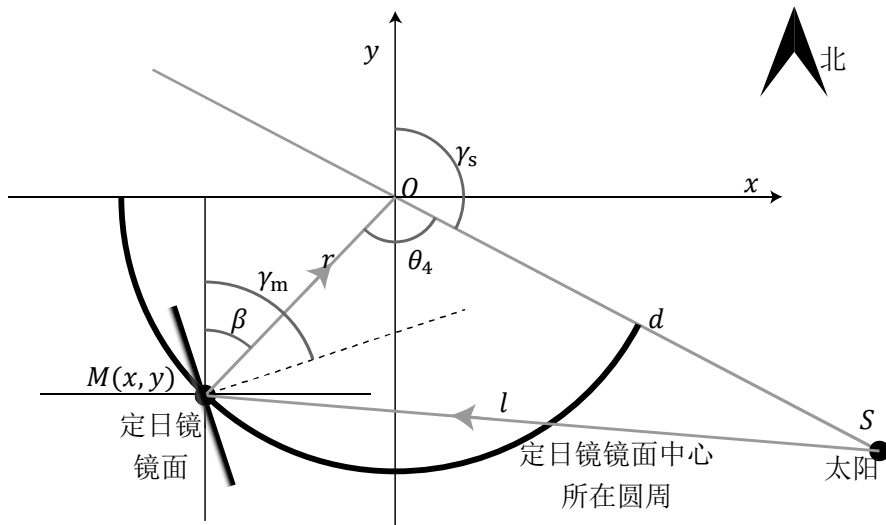


图4 定日镜中心处于第三、四象限部分区域时方位角计算示意图

③当定日镜位于第一、二象限内，①外的区域时，如图 5 所示。得

$$\gamma_m = 180^\circ - \left(\beta - \frac{1}{2} \arccos \frac{r^2 + l^2 - d^2}{2rl} \right). \quad (4)$$

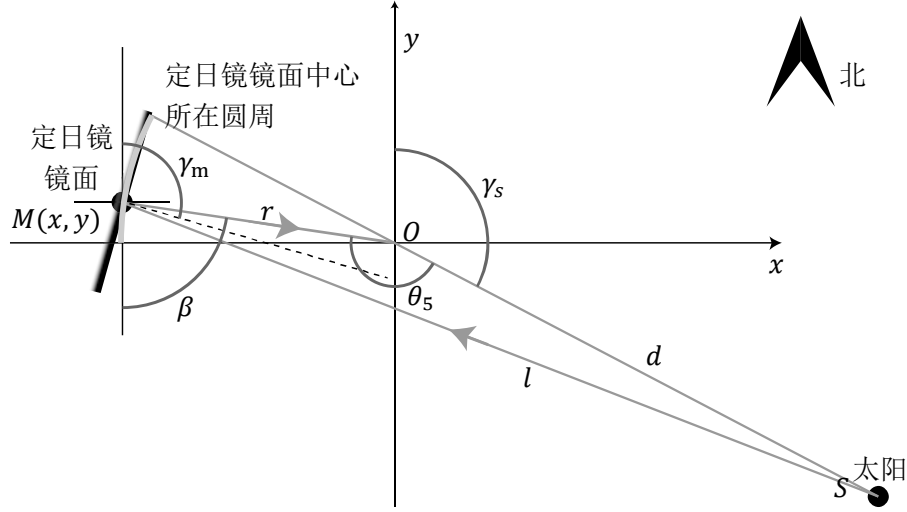


图 5 定日镜中心处于第一、二象限其余区域时方位角计算示意图

④当定日镜位于第三、四象限内，②外的区域时，与③类似，

$$\gamma_m = 180^\circ - \frac{1}{2} \arccos \frac{r^2 + l^2 - d^2}{2rl} + \beta. \quad (5)$$

综上所述，联立(2)(3)(4)(5)，得定日镜方位角为

$$\gamma_m = \begin{cases} -\left(\frac{\theta_3}{2} - (90^\circ - \beta)\right), & \text{①,} \\ \beta + \frac{1}{2} \arccos \frac{r^2 + l^2 - d^2}{2rl}, & \text{②,} \\ 180^\circ - \left(\beta - \frac{1}{2} \arccos \frac{r^2 + l^2 - d^2}{2rl}\right), & \text{③,} \\ \gamma_m = 180^\circ - \frac{1}{2} \arccos \frac{r^2 + l^2 - d^2}{2rl} + \beta, & \text{④.} \end{cases} \quad (6)$$

3. 定日镜旋转

假设定日镜旋转前正面朝北，即 $\alpha_m = 0$ ， $\gamma_m = 0$ ，此时定日镜可使用平面参数方程

$$\begin{cases} x = a, \\ y = 0, \\ z = b \end{cases}$$

描述，参数 a 与 b 的取值范围与定日镜大小有关。以定日镜中心 O' 为原点，正东方向为 x 轴正方向，正北方向为 y 轴正方向，竖直向上方向为 z 轴正方向，建立空间直角坐标系 $O'xyz$ 定日镜坐标系)。

定日镜的旋转可以使用欧拉角表示^[1]，(1)式与(6)式所求的定日镜的方位角与方向角恰为进动角与章动角，这两步旋转可分别使用矩阵

$$T_1 = \begin{bmatrix} \cos \gamma_m & \sin \gamma_m & 0 \\ -\sin \gamma_m & \cos \gamma_m & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_m & -\sin \alpha_m \\ 0 & \sin \alpha_m & \cos \alpha_m \end{bmatrix}$$

描述。故变换前定日镜坐标系上的点 $\begin{bmatrix} a \\ 0 \\ b \end{bmatrix}$ 先后经过两次旋转变换与平移后变为

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = T_2 T_1 \begin{bmatrix} a \\ 0 \\ b \end{bmatrix} + \begin{bmatrix} x \\ y \\ h_1 \end{bmatrix}. \quad (7)$$

5.1.2 光学效率计算模型的建立

根据光学效率 η 计算公式

$$\eta = \eta_{sb} \eta_{\cos} \eta_{at} \eta_{trunc} \eta_{ref}, \quad (8)$$

为计算光学效率，需要依次计算阴影遮挡效率 η_{sb} 、余弦效率 η_{\cos} 、集热器截断效率 η_{trunc} 、镜面反射率 η_{ref} 、大气透射率 η_{at} 。

1. 阴影遮挡效率 η_{sb}

考虑光线被吸收塔或其余定日镜遮挡。

在定日镜面上利用网格划分有限个区域，当区域数量足够大时，能够通过网格线上的点是否被吸收塔遮挡描述定日镜被遮挡的情况。

① 光线被吸收塔遮挡的情况

假设吸收塔坐标系下某一采样点坐标为 $N(x_1, y_1, z_1)$ ，当 N 与太阳中心 S 的连线与吸收塔存在交点时，该采样点被遮挡。这一连线的单位方向向量仅与太阳高度角和太阳方位角有关，表达式为

$$\hat{s} = \begin{bmatrix} \cos \alpha_s \sin \gamma_s \\ \cos \alpha_s \cos \gamma_s \\ \sin \alpha_s \end{bmatrix},$$

因此这一直线可以利用点向式方程

$$\frac{x - x_1}{\cos \alpha_s \sin \gamma_s} = \frac{y - y_1}{\cos \alpha_s \cos \gamma_s} = \frac{z - z_1}{\sin \alpha_s}$$

描述。

由于太阳在统计时间内始终处于第四象限的部分区域，只有处于第二象限特定区域的定日镜有可能被吸收塔遮挡。

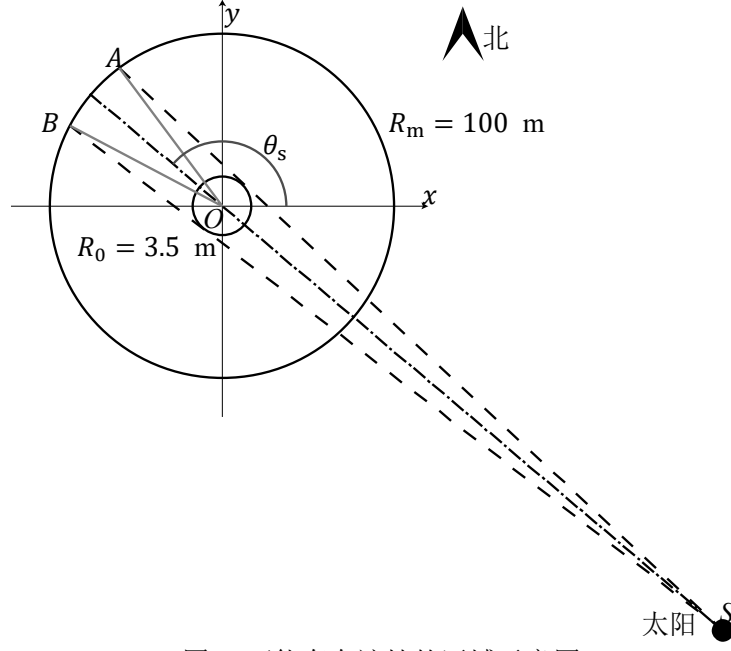


图 6 可能存在遮挡的区域示意图

如图 6 所示，最外圆所围区域外不安装定日镜，内圆为吸收塔。由于 $|OS| \gg |AB|$ ，因此视图内两条虚线 SA ， SB 与点划线 SO 平行， $\triangle AOB$ 为等腰三角形，此时 $AB = 2R_0$ 。故处于点划线上的点在以 x 为极轴、以 O 为极点的极坐标系中，极角为

$$\theta_s = (180^\circ - \gamma_s) + 90^\circ$$

处。并且

$$\angle AOB = 2 \arcsin \frac{R_0}{R_m},$$

因此极角在区间 $\left[\theta_s - \arcsin \frac{R_0}{R_m}, \theta_s + \arcsin \frac{R_0}{R_m} \right]$ 的镜面采样点存在被遮挡的可能。

由于在计算 $\angle AOB$ 时采用了近似，会导致得到的区间存在一定误差 q ，因此针对处于区间 $\left[\theta_s - \arcsin \frac{R_0}{R_m} - q, \theta_s + \arcsin \frac{R_0}{R_m} + q \right]$ 的定日镜进行判断。

遍历第二象限上述区间内所有定日镜，判断方程组

$$\begin{cases} \frac{x - x_1}{\cos \alpha_s \sin \gamma_s} = \frac{y - y_1}{\cos \alpha_s \cos \gamma_s} = \frac{z - z_1}{\sin \alpha_s}, \\ x^2 + y^2 \leq R_0^2, \\ z \leq h_2, \end{cases}$$

是否存在实根。若存在实根，则当前采样点被遮挡；反之不被遮挡。不在遍历区间的定日镜判断为不被遮挡。

② 光线被其它定日镜遮挡的情况

这一情况又分为两个可能：定日镜 A 的入射光被其余定日镜遮挡与出射光被其余定日镜遮挡。

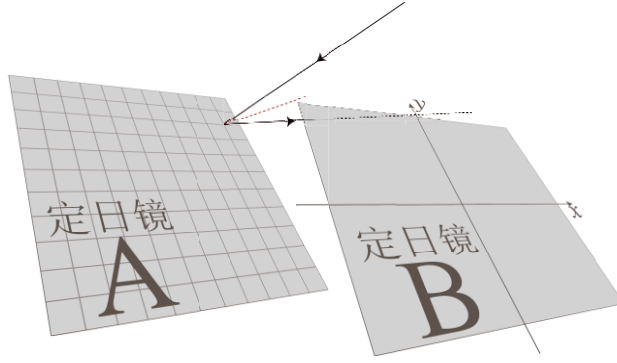


图 7 定日镜 A 采样点上反射光线被定日镜 B 遮挡示意图
设计算光线在大地坐标系下由点向式方程

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M + t\hat{s} \quad (9)$$

描述，其中 M 为定日镜 A 上的采样点， \hat{s} 为光线所在直线的单位方向向量。

联立(7)式与(9)式，消去 t ，移项整理后得

$$\begin{cases} \left(\cos \alpha_{mB} \sin \gamma_{mB} + \frac{s_y}{s_x} \cos \gamma_{mB} \right) a + \sin \alpha_{mB} b + M_y - (M_x + x_B) \frac{s_y}{s_x} - y_B = 0, \\ \left(\sin \alpha_{mB} \sin \gamma_{mB} + \frac{s_z}{s_x} \cos \gamma_{mB} \right) a - \cos \alpha_{mB} b + M_z - (M_x + x_B) \frac{s_z}{s_x} - h_{B1} = 0. \end{cases}$$

这是关于 a 与 b 的线性方程组，当光线不与定日镜 B 平行时有唯一解。使用 Cramer 法则解出 a 与 b 的值，若同时满足

$$\begin{cases} -w/2 \leq a \leq w/2, \\ -h/2 \leq b \leq h/2, \end{cases}$$

即光线与定日镜 B 所在平面的交点在定日镜 B 范围（镜 B 坐标系下 $[-w/2, +w/2] \times [-h/2, +h/2]$ ）内，认为该光线被这面定日镜阻挡，不论该光线是入射光还是反射光。

综合两遮挡因素，通过计算一块定日镜被遮挡（不论是被吸收塔遮挡还是被其余定日镜遮挡）的采样点占总采样点的比值，可以得到 η_{sb} 。

2. 余弦效率 η_{cos}

查阅文献得定日镜采样点 i 的余弦效率 η_c 的计算公式

$$\eta_c(i) = \cos \theta_i = \mathbf{s} \cdot \mathbf{n}, \quad [1]$$

式中 \mathbf{s} 为光线方向向量

$$\mathbf{s} = \begin{bmatrix} \cos \alpha_s \sin \gamma_s \\ \cos \alpha_s \sin \gamma_s \\ \sin \alpha_s \end{bmatrix},$$

\mathbf{n} 为定日镜法向量

$$\mathbf{n} = \begin{bmatrix} \cos \alpha_m \sin \gamma_m \\ \cos \alpha_m \cos \gamma_m \\ \sin \alpha_m \end{bmatrix}.$$

通过对单块定日镜所有采样点的余弦效率取平均，得到镜面的余弦效率

$$\eta_{\cos} = \frac{1}{n} \sum_{i=1}^n \eta_c(i).$$

3. 集热器截断效率 η_{trunc}

对于全部被收集塔遮挡的定日镜，由于其 η_{sb} 为0，光学效率必然为0，不必计算截断效率。

对于其他情形：考虑太阳光为光源为球体，所有发射至定日镜一点的光组成光锥。光锥的底面半径为太阳球体的半径 R_1 ，高为日地距离 d 。

① 入射光锥半顶角与轴线方向

入射光锥的半顶角 θ 与 R_1 和 d 有关：

$$\theta = \arctan \frac{R_1}{d},$$

轴线方向在5.1.1中已经给出

$$\hat{\mathbf{s}} = - \begin{bmatrix} \cos \alpha_s \sin \gamma_s \\ \cos \alpha_s \cos \gamma_s \\ \sin \alpha_s \end{bmatrix}.$$

② 定日镜法向量

定日镜的单位法向量由定日镜的高度角与方位角共同描述，结果为

$$\hat{\mathbf{n}} = \begin{bmatrix} \cos \alpha_m \sin \gamma_m \\ \cos \alpha_m \cos \gamma_m \\ \sin \alpha_m \end{bmatrix}.$$

③ 反射光锥半顶角与轴线方向

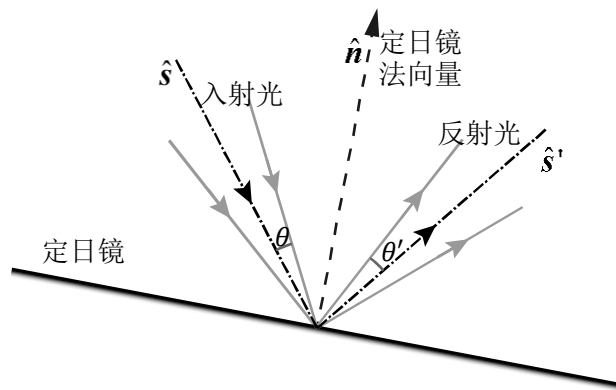


图8 入射光锥与反射光锥示意图

反射光锥的形成如图8所示。考虑入射光锥与反射光锥轴线上的单位方向向量，两者与法线夹角相同且共面，故存在关系式

$$\hat{\mathbf{s}}' = \hat{\mathbf{s}} + 2|\hat{\mathbf{s}} \cdot \hat{\mathbf{n}}|\hat{\mathbf{n}}$$

成立。由于入射光与反射光光线始终关于法线对称，故

$$\theta' = \theta.$$

④ 采样光线方程

针对每个不被遮挡的采样点，均匀选取反射光锥内的 5 条光线作为采样光线^[1]，通过确定每条采样光线是否能够射入集热器中来衡量截断效率。

假定采样点坐标为 $\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}$ ，5 条光线中的一条为光锥轴线，剩余四条为圆锥的

转向轮廓线。

对于周围四条采样光线的方向向量 s'_i ，可以认为是轴线的单位方向向量 \hat{s}' 与其它某向量 t 的矢量和。

● 上下侧光线

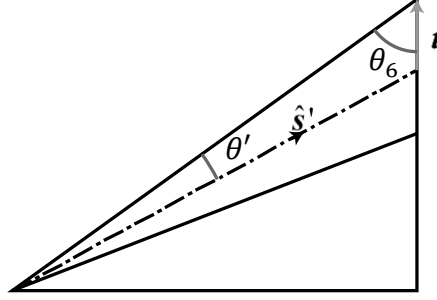


图 9 求解上侧采样光线方向向量示意图

对于上侧光线，如图 9 所示。向量 \hat{s}' 与水平面夹角 $\omega = \arcsin(\hat{s}'_z)$ ，因此有 $\theta_6 = 90^\circ - \omega - \theta'$ 。

根据公式推出

$$t_{z1} = \frac{\sin \theta'}{\sin \theta_6} = \frac{\sin \theta'}{\sin (90^\circ - \omega - \theta')}.$$

类似地，对于下侧光线，

$$t_{z2} = \frac{\sin \theta'}{\sin (90^\circ + \omega - \theta')}.$$

上下侧光线的单位方向向量

$$\hat{s}_1 = \frac{\hat{s} + t_1}{|\hat{s} + t_1|}, \hat{s}_2 = \frac{\hat{s} + t_2}{|\hat{s} + t_2|}.$$

● 左右侧光线

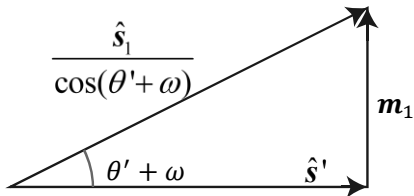


图 10 m_1 的计算示意图

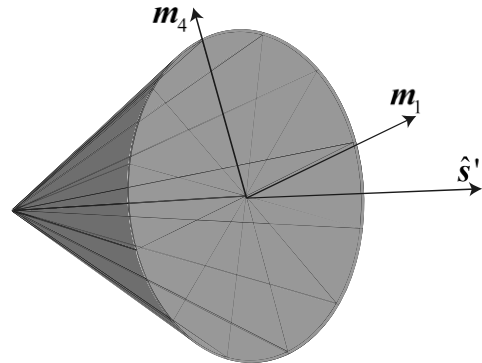


图 11 m_4 的计算示意图

如图 10 与图 11 所示，计算光锥切面向量

$$\mathbf{m}_1 = \frac{\hat{\mathbf{s}}_1}{\cos(\theta' + \omega)} - \hat{\mathbf{s}}',$$

计算垂直于 $\hat{\mathbf{s}}'$ 与 \mathbf{m}_1 的向量 \mathbf{m}_4 ，且使得 $|\mathbf{m}_4| = |\mathbf{m}_1|$ ，则

$$\mathbf{m}_4 = \frac{\hat{\mathbf{s}}' \times \mathbf{m}_1}{|\hat{\mathbf{s}}' \times \mathbf{m}_1|} |\mathbf{m}_1|.$$

\mathbf{m}_3 与 \mathbf{m}_4 等大反向：

$$\mathbf{m}_3 = -\mathbf{m}_4.$$

分别由 \mathbf{m}_3 与 \mathbf{m}_4 可以计算出左右两侧光线的单位方向向量

$$\hat{\mathbf{s}}_3 = \frac{\mathbf{m}_3 + \hat{\mathbf{s}}'}{|\mathbf{m}_3 + \hat{\mathbf{s}}'|}, \hat{\mathbf{s}}_4 = \frac{\mathbf{m}_4 + \hat{\mathbf{s}}'}{|\mathbf{m}_4 + \hat{\mathbf{s}}'|}.$$

对于五条光线，通过点向式方程描述，联立方程组

$$\begin{cases} \frac{x - x_n}{s'_{i,x}} = \frac{y - y_n}{s'_{i,y}} = \frac{z - z_n}{s'_{i,z}}, \\ x^2 + y^2 = R_0^2, \\ h_2 - h_3 \leq z \leq h_2, \end{cases} \quad (10)$$

判断是否存在实根：若存在，则该光线能够被集热器吸收；反之不能被吸收。

计算所有采样光线中能被吸收的光线占总不被遮挡的采样光线的比值，即为集热器的截断效率

$$\eta_{\text{trunc}} = \frac{n_{\text{rec}}}{N - n_{\text{sb}}}.$$

4. 镜面反射率 η_{ref}

根据题意，镜面反射率取常数 0.92.

5. 大气透射率 η_{at}

大气透射率可由公式

$$\eta_{\text{at}} = 0.99321 - 0.0001176 \times d_{\text{HR}} + 1.97 \times 10^{-8} \times d_{\text{HR}}^2$$

计算得出。式中 d_{HR} 为镜面中心 (x, y, h_1) 至集热器中心的距离

$$d_{\text{HR}} = \sqrt{x^2 + y^2 + \left(h_2 - \frac{h_3}{2} - h_1\right)^2}.$$

综上所述，光学效率为

$$\eta = \eta_{\text{sb}} \eta_{\text{cos}} \eta_{\text{at}} \eta_{\text{trunc}} \eta_{\text{ref}},$$

其中

$$\begin{cases} \eta_{\text{sb}} = 1 - \frac{n_{\text{sb}}}{N}, \\ \eta_{\text{cos}} = \frac{1}{n} \sum_{i=1}^n \mathbf{s}_i \cdot \mathbf{n}_i, \\ \eta_{\text{at}} = 0.99321 - 0.0001176 \times d_{\text{HR}} + 1.97 \times 10^{-8} \times d_{\text{HR}}^2, \\ \eta_{\text{trunc}} = \frac{n_{\text{rec}}}{N - n_{\text{sb}}}, \\ \eta_{\text{ref}} = 0.92. \end{cases}$$

5.1.3 输出热功率模型的建立

1. 镜场瞬时输出热功率

定日镜场的瞬时输出热功率为

$$E_{\text{field}} = DNI \cdot \sum_{i=1}^n A_i \eta_i,$$

式中 DNI 为法向直接辐照强度, 由公式

$$DNI = G_0 \left[a + b \exp \left(-\frac{c}{\sin \alpha_s} \right) \right]$$

近似计算, 其中 G_0 为太阳常数, 取 1.366 kW/m^2 , $a=0.4237-0.00821(6-H)^2$, $b=0.5055+0.00595(6.5-H)^2$, $c=0.2711+0.01858(2.5-H)^2$, H 为海拔高度. A_i 为第 i 面镜子的面积, η_i 为对应的光学效率.

2. 镜场年平均输出热功率

对各月、各时间的瞬时输出热功率取平均, 得年平均输出热功率为

$$\bar{E}_{\text{field}} = \frac{\sum_{p=1}^{12} \sum_{q=1}^5 E_{\text{field}}(p, q)}{12 \times 5}.$$

3. 单位镜面面积年平均输出热功率

将镜场年平均输出热功率除以镜面总面积, 得单位镜面面积年平均输出热功率为

$$\bar{E}_{\text{mirror}} = \frac{\bar{E}_{\text{field}}}{\sum_{i=1}^n A_i}.$$

5.1.4 模型求解

1. 求解过程

问题一模型求解过程已在图 1 给出.

计算过程中, 阴影遮挡效率与集热器截断效率存在关联, 具体求解步骤如下.

算法步骤

Step 1: 遍历定日镜, 产生采样点

在每面定日镜上均匀划分正方形网格, 网格交点为采样点

Step 2: 遍历采样点, 产生采样光线

在每个采样点上形成入射光线轴线, 基于光线反射定律计算出射光线轴线

Step 3: 入射光线遮挡判断

判断入射光线是否被遮挡. 若被遮挡, 则 n_{sb} 自增 1, 回到 Step 2, 遍历下一采样点; 否则继续进行 Step 4

Step 4: 出射光线遮挡判断

判断出射光线是否被遮挡. 若被遮挡, 则 n_{sb} 自增 1, 回到 Step 2, 遍历下一采样点; 否则继续进行 Step 5

Step 5: 光锥形成与截断效率计算

产生光锥, 计算截断效率

Step 6: 阴影遮挡效率计算

计算阴影遮挡效率

计算定日镜对光线的遮挡时, 为简化运算量, 只考虑距离反射定日镜最近的 11 面定日镜对光线的遮挡. 在计算截断效率时已经求得反射光锥上的 5 条采样

光线方程，在计算遮挡时考虑定日镜对光锥的遮挡。

由于采样点过多会导致运算量呈指数增长，因此在求解过程中，镜面上每隔 0.5 m 设置一采样点。

计算时得到的

$$\theta' \approx 0.267^\circ$$

较小，因此允许的角度误差赋值为 $q=0.05^\circ$ 。

2. 结果

计算得到各月 21 日平均光学效率等结果如表 2 所示，年均效率等结果如表 3 所示。

表 2 问题一每月 21 日平均光学效率及输出功率

日期	平均 光学效率	平均 余弦效率	平均阴影 遮挡效率	平均 截断效率	单位面积镜面 平均输出热功率 (kW/m ²)
1 月 21 日	0.4469	0.6661	0.8081	0.9351	0.3736
2 月 21 日	0.4743	0.6962	0.8076	0.9500	0.4335
3 月 21 日	0.4983	0.7317	0.7950	0.9648	0.4891
4 月 21 日	0.5178	0.7659	0.7809	0.9751	0.5341
5 月 21 日	0.5243	0.7846	0.7681	0.9797	0.5538
6 月 21 日	0.5268	0.7892	0.7666	0.9807	0.5594
7 月 21 日	0.5237	0.7811	0.7713	0.9789	0.5506
8 月 21 日	0.5137	0.7573	0.7853	0.9728	0.5239
9 月 21 日	0.4905	0.7202	0.7989	0.9602	0.4718
10 月 21 日	0.4642	0.6837	0.8101	0.9439	0.4108
11 月 21 日	0.4394	0.6600	0.8046	0.9319	0.3586
12 月 21 日	0.4369	0.6582	0.8029	0.9311	0.3537

表 3 问题一年平均光学效率及输出功率表

年平均 光学效率	年平均 余弦效率	年平均阴影 遮挡效率	年平均 截断效率	年平均 输出热功率 (MW)	单位面积镜面年平均 输出热功率 (kW/m ²)
0.4881	0.7245	0.7916	0.9587	29.3809	0.4677

3. 结果分析

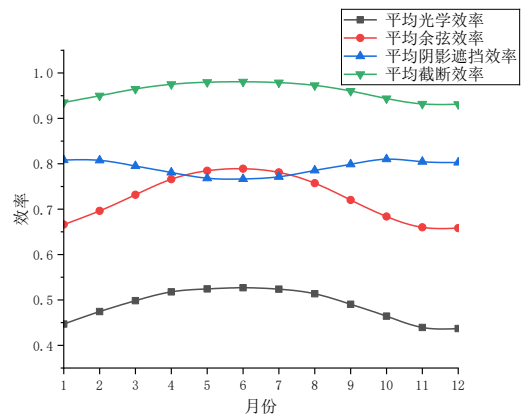


图 12 问题一各月效率曲线

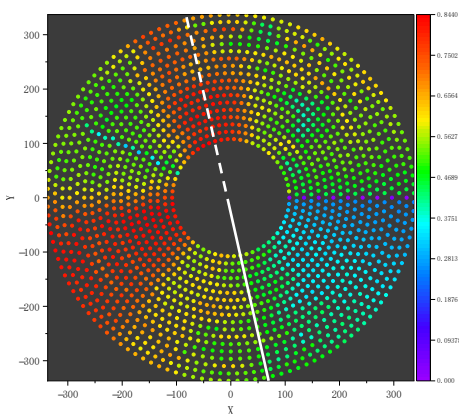


图 13 问题一各镜年平均光学效率彩点图

①针对各月相关光学效率的分析

绘制各月相关效率曲线，如图 12 所示。由图可见，平均光学效率、平均余弦效率、平均截断效率在夏至日达到峰值而夏至日的平均阴影遮挡效率则相对较高。各效率关于夏至日呈对称关系，春分日与秋分日各效率大致相同。同时，平均截断效率相对其余各效率较高，对平均光学效率的影响较小。

②针对各定日镜年平均光学效率的分析

统计问题一参数下各定日镜的年平均光学效率，如图 13 所示。由图可见，位于北侧的定日镜具有更高的平均光学效率，在后续的优化过程中应考虑在该方向架设更多的定日镜。

③针对定日镜间阴影遮挡考虑定日镜数目的分析

在计算光线被定日镜遮挡情况的过程中，为了加快求解速度，仅考虑了定日镜 A 周围的 11 面定日镜遮挡镜 A 采样点光线的情况。先分别考虑定日镜 A 周围 8、9、10、11、12、13 面定日镜 B，探究修改该参数对模型灵敏度的影响。为加快计算速度，仅计算春分日（ $D=0$ ）与秋分日（ $D=184$ ）的阴影遮挡效率 η_{sb} 。计算所得结果如表 4 所示。

表 4 针对定日镜间阴影遮挡考虑定日镜数目的分析结果表

考虑数量	8	9	10	11	12	13
η_{sb} 春分日	0.8023	0.7999	0.7950	0.7886	0.7831	0.7790
η_{sb} 秋分日	0.8067	0.8050	0.7989	0.7916	0.7860	0.7823

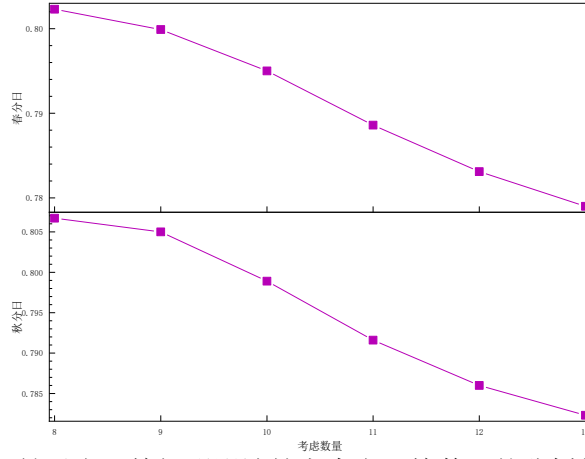


图 14 针对定日镜间阴影遮挡考虑定日镜数目的分析结果图

由图可见，阴影遮挡模型模型对定日镜考虑数量较敏感。但考虑的定日镜数量从 11 增至 12 对阴影遮挡效率的影响较小，乘以其效率后对光学效率的影响更小。

5.2 问题二模型的建立与求解

5.2.1 单目标优化模型的建立

1. 目标函数

问题一模型中给出的单位镜面面积输出热功率是问题二优化过程的目标函数，可将其写为

$$\max \bar{E}_{\text{mirror}} = \frac{\sum_{p=1}^{12} \sum_{q=1}^5 E_{\text{field}}(p, q)}{12 \times 5 \times \sum_{i=1}^n A_i} \quad (11)$$

2. 决策变量

吸收塔位置、定日镜排布方式都是本问题需要优化的决策变量。下面考虑对问题降维以减少决策变量数目。

①吸收塔位置 (x_1)

由于吸收塔位置可以不处于电站正中央, 其位置偏移可能存在更优方案使得平均输出热功率最大. 这一偏移主要是为了留出安装定日镜的优良位置.

为使更多定日镜的光学效率最高, 将吸收塔置于图 13 中白色直线位置处, 使白色虚线处的定日镜数量更多. 因此, 吸收塔位于镜场中心南偏东 12.7° , 距中心 x_1 (单位: 米).

②定日镜排布方式 ($x_2, x_3 \dots x_6$)

当定日镜呈圆形散点排布时, 塔式电站总体光学效率最高^[4], 因此仅考虑定日镜绕吸收塔呈圆形布阵的情况.

以吸收塔为原点, 自内向外, 从吸收塔正西方向开始顺时针安装定日镜, 如图 15 所示.

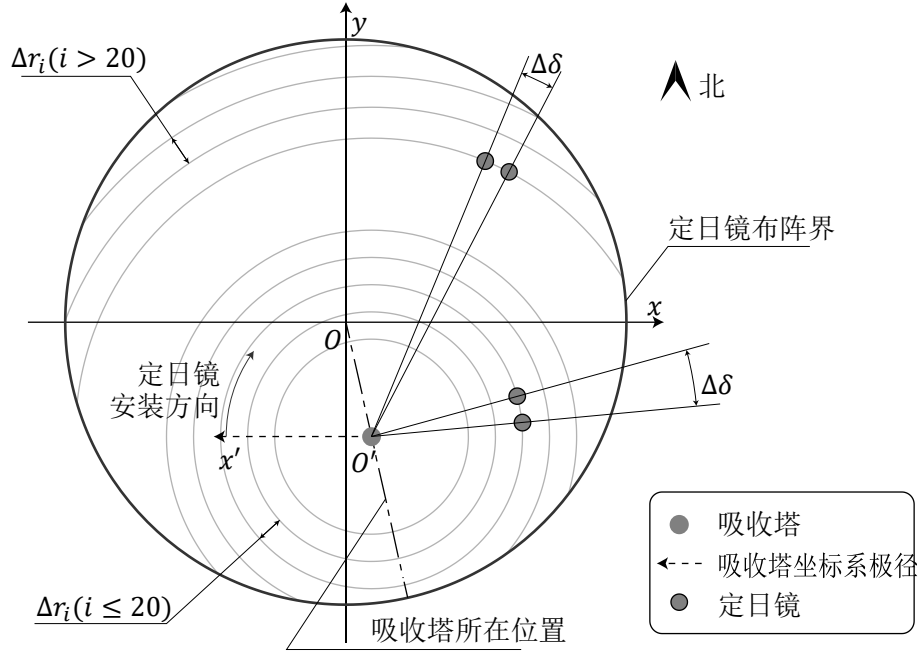


图 15 定日镜布阵示意

为尽量避免定日镜之间对阳光的遮挡, 每一定日镜架设圆与上一定日镜架设圆的径向间距由分段函数

$$\Delta r_i = \begin{cases} 15, & i \leq 20, \\ 0.5i + 5, & i > 20, \end{cases}$$

表示^[5].

每一定日镜圆上的定日镜均匀分布, 极角间距

$$\Delta \delta = \frac{2\pi}{\text{num}},$$

其中 num 为该圆安装定日镜数目

$$\text{num} = \left\lfloor \frac{2\pi r}{x} \right\rfloor,$$

式中记号 $\left\lfloor \frac{2\pi r}{x} \right\rfloor$ 代表对 $\frac{2\pi r}{x}$ 向下取整, r 为定日镜所在圆的半径, x 是左右两定

日镜最小间距.

分别针对前 20 排与后排的定日镜设计不同的 x , 分别记为 x_2, x_3 . 为避免定日镜间遮挡, 应满足 $x_3 > x_2$.

电站内所有定日镜安装高度相同,记为 x_4 ;镜面宽度与高度分别记为 x_5, x_6 。

综上所述,在本问题中共有 6 个自变量 $x_1, x_2 \dots x_6$ 会对结果产生影响。这些变量的含义与定义域如表 5 所示。

表 5 问题二遗传算法自变量说明

自变量	含义	定义域
x_1	吸收塔与原点距离（第四象限为正）	[0,350]
x_2	20 排以内定日镜间距	[7,17]
x_3	20 排以外定日镜间距	[10,20]
x_4	定日镜安装高度	[2,6]
x_5	镜面宽	[2,8]
x_6	镜面高	[2,8]

3. 约束条件

根据题意与实际情况,本问题的约束条件如下。

①相邻定日镜底座中心之间距离比镜面宽度多 5 m 以上,于是

$$x_2 > 5 + x_5.$$

②观察图 13,为增加镜场平均光学效率,应使靠近吸收塔的定日镜排布更密,即

$$x_3 > x_2.$$

③安装高度需保证镜面底部在旋转时不会触及地面,于是

$$x_4 > x_6 / 2.$$

④镜面宽度不小于镜面高度,于是

$$x_5 > x_6.$$

⑤镜场的额定年平均输出热功率超过 60 MW,于是

$$\bar{E}_{\text{field}} = \frac{\sum_{p=1}^{12} \sum_{q=1}^5 E_{\text{field}}(p, q)}{12 \times 5} \geq 60 \text{ MW}.$$

综合三者,建立问题二的单目标优化模型:

$$\max_{x_1, x_2, x_3, x_4, x_5, x_6} \bar{E}_{\text{mirror}} = \frac{\sum_{p=1}^{12} \sum_{q=1}^5 E_{\text{field}}(p, q)}{12 \times 5 \times \sum_{i=1}^n A_i} \quad (12)$$

$$\text{s.t.} \begin{cases} x_2 > 5 + x_5, \\ x_3 > x_2, \\ x_4 > x_6 / 2, \\ x_5 > x_6, \\ \bar{E}_{\text{field}} \geq 60 \text{ MW}. \end{cases}$$

5. 2. 2 模型求解

1. 遗传算法的设计

由于目标函数复杂且难以通过具体表达式表述,决策变量维度高、范围广,因此考虑使用遗传算法优化求解过程。算法具体流程如下。

算法步骤

Step 1: 初始值与定义域的确定,决策变量编码

对各决策变量赋迭代初值,依据表 5 赋定义域

对决策变量进行格雷编码转为 0-1 子串，以避免汉明悬崖

Step 2: 生成子代

基于轮盘赌选择法确定子代，基于概率进行交叉、突变，形成子代；保留亲本精英直接进入子代

Step 3: 迭代

不断重复 Step 2，直到达到最大迭代次数

确定算法参数：每代种群 50 个，编码子串长度为 10 位，最大迭代 100 轮，每代间交叉率为 0.5、变异率为 0.2。

2. 目标函数的简化

因为运算量极其庞大，在实际求解过程中对模型进行适当简化：

- ① 镜面采样点间距增大至 2 m；
- ② 忽略光锥的影响，视阳光为点光源。

3. 结果

求解得到的自变量取值如表 6 所示。

表 6 问题二遗传算法最优点

x_1	x_2	x_3	x_4	x_5	x_6
348.9736	8.9150	8.9443	2.0978	3.6481	3.4428

表 7 问题二年平均光学效率及输出功率表

年平均 光学效率	年平均 余弦效率	年平均阴影 遮挡效率	年平均 截断效率	年平均 输出热功率 (MW)	单位面积镜面年平均 输出热功率 (kW/m ²)
0.7009	0.8712	0.9991	0.9223	60.5584	0.6713

表 8 问题二参数设计表

吸收塔位置坐标	定日镜尺寸 (宽×高)	定日镜安装高度 (m)	定日镜总 面数	定日镜总面积 (m ²)
(76.721,-340.436)	3.65×3.44	2.0978	2481	3 1151.436

按照此方式求得的最优排布方式如图 16 所示。此时各月光学效率等结果如表 9 所示。

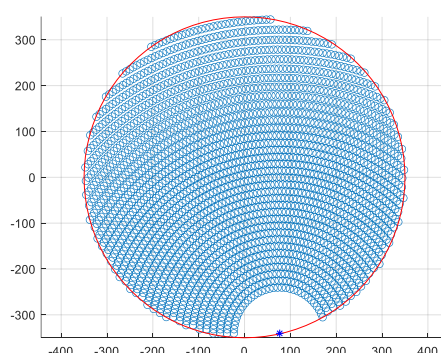


图 16 问题二所求最优定日镜布阵方式

表 9 问题二每月 21 日平均光学效率及输出功率

日期	平均 光学效率	平均 余弦效率	平均阴影 遮挡效率	平均 截断效率	单位面积镜面 平均输出热功率 (kW/m ²)
1 月 21 日	0.6634	0.8898	0.9978	0.8549	0.5631
2 月 21 日	0.6992	0.8859	0.9990	0.9040	0.6494

日期	平均 光学效率	平均 余弦效率	平均阴影 遮挡效率	平均 截断效率	单位面积镜面 平均输出热功率 (kW/m ²)
3月21日	0.7218	0.8765	0.9997	0.9426	0.7127
4月21日	0.7269	0.8596	1.0000	0.9675	0.7456
5月21日	0.7220	0.8435	1.0000	0.9793	0.7529
6月21日	0.7194	0.8378	1.0000	0.9824	0.7531
7月21日	0.7235	0.8472	1.0000	0.9771	0.7523
8月21日	0.7267	0.8649	0.9999	0.9614	0.7391
9月21日	0.7163	0.8802	0.9996	0.9316	0.6955
10月21日	0.6856	0.8880	0.9985	0.8848	0.6172
11月21日	0.6542	0.8902	0.9974	0.8430	0.5410
12月21日	0.6514	0.8903	0.9974	0.8394	0.5340

4. 结果分析

依据表 9 绘制各月相关效率曲线，并与问题一对照，如图 17 所示。由图可见，相同月份下经优化后问题二各光效效率较问题一均有提升，因此单位面积镜面年平均输出热功率较问题一而言也有提升。

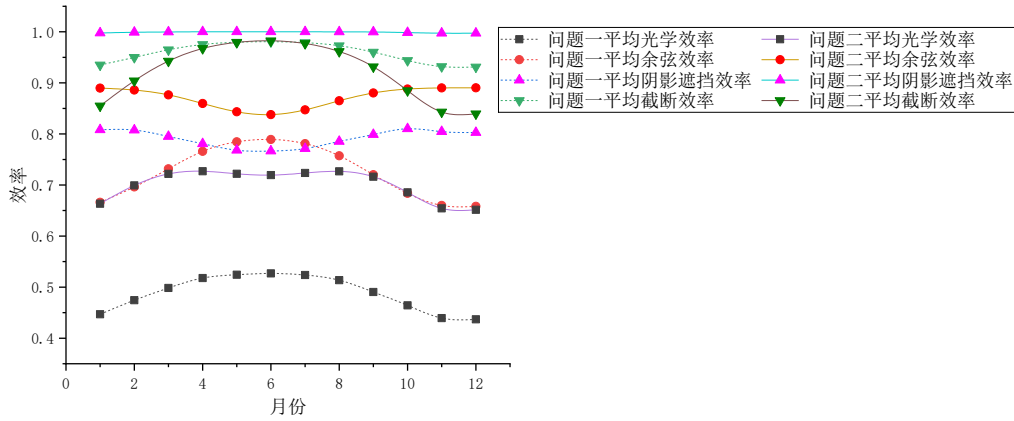


图 17 问题一、二各月效率曲线对照图

5.3 问题三模型的建立与求解

5.3.1 单目标优化模型的建立

1. 目标函数

问题三同样要求单位镜面面积年均输出热功率最大，目标函数与(11)式有相同的表达式

$$\max \bar{E}_{\text{mirror}} = \frac{\sum_{p=1}^{12} \sum_{q=1}^5 E_{\text{field}}(p, q)}{12 \times 5 \times \sum_{i=1}^n A_i}.$$

2. 决策变量

因为镜面参数条件限制的解除，在问题二的基础上需要重新确定决策变量。镜场的参数由吸收塔位置、定日镜排布方式与安装高度、各定日镜的长宽完全确定。

①吸收塔位置 (x_1)

吸收塔的排布规则与问题二相同，产生一个决策变量 x_1 。

②定日镜排布方式与安装高度 (x_2, x_3, \dots, x_5)

电站定日镜的安装高度不同主要是为了避免近日侧镜面遮挡远日侧. 由于安装高度无法随太阳移动而改变, 因此考虑其沿平均太阳方位角呈一定坡度变化, 设这一坡度为 x_2 (单位: 弧度). x_2 下限为 0; 当近日侧镜面最低、远日侧镜面最高时, x_2 取上限

$$x_{2\max} = \arctan \frac{6\text{m} - 2\text{m}}{700\text{m}} \approx 0.0057.$$

此外, 当安装坡度 x_2 距其下限较近时, 能在一定范围内抬高整体镜面安装高度, 设这一高度为 x_3 (单位: 米), 则 $x_3 \in [2\text{ m}, 6\text{ m}]$.

对于镜面的前后间距, 仍采用问题二的方式进行设计, 针对前 20 排与后排间距分别用记号 x_4, x_5 表示. 布阵时以吸收塔为原点, 顺时针最密布阵, 舍弃末端微小间隔.

③定日镜参数 ($x_6, x_7 \dots x_{7005}$)

由于镜面数目可变, 且各定日镜的镜面宽度与高度为自变量, 无法明确给出定日镜参数的决策变量个数. 经过多次试验后发现定日镜数目最大约 3300 面, 因此设定镜面数目上限为 3500 面以留出冗余空间.

综上, 在本问题中共有 7005 个自变量对结果产生影响, 这些变量的含义与定义域如表 10 所示.

表 10 问题三遗传算法自变量说明

自变量	含义	定义域
x_1	吸收塔与原点的距离	[0,350]
x_2	安装坡度	[0,0.0057]
x_3	安装高度追加	[2,6]
x_4	20 排以内定日镜间距	[7,17]
x_5	20 排以外定日镜间距	[10,20]
$x_i (6 \leq i \leq 3005)$	第 i 面定日镜镜面宽度	[2,8]
$x_j (3006 \leq j \leq 7005)$	第 j 面定日镜镜面高度	[2,8]

3. 约束条件

根据题意与实际情况, 本问题的约束条件如下.

①各镜面安装高度在 2~6 m, 于是

$$700x_1 + x_2 \leq 6.$$

②镜面绕水平轴转动时不会接触到地面, 因此安装高度 H_i 与镜面高度 h_i 还需满足

$$H_i > \frac{h_i}{2}, i = 1, 2, 3 \dots n.$$

③镜面安装高度大于镜面高度的一半, 于是

$$H_i > x_{3005+i}, i = 1, 2, 3 \dots n.$$

④镜面宽度不小于镜面高度, 于是

$$x_{5+i} > x_{3005+i}, i = 1, 2, 3 \dots n.$$

⑤两底座中心的间距大于两镜面宽度和的一半与最小间隔 5m 之和, 于是

$$x_4 > 5 + \frac{x_{5+i} + x_{5+i+1}}{2}, i = 1, 2, 3 \dots n.$$

⑥镜场的额定年平均输出热功率超过 60 MW, 于是

$$\bar{E}_{\text{field}} = \frac{\sum_{p=1}^{12} \sum_{q=1}^5 E_{\text{field}}(p, q)}{12 \times 5} \geq 60 \text{ MW}.$$

综合三者, 建立问题三的单目标优化模型:

$$\max_{x_1, x_2, x_3, x_4, x_5, x_6} \bar{E}_{\text{mirror}} = \frac{\sum_{p=1}^{12} \sum_{q=1}^5 E_{\text{field}}(p, q)}{12 \times 5 \times \sum_{i=1}^n A_i} \quad (6)$$

$$\text{s.t.} \begin{cases} 700x_1 + x_2 \leq 6, \\ H_i > \frac{h_i}{2}, i=1, 2, 3 \dots n, \\ H_i > x_{3005+i}, i=1, 2, 3 \dots n, \\ x_{5+i} > x_{3005+i}, i=1, 2, 3 \dots n, \\ x_4 > 5 + \frac{x_{5+i} + x_{5+i+1}}{2}, i=1, 2, 3 \dots n, \\ \bar{E}_{\text{field}} \geq 60 \text{ MW}. \end{cases}$$

5.3.2 模型求解

1. 遗传算法的设计与问题的简化

对模型的求解思路、简化过程与问题二相同。但由于问题三变量维度高，收敛困难，因此考虑提高遗传算法交叉率至 0.6，变异率至 0.3。

实际求解过程中，很难随机生成同时满足约束条件④的 3500 面镜面，因此当约束条件④不满足时，交换镜面宽度与高度的随机参数。

2. 结果

求解得到的部分参数如表 11 所示。按照此方式求得的最优排布方式如图 18 所示，此时各月光学效率等结果如表 14 所示。

表 11 问题三遗传算法最优点（部分数值）

x_1	x_2	x_3	x_4	x_5
-252.5037	0.0015	4.2224	7.1938	15.1983

表 12 问题三年平均光学效率及输出功率表

年平均光学效率	年平均余弦效率	年平均阴影遮挡效率	年平均截断效率	年平均输出功率 (MW)	单位面积镜面年平均输出热功率 (kW/m ²)
0.7020	0.8685	0.9991	0.9264	60.28	0.7456

表 13 问题三参数设计表

吸收塔位置坐标	定日镜总面数	定日镜总面积(m ²)
(75.7153, -335.9755)	2548	6 2832.7996

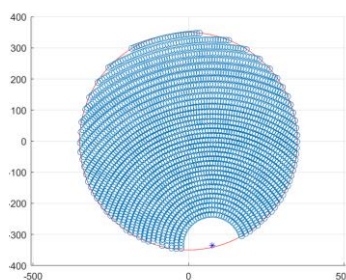


图 18 问题三所求最优定日镜布阵方式

表 14 问题三每月 21 日平均光学效率及输出功率

日期	平均 光学效率	平均 余弦效率	平均阴影 遮挡效率	平均 截断效率	单位面积镜面 平均输出热功率 (kW/m ²)
1 月 21 日	0.6682	0.8872	0.9978	0.8635	0.5735
2 月 21 日	0.6999	0.8832	0.9990	0.9074	0.6549
3 月 21 日	0.7216	0.8737	0.9998	0.9451	0.7152
4 月 21 日	0.7255	0.8569	1.0000	0.9685	0.7454
5 月 21 日	0.7200	0.8408	1.0000	0.9796	0.7519
6 月 21 日	0.7173	0.8352	1.0000	0.9825	0.7514
7 月 21 日	0.7214	0.8445	1.0000	0.9773	0.7514
8 月 21 日	0.7262	0.8622	0.9999	0.9636	0.7400
9 月 21 日	0.7168	0.8775	0.9996	0.9348	0.6996
10 月 21 日	0.6893	0.8853	0.9987	0.8919	0.6259
11 月 21 日	0.6602	0.8876	0.9974	0.8531	0.5523
12 月 21 日	0.6577	0.8876	0.9973	0.8499	0.5456

3. 结果分析

依据表 14 绘制各月相关效率曲线，并与问题二对照，如图 19 所示。由图可见，定日镜参数限制解除后平均光学效率在冬季有所提升，这是冬季平均截断效率提升所致。

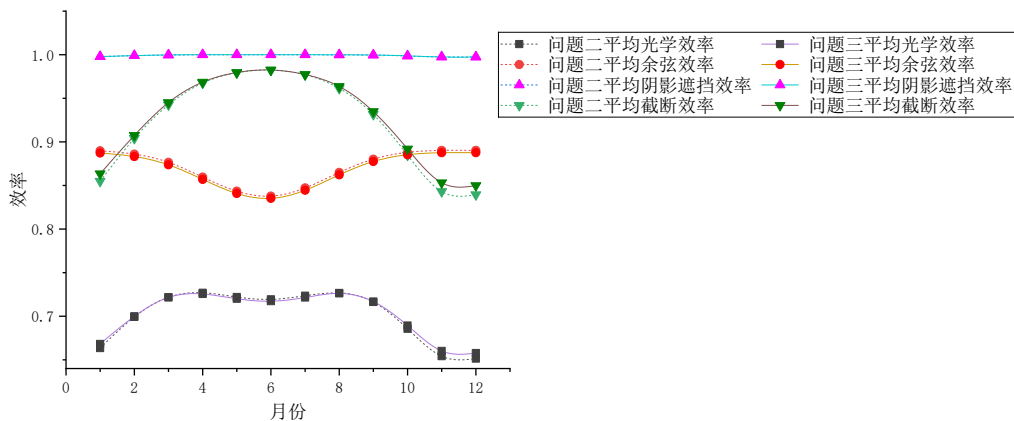


图 19 问题二、三各月效率曲线对照图

六 模型评价

6.1 模型的优点

1. 通过计算镜面采样点数据描述镜面整体数据，计算光锥中的采样光线相关数据描述光锥整体数据，在误差允许的范围内简化了模型；
2. 问题一考虑了定日镜对光线的遮挡情况；
3. 通过遗传算法智能搜索目标函数的最大值与最大值点，加快了计算速率。

6.2 模型的不足

1. 对光线的采样使计算结果精度较低；
2. 问题二与问题三中忽略了光锥的影响，仅视太阳为点光源，在一定程度上影响了截断效率的准确性。

6.3 模型推广

此模型运算简便，参数可调，可以推广至其他地理位置、其他设计需求的塔

式电站功率设计.

七 参考文献

- [1] 房淼森,逯静,沈志刚.塔式太阳能定日镜前端效率建模及其应用[J].常州信息职业技术学院学报,2021,20(03):20-24.
- [2] 周衍柏. 理论力学教程[M]. 5 版. 北京: 高等教育出版社, 2023: 121.
- [3] 张平等,太阳能塔式光热镜场光学效率计算方法[J],技术与市场,2021, 28(6):5-8.
- [4] 关弘扬,张艳梅,廖文俊,等.不同定日镜场排布形式对镜场光学效率的影响[J].装备机械, 2017(1):5.
- [5] 孙浩,高博,刘建兴.塔式太阳能电站定日镜场布局研究[J].发电技术,2021,42(06):690-698.

附录说明		
附录 1	支撑材料清单	
附录 2	各日期与 D 的关系	
附录 3	源程序代码	

附录 1 支撑材料清单

支撑材料清单		
问题 1	Q1.m	问题一求解程序
问题 2	Q2.m	问题二遗传算法主调函数
	Q2_judge.m	问题二约束条件函数
	Q2_func_dim.m	问题二生成定日镜阵列坐标
	Q2_func.m	问题二目标函数
	result2.xlsx	问题二结果
问题 3	Q3.m	问题三遗传算法主调函数
	Q3_judge.m	问题三约束条件函数
	Q3_func_dim.m	问题三生成定日镜阵列坐标
	Q3_func.m	问题三目标函数
	result3.xlsx	问题三结果
遗传算法依赖函数	b2d.m	格雷编码函数
	d2b.m	格雷解码函数
灵敏性分析	sen_re.m	灵敏性分析程序

文件说明：

- 拓展名为.xlsx 的文件为 EXCEL 文件，存放题中要求的结果
- 拓展名为.m 的文件为 MATLAB 程序文件，本文的运行环境是 MATLAB R2022b。运行程序时，需将题中给出的附录.xlsx、result2.xlsx、result3.xlsx 放置在同一文件夹下。

附录 2 各日期与 D 的关系

附表 1 各日期与 D 的关系

日期	D	日期	D
1 月 21 日	-59	7 月 21 日	122
2 月 21 日	-28	8 月 21 日	153
3 月 21 日	0	9 月 21 日	184
4 月 21 日	31	10 月 21 日	214
5 月 21 日	61	11 月 21 日	245
6 月 21 日	92	12 月 21 日	275

附录 3 源程序代码

Q1.m

问题一求解程序

```
%% 初始化
clear
clc
close all
%% 参数赋值
global data chang kuan h2 h1 as gs d h1
step=0.5; % 网格生成步长
H=3;
h1=4;
h2=80;
h3=8;
R0=3.5;
Rm=100;
chang=6;
kuan=6;
d=149597870700; % 日地距离
R1=6.963e8; % 太阳半径
fai=39.4;
q=0.05; % 遮挡角度允许误差（角度）
%% 导入数据
data=xlsread('附件.xlsx');
res={'平均光学效率','平均余弦效率','平均阴影遮挡效率','平均截断效率','单位平均热功率'};
%% 结果初始化
yita_sb_day=zeros(1,12);
yita_cos_day=zeros(1,12);
```

```

yita_at_day=zeros(1,12);
yita_trunc_day=zeros(1,12);
yita_ref_day=ones(1,12).*0.92;
%% 各镜数据存储
len=length(data);
res_of_all=zeros(len,3);
res_of_all(:,1:2)=data;
rec_time=0;
%% 反射光锥参量计算
theta_dot=atan(R1/d)*180/pi;
%% 遍历时间
D_ind=0;
for D=[-59,-28,0,31,61,92,122,153,184,214,245,275]
    D_ind=D_ind+1;
    ST_ind=0;
    %% 结果初始化
    yita_sb_time=zeros(1,5);
    yita_cos_time=zeros(1,5);
    yita_at_time=zeros(1,5);
    yita_trunc_time=zeros(1,5);
    E_time=zeros(1,5);
    %    for ST=[9]
    for ST=[9,10.5,12,13.5,15]
        ST_ind=ST_ind+1;
        E=0;    % 输出热功率初始化
        %% 计算太阳高度角与方位角
        de=asin(sin(2*pi*D/350)*sin(2*pi*23.45/360));
        omg=pi*(ST-12)/12;

        as=asin(cos(de)*cos(fai*pi/180)*cos(omg)+sin(de)*sin(fai*2*pi/360))*180/p
        i;    % 太阳高度角
        gs=acos((sin(de)-
        sin(as*pi/180)*sin(fai*2*pi/360))/cos(as*pi/180)*cos(fai*2*pi/360))*180/p
        i;    % 太阳方位角
        %% 计算 DNI
        DNI=1.366*((0.4237-0.00821*(6-H)^2)+(0.5055+0.00595*(6.5-
        H)^2)*exp(-(0.2711+0.01858*(2.5-H)^2)/sin(as*pi/180)));
        %% 遍历定日镜
        in=cell(1,len);
        yita_sb=zeros(len,1);
        yita_cos=zeros(len,1);
        yita_at=zeros(len,1);
        yita_trunc=ones(len,1).*(-1);
        for num=1:len

```

```

tic
x=data(num,1);
y=data(num,2);
[am,gm]=get_amgm(x,y);
bla_num=0;
unbla_num=0;
in_num=0;
unin_num=0;
%% 反射光锥轴线向量计算
s0=-
[cos(as*pi/180)*sin(gs*pi/180);cos(as*pi/180)*cos(gs*pi/180);sin(as*pi/18
0)];

n=[cos(am*pi/180)*sin(gm*pi/180);cos(am*pi/180)*cos(gm*pi/180);sin(am*pi/
180)];

s_dot=s0+2*abs(dot(s0,n))*n;
s_dot=s_dot./norm(s_dot);
s{5}=s_dot;
omg_dot=asin(s_dot(3))*180/pi;
%% 上下采样线方向向量计算
s{1}=s_dot+[0;0;sin(theta_dot*pi/180)/sin((90-omg_dot-
theta_dot)*pi/180)];
s{1}=s{1}./norm(s{1}); % 单位化
s{2}=s_dot+[0;0;sin(theta_dot*pi/180)/sin((90+omg_dot-
theta_dot)*pi/180)];
s{2}=s{2}./norm(s{2});
%% 左右采样线方向向量计算
m1=s{1}./cos((theta_dot+omg_dot)*pi/180)-s_dot;
temp_cr=cross(s_dot,m1);
m4=temp_cr.*norm(m1)./norm(temp_cr);
m3=-m4;
s{3}=m3+s_dot;
s{3}=s{3}./norm(s{3});
s{4}=m4+s_dot;
s{4}=s{4}./norm(s{4});
%% 划分网格
ind_max=floor(chang/step+1)*floor(kuan/step+1);
cosn=zeros(1,ind_max);
ind=1;
for i=-chang/2:step:chang/2
    for k=-kuan/2:step:kuan/2
        x0=i;
        y0=0;
        z0=k;
    end
end

```

```

%% 变换
A=[cos(gm*pi/180),sin(gm*pi/180),0;-
sin(gm*pi/180),cos(gm*pi/180),0;0,0,1];
B=[1,0,0;0,cos(am*pi/180),-
sin(am*pi/180);0,sin(am*pi/180),cos(am*pi/180)];
t=B*A*[x0;0;z0];
x1=t(1)+x;
y1=t(2)+y;
z1=t(3)+h1;
M=[x1;y1;z1];
%% 吸收塔遮挡判断

s_f=[cos(as*pi/180)*sin(gs*pi/180);cos(as*pi/180)*cos(gs*pi/180);sin(as*pi/180)];

if x1<0 && y1>0 % 仅判断第二象限是否被遮挡
    jijiao=(atan(y1/x1)+pi)*180/pi; % 极角
    theta_s=(180-gs)+90;
    det_j=atan(R0/Rm)*180/pi+q;
    if jijiao>theta_s-det_j && jijiao<theta_s+det_j
        det=4*(x1*s_f(1)+y1*s_f(2))^2-
4*(s_f(1)^2+s_f(2)^2)*(x1^2+y1^2-R0^2);
        if det<0
            %% 定日镜遮挡判断
            if jud_sb(s_f,s,num,M)
                black=1; % 不被遮挡
                unbla_num=unbla_num+1;
            else
                black=0; % 被遮挡
                bla_num=bla_num+1;
            end
        else
            t_sol=(-2*(x1*s_f(1)+y1*s_f(2))-
sqrt(det))/(2*(s_f(1)^2+s_f(2)^2));
            z_sol=z1+s_f(3)*t_sol;
            if z_sol>=0 && z_sol<=h2+h3/2
                black=0; % 被遮挡
                bla_num=bla_num+1;
            else
                %% 定日镜遮挡判断
                if jud_sb(s_f,s,num,M)
                    black=1; % 不被遮挡
                    unbla_num=unbla_num+1;
                else
                    black=0; % 被遮挡

```

```

        bla_num=bla_num+1;
    end
end
end
else
    %% 定日镜遮挡判断
    if jud_sb(s_f,s,num,M)
        black=1; % 不被遮挡
        unbla_num=unbla_num+1;
    else
        black=0; % 被遮挡
        bla_num=bla_num+1;
    end
end
else
    %% 定日镜遮挡判断
    if jud_sb(s_f,s,num,M)
        black=1; % 不被遮挡
        unbla_num=unbla_num+1;
    else
        black=0; % 被遮挡
        bla_num=bla_num+1;
    end
end
%% 截断效率计算
if black==1
    for line_ind=1:5
        det=4*(x1*s{line_ind}(1)+y1*s{line_ind}(2))^2-
4*(s{line_ind}(1)^2+s{line_ind}(2)^2)*(x1^2+y1^2-R0^2);
        if det<0
            unin_num=unin_num+1; % 不能射入
        else
            t_sol=(-
2*(x1*s{line_ind}(1)+y1*s{line_ind}(2))-
sqrt(det))/(2*(s{line_ind}(1)^2+s{line_ind}(2)^2));
            z_sol=z1+s{line_ind}(3)*t_sol;

            if z_sol>=h2-h3/2 && z_sol<=h2+h3/2
                in_num=in_num+1; % 能射入
                in{num}(ind)=0;
            else
                unin_num=unin_num+1; % 不能射入
                in{num}(ind)=1;
            end
        end
    end
end

```

```

                                end
                                end
                                end
                                %% 计算余弦效率

n_f=[cos(am*pi/180)*sin(gm*pi/180);cos(am*pi/180)*cos(gm*pi/180);sin(am*pi/180)];

                                cosn(ind)=real(dot(s_f,n_f));
                                ind=ind+1;
                                end
                                end
                                %% 统计阴影遮挡效率
                                yita_sb(num)=1-bla_num/(bla_num+unbla_num);
                                %% 计算单板平均余弦效率
                                yita_cos(num)=mean(cosn);
                                %% 计算大气透射率
                                dHR=sqrt(x^2+y^2+(h2-h1)^2);
                                yita_at(num)=0.99321-0.0001176*dHR+1.97e-8*dHR^2;
                                %% 计算截断效率
                                if yita_sb(num)~=0
                                    yita_trunc(num)=1-in_num/(in_num+unin_num);
                                else
                                    yita_trunc(num)=0;
                                end
                                toc

yita=yita_sb(num)*yita_cos(num)*yita_at(num)*yita_trunc(num)*yita_ref_day(D_ind);

                                res_of_all(num,3)=yita;
                                E=E+chang*kuan*yita;
                                fprintf('完成计算%d月%.1f时第%d块定日镜...\n',D_ind,ST,num);
                                end
                                E=E*DNI;
                                yita_sb_time(ST_ind)=mean(yita_sb);
                                yita_cos_time(ST_ind)=mean(yita_cos);
                                yita_at_time(ST_ind)=mean(yita_at);
                                yita_trunc_time(ST_ind)=mean(yita_trunc);
                                E_time(ST_ind)=E/(chang*kuan*len);
                                rec_time=rec_time+1;
                                end
                                yita_sb_day(D_ind)=mean(yita_sb_time);
                                yita_cos_day(D_ind)=mean(yita_cos_time);
                                yita_at_day(D_ind)=mean(yita_at_time);
                                yita_trunc_day(D_ind)=mean(yita_trunc_time);

```

```

E_day=mean(E_time);
%% 结果整理

res{D_ind,1}=yita_sb_day(D_ind)*yita_cos_day(D_ind)*yita_at_day(D_ind)*yita_trunc_day(D_ind)*yita_ref_day(D_ind);
res{D_ind,2}=yita_cos_day(D_ind);
res{D_ind,3}=yita_sb_day(D_ind);
res{D_ind,4}=yita_trunc_day(D_ind);
res{D_ind,5}=E_day;
fprintf("数据已记录至 res 与 res_of_all\n")
xlswrite("2023A-定日镜问题 1 结论.xlsx",res_of_all)
fprintf("文件已保存\n")
end

%% 计算定日镜高度角、方向角
function [am,gm]=get_amgm(x,y)
global h2 h1 as gs d
r=sqrt(x^2+y^2);
%% 计算定日镜高度角
theta1=atan((h2-h1)/r)*180/pi;
am=theta1+(as-theta1)/2;
%% 计算定日镜方位角
if y>0 && ( x>=0 || -atan(x/y)*180/pi+gs<=180 ) % 情况 1
    beta=180*atan(y/x)/pi;
    theta2=beta+gs-sign(beta)*90;
    l=sqrt(r^2+d^2-2*r*d*cos(theta2*pi/180));
    theta3=180*acos((r^2+l^2-d^2)/(2*r*l))/pi;
    if x>=0
        gm=180-(theta3/2-(90-beta));
    else
        gm=-(theta3/2-(90-beta));
    end
elseif y<0 && ( x<=0 || -atan(x/y)*180/pi+gs<=180 ) % 情况 2
    beta=180*atan(x/y)/pi;
    theta4=beta+180-gs;
    l=sqrt(r^2+d^2-2*r*d*cos(theta4*pi/180));
    gm=beta+1/2*180*acos((r^2+l^2-d^2)/(2*r*l))/pi;
elseif y>=0 && ( x<=0 && -atan(x/y)*180/pi+gs>=180 ) % 情况 3
    beta=180*abs(atan(x/y))/pi;
    theta5=(180-beta)+(180-gs);
    l=sqrt(r^2+d^2-2*r*d*cos(theta5*pi/180));
    gm=180-(beta-1/2*180/pi*acos((r^2+l^2-d^2)/(2*r*l)));
else % 情况 4
    beta=180*atan(x/y)/pi;

```

```

        theta6=180-gs+beta;
        l=sqrt(r^2+d^2-2*r*d*cos(theta6*pi/180));
        gm=180-1/2*180/pi*acos((r^2+l^2-d^2)/(2*r*l))+beta;
    end
end

%% 判断是否被定日镜遮挡: 1-不被遮挡, 0-被遮挡
function res=jud_sb(s_f,s,now_ind,M)
    global data chang kuan h1
    %% 寻找最近定日镜
    distance=((data(:,1)-data(now_ind,1)).^2 + (data(:,2)-
data(now_ind,2)).^2).^0.5;
    [~,near_ind]=mink(distance,10);
%     near_ind(find(distance(near_ind)<0.001))=[]; % 除去自身
%     near_ind=1:length(data);
%     near_ind=near_ind';
    %% 遍历遮挡定日镜
    for line_ind=0:5
        if line_ind==0
            s_now=s_f;
        else
            s_now=s{line_ind};
        end
        for ind=near_ind'
            if ind==now_ind
                continue
            end
            x0=data(now_ind,1);
            y0=data(now_ind,2);
            [am0,gm0]=get_amgm(x0,y0);
            x=data(ind,1);
            y=data(ind,2);
            [am,gm]=get_amgm(x,y);
            A0=[cos(gm0*pi/180),sin(gm0*pi/180),0;-
sin(gm0*pi/180),cos(gm0*pi/180),0;0,0,1];
            B0=[1,0,0;0,cos(am0*pi/180),-
sin(am0*pi/180);0,sin(am0*pi/180),cos(am0*pi/180)];
            T0=B0*A0;
            A=[cos(gm*pi/180),sin(gm*pi/180),0;-
sin(gm*pi/180),cos(gm*pi/180),0;0,0,1];
            B=[1,0,0;0,cos(am*pi/180),-
sin(am*pi/180);0,sin(am*pi/180),cos(am*pi/180)];
            T=B*A;
            b=T0\ (M-[x;y;h1]);

```



```

        ss=T0\s_now;
        xb=(ss(3)*b(1)-ss(1)*b(3))/ss(3);
        yb=(ss(3)*b(2)-ss(2)*b(3))/ss(3);
%         fprintf("xb=%f,yb=%f\n",xb,yb)
        if ((-kuan/2<=xb && xb<=kuan/2) && (-chang/2<=yb &&
yb<=chang/2))
            res=0;
            fprintf("挡")
            return
        end
    end
end
res=1;
return
end

```

Q2.m

问题二遗传算法主调函数

```

%% 问题二多维函数遗传算法与非线性优化
%% 初始化
clear;
close all;      % 清图
clc;
NP=50;          % 种群数量
L=10;           % 单个维度二进制数段长度
Dim=6;          % 维度
Pc=0.5;         % 交叉率
Pm=0.2;         % 变异率
Mc=20;          % 交叉最多尝试次数
Mm=50;          % 变异最多尝试次数
G=100;          % 最大遗传次数
Xs=[350,20,10,6,8,8];      % 定义域上限
Xx=[0,6,5.5,2,2,2];      % 定义域下限
f=randi([0,1],NP,L*Dim);  % 随机获取初始种群
boxMax=20;        % 箱型图绘制代数
boxPage=20;       % 箱型图单图显示数据数
%% 遗传算法
tic
% 预声明变量
trace1(G)=0;
trace2(G)=0;
trace3(G)=0;
maxIndex(G,Dim)=0;

```

```

fitHis(NP,boxMax)=0;
if boxMax>G      % 纠错
    boxMax=g;
end
% 修改不符合约束条件的初始种群，直至全部符合条件为止
for i=1:NP
    while ~Q2_judge(b2d(f(i,:),Xx,Xs,L))
        f(i,:)=randi([0,1],1,L*Dim);
    end
end
for k=1:G
    fprintf('正在计算问题 2 第%d 代...\n',k);
    %% 二进制解码为定义域内的十进制
    for i=1:NP
        tic
        U=f(i,:); % 获取第 i 条染色体
        x(i,:)=b2d(U,Xx,Xs,L);
        Fit(i)=Q2_func_dim(x(i,:)); % 计算每个样本适应度
        fprintf('代数%d, 种群序号%d, 适应度%f\n',k,i,Fit(i))
        toc
    end
    maxFit=max(Fit); % 最大适应度
    minFit=min(Fit); % 最小适应度
    rr=find(maxFit==Fit); % 找出最大适应度所在个体索引，返回数或数组
    fBest=f(rr(1,1),:); % 历代最优个体
    xBest=x(rr(1,1),:); % 最优个体对应的十进制染色体
    Fit1=(Fit-minFit)/(maxFit-minFit); % 适应度归一化
    %% 基于轮盘赌选择法的复制操作
    sum_Fit=sum(Fit1); % 计算种群中所有个体适应值的和
    fitvalue=Fit1/sum_Fit; % 计算每个种群的选择概率
    fitvalue=(cumsum(fitvalue)); % 计算每个种群的累计概率
    ms=sort(rand(NP,1)); % 随机生成 NP 个(0,1)的值，并排序
    fiti=1; % 旧种群当前指针
    newi=1; % 新种群当前指针
    while newi<=NP % 随机复制，并使适应度大的遗传下去
        if ms(newi) < fitvalue(fiti)
            nf(newi,:)=f(fiti,:); % 复制
            newi=newi+1;
        else
            fiti=fiti+1;
            if fiti>NP
                break;
            end
        end
    end
end

```

```

end
%% 基于概率的交叉操作
for i=1:NP-1
    p=rand; % 随机生成一个处于[0,1]的概率 p
    if p<Pc % 满足交叉条件
        t=1;
        while t<Mc
            % 开始尝试交叉
            tnf=zeros(1,L*Dim);
            q=randi([0,1],1,L*Dim); % 随机生成要交叉的基因位置
            for j=1:L*Dim
                tnf(j)=nf(i+q(j),j);
            end
            if Q2_judge(b2d(tnf,Xx,Xs,L)) % 判断当前交叉结果是否符合约束
条件
                nf(i,:)=tnf(:);
                break
            end
            t=t+1;
        end
    end
end
%% 基于概率的变异操作
i=1;
while i<=round(NP*Pm) % 控制变异染色体总数
    h=randi([1,NP],1,1); % 随机选择一个需要变异的染色体索引
    t=1;
    while t<Mm
        tnf=nf(h,:);
        for j=1:round(L*Dim*Pm) % 控制变异基因数
            g=randi([1,L*Dim],1,1); % 随机需要变异的基因索引
            tnf(g)=~tnf(g); % 取反
        end
        if Q2_judge(b2d(tnf,Xx,Xs,L)) % 判断当前变异结果是否符合约束条件
            nf(i,:)=tnf(:);
            break
        end
        t=t+1;
    end
    i=i+1;
end
%% 下一代预备
if k<=boxMax
    fitHis(:,k)=Fit';

```

```

end
f=nf; % 种群位置搬家
f(1,:)=fBest; % 保留上代最优个体
trace1(k)=maxFit; % 历代最优适应度
trace2(k)=mean(Fit); % 平均适应度
trace3(k)=minFit; % 最小适应度
maxIndex(k,:)=xBest(:); % 最大适应度索引
end
toc
%% 绘制适应度过程图
figure
subplot(2,1,1);
hold on
plot(1:k,trace1(1:k),'^r-');
plot(1:k,trace2(1:k),'ob-');
plot(1:k,trace3(1:k),'vg-');
title('历代适应度');
xlabel('进化代数');
ylabel('适应度');
legend({'最大值','平均值','最小值'},'Location','SouthEast');
hold off
subplot(2,1,2);
hold on
plot(1:G,trace1(1:G),'^r-');
plot([0,G],[trace1(G),trace1(G)],'LineWidth',1.5);
yl=ylim;
for i=1:G
    if trace1(i)==trace1(G)
        plot(i,trace1(G),'b*', 'LineWidth',1.5, 'MarkerSize',6);
        text(i,trace1(G)-(yl(2)-yl(1))/30, strcat(num2str(i),'代'));
        break
    end
end
title('历代最大适应度');
xlabel('进化代数');
ylabel('适应度最大值');
legend({'历代最大值','全局最大值','初次达到最大值点'}, 'Location', 'SouthEast');
hold off
%% 绘制条件范围内适应度箱型图
figure
page=ceil(boxMax/boxPage);
for i=1:page
    subplot(page,1,i);

```

```

    hold on
    boxplot(fitHis(:,(i-1)*boxPage+1:min([boxMax,i*boxPage])),(i-
1)*boxPage+1:min([boxMax,i*boxPage]));
    title(strcat('第',num2str((i-
1)*boxPage+1),'~',num2str(min([boxMax,i*boxPage])), '代适应度箱型图'));
    xlabel('进化代数');
    ylabel('适应度');
    plot([0,boxMax+1],[trace1(G),trace1(G)], 'LineWidth',1.5);
    legend({strcat('最大值:',num2str(trace1(G)))},'Location','SouthEast');
    set(gca,'XTickLabelRotation',70); % 倾斜 x 坐标轴数字
    hold off
end
%% 输出结果
fprintf('最大值: ');
disp(trace1(G));
fprintf('最大值点: ');
disp(maxIndex(G,:));
fprintf('精度: x±')
disp((Xs-Xx)/(2^L-1));

```

Q2_judge.m

问题二约束条件函数

```

function flag = Q2_judge( x )
%判断 x 是否符合约束条件, 若符合, 则返回 1
flag=0;
if x(2)>5+x(5) && x(3)>x(2) && x(4)>x(6)/2 && x(5)>x(6)
    flag=1;
    return
end
end

```

Q2_func_dim.m

问题二生成定日镜阵列坐标

```

function E_year = Q2_func_dim(arg)
%问题二目标函数
%% 计算吸收塔坐标
xi=arg(1)*[cos(-77.3*pi/180),sin(-77.3*pi/180)];
%% 布阵, 生成定日镜坐标
aff_num=-1; % 当前行定日镜数目
lin_ind=1; % 行号
distan=100; % 当前离吸收塔距离
coo_ls=[]; % 坐标列表

```

```

num_coo=0; % 当前坐标数
while aff_num ~= 0 % 循环行
    %% 实现 do-while 循环
    aff_num=0;
    %% 生成相对于吸收塔的极坐标
    if lin_ind<=20
        lin_num=floor(distan*2*pi/arg(2));
    else
        lin_num=floor(distan*2*pi/arg(3));
    end
    the=2*pi/lin_num;
    now_the=0;
    for the_ind=1:lin_num
        %% 转直角坐标
        now_the=now_the+the;
        coo_now=distan.*[cos(now_the),sin(now_the)];
        %% 改变参考系
        coo_now=xi-coo_now;
        %% 判断是否在场内地
        if dot(coo_now,coo_now)<=122500
            num_coo=num_coo+1;
            coo_ls(num_coo,1)=coo_now(1);
            coo_ls(num_coo,2)=coo_now(2);
            aff_num=aff_num+1;
        end
    end
    if lin_ind<=20
        distan=distan+15;
    else
        distan=distan+0.5*lin_ind+5;
    end
    lin_ind=lin_ind+1;
end
coo_tran=coo_ls;
coo_tran(:,1)=coo_tran(:,1)-xi(1);
coo_tran(:,2)=coo_tran(:,2)-xi(2);
E_year=Q2_func(coo_tran,arg(5),arg(6),arg(4));
%% 出图
close all
figure
hold on
scatter(coo_ls(:,1), coo_ls(:,2))
rectangle('Position', [0-350,0-350,2*350,2*350], 'Curvature', [1
1], 'EdgeColor', 'r');

```

```

plot(xi(1),xi(2),'b*')
axis equal
grid on
fprintf('计算完成，镜面数目%d',length(coo_ls))
end

```

Q2_func.m

问题二目标函数

```

function E_year = Q2_func(data, chang, kuan, h1)
%问题二求解函数
%% 参数赋值
step=2; % 网格生成步长
H=3;
h2=80;
h3=8;
R0=3.5;
Rm=100;
d=149597870700; % 日地距离
R1=6.963e8; % 太阳半径
fai=39.4;
q=0.05; % 遮挡角度允许误差（角度）
%% 结果初始化
yita_sb_day=zeros(1,12);
yita_cos_day=zeros(1,12);
yita_at_day=zeros(1,12);
yita_trunc_day=zeros(1,12);
yita_ref_day=ones(1,12).*0.92;
%% 遍历时间
D_ind=0;
for D=[-59,-28,0,31,61,92,122,153,184,214,245,275]
    D_ind=D_ind+1;
    ST_ind=0;
    %% 结果初始化
    yita_sb_time=zeros(1,5);
    yita_cos_time=zeros(1,5);
    yita_at_time=zeros(1,5);
    yita_trunc_time=zeros(1,5);
    E_time=zeros(1,5);
    for ST=[9,10.5,12,13.5,15]
        ST_ind=ST_ind+1;
        E=0; % 输出热功率初始化
        %% 计算太阳高度角与方位角
        de=asin(sin(2*pi*D/350)*sin(2*pi*23.45/360));
    end
end

```

```

omg=pi*(ST-12)/12;

as=asin(cos(de)*cos(fai*pi/180)*cos(omg)+sin(de)*sin(fai*2*pi/360))*180/p
i;      % 太阳高度角
gs=acos((sin(de)-
sin(as*pi/180)*sin(fai*2*pi/360))/cos(as*pi/180)*cos(fai*2*pi/360))*180/p
i; % 太阳方位角
%% 计算 DNI
DNI=1.366*((0.4237-0.00821*(6-H)^2)+(0.5055+0.00595*(6.5-
H)^2)*exp(-(0.2711+0.01858*(2.5-H)^2)/sin(as*pi/180)));
%% 遍历定日镜
len=length(data);
in=cell(1,len);
yita_sb=zeros(len,1);
yita_cos=zeros(len,1);
yita_at=zeros(len,1);
yita_trunc=ones(len,1).*(-1);
for num=1:len
    x=data(num,1);
    y=data(num,2);
    r=sqrt(x^2+y^2);
    %% 计算定日镜高度角
    theta1=atan((h2-h1)/r)*180/pi;
    am=theta1+(as-theta1)/2;
    %% 计算定日镜方位角
    if y>0 && ( x>=0 || -atan(x/y)*180/pi+gs<=180 ) % 情况 1
        beta=180*atan(y/x)/pi;
        theta2=beta+gs-sign(beta)*90;
        l=sqrt(r^2+d^2-2*r*d*cos(theta2*pi/180));
        theta3=180*acos((r^2+l^2-d^2)/(2*r*l))/pi;
        if x>=0
            gm=180-(theta3/2-(90-beta));
        else
            gm=-(theta3/2-(90-beta));
        end
    elseif y<0 && ( x<=0 || -atan(x/y)*180/pi+gs<=180 ) % 情况 2
        beta=180*atan(x/y)/pi;
        theta4=beta+180-gs;
        l=sqrt(r^2+d^2-2*r*d*cos(theta4*pi/180));
        gm=beta+1/2*180*acos((r^2+l^2-d^2)/(2*r*l))/pi;
    elseif y>=0 && ( x<=0 && -atan(x/y)*180/pi+gs>=180 ) % 情况 3
        beta=180*abs(atan(x/y))/pi;
        theta5=(180-beta)+(180-gs);
        l=sqrt(r^2+d^2-2*r*d*cos(theta5*pi/180));
    end
end

```



```

        gm=180-(beta-1/2*180/pi*acos((r^2+l^2-d^2)/(2*r*l)));
else % 情况 4
    beta=180*atan(x/y)/pi;
    theta6=180-gs+beta;
    l=sqrt(r^2+d^2-2*r*d*cos(theta6*pi/180));
    gm=180-1/2*180/pi*acos((r^2+l^2-d^2)/(2*r*l))+beta;
end
bla_num=0;
unbla_num=0;
in_num=0;
unin_num=0;
%% 反射光锥轴线向量计算
s0=-
[cos(as*pi/180)*sin(gs*pi/180);cos(as*pi/180)*cos(gs*pi/180);sin(as*pi/180)];

n=[cos(am*pi/180)*sin(gm*pi/180);cos(am*pi/180)*cos(gm*pi/180);sin(am*pi/180)];

s_dot=s0+2*abs(dot(s0,n))*n;
s_dot=s_dot./norm(s_dot);
%% 划分网格
ind_max=floor(chang/step+1)*floor(kuan/step+1);
cosn=zeros(1,ind_max);
ind=1;
for i=-chang/2:step:chang/2
    for k=-kuan/2:step:kuan/2
        x0=i;
        y0=0;
        z0=k;
        %% 变换
        A=[cos(gm*pi/180),sin(gm*pi/180),0;-
sin(gm*pi/180),cos(gm*pi/180),0;0,0,1];
        B=[1,0,0;0,cos(am*pi/180),-
sin(am*pi/180);0,sin(am*pi/180),cos(am*pi/180)];
        t=A*B*[x0;0;z0];
        x1=t(1)+x;
        y1=t(2)+y;
        z1=t(3)+h1;
        %% 遮挡判断

s_f=[cos(as*pi/180)*sin(gs*pi/180);cos(as*pi/180)*cos(gs*pi/180);sin(as*pi/180)];

        if x1<0 && y1>0 % 仅判断第二象限是否被遮挡
            jijiao=(atan(y1/x1)+pi)*180/pi; % 极角

```

```

        theta_s=(180-gs)+90;
        det_j=atan(R0/Rm)*180/pi+q;
        if jijiao>theta_s-det_j && jijiao<theta_s+det_j
            det=4*(x1*s_f(1)+y1*s_f(2))^2-
4*(s_f(1)^2+s_f(2)^2)*(x1^2+y1^2-R0^2);
            if det<0
                black=1; % 不被遮挡
                unbla_num=unbla_num+1;
            else
                t_sol=(-2*(x1*s_f(1)+y1*s_f(2))-
sqrt(det))/(2*(s_f(1)^2+s_f(2)^2));
                z_sol=z1+s_f(3)*t_sol;
                if z_sol>=0 && z_sol<=h2+h3/2
                    black=0; % 被遮挡
                    bla_num=bla_num+1;
                else
                    black=1; % 不被遮挡
                    unbla_num=unbla_num+1;
                end
            end
        else
            black=1;
            unbla_num=unbla_num+1;
        end
    else
        black=1;
        unbla_num=unbla_num+1;
    end
end
%% 截断效率计算
if black==1
    det=4*(x1*s_dot(1)+y1*s_dot(2))^2-
4*(s_dot(1)^2+s_dot(2)^2)*(x1^2+y1^2-R0^2);
    if det<0
        unin_num=unin_num+1; % 不能射入
    else
        t_sol=(-2*(x1*s_dot(1)+y1*s_dot(2))-
sqrt(det))/(2*(s_dot(1)^2+s_dot(2)^2));
        z_sol=z1+s_dot(3)*t_sol;

        if z_sol>=h2-h3/2 && z_sol<=h2+h3/2
            in_num=in_num+1; % 能射入
            in{num}(ind)=0;
        else
            unin_num=unin_num+1; % 不能射入
        end
    end
end

```

```

                                in{num}(ind)=1;
                                end
                                end
                                end
                                %% 计算余弦效率

n_f=[cos(am*pi/180)*sin(gm*pi/180);cos(am*pi/180)*cos(gm*pi/180);sin(am*pi/180)];

                                cosn(ind)=real(dot(s_f,n_f));
                                ind=ind+1;
                                end
                                end
                                %% 统计阴影遮挡效率
                                yita_sb(num)=1-bla_num/(bla_num+unbla_num);
                                %% 计算单板平均余弦效率
                                yita_cos(num)=mean(cosn);
                                %% 计算大气透射率
                                dHR=sqrt(x^2+y^2+(h2-h1)^2);
                                yita_at(num)=0.99321-0.0001176*dHR+1.97e-8*dHR^2;
                                %% 计算截断效率
                                if yita_sb(num)~=0
                                    yita_trunc(num)=1-in_num/(in_num+unin_num);
                                else
                                    yita_trunc(num)=0;
                                end

yita=yita_sb(num)*yita_cos(num)*yita_at(num)*yita_trunc(num)*yita_ref_day(D_ind);

                                E=E+chang*kuan*yita;

                                end
                                E=E*DNI;
                                yita_sb_time(ST_ind)=mean(yita_sb);
                                yita_cos_time(ST_ind)=mean(yita_cos);
                                yita_at_time(ST_ind)=mean(yita_at);
                                yita_trunc_time(ST_ind)=mean(yita_trunc);
                                E_time(ST_ind)=E/(chang*kuan*len);
                                E_all_time(ST_ind)=E;
                                end
                                fprintf('\t 完成计算%d 月...\n',D_ind);
                                yita_sb_day(D_ind)=mean(yita_sb_time);
                                yita_cos_day(D_ind)=mean(yita_cos_time);
                                yita_at_day(D_ind)=mean(yita_at_time);
                                yita_trunc_day(D_ind)=mean(yita_trunc_time);

```

```

    E_day=mean(E_time);
    E_all_day(D_ind)=2*mean(E_all_time);
    %% 结果整理

    res{D_ind,1}=yita_sb_day(D_ind)*yita_cos_day(D_ind)*yita_at_day(D_ind)*yita_trunc_day(D_ind)*yita_ref_day(D_ind);
    res{D_ind,2}=yita_cos_day(D_ind);
    res{D_ind,3}=yita_sb_day(D_ind);
    res{D_ind,4}=yita_trunc_day(D_ind);
    res{D_ind,5}=E_day;
    E_year=E_day;
end
E_all_year=mean(E_all_day(D_ind));
fprintf('年平均输出热功率%d, 单位面积功率%d\n',E_all_year,E_year)
ret=E_all_year^E_year;

```

b2d.m

格雷解码，问题二、三遗传算法依赖函数

```

function d=b2d(str,Xx,Xs,L)
%格雷解码并将二进制转为定义域内的十进制
%b2d(str,Xx,Xs,L),str-二进制行向量, Xx-下限, Xs-上限, L 单维度二进制串长度
    Dim=length(Xx);
    m(Dim)=0;
    n=zeros(1,L*Dim);
    for k=1:Dim
        n((k-1)*L+1)=str((k-1)*L+1); % 格雷编码解码
        for j=2:L
            n((k-1)*L+j)=xor(n((k-1)*L+j-1),str((k-1)*L+j)); % xor—异或运算
        end
    end
    for di=1:Dim
        for j=1:L % 二进制转十进制
            m(di)=m(di)+n((di-1)*L+j)*2^(j-1);
        end
        d(di)=Xx(di)+m(di)*(Xs(di)-Xx(di))/(2^L-1); % 十进制解码至定义域
        m(di)=0;
    end
end

```

d2b.m

格雷编码，问题三遗传算法依赖函数

```

function str = d2b (arr,Xx,Xs,L)
%格雷编码
%d2b(arr,Xx,Xs,L),arr-十进制数据数组, Xx-下限, Xs-上限, L 单维度二进制串长度;
输出 str 为二进制行向量
Dim=length(Xx);
str=zeros(1,Dim*L);
m=zeros(1,Dim);
n=zeros(1,Dim);
for k=1:Dim
    m(k)=round((arr(k)-Xx(k))*(2^L-1)/(Xs(k)-Xx(k))); % 分散至二进制定义域
    for i=1:L % 十进制转二进制
        n((k-1)*L+i)=mod(m(k),2);
        m(k)=floor(m(k)/2);
    end
end
for k=1:Dim
    str((k-1)*L+1)=n((k-1)*L+1);
    for i=2:L % 格雷编码
        str((k-1)*L+i)=xor(n((k-1)*L+i-1),n((k-1)*L+i));
    end
end
end

```

Q3.m

问题三遗传算法主调函数

```

%% 问题三多维函数遗传算法与非线性优化
%% 初始化
clear;
close all; % 清图
clc;
NP=50; % 种群数量
L=20; % 单个维度二进制数段长度
Dim=7005; % 维度
Pc=0.6; % 交叉率
Pm=0.3; % 变异率
Mc=20; % 交叉最多尝试次数
Mm=50; % 变异最多尝试次数
G=100; % 最大遗传次数
Xs=ones(1,7005).*8; % 定义域上限
Xs(1:5)=[0.0057,6,17,20,350];
Xx=ones(1,7005).*2; % 定义域下限
Xx(1:5)=[0,2,7,10,0];

```

```

f=randi([0,1],NP,L*Dim);    % 随机获取初始种群
boxMax=20;                  % 箱型图绘制代数
boxPage=20;                 % 箱型图单图显示数据数
%% 遗传算法
tic
% 预声明变量
trace1(G)=0;
trace2(G)=0;
trace3(G)=0;
maxIndex(G,Dim)=0;
fitHis(NP,boxMax)=0;
if boxMax>G    % 纠错
    boxMax=g;
end
% 修改不符合约束条件的初始种群，直至全部符合条件为止
ju_res=cell(1,NP);
for i=1:NP
    ju_res{i}=Q3_judge(b2d(f(i,:),Xx,Xs,L));
    while ~ju_res{i}{1}
        f(i,:)=randi([0,1],1,L*Dim);
        fprintf('重构_')
        ju_res{i}=Q3_judge(b2d(f(i,:),Xx,Xs,L));
    end
end
for k=1:G
    fprintf('正在计算问题 3 第%d 代...\n',k);
    %% 二进制解码为定义域内的十进制
    for i=1:NP
        tic
        if ju_res{i}{1} ~= 2
            U=f(i,:);    % 获取第 i 条染色体
            x(i,:)=b2d(U,Xx,Xs,L);
        else
            f(i,:)=d2b(ju_res{i}{2}(:),Xx,Xs,L);
            x(i,:)=ju_res{i}{2}(:);
        end
        Fit(i)=Q3_func_dim(x(i,:));    % 计算每个样本适应度
        fprintf('代数%d, 种群序号%d, 适应度%f\n',k,i,Fit(i))
        toc
    end
    maxFit=max(Fit);    % 最大适应度
    minFit=min(Fit);    % 最小适应度
    rr=find(maxFit==Fit);    % 找出最大适应度所在个体索引，返回数或数组
    fBest=f(rr(1,1),:);    % 历代最优个体

```

```

xBest=x(rr(1,1),:);      % 最优个体对应的十进制染色体
Fit1=(Fit-minFit)/(maxFit-minFit); % 适应度归一化
%% 基于轮盘赌选择法的复制操作
sum_Fit=sum(Fit1); % 计算种群中所有个体适应值的和
fitvalue=Fit1/sum_Fit; % 计算每个种群的选择概率
fitvalue=(cumsum(fitvalue)); % 计算每个种群的累计概率
ms=sort(rand(NP,1)); % 随机生成 NP 个(0,1)的值，并排序
fiti=1; % 旧种群当前指针
newi=1; % 新种群当前指针
while newi<=NP % 随机复制，并使适应度大的遗传下去
    if ms(newi) < fitvalue(fiti)
        nf(newi,:)=f(fiti,:); % 复制
        newi=newi+1;
    else
        fiti=fiti+1;
        if fiti>NP
            break;
        end
    end
end
%% 基于概率的交叉操作
for i=1:NP-1
    p=rand; % 随机生成一个处于[0,1]的概率 p
    if p<Pc % 满足交叉条件
        t=1;
        while t<Mc
            % 开始尝试交叉
            tnf=zeros(1,L*Dim);
            q=randi([0,1],1,L*Dim); % 随机生成要交叉的基因位置
            for j=1:L*Dim
                tnf(j)=nf(i+q(j),j);
            end
            ju_res{i}=Q3_judge(b2d(tnf,Xx,Xs,L));
            if ju_res{i}{1}==1 % 判断当前交叉结果是否符合约束条件
                nf(i,:)=tnf(:);
                break
            end
            t=t+1;
        end
    end
end
%% 基于概率的变异操作
i=1;
while i<=round(NP*Pm) % 控制变异染色体总数

```

```

h=randi([1,NP],1,1); % 随机选择一个需要变异的染色体索引
t=1;
while t<Mm
    tnfnf(h,:);
    for j=1:round(L*Dim*Pm) % 控制变异基因数
        g=randi([1,L*Dim],1,1); % 随机需要变异的基因索引
        tnfnf(g)=~tnfnf(g); % 取反
    end
    ju_res{i}=Q3_judge(b2d(tnfnf,Xx,Xs,L));
    if ju_res{i}{1}==1 % 判断当前变异结果是否符合约束条件
        tnfnf(i,:)=tnfnf(:);
        break
    end
    t=t+1;
end
i=i+1;
end
%% 下一代预备
if k<=boxMax
    fitHis(:,k)=Fit';
end
f=tnfnf; % 种群位置搬家
f(1,:)=fBest; % 保留上代最优个体
trace1(k)=maxFit; % 历代最优适应度
trace2(k)=mean(Fit); % 平均适应度
trace3(k)=minFit; % 最小适应度
maxIndex(k,:)=xBest(:); % 最大适应度索引
end
toc
%% 绘制适应度过程图
figure
subplot(2,1,1);
hold on
plot(1:k,trace1(1:k),'^r-');
plot(1:k,trace2(1:k),'ob-');
plot(1:k,trace3(1:k),'vg-');
title('历代适应度');
xlabel('进化代数');
ylabel('适应度');
legend({'最大值','平均值','最小值'},'Location','SouthEast');
hold off
subplot(2,1,2);
hold on
plot(1:G,trace1(1:G),'^r-');

```



```

plot([0,G],[trace1(G),trace1(G)], 'LineWidth',1.5);
yl=ylim;
for i=1:G
    if trace1(i)==trace1(G)
        plot(i,trace1(G),'b*', 'LineWidth',1.5, 'MarkerSize',6);
        text(i,trace1(G)-(yl(2)-yl(1))/30, strcat(num2str(i), '代'));
        break
    end
end
title('历代最大适应度');
xlabel('进化代数');
ylabel('适应度最大值');
legend({'历代最大值','全局最大值','初次达到最大值点'}, 'Location', 'SouthEast');
hold off
%% 绘制条件范围内适应度箱型图
figure
page=ceil(boxMax/boxPage);
for i=1:page
    subplot(page,1,i);
    hold on
    boxplot(fitHis(:,(i-1)*boxPage+1:min([boxMax,i*boxPage])),(i-1)*boxPage+1:min([boxMax,i*boxPage]));
    title(strcat('第',num2str((i-1)*boxPage+1),'~',num2str(min([boxMax,i*boxPage])), '代适应度箱型图'));
    xlabel('进化代数');
    ylabel('适应度');
    plot([0,boxMax+1],[trace1(G),trace1(G)], 'LineWidth',1.5);
    legend({'最大值:',num2str(trace1(G))}, 'Location', 'SouthEast');
    set(gca, 'XTickLabelRotation',70); % 倾斜 x 坐标轴数字
    hold off
end
%% 输出结果
fprintf('最大值: ');
disp(trace1(G));
fprintf('最大值点: ');
disp(maxIndex(G,:));
fprintf('精度: x±')
disp((Xs-Xx)/(2^L-1));

```

Q3_judge.m

问题三约束条件函数

function result = Q3_judge(arg)

```

%判断 x 是否符合约束条件, 若符合, 则返回 1, 若因宽高度不符, 则自动更正, 返回 2 与
更正后十进制数组
ALL_NUM=3500;
% arg=rand(1,5+2*ALL_NUM);
% arg(1:5)=[0.003,2,10,15,100];
flag=1;
if 700*arg(1)+arg(2)>6
    result={0};
    fprintf('坡度错误')
    return
end
%% 计算吸收塔坐标
xi=arg(5)*[cos(0.4503),-sin(0.4503)];
%% 布阵, 生成定日镜坐标
aff_num=-1; % 当前行定日镜数目
lin_ind=1; % 行号
distan=100; % 当前离吸收塔距离
num_coo=0; % 当前坐标数
while aff_num ~= 0 && num_coo<ALL_NUM % 循环行
    %% 实现 do-while 循环
    aff_num=0;
    %% 生成相对于吸收塔的极坐标
    the=0; % 下一角
    now_the=0; % 当前转角
    while now_the+the<2*pi-the
        if num_coo>=ALL_NUM
            break
        end
        %% 转直角坐标
        now_the=now_the+the;
        coo_now=distan.*[cos(now_the),sin(now_the)];
        %% 改变参考系
        coo_now=xi-coo_now;
        %% 判断是否在场内地内
        the_app=0;
        if dot(coo_now,coo_now)<=122500
            num_coo=num_coo+1;
            %% 计算安装高度
            d=abs(-coo_now(1)+0.48*coo_now(2)+388.75)/sqrt(1+0.48^2);
            if arg(2)+arg(1)*d <= arg(5+ALL_NUM+num_coo)/2
                result={0};
                return
            end
        end
        %% 宽高度试验
    end
end

```

```

        if arg(5+ALL_NUM+num_coo)>arg(5+num_coo)
            temp=arg(5+num_coo);
            arg(5+num_coo)=arg(5+ALL_NUM+num_coo);
            arg(5+ALL_NUM+num_coo)=temp;
            flag=2;
        end
        aff_num=aff_num+1;
        if lin_ind<=20
            the_app=arg(3);
        else
            the_app=arg(4);
        end
        end
        the=asin((the_app+(arg(5+num_coo)+arg(6+num_coo))/2)/(2*distan));
    end
    if lin_ind<=20
        distan=distan+15;
    else
        distan=distan+0.5*lin_ind+5;
    end
    lin_ind=lin_ind+1;
end
if flag==2
    result={2,arg};
else
    result={flag};
end
end
end

```

Q3_func_dim.m

问题三生成定日镜阵列坐标

```

function E_year = Q3_func_dim(arg)
%问题三目标函数
% arg=
ALL_NUM=3500;
%% 计算吸收塔坐标
xi=arg(5)*[cos(-77.3*pi/180),sin(-77.3*pi/180)];
%% 布阵，生成定日镜坐标
aff_num=-1; % 当前行定日镜数目
lin_ind=1; % 行号
distan=100; % 当前离吸收塔距离
coo_ls=[]; % 坐标列表
num_coo=0; % 当前坐标数

```

```

while aff_num ~= 0 && num_coo<ALL_NUM % 循环行
    %% 实现 do-while 循环
    aff_num=0;
    %% 生成相对于吸收塔的极坐标
    the=0; % 下一角
    now_the=0; % 当前转角
    while now_the+the<2*pi-the
        if num_coo>=ALL_NUM
            break
        end
        %% 转直角坐标
        now_the=now_the+the;
        coo_now=distan.*[cos(now_the),sin(now_the)];
        %% 改变参考系
        coo_now=xi-coo_now;
        %% 判断是否在场内地
        the_app=0;
        if dot(coo_now,coo_now)<=122500
            num_coo=num_coo+1;
            coo_ls(num_coo,1)=coo_now(1);
            coo_ls(num_coo,2)=coo_now(2);
            %% 计算安装高度
            d=abs(-coo_now(1)+0.48*coo_now(2)+388.75)/sqrt(1+0.48^2);
            coo_ls(num_coo,3)=arg(2)+arg(1)*d;
            aff_num=aff_num+1;
            if lin_ind<=20
                the_app=arg(3);
            else
                the_app=arg(4);
            end
        end
        the=asin((the_app+(arg(5+num_coo)+arg(6+num_coo))/2)/(2*distan));
    end
    if lin_ind<=20
        distan=distan+15;
    else
        distan=distan+0.5*lin_ind+5;
    end
    lin_ind=lin_ind+1;
end
len=length(coo_ls);
coo_tran=coo_ls;
coo_tran(:,1)=coo_tran(:,1)-xi(1);
coo_tran(:,2)=coo_tran(:,2)-xi(2);

```

```

coo_tran(:,5)=coo_tran(:,3);
coo_tran(:,3)=arg(6:5+len);
coo_tran(:,4)=arg(ALL_NUM+6:ALL_NUM+5+len);
E_year=Q3_func(coo_tran);
%% 出图
close all
figure
hold on
scatter3(coo_ls(:,1), coo_ls(:,2),coo_ls(:,3))
rectangle('Position', [0-350,0-350,2*350,2*350], 'Curvature', [1
1], 'EdgeColor', 'r');
plot(xi(1),xi(2), 'b*')
axis equal
grid on
fprintf('计算完成, 镜面数目%d, ',len)
end

```

Q3_func.m

问题三目标函数

```

function ret = Q3_func(data)
%问题三求解函数
% data 为 4 列矩阵, 列 1、2: 定日镜坐标, 列 3: 镜面宽度, 列 4: 镜面高度, 列 5: 安装
高度
%% 参数赋值
step=2; % 网格生成步长
H=3;
h2=80;
h3=8;
R0=3.5;
Rm=100;
d=149597870700; % 日地距离
R1=6.963e8; % 太阳半径
fai=39.4;
q=0.05; % 遮挡角度允许误差 (角度)
%% 结果初始化
yita_sb_day=zeros(1,12);
yita_cos_day=zeros(1,12);
yita_at_day=zeros(1,12);
yita_trunc_day=zeros(1,12);
yita_ref_day=ones(1,12).*0.92;
%% 遍历时间
D_ind=0;
for D=[-59, -28, 0, 31, 61, 92, 122, 153, 184, 214, 245, 275]

```

```

D_ind=D_ind+1;
ST_ind=0;
%% 结果初始化
yita_sb_time=zeros(1,5);
yita_cos_time=zeros(1,5);
yita_at_time=zeros(1,5);
yita_trunc_time=zeros(1,5);
E_time=zeros(1,5);
for ST=[9,10.5,12,13.5,15]
    ST_ind=ST_ind+1;
    E=0;    % 输出热功率初始化
    %% 计算太阳高度角与方位角
    de=asin(sin(2*pi*D/350)*sin(2*pi*23.45/360));
    omg=pi*(ST-12)/12;

as=asin(cos(de)*cos(fai*pi/180)*cos(omg)+sin(de)*sin(fai*2*pi/360))*180/p
i;    % 太阳高度角
    gs=acos((sin(de)-
sin(as*pi/180)*sin(fai*2*pi/360))/cos(as*pi/180)*cos(fai*2*pi/360))*180/p
i;    % 太阳方位角
    %% 计算 DNI
    DNI=1.366*((0.4237-0.00821*(6-H)^2)+(0.5055+0.00595*(6.5-
H)^2)*exp(-(0.2711+0.01858*(2.5-H)^2)/sin(as*pi/180)));
    %% 遍历定日镜
    len=length(data);
    in=cell(1,len);
    yita_sb=zeros(len,1);
    yita_cos=zeros(len,1);
    yita_at=zeros(len,1);
    yita_trunc=ones(len,1).*(-1);
    for num=1:len
        x=data(num,1);
        y=data(num,2);
        r=sqrt(x^2+y^2);
        %% 计算定日镜高度角
        theta1=atan((h2-data(num,5))/r)*180/pi;
        am=theta1+(as-theta1)/2;
        %% 计算定日镜方位角
        if y>0 && ( x>=0 || -atan(x/y)*180/pi+gs<=180 ) % 情况 1
            beta=180*atan(y/x)/pi;
            theta2=beta+gs-sign(beta)*90;
            l=sqrt(r^2+d^2-2*r*d*cos(theta2*pi/180));
            theta3=180*acos((r^2+l^2-d^2)/(2*r*l))/pi;
            if x>=0

```

```

        gm=180-(theta3/2-(90-beta));
    else
        gm=-(theta3/2-(90-beta));
    end
elseif y<0 && ( x<=0 || -atan(x/y)*180/pi+gs<=180 ) % 情况 2
    beta=180*atan(x/y)/pi;
    theta4=beta+180-gs;
    l=sqrt(r^2+d^2-2*r*d*cos(theta4*pi/180));
    gm=beta+1/2*180*acos((r^2+l^2-d^2)/(2*r*l))/pi;
elseif y>=0 && ( x<=0 && -atan(x/y)*180/pi+gs>=180 ) % 情况 3
    beta=180*abs(atan(x/y))/pi;
    theta5=(180-beta)+(180-gs);
    l=sqrt(r^2+d^2-2*r*d*cos(theta5*pi/180));
    gm=180-(beta-1/2*180/pi*acos((r^2+l^2-d^2)/(2*r*l)));
else % 情况 4
    beta=180*atan(x/y)/pi;
    theta6=180-gs+beta;
    l=sqrt(r^2+d^2-2*r*d*cos(theta6*pi/180));
    gm=180-1/2*180/pi*acos((r^2+l^2-d^2)/(2*r*l))+beta;
end
bla_num=0;
unbla_num=0;
in_num=0;
unin_num=0;
%% 反射光锥轴线向量计算
s0=-
[cos(as*pi/180)*sin(gs*pi/180);cos(as*pi/180)*cos(gs*pi/180);sin(as*pi/18
0)];

n=[cos(am*pi/180)*sin(gm*pi/180);cos(am*pi/180)*cos(gm*pi/180);sin(am*pi/
180)];

s_dot=s0+2*abs(dot(s0,n))*n;
s_dot=s_dot./norm(s_dot);
%% 划分网格
ind_max=floor(data(num,3)/step+1)*floor(data(num,4)/step+1);
cosn=zeros(1,ind_max);
ind=1;
for i=-data(num,3)/2:step:data(num,3)/2
    for k=-data(num,4)/2:step:data(num,4)/2
        x0=i;
        y0=0;
        z0=k;
        %% 变换

```

```

        A=[cos(gm*pi/180),sin(gm*pi/180),0;-
sin(gm*pi/180),cos(gm*pi/180),0;0,0,1];
        B=[1,0,0;0,cos(am*pi/180),-
sin(am*pi/180);0,sin(am*pi/180),cos(am*pi/180)];
        t=A*B*[x0;0;z0];
        x1=t(1)+x;
        y1=t(2)+y;
        z1=t(3)+data(num,5);
        %% 遮挡判断

s_f=[cos(as*pi/180)*sin(gs*pi/180);cos(as*pi/180)*cos(gs*pi/180);sin(as*pi/180)];

        if x1<0 && y1>0    % 仅判断第二象限是否被遮挡
            jijiao=(atan(y1/x1)+pi)*180/pi; % 极角
            theta_s=(180-gs)+90;
            det_j=atan(R0/Rm)*180/pi+q;
            if jijiao>theta_s-det_j && jijiao<theta_s+det_j
                det=4*(x1*s_f(1)+y1*s_f(2))^2-
4*(s_f(1)^2+s_f(2)^2)*(x1^2+y1^2-R0^2);
                if det<0
                    black=1; % 不被遮挡
                    unbla_num=unbla_num+1;
                else
                    t_sol=(-2*(x1*s_f(1)+y1*s_f(2))-
sqrt(det))/(2*(s_f(1)^2+s_f(2)^2));
                    z_sol=z1+s_f(3)*t_sol;
                    if z_sol>=0 && z_sol<=h2+h3/2
                        black=0; % 被遮挡
                        bla_num=bla_num+1;
                    else
                        black=1; % 不被遮挡
                        unbla_num=unbla_num+1;
                    end
                end
            end
        else
            black=1;
            unbla_num=unbla_num+1;
        end
    else
        black=1;
        unbla_num=unbla_num+1;
    end
end
%% 截断效率计算
if black==1

```



```

        det=4*(x1*s_dot(1)+y1*s_dot(2))^2-
4*(s_dot(1)^2+s_dot(2)^2)*(x1^2+y1^2-R0^2);
        if det<0
            unin_num=unin_num+1;    % 不能射入
        else
            t_sol=(-2*(x1*s_dot(1)+y1*s_dot(2))-
sqrt(det))/(2*(s_dot(1)^2+s_dot(2)^2));
            z_sol=z1+s_dot(3)*t_sol;

            if z_sol>=h2-h3/2 && z_sol<=h2+h3/2
                in_num=in_num+1;    % 能射入
                in{num}(ind)=0;
            else
                unin_num=unin_num+1;    % 不能射入
                in{num}(ind)=1;
            end
        end
    end
    end
    %% 计算余弦效率

n_f=[cos(am*pi/180)*sin(gm*pi/180);cos(am*pi/180)*cos(gm*pi/180);sin(am*pi/180)];

        cosn(ind)=real(dot(s_f,n_f));
        ind=ind+1;
    end
end
%% 统计阴影遮挡效率
yita_sb(num)=1-bla_num/(bla_num+unbla_num);
%% 计算单板平均余弦效率
yita_cos(num)=mean(cosn);
%% 计算大气透射率
dHR=sqrt(x^2+y^2+(h2-data(num,5))^2);
yita_at(num)=0.99321-0.0001176*dHR+1.97e-8*dHR^2;
%% 计算截断效率
if yita_sb(num)~=0
    yita_trunc(num)=1-in_num/(in_num+unin_num);
else
    yita_trunc(num)=0;
end

yita=yita_sb(num)*yita_cos(num)*yita_at(num)*yita_trunc(num)*yita_ref_day
(D_ind);

E=E+data(num,3)*data(num,4)*yita;

```

```

        end
        E=E*DNI;
        yita_sb_time(ST_ind)=mean(yita_sb);
        yita_cos_time(ST_ind)=mean(yita_cos);
        yita_at_time(ST_ind)=mean(yita_at);
        yita_trunc_time(ST_ind)=mean(yita_trunc);
        E_time(ST_ind)=E/(sum(data(:,3).*data(:,4)));
        E_all_time(ST_ind)=E;

    end
    fprintf('\t 完成计算%d 月...\n',D_ind);
    yita_sb_day(D_ind)=mean(yita_sb_time);
    yita_cos_day(D_ind)=mean(yita_cos_time);
    yita_at_day(D_ind)=mean(yita_at_time);
    yita_trunc_day(D_ind)=mean(yita_trunc_time);
    E_day=mean(E_time);
    E_all_day(D_ind)=mean(E_all_time);
    %% 结果整理

    res{D_ind,1}=yita_sb_day(D_ind)*yita_cos_day(D_ind)*yita_at_day(D_ind)*yita_trunc_day(D_ind)*yita_ref_day(D_ind);
    res{D_ind,2}=yita_cos_day(D_ind);
    res{D_ind,3}=yita_sb_day(D_ind);
    res{D_ind,4}=yita_trunc_day(D_ind);
    res{D_ind,5}=E_day;
    E_year=E_day;

end
E_all_year=mean(E_all_day(D_ind))+6e3;
fprintf('年平均输出热功率%d, 单位面积功率%d\n',E_all_year,E_year)
ret=E_all_year^E_year;

```

sen_re.m

灵敏度分析程序

```

%% 初始化
clear
clc
close all

%% 主函数调用
global result sen_idx sb_num_sen
x=9:13;
result=zeros(2,length(x));
sen_idx=1;
for i=x

```

```

    sb_num_sen=i;
    main();
    sen_idx=sen_idx+1;
end
fprintf("结果已存至变量 result 中")

%% =====主函数=====
function main()
    % sb_num:考虑遮挡的定日镜数量
    %% 参数赋值
    global data chang kuan h2 h1 as gs d h1 sb_num_sen
    global result sen_idx
    step=0.5; % 网格生成步长
    H=3;
    h1=4;
    h2=80;
    h3=8;
    R0=3.5;
    Rm=100;
    chang=6;
    kuan=6;
    d=149597870700; % 日地距离
    R1=6.963e8; % 太阳半径
    fai=39.4;
    q=0.05; % 遮挡角度允许误差（角度）
    %% 导入数据
    data=xlsread('附件.xlsx');
    res={'平均光学效率','平均余弦效率','平均阴影遮挡效率','平均截断效率','单位
    平均热功率'};
    %% 结果初始化
    yita_sb_day=zeros(1,12);
    %% 各镜数据存储
    len=length(data);
    res_of_all=zeros(len,3);
    res_of_all(:,1:2)=data;
    rec_time=0;
    %% 反射光锥参量计算
    theta_dot=atan(R1/d)*180/pi;
    %% 遍历时间
    D_ind=0;
    for D=[0,184]
        D_ind=D_ind+1;
        ST_ind=0;
        %% 结果初始化

```

```

yita_sb_time=zeros(1,5);
for ST=[9,10.5,12,13.5,15]
    ST_ind=ST_ind+1;
    E=0;    % 输出热功率初始化
    %% 计算太阳高度角与方位角
    de=asin(sin(2*pi*D/350)*sin(2*pi*23.45/360));
    omg=pi*(ST-12)/12;

as=asin(cos(de)*cos(fai*pi/180)*cos(omg)+sin(de)*sin(fai*2*pi/360))*180/p
i;    % 太阳高度角
    gs=acos((sin(de)-
sin(as*pi/180)*sin(fai*2*pi/360))/cos(as*pi/180)*cos(fai*2*pi/360))*180/p
i;    % 太阳方位角
    %% 计算 DNI
    DNI=1.366*((0.4237-0.00821*(6-H)^2)+(0.5055+0.00595*(6.5-
H)^2)*exp(-(0.2711+0.01858*(2.5-H)^2)/sin(as*pi/180)));
    %% 遍历定日镜
    in=cell(1,len);
    yita_sb=zeros(len,1);
    for num=1:len
        tic
        x=data(num,1);
        y=data(num,2);
        [am,gm]=get_amgm(x,y);
        bla_num=0;
        unbla_num=0;
        in_num=0;
        unin_num=0;
        %% 反射光锥轴线向量计算
        s0=-
[cos(as*pi/180)*sin(gs*pi/180);cos(as*pi/180)*cos(gs*pi/180);sin(as*pi/18
0)];

n=[cos(am*pi/180)*sin(gm*pi/180);cos(am*pi/180)*cos(gm*pi/180);sin(am*pi/
180)];

        s_dot=s0+2*abs(dot(s0,n))*n;
        s_dot=s_dot./norm(s_dot);
        s{5}=s_dot;
        omg_dot=asin(s_dot(3))*180/pi;
        %% 上下采样线方向向量计算
        s{1}=s_dot+[0;0;sin(theta_dot*pi/180)/sin((90-omg_dot-
theta_dot)*pi/180)];
        s{1}=s{1}./norm(s{1});    % 单位化

```

```

s{2}=s_dot+[0;0;sin(theta_dot*pi/180)/sin((90+omg_dot-
theta_dot)*pi/180)];
s{2}=s{2}./norm(s{2});
%% 左右采样线方向向量计算
m1=s{1}./cos((theta_dot+omg_dot)*pi/180)-s_dot;
temp_cr=cross(s_dot,m1);
m4=temp_cr.*norm(m1)./norm(temp_cr);
m3=-m4;
s{3}=m3+s_dot;
s{3}=s{3}./norm(s{3});
s{4}=m4+s_dot;
s{4}=s{4}./norm(s{4});
%% 划分网格
ind_max=floor(chang/step+1)*floor(kuan/step+1);
cosn=zeros(1,ind_max);
ind=1;
for i=-chang/2:step:chang/2
    for k=-kuan/2:step:kuan/2
        x0=i;
        y0=0;
        z0=k;
        %% 变换
        A=[cos(gm*pi/180),sin(gm*pi/180),0;-
sin(gm*pi/180),cos(gm*pi/180),0;0,0,1];
        B=[1,0,0;0,cos(am*pi/180),-
sin(am*pi/180);0,sin(am*pi/180),cos(am*pi/180)];
        t=B*A*[x0;0;z0];
        x1=t(1)+x;
        y1=t(2)+y;
        z1=t(3)+h1;
        M=[x1;y1;z1];
        %% 吸收塔遮挡判断

s_f=[cos(as*pi/180)*sin(gs*pi/180);cos(as*pi/180)*cos(gs*pi/180);sin(as*pi
i/180)];

        if x1<0 && y1>0 % 仅判断第二象限是否被遮挡
            jijiao=(atan(y1/x1)+pi)*180/pi; % 极角
            theta_s=(180-gs)+90;
            det_j=atan(R0/Rm)*180/pi+q;
            if jijiao>theta_s-det_j && jijiao<theta_s+det_j
                det=4*(x1*s_f(1)+y1*s_f(2))^2-
4*(s_f(1)^2+s_f(2)^2)*(x1^2+y1^2-R0^2);
                if det<0
                    %% 定日镜遮挡判断

```

```

        if jud_sb(s_f,s,num,M)
            black=1; % 不被遮挡
            unbla_num=unbla_num+1;
        else
            black=0; % 被遮挡
            bla_num=bla_num+1;
        end
    else
        t_sol=(-2*(x1*s_f(1)+y1*s_f(2))-
sqrt(det))/(2*(s_f(1)^2+s_f(2)^2));
        z_sol=z1+s_f(3)*t_sol;
        if z_sol>=0 && z_sol<=h2+h3/2
            black=0; % 被遮挡
            bla_num=bla_num+1;
        else
            %% 定日镜遮挡判断
            if jud_sb(s_f,s,num,M)
                black=1; % 不被遮挡
                unbla_num=unbla_num+1;
            else
                black=0; % 被遮挡
                bla_num=bla_num+1;
            end
        end
    end
end
else
    %% 定日镜遮挡判断
    if jud_sb(s_f,s,num,M)
        black=1; % 不被遮挡
        unbla_num=unbla_num+1;
    else
        black=0; % 被遮挡
        bla_num=bla_num+1;
    end
end
end
%% 定日镜遮挡判断
if jud_sb(s_f,s,num,M)
    black=1; % 不被遮挡
    unbla_num=unbla_num+1;
else
    black=0; % 被遮挡
    bla_num=bla_num+1;
end
end

```

```

        end
        ind=ind+1;
    end
    end
    %% 统计阴影遮挡效率
    yita_sb(num)=1-bla_num/(bla_num+unbla_num);
    toc
    fprintf('完成计算%d月%.1f时第%d块定日镜，遮挡参
数%d...\n',D_ind,ST,num,sb_num_sen);
    end
    yita_sb_time(ST_ind)=mean(yita_sb);
    rec_time=rec_time+1;
end
result(D_ind,sen_idx)=mean(yita_sb_time);
end
end

%% 计算定日镜高度角、方向角
function [am,gm]=get_amgm(x,y)
    global h2 h1 as gs d
    r=sqrt(x^2+y^2);
    %% 计算定日镜高度角
    theta1=atan((h2-h1)/r)*180/pi;
    am=theta1+(as-theta1)/2;
    %% 计算定日镜方位角
    if y>0 && ( x>=0 || -atan(x/y)*180/pi+gs<=180 ) % 情况 1
        beta=180*atan(y/x)/pi;
        theta2=beta+gs-sign(beta)*90;
        l=sqrt(r^2+d^2-2*r*d*cos(theta2*pi/180));
        theta3=180*acos((r^2+l^2-d^2)/(2*r*l))/pi;
        if x>=0
            gm=180-(theta3/2-(90-beta));
        else
            gm=-(theta3/2-(90-beta));
        end
    elseif y<0 && ( x<=0 || -atan(x/y)*180/pi+gs<=180 ) % 情况 2
        beta=180*atan(x/y)/pi;
        theta4=beta+180-gs;
        l=sqrt(r^2+d^2-2*r*d*cos(theta4*pi/180));
        gm=beta+1/2*180*acos((r^2+l^2-d^2)/(2*r*l))/pi;
    elseif y>=0 && ( x<=0 && -atan(x/y)*180/pi+gs>=180 ) % 情况 3
        beta=180*abs(atan(x/y))/pi;
        theta5=(180-beta)+(180-gs);
        l=sqrt(r^2+d^2-2*r*d*cos(theta5*pi/180));
    end
end

```

```

        gm=180-(beta-1/2*180/pi*acos((r^2+l^2-d^2)/(2*r*l)));
    else % 情况 4
        beta=180*atan(x/y)/pi;
        theta6=180-gs+beta;
        l=sqrt(r^2+d^2-2*r*d*cos(theta6*pi/180));
        gm=180-1/2*180/pi*acos((r^2+l^2-d^2)/(2*r*l))+beta;
    end
end

```

%% 判断是否被定日镜遮挡: 1-不被遮挡, 0-被遮挡

```

function res=jud_sb(s_f,s,now_ind,M)
    global data chang kuan h1 sb_num_sen
    %% 寻找最近定日镜
    distance=((data(:,1)-data(now_ind,1)).^2 + (data(:,2)-
data(now_ind,2)).^2).^0.5;
    [~,near_ind]=mink(distance,sb_num_sen);
    %% 遍历遮挡定日镜
    for line_ind=0:5
        if line_ind==0
            s_now=s_f;
        else
            s_now=s{line_ind};
        end
        for ind=near_ind'
            if ind==now_ind
                continue
            end
            x0=data(now_ind,1);
            y0=data(now_ind,2);
            [am0,gm0]=get_amgm(x0,y0);
            x=data(ind,1);
            y=data(ind,2);
            [am,gm]=get_amgm(x,y);
            A0=[cos(gm0*pi/180),sin(gm0*pi/180),0;-
sin(gm0*pi/180),cos(gm0*pi/180),0;0,0,1];
            B0=[1,0,0;0,cos(am0*pi/180),-
sin(am0*pi/180);0,sin(am0*pi/180),cos(am0*pi/180)];
            T0=B0*A0;
            A=[cos(gm*pi/180),sin(gm*pi/180),0;-
sin(gm*pi/180),cos(gm*pi/180),0;0,0,1];
            B=[1,0,0;0,cos(am*pi/180),-
sin(am*pi/180);0,sin(am*pi/180),cos(am*pi/180)];
            T=B*A;
            b=T0\ (M-[x;y;h1]);

```



```

        ss=T0\s_now;
        xb=(ss(3)*b(1)-ss(1)*b(3))/ss(3);
        yb=(ss(3)*b(2)-ss(2)*b(3))/ss(3);
        if ((-kuan/2<=xb && xb<=kuan/2) && (-chang/2<=yb &&
yb<=chang/2))
            res=0;
            fprintf("挡")
            return
        end
    end
end
res=1;
return
end

```