

定日镜场的优化设计

摘要

本文在求解定日镜场的优化设计中，通过讨论阴影遮挡损失和截断效率，求得定日镜场的光学效率和输出热功率模型；根据题目所给定日镜场布局的影响因素，利用粒子群算法，确定最佳的定日镜场参数配置；通过定日镜场优化设计，达到指定的输出热功率。

针对问题一：基于反射定律求得定日镜法向量和余弦效率；通过点法式得到入射和反射方程，平面点法式方程得到定日镜平面方程，利用入射方程和定日镜方程联立求得交点，利用矩形近似求得遮挡损失；通过蒙特卡洛法生成 2000 条反射光线，利用反射直线方程和集热器的截面方程联立求得与反射光线的交点数 n ，得到每面定日镜截断效率 $n/2000$ ；将以上参数相乘得到光学效率为 0.72，年平均输出热功率为 49MW，单位面积镜面年平均输出热功率为 0.89kW/m^2 。

针对问题二：在第一问的定日镜光学效率和输出热功率模型基础上，建立了以输出热功率不小于 60MW 的多目标规划模型，利用粒子群算法，根据问题的具体需求和限制，定义了包含定日镜位置、定日镜尺寸和定日镜高度的一些约束条件和适应度函数，通过不断迭代更新粒子的速度和位置，在满足约束条件的前提下，寻找最优解，计算结果见表 3，其中输出热功率为 。

针对问题三：在第二问的基础上，将放射栅格模型的每一圈定日镜做同一化处理，定日镜的高度和尺寸随放射栅格模型的圈数增大依次增加，以此修改第二问的约束条件，仍利用粒子群算法得到最优解，其中输出热功率为 。

关键词：阴影遮挡损失 截断效率 蒙特卡洛法 粒子群算法 放射栅格模型

一、问题重述

1.1 问题背景

随着能源消耗的增加和环境问题的日益严重，人们对可再生能源的开发和利用越来越重视。为响应国家“碳达峰”“碳中和”目标，我们引入一种低碳环保的新型清洁能源技术——塔式太阳能光热发电，其收集太阳能的基本组件定日镜场是一种利用镜面反射原理来聚焦太阳光的设备。通过进一步研究和应用定日镜场技术，可以实现可持续能源的利用，为人类创造更加清洁、可持续的未来。

张平^[1]等人采用几何光学算法分析阐述镜场性能的计算理论模型，其中详细介绍了镜场阴影挡光效率和截断效率的计算模型，为模型建立提供了帮助。杜宇航^[2]等人根据电站的主要仿真参数，构建了光热电站的定日镜场模型和圆柱形外露式吸热器模型，采取光线追迹法，计算了不同条件下吸热器表面能流密度的分布情况。谢飞^[3]在文章中详细介绍了塔式太阳能热点系统定日静场光学仿真与应用研究，并分别研究了基于平行光与非平行光的模型的镜场效率模型。刘建新^[4]在研究塔式光热电站光学效率的基础上，进一步研究了定日静场优化布置，为本文研究提供了参考和帮助。

1.2 问题提出

定日镜是塔式电站收集太阳能的基本组件，由纵向转轴和水平转轴控制定日镜的方向。其中纵向转轴控制反射镜的方位角，水平转轴控制反射镜的俯仰角。定日镜中心为两转轴的焦点，其离地高度为定日镜的安装高度。定日镜场由大量定日镜组成阵列构成，太阳光呈锥形光线经过定日镜反射汇聚至集热器中，并将太阳能存储实现热能向电能的转化。

现位于东经 98.5° ，北纬 39.4° ，海拔 3000 m，半径 350 m 的圆形区域内规划建设圆形定日镜场，建立经常坐标系，规定圆形区域中心为原点，x 轴为正东方向，y 轴为正北方向，z 轴垂直于地面向上方向。吸收塔是圆柱形外表受光式集热器，且周围 100m 范围内不安装定日镜。定日镜为平面矩形，且镜面宽度不小于镜面高度。镜面边长在 2m 至 8m 之间，安装高度在 2m 至 6m 之间，安装时必须确保镜面在水平转轴旋转时不会接触地面。且要求相邻镜面底座中心之间的距离比镜面宽度多 5m 以上。

基于以上背景信息与附件内容，需建立数学模型解决以下问题。

问题一：吸收塔建在圆形定日静场中心，定日镜为边长为 6m 的正方形，安装高度为 4m，所有定日镜中心的位置在附件中给出。求解该定日镜场的年平均光学效率、年平均输出热功率以及单位镜面面积年平均输出热功率。

问题二：规定所有定日镜尺寸与安装高度相同，额定功率为 60MW，要求在定日静场在达到额定功率的条件下，单位镜面面积年平均输出热功率尽量大，设计并确定吸收塔的位置坐标、定日镜尺寸、安装高度、定日镜数目、定日镜位置使该定日静场满足上述要求。

问题三：规定定日镜尺寸与安装高度可以不同，额定功率同问题二，重新设计定日静场的参数，使其在达到额定功率条件下，单位镜面面积年平均输出热功率尽量大。

二、问题分析

2.1 问题一分析

问题一中需要建立阴影挡光效率模型和截断效率模型，首先根据题目所给数据确定六十个时刻的太阳方位，通过确定定日镜位置与太阳方位的关系，求出入射光线与反射光线的方向向量，计算出余弦损失效率。建立阴影挡光效率模型，考虑挡光损失和反射损失计算结果。计算截断效率模型时，本文使用蒙特卡洛法在定日镜生成随机点，计算光线从定日镜反射到集热塔的数量，得到截断效率，最后根据对所求得分量进行计算得到结果。

2.2 问题二分析

问题二是基于定日镜坐标的连续型优化问题，规定所有定日镜的尺寸与高度相同，使得定日镜场的平均输出热功率最大，本题定日镜场采用放射状栅格形排列，相邻圈之间的定日镜间隔分布，根据题目所给条件列出该模型的目标函数与优化条件，通过粒子群算法得到问题结果。

2.3 问题三分析

问题三在问题二的基础上，定日镜的尺寸与高度不同，故考虑外圈定日镜高度高于内圈高度，同时内圈定日镜的尺寸较小，外圈定日镜尺寸逐渐增大。与问题二相似建立目标函数与优化条件，通过粒子群算法得到数据结果。

三、模型假设

为了降低问题的复杂度，本文在建立模型之前，对模型进行了必要的简化与假设，具体假设如下：

（1）太阳光为平行光束假设

假设入射太阳光为平行光束。太阳光线以锥形光射入，由于日地距离足够远，入射定日镜的各光束近似平行，模型中可以忽略锥形光线展角对模型结果的影响，将入射太阳光等效为平行光线计算。

（2）镜面中心假设

假设镜面中心与镜面支撑点重合，忽略定日镜厚度对镜面中心坐标影响。

（3）阴影遮挡假设

假设目标定日镜的阴影遮挡只受附近一定范围内其它定日镜影响，不考虑较远距离其它定日镜对目标定日镜的影响。

（4）集热器热量吸收假设

假设反射到集热器上的所有光线均被吸收。

四、符号说明

符号	说明	单位
α_s	太阳高度角	°
γ_s	太阳方位角	°
δ	太阳赤纬角	°
φ	当地纬度	°
ω	太阳时角	°
η	定日镜的光学效率	/
η_{sb}	阴影遮挡效率	/
η_{\cos}	余弦效率	/
η_{at}	大气透射率	/
η_{trunc}	集热器截断效率	/
η_{ref}	镜面反射率	/
d_{HR}	镜面中心到集热器中心的距离	m
W_i	第 i 面定日镜的镜面宽度	m
H_i	第 i 面定日镜的镜面高度	m
Z_i	第 i 面定日镜的中心高度	m
x_i	第 i 面定日镜的横坐标	m
y_i	第 i 面定日镜的纵坐标	m
E_{field}	定日镜场的输出热功率	kW/m ²
Q	定日镜的数目	个

五、模型建立与求解

5.1 问题一模型建立与求解

根据题意，画出集热器与定日镜的示意图，其中 α_s 为太阳高度角， τ_s 为太阳方位角，如图 1 所示

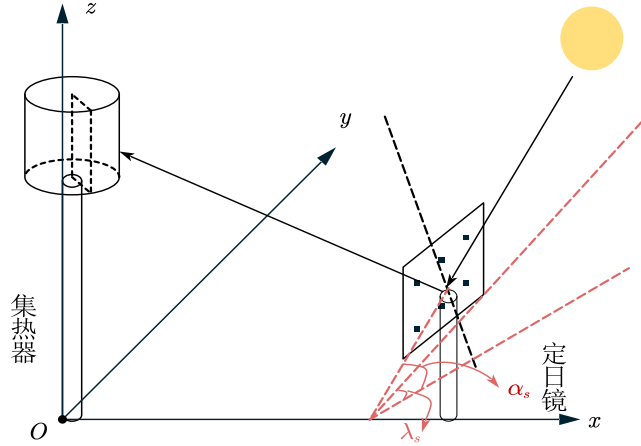


图 1 定日静场坐标及角度分析

根据太阳时角的定义

$$\omega = \frac{\pi}{12}(ST - 12) \quad (1)$$

其中 ST 为当地时间。根据题意 ST 取 9、10.5、12、13.5、15，单位为 h。

可求得 ω 依次为-0.785、-0.393、0、0.393、0.785。

太阳赤纬角 δ 是太阳光线与地球赤道面所夹的角，根据题意所给的计算公式为：

$$\sin \delta = \sin \frac{2\pi D}{365} \sin(\frac{2\pi}{365} 23.45) \quad (2)$$

其中 D 为以春分作为第 0 天起算的天数，如下表所示

D	306	337	0	31	61	92	122	153	184	214	245	275
---	-----	-----	---	----	----	----	-----	-----	-----	-----	-----	-----

已知当地纬度 φ 、太阳赤纬角 δ 和太阳时角 ω 后，分别代入式（3）和式（4）便可求得太阳高度角 α_s 和太阳方位角 γ_s 。

$$\sin \alpha_s = \cos \delta \cos \varphi \cos \omega + \sin \delta \sin \varphi \quad (3)$$

$$\cos \gamma_s = \frac{\sin \delta - \sin \alpha_s \sin \varphi}{\cos \alpha_s \cos \varphi} \quad (4)$$

在该模型中，由于太阳呈锥形光线发散入射，如图所示，当照射到定日镜上时，每一束光线之间的夹角都足够小，故为了简便计算，我们将太阳光线入射看作平行光入射，如图 2 所示。

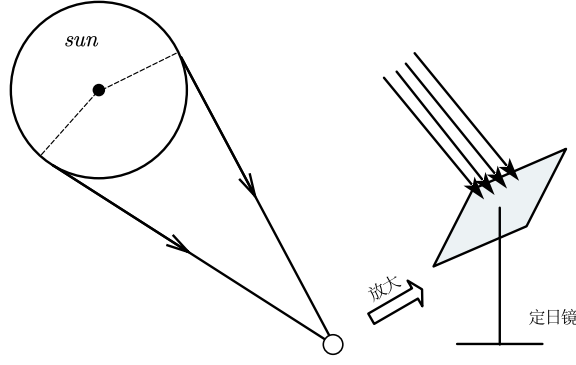


图 2 太阳光为平行光示意图

建立定日镜场坐标系（见图 1），规定集热塔底部中心为坐标原点，根据题目所给信息，集热器的坐标可表示为 (x_0, y_0, z_0) ，其中 $x_0 = 0$ ， $y_0 = 0$ ， $z_0 = 80$ ， z_0 为集热器中心相对于地面的高度。假设定日镜镜面中心在基座中轴线上，假设定日镜 1 的镜面中心坐标为 (x, y, z) ， x, y 坐标可由附件给出， z 为定日镜中心的高度为 4m。

由附件所给数据生成散点图，得到定日镜的分布图，如图 3 所示

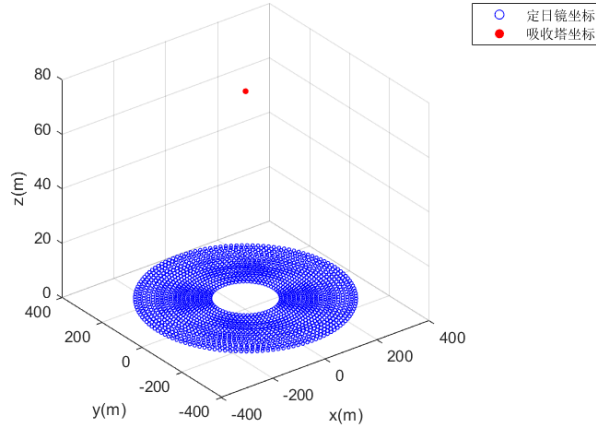


图 3 定日镜与吸收塔的分布示意图

设反射光线的坐标为 $(\Delta x, \Delta y, \Delta z)$

$$\begin{cases} \Delta x = x_0 - x \\ \Delta y = y_0 - y \\ \Delta z = z_0 - z \end{cases} \quad (5)$$

根据反射光线坐标可根据两点间的距离公式求得镜面中心到集热器中心的距离 d_{HR} ，计算公式如下

$$d_{HR} = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2} \quad (6)$$

接着求解反射光线得单位向量 $\vec{n}_2 = (X_2, Y_2, Z_2)$

$$X_2 = \frac{\Delta x}{d_{HR}}, Y_2 = \frac{\Delta y}{d_{HR}}, Z_2 = \frac{\Delta z}{d_{HR}} \quad (7)$$

根据式 (3) (4) 可求得每一天不同时刻的太阳高度角和太阳方位角，根据太阳高度角和太阳方位角计算太阳入射光线的单位向量，设单位向量的坐标为 $\vec{n}_1 = (X_1, Y_1, Z_1)$ ，

结合设定的坐标系得到具体公式为

$$\begin{cases} X_1 = \cos \alpha \sin \gamma \\ Y_1 = \cos \alpha \cos \gamma \\ Z_1 = \sin \alpha \end{cases} \quad (8)$$

得到每一块定日镜的入射光线和反射光线的单位向量后，接着计算法线的单位向量 $\vec{n} = (X, Y, Z)$

$$\vec{n} = \frac{\vec{n}_2 - \vec{n}_1}{|\vec{n}_2 - \vec{n}_1|} \quad (9)$$

5.1.1 年平均光学效率

定日镜的光学效率 η_i 为

$$\eta = \eta_{sb} \eta_{\cos} \eta_{at} \eta_{trunc} \eta_{ref} \quad (10)$$

其中 η_{sb} 为阴影遮挡效率， η_{\cos} 余弦效率， η_{at} 为大气透射率， η_{trunc} 为集热器截断效率， η_{ref} 为镜面反射率。下面对各个分量逐一求解。

(1) 余弦效率

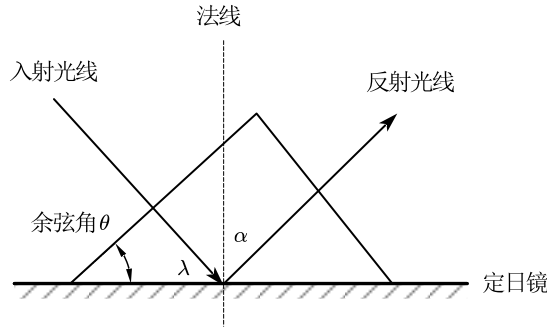


图 4 余弦效率中余弦角示意图

余弦效率定义为太阳入射角的余弦值，由直角三角形内角关系式，可知 θ 与 λ 互余， α 与 λ 互余，则可知 θ 等于 α 。

通过余弦定理计算得到余弦角的值，余弦角即为反射光线与法线的夹角。

$$\cos \theta = \frac{\vec{n} \cdot \vec{n}_2}{|\vec{n}| \cdot |\vec{n}_2|} \quad (11)$$

将每一个定日镜在不同时刻的反射光线坐标与法线坐标代入 (11) 式即得余弦效率结果。

(2) 大气透射率

大气透射率是指光线在穿过大气层时被吸收和散射的程度，由题意知大气透射率的计算公式为

$$\eta_{at} = 0.99321 - 0.000117d_{HR} + 1.97 \times 10^{-8} \times d_{HR}^2 \quad (d_{HR} \leq 1000) \quad (12)$$

将 (6) 结果代入 (12) 式，即可求得大气透射率的值。

(3) 阴影遮挡效率

在计算阴影遮挡效率中，我们考虑后排定日镜接收的太阳光被前方定日镜所阻挡和后排定日镜在反射太阳光时被前方定日镜遮挡两种情况，关键在于分析定日镜之间的相互影响，我们通过蒙特卡洛法更好地估计光线在定日镜场中的传播和反射路径，

得到定日镜之间的阴影损失和遮挡损失，求得结果。

根据附件所给信息已知定日镜中心的坐标，便可求得定日镜四个顶点的坐标，如图 5 所示。

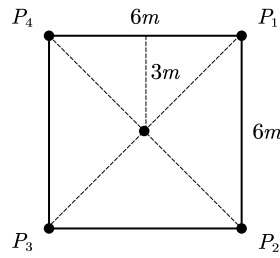


图 5 四点坐标示意

求得四点坐标后，首先通过式（8）的入射光线的单位向量与定日镜中任意一点的坐标组合成点向式方程

$$\frac{x - P_{ix}}{X_1} = \frac{y - P_{iy}}{Y_1} = \frac{z - P_{iz}}{Z_1} = t \quad (13)$$

得到

$$\begin{cases} x = tX_1 + P_{ix} \\ y = tY_1 + P_{iy} \\ z = tZ_1 + P_{iz} \end{cases} \quad (14)$$

其中 P_{ix} 表示其中一个顶点的横坐标， i 为 1, 4 中随机一个数。

根据四个点的坐标，通过点斜式计算出四条边的直线方程

$$\frac{x - P_{1x}}{P_{2x} - P_{1x}} = \frac{y - P_{1y}}{P_{2y} - P_{1y}} = \frac{z - P_{1z}}{P_{2z} - P_{1z}} \quad (15)$$

将（9）求得的法向量与定日镜中任意一点组合成点法式求定日镜平面方程

$$(x - P_{ix})X + (y - P_{iy})Y + (z - P_{iz})Z = 0 \quad (16)$$

联立入射光的直线方程（12）与定日镜平面方程（14），计算方程是否有解，若求得方程存在实数解，则证明该光线穿过定日镜，若方程无实数根，则证明光线与定日镜不相交。示意图如图 6 所示。

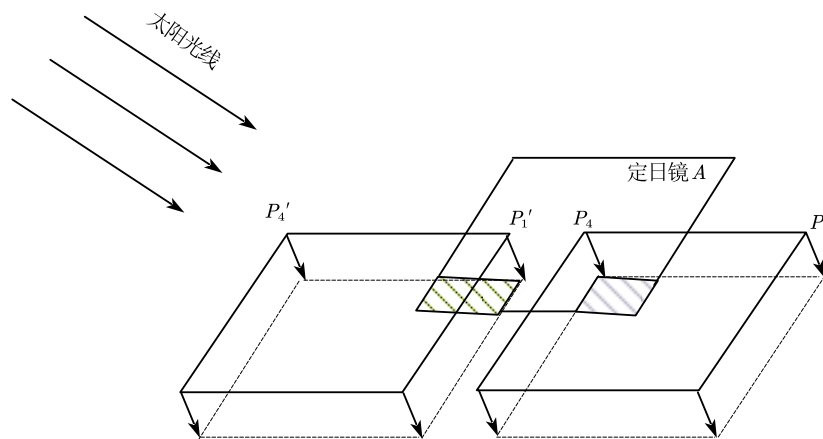


图 6 阴影损失镜面部分示意图

以一个定日镜为对象，搜索定日镜周围最近的6个定日镜，取离集热塔近的两面镜子为研究对象。为了简化模型，将阴影部分简化为矩形，根据太阳方位角的变化，对阴影部分讨论以下三种情况

- a) P_4 与 P_1' 其中一个在定日镜 A 上
- b) P_4 与 P_1' 都在定日镜 A 上
- c) P_4 与 P_1' 在定日镜 A 上， P_1 与 P_4' 其中一个在定日镜 A 上

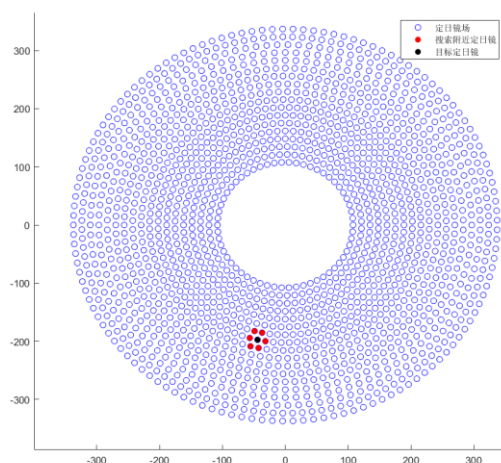
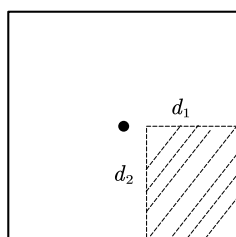


图 7 搜索定日镜周围最近的六个点示意图

三种情况中，根据投影在定日镜 A 上的点，得到点的坐标，根据点到直线的距离求出交点到定日镜 A 两条边的距离 d_1 和 d_2 ， $d_1 \times d_2$ 得到定日镜的阴影面积。



阴影遮挡效率即为

$$\text{阴影效率} = 1 - \frac{\text{阴影面积}}{\text{定日镜面积}}$$

(4) 集热器截断效率

在一面定日镜上，除去阴影部分，剩余部分通过蒙特卡洛法生成 2000 个随机点，阴影部分如图 8 所示，

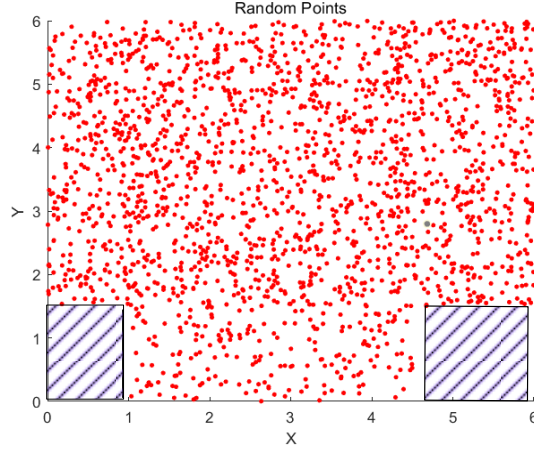


图 8 蒙特卡洛法生成随机点

当光线从定日镜反射到集热器上时，反射光线以平行光束射出，将集热器圆柱状等效为矩形，如图 9 所示

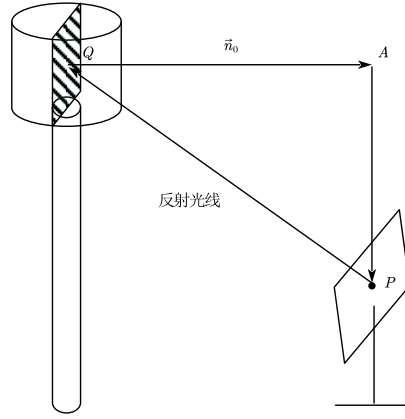


图 9 截断效率模型示意图

集热器中心坐标为 $Q(0,0,80)$ ，定日镜的镜面中心坐标为 $P(x,y,4)$ ， x 坐标和 y 坐标由附录给出， A 点坐标为 $(x,y,80)$ ，其中 \overrightarrow{QA} 为集热器法向量，得到 $\overrightarrow{QA} = (x,y,0)$ ，根据法向量 \vec{n}_0 和 Q 点坐标求得集热器平面方程。

根据蒙特卡洛方程生成得随机点坐标即可得到发射光线的坐标，同文章上述方法，联立集热器平面方程与反射光线方程，判断方程是否有解，若有解则该条发射光线被集热器吸收，若方程无解，则该条反射光线未被集热器吸收。

计算被集热器吸收的发射光线条数，即可得到集热器截断效率为

$$\eta_{\text{trunc}} = \frac{\text{被集热器吸收的光线总数}}{\text{总光线条数}}$$

(5) 镜面反射率

镜面反射率 η_{ref} 是指光线射入镜面后被反射的比例，取常数 0.92。即

$$\eta_{\text{ref}} = 0.92$$

5.1.2 年平均输出热功率

由题意可知，定日静场的输出热功率

$$E_{\text{field}} = \text{DNI} \cdot \sum_i^N A_i \eta_i \quad (17)$$

其中 DNI 为法向直接辐射辐照度；N 为定日镜总数（单位：面）； A_i 为第 i 面定日镜采光面积（单位： m^2 ）； η_i 为第 i 面镜子的光学效率。

法向直接辐射辐照度 DNI（单位 kW/m^2 ）是指地球上垂直于太阳光线的平面单位面积上、单位时间内接收到的太阳辐射能量，可按以下公式近似计算

$$\text{DNI} = G_0 \left[a + b \exp \left(-\frac{c}{\sin \alpha_s} \right) \right] \quad (18)$$

$$a = 0.4327 - 0.00821(6 - H)^2$$

$$b = 0.5055 + 0.00595(6.5 - H)^2$$

$$c = 0.2711 + 0.01858(2.5 - H)^2$$

其中 G_0 为太阳常数，其值去为 $1.366 \text{ kW}/\text{m}^2$ ，H 为海拔高度（单位：km）。

由（3）式求得的太阳高度角以及题目所给海拔高度 $H=3000\text{m}$ 代入得到 DNI 的值。则平均输出热功率为所有定日镜所有时间的平均，即

$$\text{平均输出热功率} = \frac{\sum_{i=1}^{1745 \times 60} E_{\text{field}}(i,1)}{1745 \times 60} \quad (19)$$

其中 $E_{\text{field}}(i,1)$ 为一个 i 行 1 列的矩阵。

5.1.3 单位镜面面积年平均输出热功率

单位镜面面积年平均输出热功率即为平均输出热功率除以所有定日镜的面积和，即

$$\text{单位镜面面积年平均输出热功率} = \frac{\text{平均输出热功率}}{6 \times 6 \times 1745}$$

5.1.4 求解过程及结果分析

如图 10 所示，该图为平均余弦效率随时间的变化曲线，可以观察到 6、7 月份的余弦效率最大。

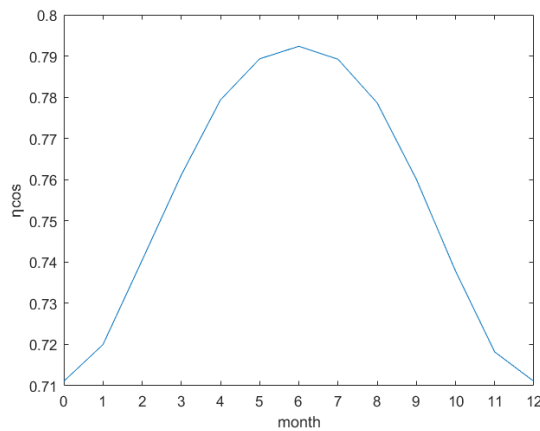


图 10 太阳时与平均余弦效率的关系

下表为每月 21 日日平均光学效率及输出功率。

表 1 问题一每月 21 日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	0.67	0.72	0.62	0.85	0.289
2 月 21 日	0.56	0.74	0.76	0.89	0.312
3 月 21 日	0.59	0.76	0.68	0.87	0.351
4 月 21 日	0.73	0.78	0.73	0.94	0.325
5 月 21 日	0.68	0.79	0.69	0.92	0.291
6 月 21 日	0.71	0.79	0.75	0.91	0.387
7 月 21 日	0.72	0.80	0.72	0.88	0.394
8 月 21 日	0.59	0.78	0.77	0.93	0.278
9 月 21 日	0.62	0.77	0.74	0.85	0.322
10 月 21 日	0.68	0.74	0.71	0.86	0.310
11 月 21 日	0.69	0.72	0.69	0.89	0.279
12 月 21 日	0.59	0.71	0.65	0.90	0.311

表 2 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)
0.72	0.76	0.74	0.83	49	0.89

5.2 问题二模型建立与求解

目前, 关于定日镜场的布置方法, 有传统的放射状栅格法, Pylkkanen 等人提出全年无遮挡的镜场布局方法, Crescent Dunes 镜场以圆形布局, Sun Tower 电站镜场以直线型布局, 本文采用传统的放射状栅格法布置定日静场。镜场布置实质上是多目标优化模型, 下面建立该题的优化模型。

5.2.1 目标函数的建立

根据题目所给条件, 需要确定日镜场的参数, 使得定日静场在达到额定功率的条件下, 单位镜面面积年平均输出热功率尽量大, 根据 (17) 式输出热功率的定义, 目标函数即为

$$\text{目标函数: } E_{\text{field}} = \text{DNI} \cdot \sum_i^N A_i \eta_i \geq 60 \quad (20)$$

5.2.2 约束条件的确定

根据题目所给条件, 设定日镜的尺寸为 $m \times n$, m 为定日镜长度, n 为定日镜宽度, 定日镜场布局以吸热塔为中心布置, 集热器的坐标为 $(0, 0, 80)$, 以一面定日镜为研究对象, 该定日镜中心坐标为 (x, y, z) , 根据题目要求, 吸收塔周围 100m 范围内不安装定日镜, 即为

$$\sqrt{(x-x_0)^2+(y-y_0)^2} \geq 100 \quad (21)$$

镜面宽度不小于镜面高度，且镜面边长在 2m 到 8m 之间，即为

$$2 < w \leq h < 8 \quad (22)$$

安装高度在 2m 至 6m 之间，即

$$2 < z < 6 \quad (23)$$

安装高度必须保证镜面在绕水平轴转轴旋转时不会触及地面，即

$$z \geq \frac{h}{2} \quad (24)$$

要求相邻定日镜底座中心之间的距离比镜面宽度多 5m 以上，即

$$\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2} > h+5 \quad (25)$$

设定日镜以环状排列， r_0 为集热塔到第一圈定日镜的距离，后排定日镜以 Δr 等间隔排列，由于定日镜场的半径为 350m，假设定日镜场由 n 圈定日镜组成，即

$$r_0 + n\Delta r \leq 350 \quad (26)$$

该排列方式为放射状栅格法，定日镜在同一圈内以 α 的角度间隔排列，设定日镜的数目为 Q ，

$$Q = \frac{360}{\alpha} n \quad (27)$$

定日镜场布置图如图 11 所示

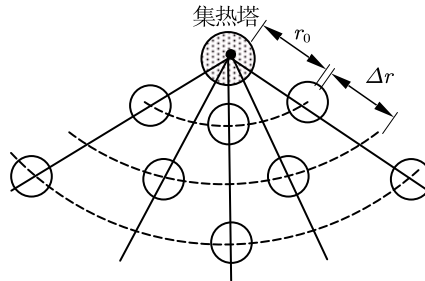


图 11 放射状栅格布置法示意图

5.2.3 模型求解

该定日镜场优化模型能够使用粒子群算法、遗传算法、模拟退火算法等求解，对于此问题，我们希望找到一个全局最优的解，而不是陷入局部最优解，使得粒子能够在搜索空间中不断探索，并且通过信息共享来引导粒子向更优的方向移动，同时该模型涉及到大量的变量和约束条件，故本文使用粒子群算法求解，提高全局搜索能力和计算效率。

下面是使用粒子群算法求解定日镜场的优化模型的具体步骤：

Step1: 初始化粒子群： 确定粒子数量和每个粒子的位置和速度。对于定日镜场的优化模型，可以将每个粒子的位置表示为一组参数，例如太阳光照的角度和强度等。位置和速度可以随机初始化，也可以根据问题的具体要求进行设置。

Step2: 计算适应度： 根据每个粒子的位置计算适应度值，即目标函数的值。在定日

静场的优化模型中，适应度值可以根据问题的具体要求进行定义，例如最大化太阳能的收集量或最小化阴影的覆盖面积。

Step3. 更新个体最优解：对于每个粒子，根据其当前位置和适应度值更新其个体最优解。如果当前适应度值优于个体最优解，则更新个体最优解。

Step4. 更新群体最优解：根据所有粒子的个体最优解，确定群体最优解。如果某个粒子的个体最优解优于群体最优解，则更新群体最优解。

Step5. 更新速度和位置：根据个体最优解和群体最优解，更新每个粒子的速度和位置。速度的更新可以通过考虑个体和群体最优解的差异来引导粒子的移动方向和速度。

Step6. 判断停止条件：判断是否满足停止条件，如达到最大迭代次数或适应度值收敛等。如果满足停止条件，则算法结束；否则，返回步骤 3 继续迭代。

Step7. 输出最优解：根据停止条件确定最优解，即群体最优解或个体最优解。在定日静场的优化模型中，最优解可以表示为一组参数，例如最佳的太阳光照角度和强度等。

结果如下表 3 所示

表 3 问题三设计参数表

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)
(17.3,12.7)	5.27×3.7	4.7	2125	42435.375

5.3 问题三模型建立与求解

5.3.1 模型建立

与问题二所用布置方式相同，定日镜场仍然采用放射栅格法排列，由于定日镜的高度和大小可以改变，由内向外，定日镜的高度依次增大。目标函数与问题一相同，变量与问题二相同，其中定日镜的尺寸改为 $W_i \times H_i$, W_i 为定日镜宽度， H_i 为定日镜高度，定日镜的高度为 Z_i , i 为定日镜序数，目标函数与约束条件如下。

目标函数：

$$E_{\text{field}} = \text{DNI} \cdot \sum_i^N A_i \eta_i \geq 60 \quad (28)$$

约束条件：

$$\sqrt{(x - x_0)^2 + (y - y_0)^2} \geq 100 \quad (29)$$

$$2 < W_i < H_i < 8 \quad (30)$$

$$Z_i \geq \frac{H_i}{2} \quad (31)$$

$$\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} > H + 5 \quad (32)$$

$$r_0 + n\Delta r \leq 350 \quad (33)$$

$$Q = \frac{360}{\alpha} n \quad (34)$$

由于高度可调，增加约束条件使得外圈定日镜的高度、宽度、定日镜中心的高度比相邻内圈定日镜的高度、宽度、定日镜中心的高度分别高 ΔH 、 ΔW 、 ΔZ ，即

$$Z_{(i+360/\alpha)} = Z_i + \Delta Z \quad (35)$$

$$W_{(i+360/\alpha)} = W_i + \Delta W \quad (36)$$

$$H_{(i+360/\alpha)} = H_i + \Delta H \quad (37)$$

5.3.2 模型求解

与第二问相同，本题仍然采用粒子群算法求解该多目标优化模型，步骤同问题二。求解结果如下表 4 所示。

表 4 问题三设计参数表

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)
(3.5,4.6)	/	/	1895	/

六、模型检验

由于在正午十二点时，太阳位于天顶正上方，光线垂直照射地面。在这种情况下，太阳光线与地面法线的夹角对称分布在东西两个方向。故太阳光线在正午十二点时的余弦效率是东西对称的，通过 MATLAB 画图，我们验证了模型的正确性，春分时刻一天内余弦效率分布图如图 12 所示。

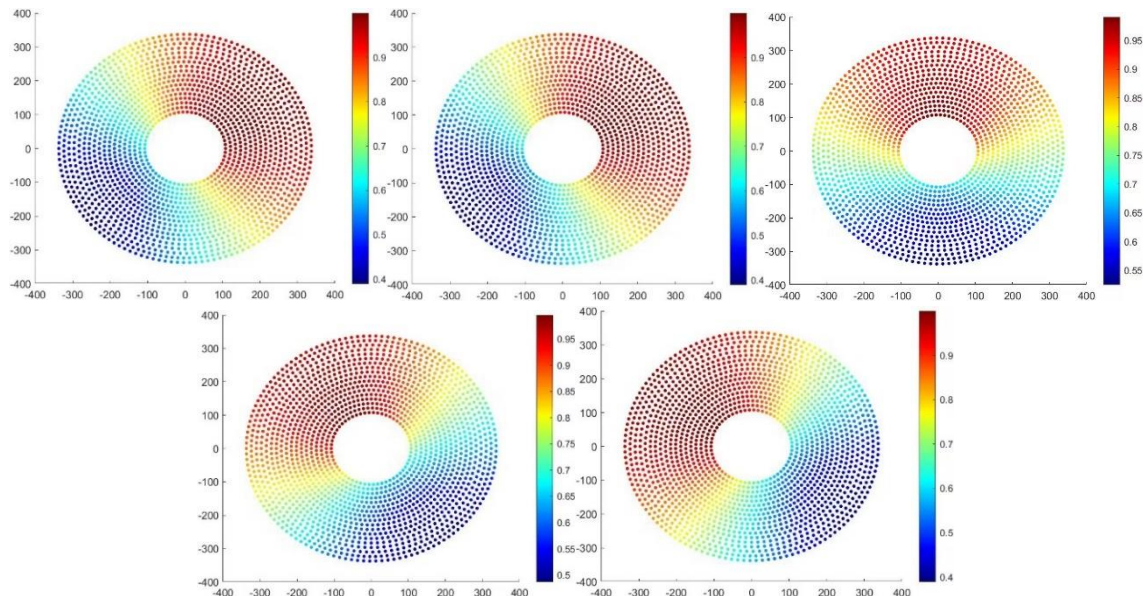


图 12 春分时刻一天内余弦效率分布图

七、模型总结

7.1 模型优点

1、本文基于反射定理，在定日镜模型中利用入射光线的方向向量和反射光线的方向向量求得各定日镜的单位法向量，余弦效率。后续取得的结果较为符合实际。

2、本文采用粒子群算法，并行性高。粒子群算法中的粒子可以同时并行地搜索解空间，加快了算法的收敛速度。搜索全局最优解能力较强。通过粒子之间的信息交流，可以有效地避免局部最优解的陷阱。

3、本文通过对目标函数的处理，降低了计算的复杂度，为模型求解提供便利。

7.2 模型缺点

1、本文在考虑太阳光时，将太阳光射出近似为平行光处理，但实际上，太阳上各点任意方向都有光线射出，故应当将太阳照射到地球上一点的光线看作锥形光，这样更加符合实际情况。

2、在第一问中考虑定日镜的光学效率时，仅考虑后排定日镜接收的太阳光被前方定日镜所阻挡这种情况和后排定日镜在反射太阳光时被前方定日镜阻挡两种情况，实际还有塔对镜场造成的阴影损失，故对于更加精确的光学效率应该考虑更加全面的情况。

7.3 模型推广与改进

本模型对在实际生活中定日镜场的布局优化问题具有很大的借鉴意义，实际问题中，在阴影效率计算中，应该同时考虑塔对镜场造成的阴影损失，另外，在太阳光束判断中，应当将太阳光看作锥形光线而非平行光束，这样更加符合实际情况。

同时定日镜场优化模型可以在太阳能利用、能源系统优化和实验研究等领域进行推广应用，以提高能源利用效率和可持续发展。

参考文献

- [1]张平,奚正稳,华文瀚等.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(06):5-8.
- [2]杜宇航,刘向民,王兴平等.塔式光热电站定日镜不同聚焦策略的影响分析[J].动力工程学报,2020,40(05):426-432.
- [3]谢飞. 塔式太阳能热电系统定日镜场光学仿真与应用研究[D].浙江大学,2013.
- [4]刘建兴. 塔式光热电站光学效率建模仿真及定日镜场优化布置[D].兰州交通大学,2023.

附录

附录一

问题一求解代码

```

clc;clear;close all;
n = 1745; % 定日镜总数
H = 3; % 海拔高度(km)
A = 6*6; % 定日镜面积

% 计算每个月的 21 日到 3 月 21 日的天数差
end_date = datetime('03/21', 'InputFormat', 'MM/dd');
days_difference = calculateDaysDifference(end_date);
D = days_difference; % 以春分作为第 0 天起算的天数

ST = [9, 10.5, 12, 13.5, 15]; % 当地时间
phi = deg2rad(39.4); % 当地纬度(北纬+39.4° )
delta = asin(sin(2*pi*D/365)*sin((2*pi/360)*23.45)); % 太阳赤纬角
omega = pi/12*(ST - 12); % 太阳时角

alpha_s = zeros(12,5);
for i = 1:12
    for j = 1:5
        alpha_s(i,j) = asin(cos(delta(i,1)).*cos(phi).*cos(omega(j,1)) +
sin(delta(i,1)).*sin(phi));
    end
end

gamma_s = zeros(12,5);
for i = 1:12
    for j = 1:5
        gamma_s(i,j) = acos((sin(delta(i,1)) -
sin(alpha_s(i,j)).*sin(phi))./cos(alpha_s(i,j)).*cos(phi)); %太阳方位角
    end
end

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;
G0 = 1.366; % 太阳常数(kW/m^2)
for i = 1:12
    for j = 1:5
        DNI = G0*(a + b*exp(-c./sin(alpha_s))); % 法相直接辐射辐照度(kW/m^2)
    end
end

```

```

    end
end

N = zeros(60*1745,3); % 单位法向量

j = 1;
for i = 1:60
    for k = 1:1745
        % 计算反射平面的单位法向量
        N(j,:) = (Reflected_ray(k,:)-Incident_ray(i,:))/norm(Reflected_ray(k,:)-Incident_ray(i,:));
        % 计算入射角
        theta(j,1) =
acos(dot(N(j,:),Reflected_ray(k,:))/norm(N(j,:))*norm(Reflected_ray(k,:))) ;
    end
end

eta_cos = cos(theta); % 余弦效率
Eta_cos = reshape(eta_cos, 1745, 60); % 余弦效率
Theta = reshape(theta, 1745, 60);
Average_eta_cos = zeros(12,1);
for i = 1:12
    Average_eta_cos(i) = real(sum(sum(Eta_cos(:,1+5*(i-1):5*i))/(n*5))); % 平均余弦效率
end
Average_eta_cos_year = sum(Average_eta_cos)/12; % 年平均余弦效率

Average_eta_sb = 1 - Shadow_occlusion_loss; % 平均阴影遮挡效率
Average_eta_sb_year = sum(eta_sb)/12; % 年平均阴影遮挡效率

eta_ref = 0.92; % 镜面反射率
E_cl = eta_ref*sum(DNI')*A; % 集热器接收能量
E_rf = sum(DNI')*A; % 镜面全反射能量
E_ls = Shadow_occlusion_loss.*sum(DNI')*A; % 阴影遮挡损失能量
Average_eta_trunc = E_cl./(E_rf - E_ls); % 平均集热器截断效率
Average_eta_trunc_year = sum(Average_eta_trunc)/12; % 年平均集热器截断效率

eta_at1 = eta_at(1:12,1);
eta = (eta_sb.*Average_eta_cos.*eta_at1.*eta_trunc).*eta_ref;
eta_year = sum(eta)/12;
E_field = sum(DNI')*.A.*eta; % 平均输出热功率(MV)

```

```

E_field_year = sum(E_field)/12; % 年平均输出热功率(MV)
Unit_area_E_field = E_field/A; % 单位面积定日镜平均输出热功率
Unit_area_E_field_year = sum(Unit_area_E_field)/12;% 单位面积镜面年平均输出热功率(kW/m2)

%% 遮挡识别模型
clc;clear;close all;
tic
% 定日镜坐标数据
coordinate = xlsread("附件.xlsx",1,'A2:B1746'); % 读取 Excel 附录文件

% 计算每个月的 21 日到 3 月 21 日的天数差
end_date = datetime('03/21', 'InputFormat', 'MM/dd');
days_difference = calculateDaysDifference(end_date);
D = days_difference; % 以春分作为第 0 天起算的天数

% 当地时间
ST = [9, 10.5, 12, 13.5, 15]';

% 入射光线方向向量
Incident_ray = zeros(60,3);
k = 1;
for i = 1 : 12
    for j = 1 : 5
        Incident_ray(k,:) = -n_sum_fun(D(i,1),ST(j,1));
        k = k + 1;
    end
end

% 定日镜边长 & 半边长
mirrorWidth = 6;
half_mirrorWidth = mirrorWidth/2;

% 定日镜中心坐标
center_mirror = zeros(60*1745,3);

centerX_mirror = coordinate(:,1);
centerY_mirror = coordinate(:,2);
centerZ_mirror = 4*ones(1745,1);
center_m(1:1745,:) = [centerX_mirror, centerY_mirror, centerZ_mirror];

```

```

M = cell(60, 1);
for i = 1:60
    M{i} = center_m;
end
center_mirror = cat(1,M{:});

% 吸热塔中心坐标
centerX_tower = 0;
centerY_tower = 0;
centerZ_tower = 80;

% 镜面中心(X,Y,Z 方向上)到集热器中心的距离
DeltaX = centerX_tower - centerX_mirror;
DeltaY = centerY_tower - centerY_mirror;
DeltaZ = centerZ_tower - centerZ_mirror;

% 镜面中心到集热器中心的距离 d_HR (m)
d_HR = sqrt(DeltaX.^2 + DeltaY.^2 + DeltaZ.^2);
eta_at = 0.99321 - 0.0001176*d_HR + 1.97*10^(-8)*(d_HR).^2; % 大气透射率

% 反射光线方向向量 Reflected_ray
reflected_rayX = DeltaX./d_HR;
reflected_rayY = DeltaY./d_HR;
reflected_rayZ = DeltaZ./d_HR;
Reflected_ray = [reflected_rayX, reflected_rayY, reflected_rayZ];

% 定日镜的单位法向量 Unit_Normal_Vector
Unit_Normal_Vector = zeros(104700,3);

% 计算反射平面的单位法向量 Unit_Normal_Vector & 入射角 theta
j = 1;
for i = 1:60
    for k = 1:1745
        Unit_Normal_Vector(j,:) = (Reflected_ray(k,:)-
Incident_ray(i,:))/norm(Reflected_ray(k,:)-Incident_ray(i,:)); % 计算反射平面的单位法向量
        theta(j,1) =
acos(dot(Unit_Normal_Vector(j,:),Reflected_ray(k,:))/norm(Unit_Normal_Vector(j,:))*norm(Reflected_ray(k,:))); % 计算入射角
        j = j + 1;
    end
end
end

```

```

%% 计算定日镜的边界向量 boundary_Vector1 & boundary_Vector2
unit_z = zeros(60*1745,3);

boundary_Vector1 = zeros(60*1745,3);
for j = 1:60*1745
    unit_z(j,:) = [0,0,1];
    boundary_Vector1(j,:) = cross(Unit_Normal_Vector(j,:),unit_z(j,:));
    boundary_Vector2(j,:) = cross(Unit_Normal_Vector(j,:),boundary_Vector1(j,:));
end

%% 计算定日镜边界的四个顶点的坐标
% 定日镜边界的四个顶点的坐标
% 注释： 1~3 列 1~1745 行为 1745 个定日镜顶点 1 第 1 时刻的坐标 x,y,z...
mirrorVertices = [];
mirrorVertices(1:1745*60,1:3) = [center_mirror + half_mirrorWidth*(boundary_Vector1 +
boundary_Vector2)];
mirrorVertices(1:1745*60,4:6) = [center_mirror + half_mirrorWidth*(boundary_Vector1 -
boundary_Vector2)];
mirrorVertices(1:1745*60,7:9) = [center_mirror + half_mirrorWidth*(-boundary_Vector1 -
boundary_Vector2)];
mirrorVertices(1:1745*60,10:12) = [center_mirror + half_mirrorWidth*(-boundary_Vector1
+ boundary_Vector2)];

toc

%% %% 计算阴影遮挡面积
%% % 初始化变量
% shadowArea = 0; % 初始化阴影面积为 0
%
%% % 计算被阴影遮挡的面积
% for i = 1:size(mirrorSegments, 1) - 1
%     % 取出相邻的两个顶点坐标
%     vertex1 = mirrorSegments(i, :);
%     vertex2 = mirrorSegments(i+1, :);
%
%     % 计算线段的法向量（逆时针旋转 90 度）
%     normalVector = [-(vertex2(2) - vertex1(2)), vertex2(1) - vertex1(1)];
%
%     % 检查太阳光线和线段的相对方向
%     sunDirection = [targetX - sunX, targetY - sunY];

```

```

% if dot(sunDirection, normalVector) > 0
%     % 太阳光线指向线段的方向，存在阴影遮挡
%     segmentLength = norm(vertex2 - vertex1); % 计算线段长度
%     shadowArea = shadowArea + segmentLength * precision; % 累加被阴影遮挡的面积
% end
% end
%
% % 输出被阴影遮挡的面积
% disp("被阴影遮挡的面积（Shadow Area）:");
% disp(shadowArea);

Shadow_occlusion_loss = ShadowArea/A;
%% 画图
% 二维
figure
scatter(centerX_mirror, centerY_mirror, 10, 'blue', 'o');
hold on
scatter(centerX_tower, centerY_tower, 15, 'filled');
hold on

legend('定日镜坐标', '吸收塔坐标');
xlabel('x(m)');
ylabel('y(m)');

% 三维
figure
scatter3(centerX_mirror, centerY_mirror, centerZ_mirror, 5, 'blue', 'o')
hold on
scatter3(centerX_tower, centerY_tower, centerZ_tower, 15, 'red', 'filled', 'o')
hold on

legend('定日镜坐标', '吸收塔坐标');
xlabel('x(m)');
ylabel('y(m)');
zlabel('z(m)');

%% 函数
% 计算 D
function days_difference = calculateDaysDifference(end_date)
% 设定起始日期为 3 月 21 日
reference_date = datetime('03/21', 'InputFormat', 'MM/dd');

```

```

% 指定目标日期为每个月的 21 日（包括 1 月至 12 月）
months = 1:12;
target_dates = datetime(year(reference_date), months, 21);

% 计算每个目标日期距离指定日期的天数差
days_difference = mod(days(target_dates - end_date), 365);
days_difference(days_difference < 0) = days_difference(days_difference < 0) + 365;

% 转置为列矩阵输出
days_difference = days_difference';
end

% 修正角度
function n_sum=n_sum_fun(D,ST)
%%输出未太阳光的方向向量，看成是日期还有时间的函数
%D=1;%日期，1~365 天内变化
%ST=9;%当地时间，按照题目要求取 9，10.5,12,13.5,15 几个时间点
fi=39.4;%纬度，单位°
deta=asind(sin(2*pi*D/365)*sin(2*pi*23.45/360));%赤纬角，单位°
w=180/12*(ST-12);%太阳时角,单位°
%%太阳高度角，角度制
afa_s=asind(cosd(deta)*cosd(fi).*cosd(w)+sind(deta)*sind(fi));
%太阳方位角，角度制
gama_s=acosd((sind(deta)-sind(afa_s).*sind(fi))./cosd(afa_s)/cosd(fi));
%%太阳方位角的修正
if w<0
%%时角小于 0，表示上午，若北位 x 轴方向要把方位角取成 360 的补角，
%%题目中是东为 x 轴方向，所以还要+90 度
    gama_s=-gama_s+90;
else
%%时角小于 0，表示再上午，若北位 x 轴方向要把方位角取成负的，
%%题目中是东为 x 轴方向，所以还要+90 度
    gama_s=gama_s+90-360;
end
%%太阳光的方向向量
n_sum=[cosd(afa_s)*cosd(gama_s),cosd(afa_s)*sind(gama_s),sind(afa_s)];
n_sum=n_sum/norm(n_sum);%%单位化
end

```

附录二

问题二代码

```
% PSO 算法的参数设置
num_particles = 1000; % 粒子数量
num_dimensions = 2; % 优化问题的维度
max_iterations = 10000; % 迭代次数
c1 = 2; % 加速因子 1（认知因子）
c2 = 2; % 加速因子 2（社会因子）
w = 0.7; % 惯性权重

% 定义目标函数
fitness_function = fun_E_field;
fun_E_field = DNI.*sum(A2.*eta);

% 初始化粒子的位置和速度
positions = rand(num_particles, num_dimensions); % 随机初始化粒子位置
velocities = zeros(num_particles, num_dimensions); % 初始速度设为零

% 设置个体最佳位置和适应值
personal_best_positions = positions;
personal_best_fitness = fitness_function(positions);

% 设置全局最佳位置和适应值
[global_best_fitness, global_best_index] = min(personal_best_fitness);
global_best_position = personal_best_positions(global_best_index, :);

% PSO 主循环
for iteration = 1:max_iterations
    % 更新粒子速度和位置
    r1 = rand(num_particles, num_dimensions);
    r2 = rand(num_particles, num_dimensions);

    velocities = w*velocities + c1*r1.*(personal_best_positions - positions) + ...
        c2*r2.*( repmat(global_best_position, num_particles, 1) - positions);

    positions = positions + velocities;

    % 限制粒子位置在定义域范围内
    positions(positions > 1) = 1;
    positions(positions < 0) = 0;
end
```



```

% 更新个体最佳位置和适应值
current_fitness = fitness_function(positions);
update_indices = current_fitness < personal_best_fitness;

personal_best_positions(update_indices, :) = positions(update_indices, :);
personal_best_fitness(update_indices) = current_fitness(update_indices);

% 更新全局最佳位置和适应值
[current_best_fitness, current_best_index] = min(personal_best_fitness);

if current_best_fitness < global_best_fitness
    global_best_fitness = current_best_fitness;
    global_best_position = personal_best_positions(current_best_index, :);
end

% 显示迭代过程
disp(['Iteration: ', num2str(iteration), ' Best Fitness: ', num2str(global_best_fitness)]);
end

E_field_Best = global_best_position;

% 显示优化结果
disp('Optimization result:');
disp('Best Fitness: ');
disp(global_best_fitness);
disp('Best Position: ');
disp(global_best_position);

```

附录三

问题三求解代码

```

clc;clear all;

fun_E_field = DNI.*sum(A2.*eta);
sqrt((x - x0)^2 + (y - y0)^2) >= 100;
2 < omega <= height < 8;
2 < z < 6;
z >= height/2;
sqrt((x(i+1,1) - x(i,1))^2 + (y(i+1,1) - y(i,1))) > heiht + 5;
sqrt((xi+1,1)-x(i,1))^2 + (Y(i+1,1) - y(i,1))^2 > H + 5;
r0 + n*delta_r <= 350;
Q = 360*n/alpha;
num_particles = 1000;
num_dimensions = 2;

```

```

max_iterations = 10000;
c1 = 2;
c2 = 2;
w = 0.6;
fitness_function = fun_E_field;
positions = rand(num_particles, num_dimensions); % 随机初始化粒子位置
velocities = zeros(num_particles, num_dimensions); % 初始速度设为零
personal_best_positions = positions;
personal_best_fitness = fitness_function(positions);
[global_best_fitness, global_best_index] = min(personal_best_fitness);
global_best_position = personal_best_positions(global_best_index, :);
for iteration = 1:max_iterations
    r1 = rand(num_particles, num_dimensions);
    r2 = rand(num_particles, num_dimensions);
    velocities = w*velocities + c1*r1.*(personal_best_positions - positions) + ...
        c2*r2.*( repmat(global_best_position, num_particles, 1) - positions);
    positions = positions + velocities;
    positions(positions > 1) = 1;
    positions(positions < 0) = 0;
    current_fitness = fitness_function(positions);
    update_indices = current_fitness < personal_best_fitness;
    personal_best_positions(update_indices, :) = positions(update_indices, :);
    personal_best_fitness(update_indices) = current_fitness(update_indices);
    [current_best_fitness, current_best_index] = min(personal_best_fitness);
    if current_best_fitness < global_best_fitness
        global_best_fitness = current_best_fitness;
        global_best_position = personal_best_positions(current_best_index, :);
    end

    disp(['Iteration: ', num2str(iteration), ' Best Fitness: ', num2str(
global_best_fitness)]);
end

```