

基于极坐标沃罗诺伊图中心镶嵌等 数据同化模型的定日镜场优化策略

摘要

作为塔式太阳能光热发电站的基本组件，定日镜的尺寸、位置、安装高度、数目等因素极大程度影响着定日镜场的输出功率。本文建立了四种坐标系，依据主要公式分割问题所需的功能模块，通过几何知识和相关近似理论刻画了物理情境下不同变量之间显示或者隐式的关系；面对高位输入的不适定性情况，我们通过构筑简化模型，最终将输入参数与输出参数的联系表达出来，进而求出在特定条件下使得定日镜的单位镜面面积输出热功率尽可能大的最优解。

针对问题一 首先，我们根据某一时刻的时间和空间信息，通过模型一求出了太阳光的方向向量，同时，更新了定日镜场每面镜子的方向坐标。其次，我们明确了定日镜光学效率的影响因素，找出它和输出功率的关系。为了计算定日镜遮盖面积这个重要变量，我们讨论了一种一般性的情况，根据**光线追迹法**来追迹任意一条经过定日镜 i 的光线，采用了 **K 近邻算法 (K - NearestNeighbor, KNN)** 找出对遮挡镜 i 贡献率最大的 k 个投影来数值上近似总的遮挡面积，在具体计算过程中，我们通过用镜面四个顶点表征镜面信息。然后，我们讨论了塔对镜，镜对镜的出入射光线遮挡影响。在求解集热器截断效率时，我们比较了两种建模方式，最终选取了在数值上更接近真实解的模型。

针对问题二 考虑到题中涉及的参数维度较高，精确求取最优解对计算机的算力有很高的要求。为进一步优化算法，我们在第一问的基础上进行了**模型数据同化**。首先，我们设计了**辐射状环形定日镜场的排列方式** [1]，建立了环的层数和每个定日镜位置以及定日镜总数的关系，通过计算图的形式整理成编码器的形式，将高维输入参数不适定性问题转化为单变量 n_c (定日镜排布环数) 的**一维离散问题**。最后，我们使用**粒子群加速算法**在可行域内搜索最优的定日镜尺寸 [8.0000, 6.9718]，安装高度，定日镜排布环数和吸收塔位置坐标 [0.017,-14.900,84.000]，最后再使用解码器，将 $n_c = 18$ 解码成最优的定日镜场的位置排布，详见表 3。

针对问题三 考虑到定日镜尺寸和安装高度约束条件的去除，计算的复杂度进一步提升。我们采用了工程上常用 [5] 的**超立方拉丁采样 (LHS - Latin Hypercube Sampling)** 和**沃罗诺伊图中心镶嵌 (CVT - Centroid Voronoi Tessellation)** [6] 来进行模型数据同化，在极坐标系下通过**超立方拉丁采样**划分出了 5 个定日镜群，根据**沃罗诺伊图中心镶嵌** [6] 将每个定日镜群内部的定日镜的尺寸和安装高度都设置成常值。最后，我们使用**粒子群加速算法**在可行域内搜索最优的定日镜群的尺寸和安装高度，详细结果见表 6。

关键字： 模型数据同化，定日镜场，光线追迹法，K 近邻算法，超立方拉丁采样，沃罗诺伊图中心镶嵌，粒子群加速算法

一、问题重述

1.1 问题背景

作为一种低碳环保的新型清洁能源技术 [1]，塔式太阳能光热发电技术对我国进一步减少环境污染有重要意义。在塔式太阳能光热发电系统中，定日镜是关键组件之一。定日镜通过反射和聚焦太阳光线，将其汇聚到吸收塔的集热器上，使集热器中的导热介质被加热。这一过程可将太阳能转化为热能，并随后通过热交换转化为电能。因此，最大程度地捕获和利用太阳光的能量，提升单位面积年平均输出热功率是提升塔式太阳能光热发电技术效率的关键，这也将对减少我国碳排放和推动可持续发展产生重要影响。

1.2 问题提出

依据题目信息和有关知识，通过建立数学模型，需要解决如下问题：

问题一：已知吸收塔的位置坐标、定日镜尺寸、安装高度、定日镜数目、定日镜位置，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：在达到额定年平均输出热功率的情况下，假定所有定日镜尺寸及安装高度相同，求解使得单位镜面面积年平均输出热功率最大时的吸收塔位置坐标、定日镜尺寸、安装高度、定日镜数目以及定日镜位置。

问题三：去除问题二中所有定日镜尺寸及安装高度相同的约束条件，求解使得单位镜面面积年平均输出热功率最大时的吸收塔位置坐标、定日镜尺寸、安装高度、定日镜数目以及定日镜位置。

二、模型假设

1. **假设：**传统定日镜的旋转机制。

解释：依据题意，本题中分析的定日镜旋转机制符合传统机制，即纵向转轴的轴线控制反射镜的方位角，水平转轴的轴线控制反射镜的俯仰角。将定日镜的运动以两个轴为中心分析也更方便进行后续的简化计算。

2. **假设：**吸收塔直径与集热器直径相同。

解释：由于题干中并未给出吸收塔的直径，考虑到现实中吸收塔的结构，本文默认吸收塔直径与集热器直径相同。

三、符号说明

符号	意义	单位
O_i	定日镜中心在镜场坐标系下的坐标	(m,m,m)
w_i, h_i	第 i 个定日镜的宽, 高	m
α_s	太阳高度角	°
γ_s	太阳方位角	°
φ	当地纬度	°
ω	太阳时角	°
δ	太阳赤纬角	°
η_{sb}	阴影遮挡效率	1
A_s	定日镜总体被遮挡面积	m ²
η_{\cos}	余弦效率	1
η_{at}	大气透射率	1
η_{trunc}	集热器截断效率	1
η_{ref}	镜面反射率	1
A_s	定日镜总体被遮挡面积	m ²
ε	吸热器设计效率	1
n_c	定日镜排布环数	1
d_{HR}	镜面中心到集热器中心的距离	m
d	定日镜中心距离	m
$r_{i,c}$	第 i 个环的半径	m
n_{total}	定日镜总数	1
$\overline{E_{field}}$	单位镜面面积年平均输出热功率	kW/m ²

四、问题分析

4.1 问题一分析

4.1.1 物理模型表征

在我们的建模过程中,对于吸收塔:我们根据假设定义 O 为镜场坐标系的原点,同时根据问题一的题干描述, O 也是吸收塔的底座中心点,我们定义 S 为吸收塔顶端圆柱形集热器的中心点。对于定日镜,我们定义一个五元组用以表征定日镜的物理和几何信息:

$$\{i, O_i, w_i, h_i, \{\vec{e}_x, \vec{e}_y, \vec{e}_z\}\} \quad (1)$$

其中, i 表示定日镜的下标,对于问题一,我们有 $i \in \{1, 2, 3, \dots, 1745\}$, O_i 表示定日镜中心在镜场坐标系下的坐标, w_i 和 h_i 分别表示第 i 个定日镜的宽和高, $\{\vec{e}_x, \vec{e}_y, \vec{e}_z\}$ 表示第 i 个定日镜的三位坐标方向向量在镜场坐标系下的表示。

4.1.2 流程分析

为了进一步解决问题一涉及的问题,我们建立了三个大模型并包含若干小模型来得到相关数据,获得最终结果。流程如下所示:

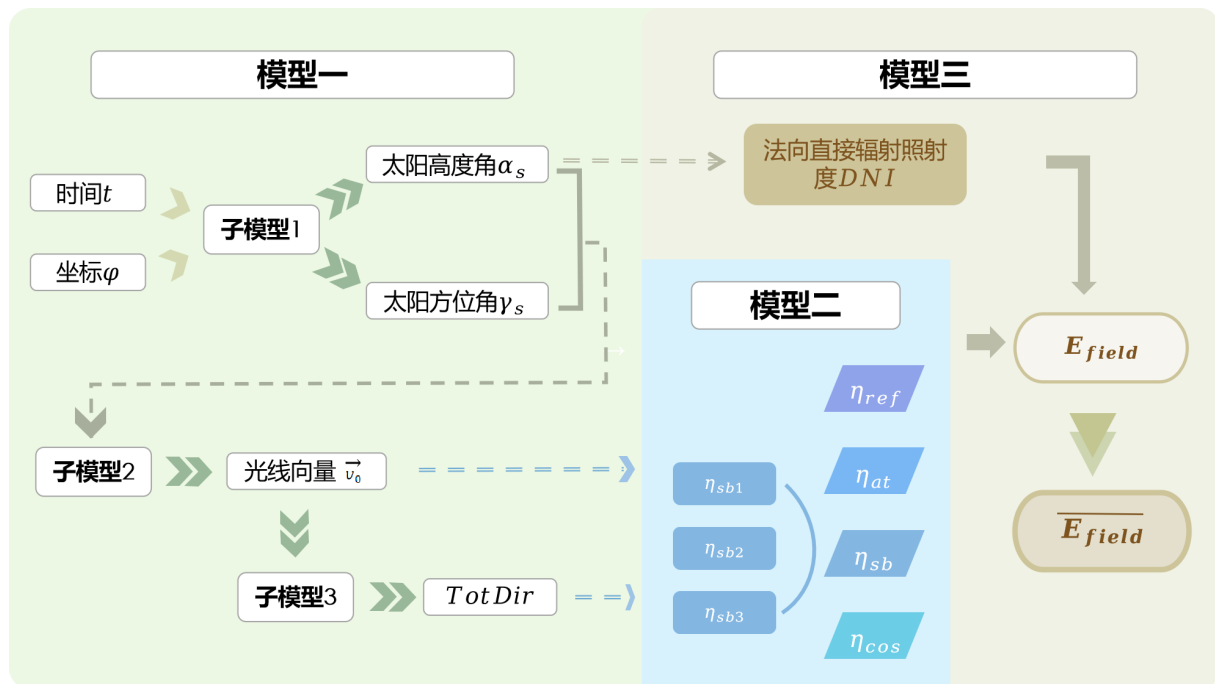


图 1 问题一求解流程

4.2 问题二分析

去除问题一中对定日镜尺寸和安装高度的约束条件,在第一问模型的基础上求出所有定日镜尺寸及安装高度相同时的最优解。

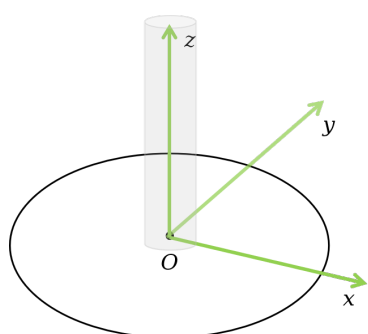
4.3 问题三分析

在第二问的基础上，进一步去除所有定日镜尺寸及安装高度相同的约束条件，在已有模型上求出使得单位镜面面积年平均输出热功率最大的解。

五、问题一建模与求解

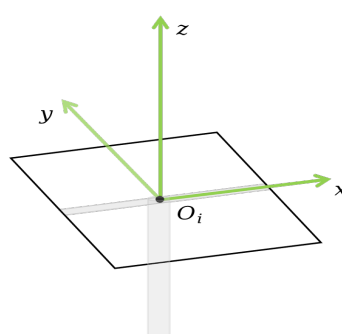
5.1 坐标系的建立

为了便于分析，我们建立了四个坐标，镜场坐标系（笛卡尔坐标系）、定日镜坐标系（笛卡尔坐标系）、地球坐标系（球坐标系）以及光锥坐标系（锥坐标系）。四个坐标系的图示如下所示：



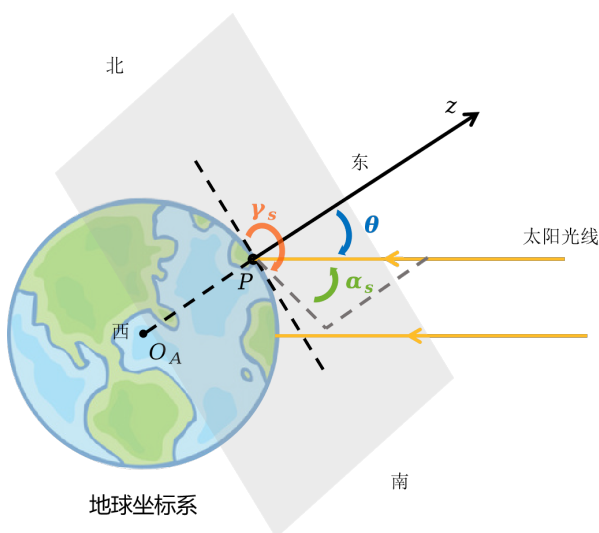
镜场坐标系

图2 镜场坐标系示意图



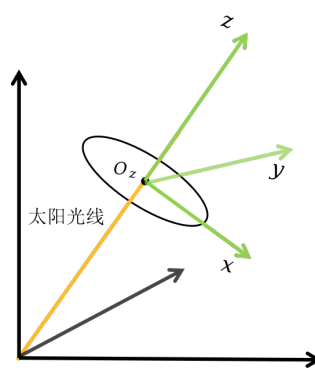
定日镜坐标系

图3 定日镜坐标系示意图



地球坐标系

图4 地球坐标系示意图



光锥坐标系

图5 光锥坐标系示意图

对于任意一个坐标系（以镜场坐标系 R_m 为例），我们称 S_m 为镜场坐标系三个轴向量张成的空间，对于此空间内经过 i 点的一个仿射子空间（平面），我们记作 $U_{m,i}$ ，其他的坐标系也类似。

5.2 模型一: 地球坐标系下时间空间与太阳光主轴方向向量的关系

5.2.1 模型 1.1: 求解太阳高度角和太阳方位角

首先, 根据几何知识, 我们可通过如下表达式求解太阳高度角 α_s [2]:

$$\sin \alpha_s = \cos \delta \cos \varphi \cos \omega + \sin \delta \sin \varphi \quad (2)$$

同理, 我们求解太阳方位角 γ_s [3]:

$$\cos \gamma_s = \frac{\sin \delta - \sin \alpha_s \sin \varphi}{\cos \alpha_s \cos \varphi} \quad (3)$$

其中 φ 为当地纬度, 北纬为正; ω 为太阳时角。

$$\omega = \frac{\pi}{12}(ST - 12) \quad (4)$$

其中 ST 为当地时间, δ 为太阳赤纬角 [4]。

$$\sin \delta = \sin \frac{2\pi D}{365} \sin \left(\frac{2\pi}{360} 23.45 \right) \quad (5)$$

其中 D 为以春分作为第 0 天起算的天数, 例如, 若春分是 3 月 21 日, 则 4 月 1 日对应 $D = 11$ 。

5.2.2 模型 1.2: 转换太阳高度角和太阳方位角为方向向量

为了转换地球坐标系 (球坐标系) 中的角度为镜场坐标系 (笛卡尔坐标系) 下的向量, 我们做了如下变换:

$$\begin{aligned} \vec{V}_0 &= (-\cos \alpha_s \sin \gamma_s, -\cos \alpha_s \cos \gamma_s, \sin \alpha_s) \\ \gamma_s &\in [0, 2\pi), \alpha_s \in \left(0, \frac{\pi}{2}\right] \end{aligned} \quad (6)$$

5.2.3 模型 1.3: 根据太阳方向向量更新所有定日镜的法向向量

由题中所给背景条件, 我们设计了如下的定日镜追光策略:

- 步骤一: 由光的反射定律和反射光线经过吸收塔的限制条件求出每个定日镜的法向向量。

$$\vec{e}_z = \frac{\frac{\vec{O}_i \vec{S}}{\|\vec{O}_i \vec{S}\|_{L_2}} - \vec{V}_0}{\left\| \frac{\vec{O}_i \vec{S}}{\|\vec{O}_i \vec{S}\|_{L_2}} - \vec{V}_0 \right\|_{L_2}} \quad (7)$$

- 步骤二：由定日镜坐标系的坐标正交性和 x 轴与地面平行的信息，得到定日镜的 x 轴方向向量。

$$\begin{aligned} & \text{求 } \vec{e}_x \in \mathbb{R}^3, \text{ 使得} \\ & \begin{cases} \|\vec{e}_x\|_{L_2} = 1 \\ \langle \vec{e}_x, \vec{e}_\perp \rangle = 0 \\ \langle \vec{e}_x, \vec{e}_z \rangle = 0 \end{cases} \end{aligned} \quad (8)$$

其中 $\vec{e}_\perp = [0, 0, 1]^\top$ 。为了限制 \vec{e}_x 与地面平行，即垂直于镜场的法向方向向量。

- 步骤三：由定日镜坐标系的坐标正交性得到定日镜的 y 轴方向向量。

$$\begin{aligned} & \text{求 } \vec{e}_y \in \mathbb{R}^3, \text{ 使得} \\ & \begin{cases} \|\vec{e}_y\|_{L_2} = 1 \\ \langle \vec{e}_y, \vec{e}_x \rangle = 0 \\ \langle \vec{e}_y, \vec{e}_z \rangle = 0 \end{cases} \end{aligned} \quad (9)$$

至此，我们完成了太阳方向向量的更新。我们还需要补充一点：对于模型一的方程组的求解，我们第一次使用了方程直接求解的做法，第二次，为了加速定日镜方向随太阳方向更新模型，我们通过数学上的方程简化，解析地求解出了太阳方向向量。

5.3 模型二：定日镜场的光学效率

5.3.1 求解阴影遮挡效率

$$\begin{cases} \eta_{sb} = 1 - L_s \\ L_s \triangleq \frac{A_s}{A_{totale}} \\ A_s = A_{s1} + A_{s2} + A_{s3} \end{cases} \quad (10)$$

根据相关文献 [3]，我们将阴影遮挡分成以下三类：

- 吸收塔对第 i 个定日镜的遮挡影响 A_{s1}
- 第 j 个定日镜对第 i 个定日镜的入射光遮挡影响 A_{s2}
- 第 j 个定日镜对第 i 个定日镜的反射光遮挡影响 A_{s3}

接下来我们对这三类情况进行讨论分析。

1. 考虑吸收塔对第 i 个定日镜的遮挡影响

我们想要找到吸收塔阴影遮挡的定日镜，考虑先找到吸收塔或者吸收塔的简化表征结构在水平面上的投影，然后衡量定日镜是否会被遮挡。我们基于 Kevin Suffern 教授团队的光线追迹法 [4]，追迹一条光线从射入定日镜到反射到集热器。根据几何学公式设计了如下的算法：

首先,求解吸收塔顶端集热器中心点 S 在镜场坐标系下,在太阳光方向上落在 $X-Y$ 平面上的投影 S' , 然后求解 OO_i , 也就是镜场坐标系原点到第 i 个定日镜的中心的距离, 与在 OS' 上投影的残差 \vec{e} , 判断 \vec{e} 的模长与吸收塔半径的大小关系; 接下来考虑影长和问题的约束, 判断 OO_i 在 OS' 上的投影是否在可行区间内。如果基于算法判定定日镜 i 被遮挡, 则默认其遮挡面积为自身面积。我们设:

$$k\vec{V}_0 = \vec{OS}' - \vec{OS}, k \in \mathbb{R}^+. \quad (11)$$

为了方便, 此处向量范数 $\|\cdot\|$ 均为 L_2 范数, 由几何关系知:

$$\|k\vec{V}_0\|^2 = \|\vec{OS}'\|^2 + \|\vec{OS} + k\vec{V}_0\|^2 \quad (12)$$

化简得:

$$\begin{aligned} -2k \langle \vec{v}_0, \vec{OS}' \rangle &= 2 \cdot \|\vec{OS}'\|^2 \\ k &= -\frac{\|\vec{OS}'\|^2}{\langle \vec{v}_0, \vec{OS}' \rangle} \end{aligned} \quad (13)$$

代入公式 (11) 可得 \vec{OS} 的投影向量。

$$\begin{cases} \vec{OS}' = k\vec{V}_0 + \vec{OS} \\ \left\{ \begin{aligned} \overrightarrow{OO_i} \cdot \vec{OS}' - \frac{\overrightarrow{OO_i} \cdot \overrightarrow{OS}'}{\|\overrightarrow{OS}'\|^2} \cdot \overrightarrow{OS}' < r_t. \\ \frac{\overrightarrow{OO_i} \cdot \overrightarrow{OS}'}{\|\overrightarrow{OS}'\|} \cdot \overrightarrow{OS}' \in [100, 350] \end{aligned} \right. \end{cases} \quad (14)$$

下图为吸收塔对光线遮挡的示意图。

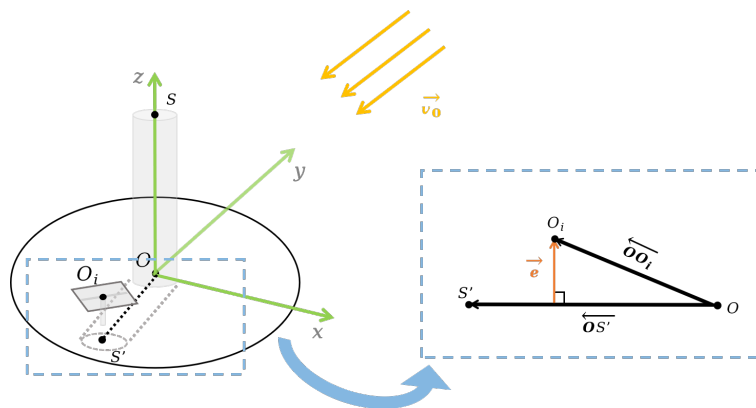


图 6 吸收塔遮挡示意图

2. 考虑第 j 个定日镜对第 i 个定日镜的入射光遮挡影响

首先，我们定义坐标系转换算子矩阵 T:

$$T = \begin{pmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{pmatrix} \quad (15)$$

其中， (l_x, m_x, n_x) 分别为定日镜坐标系的 x 轴向单位向量在镜场坐标系中沿 X,Y,Z 轴的分量，同理， (l_y, m_y, n_y) 和 (l_z, m_z, n_z) 分别对应 y 轴和 z 轴单位向量沿 X,Y,Z 轴的分量，我们可以通过左乘 T 算子矩阵实现一个镜面坐标系中的向量到镜场坐标系的转换。例如，我们称镜面坐标系中的光线向量为 \vec{V}_H ，称镜场坐标系中的光线向量为 \vec{V}_0 ，有以下关系：

从镜面坐标系转换到镜场坐标系：

$$\vec{V}_0 = T \cdot \vec{V}_H \quad (16)$$

从镜场坐标系转换到镜面坐标系：

$$\vec{V}_H = T^T \cdot \vec{V}_0 \quad (17)$$

值得注意的是，T 算子矩阵的列向量是标准正交的，所以有 $T^T \cdot T = I$ 其中，I 代表和 T 的列空间维数相同的单位矩阵。因此，我们在后面的步骤中计算投影的时候，对形如 $T^T \cdot T$ 的式子进行了化简。

对于一般情况，在镜场坐标系下，我们研究一条光线的路径，以第 i 个定日镜作为研究对象分析第 j 个定日镜对其在所研究光线方向向量上投影到镜 i 遮挡面积。我们定义 U_{O_i} 和 U_{O_j} 为经过 i, j 镜中心点的两个平面， U_i 和 U_j 分别为 U_{O_i} 和 U_{O_j} 平面中由宽和高的限制条件下的封闭矩面，即 $(U_i, U_j) \subset (U_{O_i}, U_{O_j})$ ，按照 1. 中的方法做投影：

$$P_{j \rightarrow i}(U_j) = U'_j, \quad U'_j \subset U_{O_i} \quad (18)$$

$$\forall H \in U_j, P_{j \rightarrow i}(H) = H', \quad H' \subset U_{O_i} \quad (19)$$

其中， $P_{j \rightarrow i}(\cdot)$ 为 U_{O_j} 中某点投影到 U_{O_i} 的映射。所以，镜 i 上被遮挡的总面积为：

$$U_{i, total} = U_i \cap \left(\bigcup_{j=1, j \neq i}^{n_c} U'_j \right) \quad (20)$$

具体一点，针对定日镜场的数学模型，我们先基于定日镜 j 中心坐标，在镜面坐标系中快速地求出定日镜 j 四个顶点的位置向量，然后通过公式 (16) 转换到镜场坐标系中，然后完成上述的投影，取并集和取交集操作，最后通过公式 (17) 转换到定日镜 i 的坐标系中。

根据常识和少量的数值模拟，我们发现距离镜 i 较远的定日镜对镜 i 的影响非常小，在数值上可以忽略不计，所以我们采用 KNN(K - NearestNeighbor) 算法找到与镜 i

的欧式距离最近的 K 个镜子，按照距离从小到大依次向镜 i 投影，取交集，从而找出对遮挡镜 i 贡献率最大的 k 个投影来近似总的入射光遮挡面积，具体的算法如下：

Algorithm 1: 一种光线追迹法求解遮挡阴影的算法

Input: k, id, \vec{V}

Output: A_{stemp}

根据 \vec{V} 求出镜 i 和镜 j 的 T 矩阵。 $\{O_1, O_1 \dots O_k\} \leftarrow \text{AlgorithmKNN}(id, k)$

找离目标定日镜最近的 k 个定日镜的函数， k 个定日镜中心点坐标记为 $\{O_1, O_1 \dots O_k\}$

$[ul_j, ur_j, ll_j, ll_j] \leftarrow \text{AlgorithmFindCorner}(\{O_1, O_2 \dots O_k\})$

对于给定的定日镜中心，找到在定日镜坐标系下的四个顶角，分别记作 (ul_j, ur_j, ll_j, ll_j)

for i **in** $1, 2, \dots, k$ **do**

$[ul'_j, ur'_j, ll'_j, ll'_j] \leftarrow \text{AlgorithmProject}([ul_j, ur_j, ll_j, ll_j])$

将镜场坐标系下的镜 j 顶点投影到镜 i 平面上

$[\tilde{ul}'_j, \tilde{ur}'_j, \tilde{ll}'_j, \tilde{ll}'_j] \leftarrow \text{AlgorithmUpdate}[ul'_j, ur'_j, ll'_j, ll'_j]$

对新产生的顶点投影和原先的投影并集取并集，然后与镜 i 取交集

$$A_{stemp} = \sum_{i=1}^k (\tilde{ur}'_i - \tilde{ul}'_i) \cdot (\tilde{ur}'_i - \tilde{lr}'_i)$$

end

return A_{stemp}

定日镜的遮挡示意如下图所示。

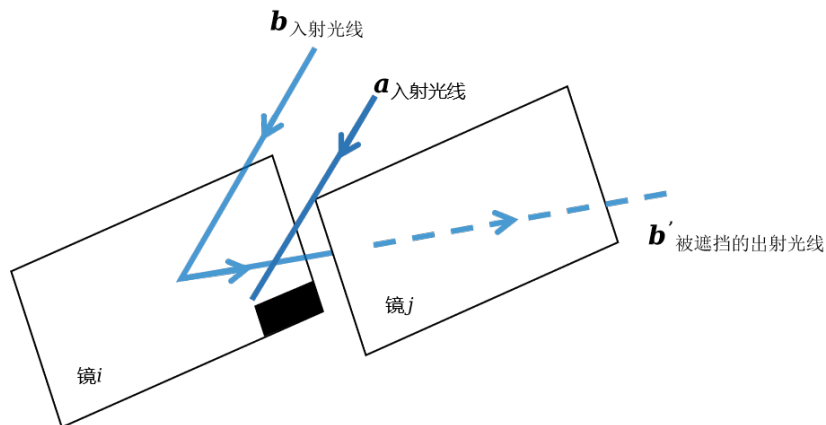


图 7 定日镜遮挡示意图

3. 考虑第 j 个定日镜对第 i 个定日镜的反射光遮挡影响

第三种情况与第二种情况唯一的不同是追踪光线的选择：第二种情况直接是太阳光方向向量，第三种情况是太阳光反射向量作为追踪光线。所以针对第三种情况，我们只

需要现根据定日镜法向向量和太阳光入射方向向量求解反射光线向量，后续操作同第二种情况中我们对一般情况的讨论。

5.3.2 求解余弦效率

根据几何学知识，我们可以通过太阳光方向向量向量 \vec{V}_0 和定日镜法向量 \vec{e}_z 的叉乘，然后归一化处理的方法得到余弦角的余弦值，进而得到余弦效率。

$$\eta_{cos} = \frac{\|\vec{e}_z \times \vec{V}_0\|}{\|\vec{e}_z\| \cdot \|\vec{V}_0\|} \quad (21)$$

5.3.3 求解大气透射率

根据题面材料，可知大气透射率 (η_{at}) 的计算公式为：

$$\eta_{at} = 0.99321 - 0.0001176d_{HR} + 1.97 \times 10^{-8} \times d_{HR}^2 \quad (d_{HR}^2 \leq 1000)$$

其中， d_{HR} 表示镜面中心到集热器中心的距离 (m)，在我们的问题中是 O_iS 。通过带入题中给出的数值，可得出数值结果。

5.3.4 求解集热器截断效率

根据题面材料，可知集热器截断效率 (η_{trunc}) 的计算公式为：

$$\eta_{trunc} = \frac{\text{集热器接收能量}}{\text{镜面全反射能量} - \text{阴影遮挡损失能量}}$$

吸热器溢出损失指由定日镜反射但没有到达吸热器吸热面的能量（去除遮挡损失）。在这里，我们提出了两种模型：

1. 模型 2.1 是考虑太阳光并非平行光线，而是具有一定锥形角的一束锥形光线。因此太阳入射光线经定日镜任意一点的反射光线也是一束锥形光线，又由于太阳的半展角宽为 4.65mrad ，我们将模型简化为：以定日镜中心为锥形坐标系原点发出的一束半展角宽为 4.65mrad 的锥形光束，因此，当光束落到集热器在太阳光反射光线主光轴的正交截面上时，总体辐照区域近似为一个半径为 O_iS 乘半展角宽的圆形，我们记录了模型相应的辐照面积。

2. 模型 2.2 是忽略太阳光的锥形影响，将太阳光主光轴近似为太阳光方向，将定日镜完整地沿着反射方向向量投影到经过集热器中心 S 点的平面上，计算相应的辐照面积。

我们为了评估两种模型的正确性，基于所有定日镜的两种模型的辐照面积之比，画出密度图，如下：

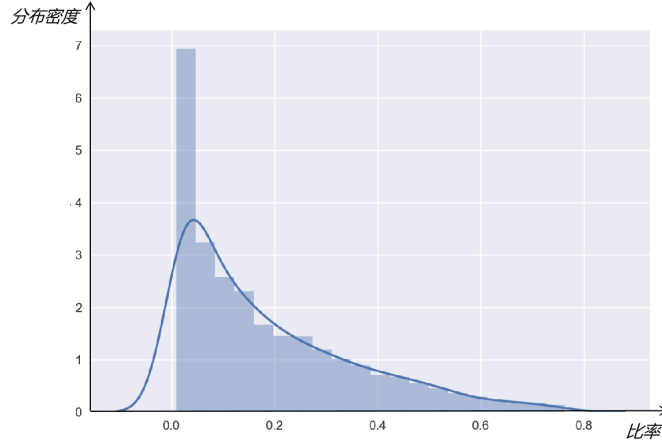


图 8 两种建模方法的比值密度函数

由此可见，相比于第二种模型，第一种模型对于辐照面积的贡献率大约是 1/10。因此，我们选取了第二种模型接入计算模块，考虑到实际工程上因为吸热器表面反射损失和吸热器散热效应，导致集热器（吸热器）阶段效率需要乘以一个比例系数，国内外吸热器设计效率值 [1][5] 大约为 88.0%，具体的计算函数如下：

$$\eta_{\text{trunc}} = \varepsilon \cdot \min \left\{ \frac{D_j \cdot h_j}{w_i \cdot h_i}, 1 \right\}. \quad (22)$$

其中， ε 是吸热器设计效率， D_j, h_j, w_i 和 h_i 分别表示集热器的直径和高，以及第 i 个定日镜的宽和高。

5.3.5 求解镜面反射效率

镜面反射率 η_{ref} 可取为常数，我们选取的是题干中给定的参考值 0.92。

5.4 模型三：定日镜场的输出功率

定日镜场的输出功率可以表示成一个比例系数乘以所有定日镜的光学效率和其面积的乘积求和，公式如下：

$$E_{\text{field}} = \text{DNI} \cdot \sum_i^N A_i \eta_i \quad (23)$$

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。DNI 的计算公式如下：

$$\begin{aligned} \text{DNI} &= G_0 \left[a + b \exp \left(-\frac{c}{\sin \alpha_s} \right) \right] \\ a &= 0.4237 - 0.00821(6 - H)^2 \\ b &= 0.5055 + 0.00595(6.5 - H)^2 \\ c &= 0.2711 + 0.01858(2.5 - H)^2 \end{aligned} \quad (24)$$

其中由题干可知, G_0 为太阳常数, 其值取为 1.366 kW/m^2 , H 为海拔高度 (单位: km)。

5.5 问题一结果呈现

我们针对模型 1.3 中的方程组的求解进行了解析的优化, 优化前和优化后得到的平均光学效率, 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分别如下图所示:

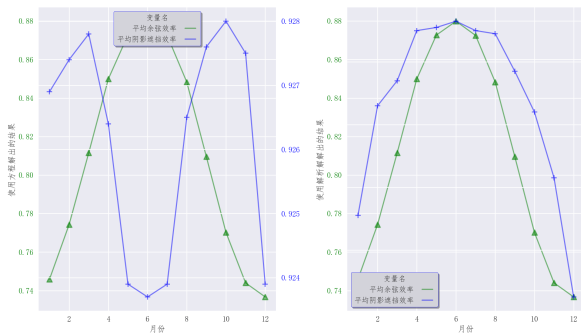


图 9 方程解与解析解在平均余弦效率和平均阴影遮挡效率上的差异

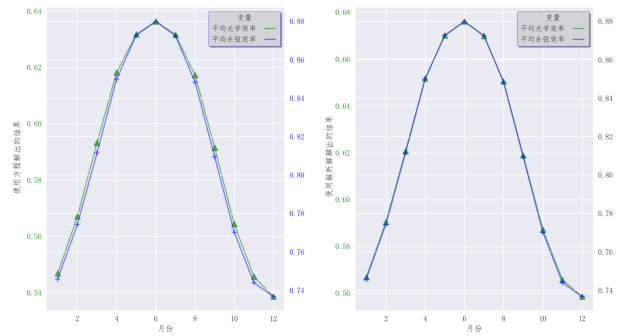


图 10 方程解与解析解在平均光学效率和平均余弦效率上的差异

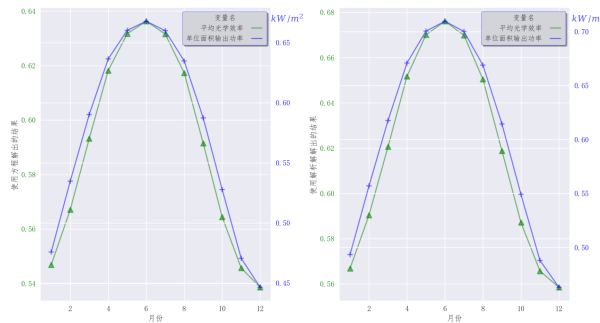


图 11 方程解与解析解在平均光学效率和单位面积输出功率上的差异

可以看到方程解在平均阴影遮挡效率上有一定的反常, 但是由于平均光学效率, 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是正相关的, 总体上呈现出冬季较低, 夏季较高的趋势。这也符合我们的预期, 夏季太阳高度角较大, 总体来讲阴影较小, 定日镜的余弦效率较高; 冬季反之。值得注意的是, 对于我们优化过的解析解, 四个参数的变化趋势大致相同, 大小上略有不同, 较为符合我们的模型预期。

在这里, 我们需要补充一点: 定日镜的平均截断效率结果是常值, 也就是吸热器效率常数, 原因如下: 根据 5.3.5 中的计算, 由于模型 2.1 计算的辐照面积相较于模型 2.2

计算的辐照面积较小，所以我们只选取了定日镜的正交投影作为辐照总面积，经过几何关系的计算，我们发现，根据第一问给出的定日镜尺寸数据，即 $6\text{ m}\times 6\text{ m}$ ，辐照总面积总小于集热器在反射光线方向上的投影，所以定日镜的平均阶段效率是吸热器效率常数。根据建立的模型，我们得到如下的结果。

表 1 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 (km/m^2)
1 月 21 日	0.5550	0.7227	0.9868	0.8800	0.4831
2 月 21 日	0.5715	0.7436	0.9875	0.8800	0.5389
3 月 21 日	0.5891	0.7647	0.9894	0.8800	0.5861
4 月 21 日	0.6012	0.7832	0.9854	0.8800	0.6189
5 月 21 日	0.6049	0.7934	0.9781	0.8800	0.6322
6 月 21 日	0.6074	0.7965	0.9778	0.8800	0.6375
7 月 21 日	0.6050	0.7933	0.9784	0.8800	0.6322
8 月 21 日	0.6009	0.7825	0.9858	0.8800	0.6179
9 月 21 日	0.5881	0.7636	0.9892	0.8800	0.5838
10 月 21 日	0.5694	0.7409	0.9875	0.8800	0.5325
11 月 21 日	0.5540	0.7209	0.9877	0.8800	0.4779
12 月 21 日	0.5471	0.7136	0.9843	0.8800	0.4536

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)	单位面积镜面平均输出热功率 (km/m^2)
0.6189	0.8096	0.9828	0.8800	37.8595	0.6027

六、问题二建模与求解

6.1 问题二建模

由于待优化的定日镜场数目和位置维度过大，且相互之间有较强的依赖性，暴力解决较为困难，所以经过文献调研 [2]，我们设计了辐射状环形定日镜场的排列方式，我们建立了环的层数和每个定日镜位置以及定日镜总数的关系，这些变量如下：

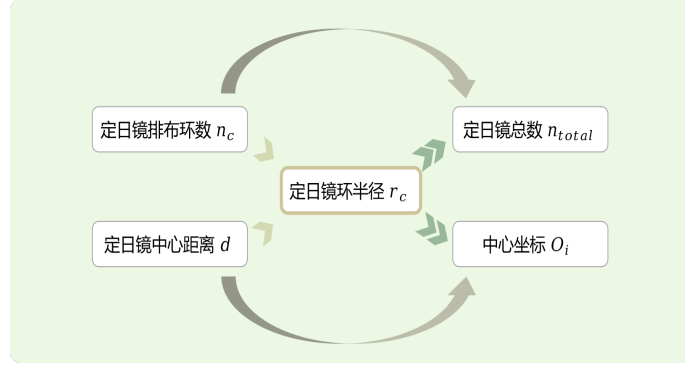


图 12 定日镜场排布设计

其中， n_c 表示我们使用的定日镜排布的环数， d 表示定日镜中心之间的距离， r_c 表示定日镜环的半径，并且我们把第 i 个环的半径记作 $r_{c,i}$ ， O_i 表示第 i 个定日镜的中心坐标 (前两维坐标分别为 $X_{(i,j)}$ 和 $Y_{(i,j)}$)， n_{total} 表示定日镜的总数，他们有如下的关系：

$$d = \sup_{(O_i, O_j) \in S_m^2} \{\text{dist}(O_i, O_j)\} = 2 \cdot \max\{w_i/2\} \cdot + 5 \quad (25)$$

其中， $\text{dist}(\bullet, *)$ 表示 \bullet 到 $*$ 的欧几里得距离。

$$n_{total} = \sum_{i=1}^{n_c} \left[\frac{2\pi[(100 + d(i-1))]}{d} \right] \quad (26)$$

$$X_{(i,j)} = r \cos \left[2j \arctan \left(\frac{d}{2r} \right) \right] \quad (27)$$

$$Y_{(i,j)} = r \sin \left[2j \arctan \left(\frac{d}{2r} \right) \right] \quad (27)$$

因为 $d \ll r$ ，故由泰勒一阶展开得：

$$\begin{aligned} X_{(i,j)} &= r \cos \left\{ j \left[\frac{d}{r} + o \left(\frac{d}{r} \right) \right] \right\} = r \cos \left(\frac{d}{r} j \right) \\ Y_{(i,j)} &= r \sin \left\{ j \left[\frac{d}{r} + o \left(\frac{d}{r} \right) \right] \right\} = r \sin \left(\frac{d}{r} j \right) \end{aligned} \quad (28)$$

我们结合了经典定日镜场排布方式，通过设置调节变量 n_c 简化了问题二中的定日镜位置坐标和数目，在搜索算法适应度函数建立之前，通过上述几何学公式，编码了定日镜场的位置坐标和数目，在搜索算法结束后，通过反函数构建解编码器，重构出满足条件的定日镜场的位置坐标和数目。

6.2 问题二求解

问题二的数学规划目标函数可以写成：

$$\text{Max } \overline{E_{field}}(O_T, H_M, W_M, H_A, n_c) \quad (29)$$

Subject to:

$$\begin{cases} \text{dist}(O, O_T) \in [100, 350]. \\ (H_M, W_M) \in [2, 8]^2. \\ H_A \in [2, 6]. \\ \frac{H_M}{2} < H_A \\ n_C \in [1, 19]. \end{cases}$$

其中, $\overline{E_{field}}$ 为单位镜面面积年平均输出热功率, O_T 为吸收塔的中心坐标, H_M 和 W_M 分别为定日镜的高和宽, H_A 为定日镜安装高度, n_c 为定日镜按照辐射状排布方式下的环数, 通过初步计算, 我们确定 n_c 的最大值为 19。

我们考虑:

(1) 由公式可见, 问题二的输入参数维度较高, 所以我们需要找到一种搜索算法增加解的随机性, 使得我们的数值模拟能够高效快速地搜索整个可行域空间。

(2) 因为问题二的限制条件是区间的形式, 所以可行域是一个超立方体, 也是一个紧集, 即从参数到目标函数的正向映射是一个 Hilbert 空间上的紧算子。

综合以上两个考量, 结合元启发式算法的优点, 我们选用了粒子群加速算法作为搜索算法。为了在数学上刻画的额定功率的限制条件, 我们设计适应度函数如下:

$$\text{argmin } E_{field} \text{sign}(E_{field} - E_0) \quad (30)$$

其中, $E_0 = 60MW$ 为要达到的额定功率, $\text{sign}(\bullet)$ 为符号函数。

6.3 问题二结果呈现

以下结果分别是问题一中给出的定日镜排布方式, 问题二中我们调整输入定日镜场环数 n_c 分别为 18 和 19 时, 定日镜场的排布方式, 可以看到, 随着层数的增大, 定日镜密度也随之增加; 相比于问题一给定的定日镜排布方式, 我们按照辐射状环形结构, 通过贪心策略得到的排布方式更加紧密, 空间利用率更高。

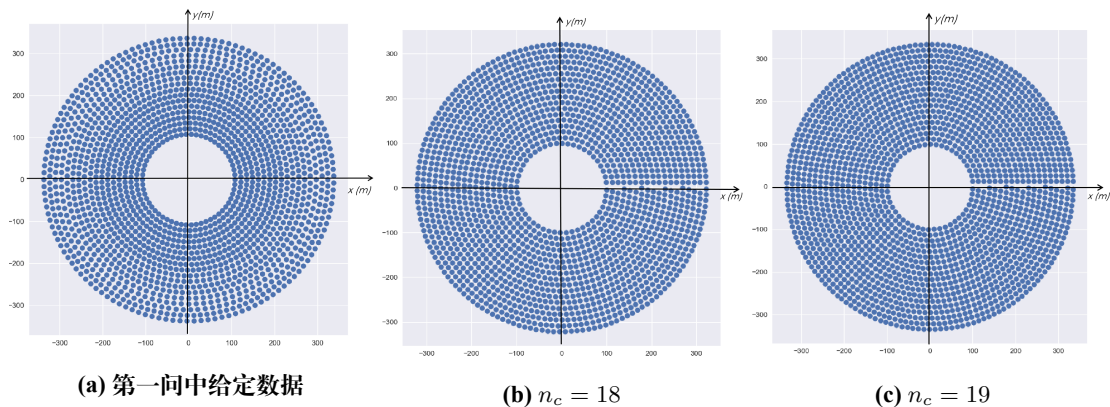


图 13 不同 n_c 的模拟图

根据问题二建立的模型，我们得到如下的结果。

表 3 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 (km/m^2)
1 月 21 日	0.5436	0.7077	0.9887	0.8800	0.4730
2 月 21 日	0.5619	0.7294	0.9919	0.8800	0.5298
3 月 21 日	0.5794	0.7517	0.9920	0.8800	0.5764
4 月 21 日	0.5916	0.7718	0.9865	0.8800	0.6090
5 月 21 日	0.5977	0.7833	0.9871	0.8800	0.6247
6 月 21 日	0.5995	0.7869	0.9797	0.8800	0.6292
7 月 21 日	0.5977	0.7832	0.9819	0.8800	0.6246
8 月 21 日	0.5912	0.7710	0.9868	0.8800	0.6079
9 月 21 日	0.5787	0.7506	0.9923	0.8800	0.5744
10 月 21 日	0.5595	0.7266	0.9913	0.8800	0.5231
11 月 21 日	0.5422	0.7058	0.9891	0.8800	0.4678
12 月 21 日	0.5360	0.6984	0.9877	0.8800	0.4445

表 4 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)	单位面积镜面平均输出热功率 (km/m^2)
0.5733	0.7472	0.9875	0.8800	63.6573	0.5552

表 5 问题二设计参数表

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)
(0.017,-14.934,84)	8×6.9718	3.4859	2049	114281.7456

七、问题三建模与求解

对于问题三，相比于问题二，我们需要多考虑不同定日镜尺寸和安装高度的变化对单位镜面面积年平均输出热功率，但是如果遍历优化每一个定日镜，会造成很大的算力依赖和内存占用，又因为我们只考虑辐射状环形定日镜场 [2]，所以我们希望设计出一种在图形学上近似：

- (1) 不同环（半径）之间的定日镜尺寸和安装高度差异所带来的改变和优化
- (2) 不同阵列（角度）之间的定日镜尺寸和安装高度差异所带来的改变和优化

综上所述，我们采用了工程上比较常用 [5] 的超立方拉丁采样和沃罗诺伊图中心镶嵌 [6]，按照极坐标系的位置关系划分出了 5 个定日镜群，将每个定日镜群的定日镜的尺寸和安装高度都设置成了常值。值得注意的是，我们没有用笛卡尔坐标系下直接进行采样差值或者聚类操作，因为我们认为，辐射状环形定日镜场 [2] 的空间结构转换为极坐标系后，定日镜位点分布会更加均匀，继而使得我们的采样更具有代表性。

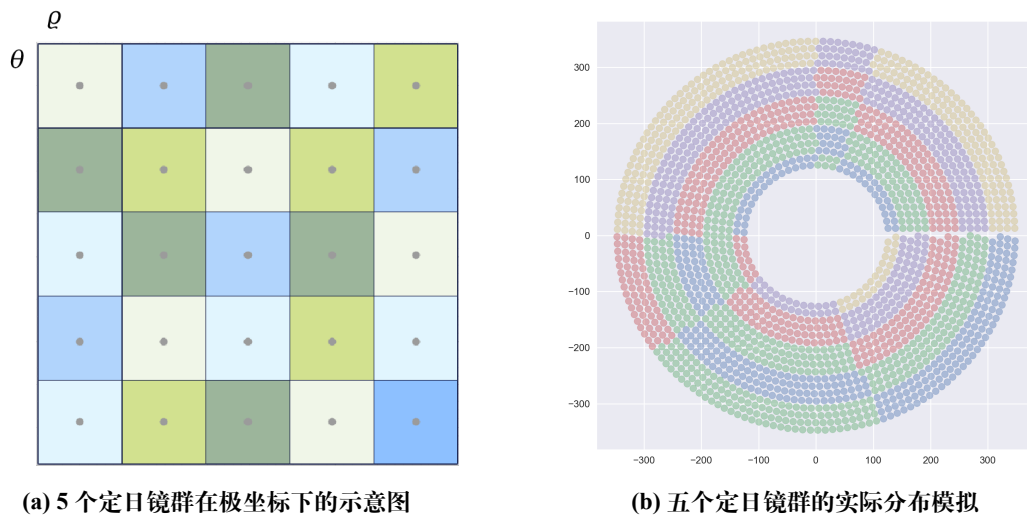


图 14 定日镜群分布

如上图所示，我们在定日镜场的极坐标系的展开图下进行了多次采样数为 5 的超立方拉丁采样，每次每个超立方拉丁采样获得的采样点分别并入五个定日镜群，最后直至对 25 (5×5) 个网格中心都进行了采样，保证我们的模型在不同定日镜群的参数设计上能最大似然地模拟所有定日镜的参数差异所带来的改变和优化。

接下来，我们将上图中的所有定日镜极坐标下的 (ρ, θ) 二元组信息转化为笛卡尔坐标系下的 $(x = \rho \cos \theta, y = \rho \sin \theta)$ 二元组信息，可以看到，这种变换在自由度上是没有损失的。下图展示了在笛卡尔坐标系中我们的采样和插值方法得到的 5 个定日镜群，可以看到，这种采样方式较为均匀。

最后，我们使用粒子群加速算法在可行域内搜索最优的定日镜群的尺寸和安装高度。值得注意的是，根据常理，我们可知，最优解大致分布在第二问的参数附近，也就是在粒子群加速算法的初始化中，我们给定各个定日镜群的参数相等，一致取第二问的结果。

7.1 问题三结果呈现

根据问题三建立的模型，我们得到如下的结果。

表 6 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 (km/m^2)
1 月 21 日	0.5450	0.7117	0.9861	0.8800	0.4742
2 月 21 日	0.5631	0.7329	0.9895	0.8800	0.5309
3 月 21 日	0.5799	0.7545	0.9896	0.8800	0.5769
4 月 21 日	0.5909	0.7738	0.9833	0.8800	0.6083
5 月 21 日	0.5948	0.7846	0.9756	0.8800	0.6217
6 月 21 日	0.5978	0.7880	0.9754	0.8800	0.6274
7 月 21 日	0.5950	0.7845	0.9760	0.8800	0.6218
8 月 21 日	0.5906	0.7730	0.9837	0.8800	0.6073
9 月 21 日	0.5792	0.7534	0.9899	0.8800	0.5750
10 月 21 日	0.5606	0.7302	0.9889	0.8800	0.5242
11 月 21 日	0.5436	0.7099	0.9865	0.8800	0.4689
12 月 21 日	0.5379	0.7026	0.9855	0.8800	0.4460

表 7 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	年平均输出热功率 (MW)	单位面积镜面平均输出热功率 (km/m^2)
0.5732	0.7499	0.9842	0.8800	60.9714	0.5569

表 8 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)
(0.017,-14.934,84)	空 (尺寸不同)	空 (安装高度不同)	2049	117554.2672

八、总结

8.1 敏感性和鲁棒性分析

为了验证模型的性能，我们根据物理问题设计了如下的敏感性测试和模型鲁棒性测试。思路如下：敏感性测试：我们对于问题一求解集热器截断效率部分，对模型进行了一步修正， δ_0 代表模型修正参数，代表由于我们舍弃了太阳光是圆锥形的假设模型，所以我们添加了一项大小和该模型数量级相等的修正项； δ_s 代表外界噪声水平，表示外界光污染或者物体遮挡所导致的集热器截断效率损失。

$$\eta_{\text{trunc}} = \varepsilon \cdot \min \left\{ \frac{D_j \cdot h_j}{\delta_s + \delta_0 + w_i \cdot h_i}, 1 \right\} \quad (31)$$

其中, $\delta_o = \sigma\% \times (w_i \cdot h_i)$

这里, $\sigma \in \{1, 2, 3, 4, 5\}$, $rand(-1, 1)_{gaussian}$ 表示遵循高斯分布随机产生 $[-1, 1]$ 的伪随机数。

对于模型鲁棒性测试, 由于我们在问题二中, 进行了一步模型数据同化, 导致大量信息压缩到中间变量 r 上, 所以在 r 上的噪音攻击可以有效测试模型的稳定性和鲁棒性。具体过程如下:

$$\begin{aligned} X_{(i,j)} &= (r + \delta_r) \cos \frac{dj}{r + \delta_r} \\ Y_{(i,j)} &= (r + \delta_r) \sin \frac{dj}{r + \delta_r} \\ \delta_r &= r \times \sigma\% \times rand(-1, 1)_{gaussian} \end{aligned} \quad (32)$$

这里, $\sigma \in 1, 2, 3, 4, 5$, $rand(-1, 1)_{gaussian}$ 表示遵循高斯分布随机产生 $[-1, 1]$ 的伪随机数。

表 9 模型敏感性和鲁棒性分布表

		敏感性分析					模型鲁棒性分布				
		有效辐照面积					半径大小				
噪声水平	年平均	年平均	年平均阴影	年平均	年平均输出热	年平均	年平均	年平均阴影	年平均	年平均输出热	
	光学效率	余弦效率	遮挡效率	截断效率	功率 (MW)	光学效率	余弦效率	遮挡效率	截断效率	功率 (MW)	
1%	0.007675	0.001874	0.000000	0.009886	0.004323	0.137973	0.099706	0.009722	0.000000	0.024316	
2%	0.017443	0.001874	0.000000	0.019545	0.014049	0.138322	0.099572	0.010228	0.000000	0.023955	
3%	0.026862	0.001874	0.000000	0.029091	0.023595	0.138671	0.099438	0.010633	0.000000	0.023775	
4%	0.036281	0.001874	0.000000	0.038295	0.032961	0.138496	0.098903	0.010937	0.000000	0.023955	
5%	0.045351	0.001874	0.000000	0.047500	0.042147	0.138148	0.098501	0.010937	0.000000	0.024316	

由表 9. 中经过敏感度和鲁棒性攻击测试后相关参数的相对误差。我们不难看出, 模型对于集热器阶段效率敏感度较低, 模型在收到对定日镜场半径的噪音攻击时, 稳定性和鲁棒性较差, 有待我们进一步改进。

8.2 模型优点

1. 模型数据同化, 有效降维: 我们在问题一中计算镜 j 对镜 i 的遮挡, 将模型简化到讨论顶点四个点的遮挡效应; 在问题二中, 我们将超高维的定日镜坐标通过简单的, 符合客观物理情景的编码映射, 降维度压缩到一维; 在问题三中, 我们通过超立方拉丁采样和样和沃罗诺伊图中心镶嵌 [6] 得到的 5 个定日镜群, 将问题在尽可能少地失去表征能力的前提下, 简化为多个定日镜群的差异优化问题。

2. 节省计算内存: 针对问题一, 我们用 KNN 算法得到了在欧几里得空间上最近的 K 个定日镜, 分别和整体地讨论其对定日镜 i 的影响; 针对问题一, 我们在更新定日镜法向向量的一步, 通过求解解析解化简方程, 极大地增大了求解效率。

3. 快速高效：针对问题二和问题三，我们的模型中使用初始条件加速的元启发式算法收敛速度较快，可以看到，问题二的适应度函数在 50 代种群之内就完成了收敛；问题三的适应度函数在 40 代之内就完成了收敛。

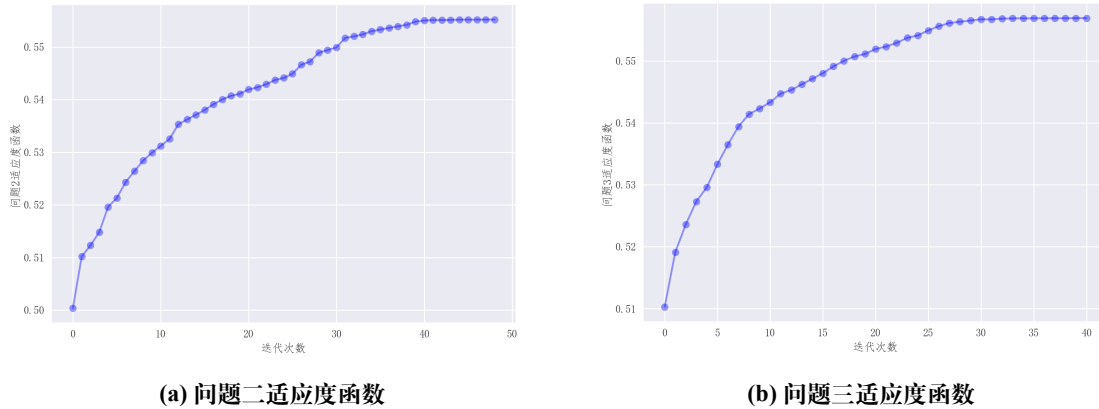


图 15 适应度函数

8.3 模型缺点

1. 模型假设和模型数据同化造成一定的误差：在确定如何摆放镜阵时，直接确定了每一层之间的距离，使得镜子尺寸的选择与安装高度的确定收到了影响；在模型数据同化过程中，不可避免地会导致模型存在误差；在镜子尺寸可以不同的情况下，直接将镜子归为五类，没有考虑其他的分类数目与方法对于最终结果的影响。

2. 算法仍旧存在改进空间：在进行 knn 算法操作时，可以尝试采用快速排序，双向扫描等先进排序算法，有待进一步简化，将求解复杂度从 $O(n^2)$ 降低到 $O(n \log n)$ ；有待继续探索其他的元启发式算法，例如基于 lefy 飞行的布谷鸟算法 [7] 等；有待探索新的模型数据同化方法，用来解决问题三中定日镜群的划分问题。

参考文献

[1] 李心, 许粲玲, 纪培栋, 肖刚, 章涵, 金建祥. (2020). 塔式光热电站集热场设计综述及经济性研究. 南方能源建设, 7(2), 51-59.

[2] Siala, F. M. F., Elayeb, M. E. (2001). Mathematical formulation of a graphical method for a no-blocking heliostat field layout. Renewable energy, 23(1), 77-92.

[3] 张平, 奚正稳, 华文瀚, 王娟娟, 孙登科. (2021). 太阳能塔式光热镜场光学效率计算方法. 技术与市场.

[4] Suffern, K. (2016). Ray Tracing from the Ground up. CRC Press.

- [5] Liu, W., Xu, Y., He, M., Yan, C., Wang, Q., Piao, Y. (2022). Research on infilling strategy in surrogate-aided optimization for axial compressor blades. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy, 236(6), 1048-1058.
- [6] Lévy, B., Liu, Y. (2010). L p centroidal voronoi tessellation and its applications. ACM Transactions on Graphics (TOG), 29(4), 1-11.
- [7] Yang, X. S., Deb, S. (2014). Cuckoo search: recent advances and applications. Neural Computing and applications, 24, 169-174.

附录 A 支撑材料列表

支撑材料内容	文件名
问题二中要求的 result2.xlsx	result2.xlsx
问题三中要求的 result3.xlsx	result3.xlsx
问题一中涉及的程序代码及数据	T1 的模型
问题二中涉及的程序代码及数据	T2 的模型
问题三中涉及的程序代码及数据	T3 的模型

附录 B 问题一的模型算法

```
function y=A_s(id,v0,S,tot_center,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z)
    daan=A_s1(id,v0,S,tot_center,w,h);
    ans1=daan;
    [hang,lie]=size(tot_center);
    num=hang;
    if ans1==0
        ans2=A_s2(v0,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z,tot_center,id);
        ans3=A_s3(v0,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z,tot_center,id);
    else
        ans2=0;
        ans3=0;
    end
    y=ans1+ans2+ans3;
end

function y=A_s1(id,v0,S,tot_center,w,h)
    %塔的遮挡
```

```

ez=[0,0,-1];
cos_ceta=dot(v0,ez)/(norm(v0)*norm(ez));
tan_ceta=sqrt(1-cos_ceta*cos_ceta)/cos_ceta;
% e=v0-dot(v0,ez)/(norm(v0)*norm(ez))*ez;
e=v0-cos_ceta*ez;
os=S;
tmp=norm(os)*sqrt(1/(cos_ceta*cos_ceta)-1);
e=e/norm(e);
[m,n]=size(tot_center);
rt=3.5;%塔的半径
now=0;
oob=tot_center(id,:);%oob是镜子与塔的向量
oob(3)=0;
if norm(oob-dot(oob,e)/norm(e)*e)>rt
    now=0;
elseif norm(dot(oob,e)/norm(e)*e)<100
    now=0;
elseif norm(dot(oob,e)/norm(e)*e)>tmp
    now=0;
else
    now=now+w(id)*h(id);
end
y=now;
end

function ans=A_s2(v0,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z,tot_center,id)
    %%镜面相互的遮挡，入射
    %求v0
    ez=tot_Dir_Z(id,:);
    v=v0+2*dot(v0,ez)/norm(v0)*ez;
    T_b=[tot_Dir_X(id,1),tot_Dir_Y(id,1),tot_Dir_Z(id,1);
        tot_Dir_X(id,2),tot_Dir_Y(id,2),tot_Dir_Z(id,2);
        tot_Dir_X(id,3),tot_Dir_Y(id,3),tot_Dir_Z(id,3)];%镜面算子3*3
    k=1;
    idx=KNN(k,id,tot_center);%距离id处距离最小的k个镜子的编号,tot_center指的是所有镜子的中心下标
    corner_b=find_corner(tot_center(id),w(id),h(id),tot_Dir_X(id),tot_Dir_Y(id),tot_Dir_Z(id),T_b);%4*3的矩阵
    %center是中心点的坐标,
    %w、h分别代表长和宽,是总数(所有镜子的)
    now=0;
    for j=1:k
        center=tot_center(idx,:);
        T_a=[tot_Dir_X(idx,1),tot_Dir_Y(idx,1),tot_Dir_Z(idx,1);
            tot_Dir_X(idx,2),tot_Dir_Y(idx,2),tot_Dir_Z(idx,2);
            tot_Dir_X(idx,3),tot_Dir_Y(idx,3),tot_Dir_Z(idx,3)];%镜面算子3*3
        corner=find_corner(center,w(idx),h(idx),tot_Dir_X(idx),tot_Dir_Y(idx),tot_Dir_Z(idx),T_a);%4*3的矩阵
        corner=corner*transpose(T_a);
        %求投影

```

```

    for i=1:4
        corner(i,:)=corner(i,.)+center-tot_center(id,:);
    end
    corner=corner*T_b;
    now=now+Update_intersection(corner,corner_b);
end
ans=now;
end

function ans=A_s3(v0,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z,tot_center,id)
    %v0
    %%反射的遮挡
    %v看是哪种情况
    v=v0;
    T_b=[tot_Dir_X(id,1),tot_Dir_Y(id,1),tot_Dir_Z(id,1);
        tot_Dir_X(id,2),tot_Dir_Y(id,2),tot_Dir_Z(id,2);
        tot_Dir_X(id,3),tot_Dir_Y(id,3),tot_Dir_Z(id,3)];%镜面算子3*3
    center_b=tot_center(id,:);
    k=1;
    idx=KNN(k,id,tot_center);%距离id处距离最小的k个镜子的编号,tot_center指的是所有镜子的中心下标
    corner_b=find_corner(tot_center(id),w(id),h(id),tot_Dir_X(id),tot_Dir_Y(id),tot_Dir_Z(id),T_b);%4*3的矩阵
    %center是中心点的坐标,
    %w、h分别代表长和宽,是总数(所有镜子的)
    now=0;
    for j=1:k
        center=tot_center(idx,:);
        T_a=[tot_Dir_X(idx,1),tot_Dir_Y(idx,1),tot_Dir_Z(idx,1);
            tot_Dir_X(idx,2),tot_Dir_Y(idx,2),tot_Dir_Z(idx,2);
            tot_Dir_X(idx,3),tot_Dir_Y(idx,3),tot_Dir_Z(idx,3)];%镜面算子3*3
        corner=find_corner(center,w(idx),h(idx),tot_Dir_X(idx),tot_Dir_Y(idx),tot_Dir_Z(idx),T_a);%4*3的矩阵
        corner=corner*transpose(T_a);
        %求投影(有疑问)
        OaOb=center_b-center;
        tmp=sum(OaOb.^2)/(dot(OaOb,v))*v;
        for i=1:4
            corner(i,:)=corner(i,.)+tmp;
        end
        corner=corner*T_b;
        now=now+Update_intersection(corner,corner_b);
    end
    ans=now;
end

function ans=adjust_coordinate(v0,id,S,tot_center)
    ObS=S-tot_center(id,:);
    ObS=ObS/norm(ObS);
    e_tmp=ObS-v0;

```



```

ez=e_tmp/norm(e_tmp);
z1=ez(1);
z2=ez(2);
ex=[sqrt(z1*z1/(z1*z1+z2*z2)), -sqrt(z2*z2/(z1*z1+z2*z2)), 0];%取哪一个解
ey=cross(ez,ex);
if ey(3)<0
    ex=[-sqrt(z1*z1/(z1*z1+z2*z2)), sqrt(z2*z2/(z1*z1+z2*z2)), 0];
    ey=cross(ez,ex);
end
%%
%syms x y
%[x,y]=solve([x*x+y*y==1,x*ez(1)+y*ez(2)==0],[x y]);
%ex=[x(1),y(1),0];
%ey=cross(ez,ex);
%if ey(3)<0
%    ex=[x(2),y(2),0];
%    ey=cross(ez,ex);
%end
ans=[ex;ey;ez];
end

function y=Date(month,day)
M=[31,28,31,30,31,30,31,31,30,31,30,31];
now=0;
for i=1:month-1
    now=M(i)+now;
end
now=now+day;
now=now-31-28-21;
if now<0
    now=now+365;
end
y=now;
end

function ans=Dis(tot,i,id)
ans=(tot(i,1)-tot(id,1))*(tot(i,1)-tot(id,1))+(tot(i,2)-tot(id,2))*(tot(i,2)-tot(id,2))+(tot(i,3)-tot(id,3))
end

function corner=find_corner(center,w,h,Dir_X,Dir_Y,Dir_Z,T)
%ul全称upper_left,其余同理,为1*3的矩阵
ul=center-w/2*Dir_X+h/2*Dir_Y;
ur=center+w/2*Dir_X+h/2*Dir_Y;
ll=center-w/2*Dir_X-h/2*Dir_Y;
lr=center+w/2*Dir_X-h/2*Dir_Y;
ul=transpose(T)*transpose(ul);
ur=transpose(T)*transpose(ur);

```

```

    ll=transpose(T)*transpose(ll);
    lr=transpose(T)*transpose(lr);
    ul=transpose(ul);
    ur=transpose(ur);
    ll=transpose(ll);
    lr=transpose(lr);
    corner=[ul;ur;ll;lr];
end

%
filename = '附件.xlsx';
[tot_center,txt,row] = xlsread(filename,'A2:C2104');%tot_center代表的是每个镜子的坐标
[hang,lie]=size(tot_center);
filename1 = 'wh.xlsx';
[w,txt,row] = xlsread(filename1,'A2:A2104');
[h,txt,row] = xlsread(filename1,'B2:B2104');

%%
%%prepare
Time=[9,10.5,12.0001,13.5,15];

for month=1:12
    month
    %ANS代表表一中的五个元素
    ANS1=0;
    ANS2=0;
    ANS3=0;
    ANS4=0;
    ANS5=0;
    E_field_aver=0;
    for CLOCK=1:5
%%
%%model_1

day=21;%日期
D=Date(month,day);%从春分起的天数
sin_delta=sin(2*pi*D/365)*sin(2*pi/360*23.45);
delta=asin(sin_delta);%太阳赤纬角
phi=39.4/180*pi;
ST=Time(CLOCK);%当地时间
omega=pi/12*(ST-12);%太阳时角
sin_phi=sin(phi);
sin_alpha_s=cos(delta)*cos(phi)*cos(omega)+sin(delta)*sin(phi);%太阳高度角
alpha_s=asin(sin_alpha_s);
%太阳方位角
cos_gama_s=(sin_delta-sin_alpha_s*sin(phi))/cos(alpha_s)/cos(phi);
gama_s=acos(cos_gama_s);

```

```

%求v0
v0=[-cos(alpha_s)*sin(gama_s),-cos(alpha_s)*cos(gama_s),sin(alpha_s)];
v0=v0/norm(v0);
%month=5;
%day=21;%日期
%D=Date(month,day);%从春分起的天数
%sin_delta=sin(2*pi*D/365)*sin(2*pi/360*23.45);
%cos_delta=sqrt(1-sin_delta*sin_delta);
%phi=39.4/180*pi;
%ST=12;%当地时间
%omega=pi/12*(ST-12);%太阳时角
%sin_phi=sin(phi);
%sin_alpha_s=cos_delta*cos(phi)*cos(omega)+sin_delta*sin(phi);%太阳高度角
%cos_alpha_s=sqrt(1-sin_alpha_s*sin_alpha_s);
%太阳方位角
%sin_delta
%sin_alpha_s
%sin_phi=sin(phi)
%cos_alpha_s
%cos_phi=cos(phi)
%cos_gama_s=(sin_delta-sin_alpha_s*sin(phi))/(cos_alpha_s*cos(phi))
%sin_gama_s=sqrt(1-cos_gama_s*cos_gama_s)
%求v0
v0=[-cos_alpha_s*sin_gama_s,-cos_alpha_s*cos_gama_s,sin_alpha_s];
v0=v0/norm(v0)

%%
%%model_3
H=3;%海拔高度(单位:km)
G0=1.366;%太阳常数
a=0.4237-0.00821*(6-H)*(6-H);
b=0.5055+0.00595*(6.5-H)*(6.5-H);
c=0.2711+0.01858*(2.5-H)*(2.5-H);
DNI=G0*[a+b*exp(-c/sin_alpha_s)];

%%
%%coordinate system
S=[0,-15,84];%塔的坐标
tot_Dir_X=zeros(hang,3);
tot_Dir_Y=zeros(hang,3);
tot_Dir_Z=zeros(hang,3);
for i=1:hang
    tmp=adjust_coordinate(v0,i,S,tot_center);
    tot_Dir_X(i,:)=tmp(1,:);

```

```

    tot_Dir_Y(i,:)=tmp(2,:);
    tot_Dir_Z(i,:)=tmp(3,:);
end

%%
%%model_2
%A_tot=0;
%for i=1:hang
%   A_tot=A_tot+w(i)*h(i);
%end
%eta_sb=1-A_s(v0,S,tot_center,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z)/A_tot
%求eta_cos
%eta_cos=0;
%for i=1:hang
%
%   eta_cos=eta_cos+norm(cross(tot_Dir_Z(i,:),v0))/norm(tot_Dir_Z(i,:))/norm(v0)*w(i)*h(i)/A_tot;
%end
%eta_cos
%eta_at=0.99321-0.0001176*d_HR+1.97*0.0000001*d_HR*d_HR;
%eta_trunc;
%eta_ref;
%eta=eta_sb*eta_cos*eta_at*eta_trunc*eta_ref;

%%
A_tot=zeros(hang,1);
eta_sb=zeros(hang,1);
eta_cos=zeros(hang,1);
eta_at=zeros(hang,1);
eta_trunc=zeros(hang,1);
eta_ref=zeros(hang,1);
R=7;
H=8;
%total为总的表面积
total=0;
%ans都为答案，以表一的顺序推进
ans1=0;
ans2=0;
ans3=0;
ans4=0;
ans5=0;
for id=1:hang
    A_tot(id)=w(id)*h(id);
    total=total+A_tot(id);
end
for id=1:hang
    %eta_sb

```

```

eta_sb(id)=1-A_s(id,v0,S,tot_center,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z)/A_tot(id);
if eta_sb(id)<0
    eta_sb(id)=0;
end
%eta_at
d_HR=norm(S-tot_center(id));
eta_at(id)=0.99321-0.0001176*d_HR+1.97*0.00000001*d_HR*d_HR;
%eta_ref
eta_ref(id)=0.92;
%eta_cos
eta_cos(id)=norm(cross(tot_Dir_Z(id,:),v0))/norm(tot_Dir_Z(id,:))/norm(v0);
%eta_trunc
epsilon=0.88;
eta_trunc(id)=epsilon*min(1,R*H/w(id)/h(id));
%eta
eta(id)=eta_sb(id)*eta_cos(id)*eta_at(id)*eta_trunc(id)*eta_ref(id);
ans1=ans1+eta(id)*w(id)*h(id)/total;
ans2=ans2+eta_cos(id)*w(id)*h(id)/total;
ans3=ans3+eta_sb(id)*w(id)*h(id)/total;
ans4=ans4+eta_trunc(id)*w(id)*h(id)/total;
end

%求E_field
E_field=0;
for i=1:hang
    E_field=E_field+DNI*h(i)*w(i)*eta(i);
end
E_field;
ANS1=ANS1+ans1;
ANS2=ANS2+ans2;
ANS3=ANS3+ans3;
ANS4=ANS4+ans4;
ANS5=ANS5+E_field/total;
E_field_aver=E_field_aver+E_field;
    end
    ANS1=ANS1/5
    ANS2=ANS2/5
    ANS3=ANS3/5
    ANS4=ANS4/5
    ANS5=ANS5/5
    E_field_aver=E_field_aver/5
end

%
filename = '附件.xlsx';
[tot_center,txt,raw] = xlsread(filename,'A2:C1746');%tot_center代表的是每个镜子的坐标
[hang,lie]=size(tot_center);

```

```

filename1 = 'wh.xlsx';
[w,txt,row] = xlsread(filename1,'A2:A1746');
[h,txt,row] = xlsread(filename1,'B2:B1746');

%%
%%prepare
Time=[9,10.5,12.0001,13.5,15];

for month=1:12
    month
    %ANS代表表一中的五个元素
    ANS1=0;
    ANS2=0;
    ANS3=0;
    ANS4=0;
    ANS5=0;
    for CLOCK=1:5
%%
%%model_1

day=21;%日期
D=Date(month,day);%从春分起的天数
sin_delta=sin(2*pi*D/365)*sin(2*pi/360*23.45);
delta=asin(sin_delta);%太阳赤纬角
phi=39.4/180*pi;
ST=Time(CLOCK);%当地时间
omega=pi/12*(ST-12);%太阳时角
sin_phi=sin(phi);
sin_alpha_s=cos(delta)*cos(phi)*cos(omega)+sin(delta)*sin(phi);%太阳高度角
alpha_s=asin(sin_alpha_s);
%太阳方位角
cos_gama_s=(sin_delta-sin_alpha_s*sin(phi))/cos(alpha_s)/cos(phi);
gama_s=acos(cos_gama_s);
%求v0
v0=[-cos(alpha_s)*sin(gama_s),-cos(alpha_s)*cos(gama_s),sin(alpha_s)];
v0=v0/norm(v0);
%month=5;
%day=21;%日期
%D=Date(month,day);%从春分起的天数
%sin_delta=sin(2*pi*D/365)*sin(2*pi/360*23.45);
%cos_delta=sqrt(1-sin_delta*sin_delta);
%phi=39.4/180*pi;
%ST=12;%当地时间
%omega=pi/12*(ST-12);%太阳时角
%sin_phi=sin(phi);
%sin_alpha_s=cos_delta*cos(phi)*cos(omega)+sin_delta*sin(phi);%太阳高度角
%cos_alpha_s=sqrt(1-sin_alpha_s*sin_alpha_s);

```

```

%太阳方位角
%sin_delta
%sin_alpha_s
%sin_phi=sin(phi)
%cos_alpha_s
%cos_phi=cos(phi)
%cos_gama_s=(sin_delta-sin_alpha_s*sin(phi))/(cos_alpha_s*cos(phi))
%sin_gama_s=sqrt(1-cos_gama_s*cos_gama_s)
%求v0
v0=[-cos_alpha_s*sin_gama_s,-cos_alpha_s*cos_gama_s,sin_alpha_s];
v0=v0/norm(v0)

%%
%%model_3
H=3;%海拔高度(单位:km)
G0=1.366;%太阳常数
a=0.4237-0.00821*(6-H)*(6-H);
b=0.5055+0.00595*(6.5-H)*(6.5-H);
c=0.2711+0.01858*(2.5-H)*(2.5-H);
DNI=G0*[a+b*exp(-c/sin_alpha_s)];

%%
%%coordinate system
S=[0,0,84];%塔的坐标
tot_Dir_X=zeros(hang,3);
tot_Dir_Y=zeros(hang,3);
tot_Dir_Z=zeros(hang,3);
for i=1:hang
    tmp=adjust_coordinate(v0,i,S,tot_center);
    tot_Dir_X(i,:)=tmp(1,:);
    tot_Dir_Y(i,:)=tmp(2,:);
    tot_Dir_Z(i,:)=tmp(3,:);
end

%%model_2
%%
A_tot=zeros(hang,1);
eta_sb=zeros(hang,1);
eta_cos=zeros(hang,1);
eta_at=zeros(hang,1);
eta_trunc=zeros(hang,1);
eta_ref=zeros(hang,1);
R=7;
H=8;

```

```

%total为总的表面积
total=0;
%ans都为答案，以表一的顺序推进
ans1=0;
ans2=0;
ans3=0;
ans4=0;
ans5=0;
for id=1:havg
    A_tot(id)=w(id)*h(id);
    total=total+A_tot(id);
end
for id=1:havg
    %eta_sb
    eta_sb(id)=1-A_s(id,v0,S,tot_center,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z)/A_tot(id);
    if eta_sb(id)<0
        eta_sb(id)=0;
    end
    %eta_at
    d_HR=norm(S-tot_center(id));
    eta_at(id)=0.99321-0.0001176*d_HR+1.97*0.00000001*d_HR*d_HR;
    %eta_ref
    eta_ref(id)=0.92;
    %eta_cos
    eta_cos(id)=norm(cross(tot_Dir_Z(id,:),v0))/norm(tot_Dir_Z(id,:))/norm(v0);
    %eta_trunc
    epsilon=0.88;
    eta_trunc(id)=epsilon*min(1,R*H/w(id)/h(id));
    %eta
    eta(id)=eta_sb(id)*eta_cos(id)*eta_at(id)*eta_trunc(id)*eta_ref(id);
    ans1=ans1+eta(id)*w(id)*h(id)/total;
    ans2=ans2+eta_cos(id)*w(id)*h(id)/total;
    ans3=ans3+eta_sb(id)*w(id)*h(id)/total;
    ans4=ans4+eta_trunc(id)*w(id)*h(id)/total;
end

%求E_field
E_field=0;
for i=1:havg
    E_field=E_field+DNI*h(i)*w(i)*eta(i);
end
E_field
ANS1=ANS1+ans1;
ANS2=ANS2+ans2;
ANS3=ANS3+ans3;
ANS4=ANS4+ans4;
ANS5=ANS5+E_field/total;

```



```

end
ANS1=ANS1/5
ANS2=ANS2/5
ANS3=ANS3/5
ANS4=ANS4/5
ANS5=ANS5/5
end

function ans=Update_intersection(corner_a,corner_b)
x=zeros(4);
y=zeros(4);
x(1)=corner_a(3,1);
x(2)=corner_a(4,1);
x(3)=corner_b(3,1);
x(4)=corner_b(4,1);
y(1)=corner_a(3,2);
y(2)=corner_a(4,2);
y(3)=corner_b(3,2);
y(4)=corner_b(4,2);
if (y(4)-y(1))*(y(3)-y(2))>0
    ans=0;
elseif (x(4)-x(1))*(x(3)-x(2))>0
    ans=0;
else
    x=paixu(x);
    y=paixu(y);
    ans=(x(3)-x(2))*(y(3)-y(2));
end
end
end

```

附录 C 问题二的模型算法

```

function ans=T2_da(now)
nc=now(1);
tower_x=now(2);
tower_y=now(3);
h=now(4);
w=now(5);
H=now(6);
if h>H*2
    h=H*2;
end
tot_center=T2_location(nc,H);
[hang,lie]=size(tot_center);

```

```

location_tower=[tower_x,tower_y,84];
ans=T2_func(tot_center,h*ones(hang,1),w*ones(hang,1),location_tower,hang);
end

function DAAN=T2_func(tot_center,h,w,location_tower,hang)
%%
%%prepare
ANS6=0;
ANS7=0;
for month=1:12
    month;
    Time=[9,10.5,12.0001,13.5,15];
    %ANS代表表一中的五个元素
    ANS1=0;
    ANS2=0;
    ANS3=0;
    ANS4=0;
    ANS5=0;
    for CLOCK=1:5
        %%
        %%model_1
        day=21;%日期
        D=Date(month,day);%从春分起的天数
        sin_delta=sin(2*pi*D/365)*sin(2*pi/360*23.45);
        delta=asin(sin_delta);%太阳赤纬角
        phi=39.4/180*pi;
        ST=Time(CLOCK);%当地时间
        omega=pi/12*(ST-12);%太阳时角
        sin_phi=sin(phi);
        sin_alpha_s=cos(delta)*cos(phi)*cos(omega)+sin(delta)*sin(phi);%太阳高度角
        alpha_s=asin(sin_alpha_s);
        %太阳方位角
        cos_gama_s=(sin_delta-sin_alpha_s*sin(phi))/cos(alpha_s)/cos(phi);
        gama_s=acos(cos_gama_s);
        %求v0
        v0=[-cos(alpha_s)*sin(gama_s),-cos(alpha_s)*cos(gama_s),sin(alpha_s)];
        v0=v0/norm(v0);

        %%
        %%model_3
        H=3;%海拔高度(单位:km)
        G0=1.366;%太阳常数
        a=0.4237-0.00821*(6-H)*(6-H);
        b=0.5055+0.00595*(6.5-H)*(6.5-H);
        c=0.2711+0.01858*(2.5-H)*(2.5-H);
        DNI=G0*[a+b*exp(-c/sin_alpha_s)];
    end
end

```

```

%%
%%coordinate system
S=location_tower;%塔 的坐标
tot_Dir_X=zeros(hang,3);
tot_Dir_Y=zeros(hang,3);
tot_Dir_Z=zeros(hang,3);
for i=1:hang
    tmp=adjust_coordinate(v0,i,S,tot_center);
    tot_Dir_X(i,:)=tmp(1,:);
    tot_Dir_Y(i,:)=tmp(2,:);
    tot_Dir_Z(i,:)=tmp(3,:);
end

%%
A_tot=zeros(hang,1);
eta_sb=zeros(hang,1);
eta_cos=zeros(hang,1);
eta_at=zeros(hang,1);
eta_trunc=zeros(hang,1);
eta_ref=zeros(hang,1);
R=7;
H=8;
%total为总的表面积
total=0;
%ans都为答案，以表一的顺序推进
ans1=0;
ans2=0;
ans3=0;
ans4=0;
ans5=0;
for id=1:hang
    A_tot(id)=w(id)*h(id);
    total=total+A_tot(id);
end
for id=1:hang
    %eta_sb
    eta_sb(id)=1-A_s(id,v0,S,tot_center,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z)/A_tot(id);
    if eta_sb(id)<0
        eta_sb(id)=0;
    end
    %eta_at
    d_HR=norm(S-tot_center(id));
    eta_at(id)=0.99321-0.0001176*d_HR+1.97*0.00000001*d_HR*d_HR;
    %eta_ref
    eta_ref(id)=0.92;
    %eta_cos
    eta_cos(id)=norm(cross(tot_Dir_Z(id,:),v0))/norm(tot_Dir_Z(id,:))/norm(v0);

```

```

    %eta_trunc
    epsilon=0.88;
    eta_trunc(id)=epsilon*min(1,R*H/w(id)/h(id));
    %eta
    eta(id)=eta_sb(id)*eta_cos(id)*eta_at(id)*eta_trunc(id)*eta_ref(id);
    ans1=ans1+eta(id)*w(id)*h(id)/total;
    ans2=ans2+eta_cos(id)*w(id)*h(id)/total;
    ans3=ans3+eta_sb(id)*w(id)*h(id)/total;
    ans4=ans4+eta_trunc(id)*w(id)*h(id)/total;
end

%求E_field
E_field=0;
for i=1:hang
    E_field=E_field+DNI*h(i)*w(i)*eta(i);
end
E_field;
ANS1=ANS1+ans1;
ANS2=ANS2+ans2;
ANS3=ANS3+ans3;
ANS4=ANS4+ans4;
ANS5=ANS5+E_field/total;
ANS6=ANS6+E_field/60;
ANS7=ANS7+E_field/total/60;
end
end
DAAN=-(ANS6-60000)/abs(ANS6-60000)*ANS7;
if ANS6>60000 && DAAN<-0.52
    ANS6
    chang=h(1) ~
    kuan=w(1)
    gao=tot_center(1,3)
    DAAN
    location_tower
end
end
gao=3.4859;
tot_center=T2_location(18,gao)

%nc表示几个环
%rc表示半径
%d表示sup(distance)
d=13;
nc=10;
rc=87+d*nc;
n_total=0;

```

```

for i=1:nc
    n_total=n_total+floor(2*pi*(87+d*i)/d);
end
x=zeros(30,300);
y=zeros(30,300);
for i=1:nc
    for j=1:floor(2*pi*(87+d*i)/d)
        x(i,j)=(87+d*i)*cos(d*j/(87+d*i));
        y(i,j)=(87+d*i)*sin(d*j/(87+d*i));
    end
end

function daan=T2_location(nc,h)
%nc表示几个环
%rc表示半径
%d表示sup(distance)
%%
%d=13;
%rc=100+d*nc;
%n_total=0;
%for i=1:nc
%    n_total=n_total+floor(2*pi*(100+d*i)/d);
%end
%tot_center=zeros(n_total,3);
%num=0;
%for i=1:nc
%    for j=1:floor(2*pi*(100+d*i)/d)
%        num=num+1;
%        tot_center(num,1)=(100+d*i)*cos(d*j/(100+d*i));
%        tot_center(num,2)=(100+d*i)*sin(d*j/(100+d*i));
%        tot_center(num,3)=h;
%    end
%end
%daan=tot_center;
%%
d=13;
rc=87+d*nc;
n_total=0;
for i=1:nc
    n_total=n_total+floor(2*pi*(360-d*i)/d);
end
tot_center=zeros(n_total,3);
num=0;
for i=1:nc
    for j=1:floor(2*pi*(360-d*i)/d)
        num=num+1;
        tot_center(num,1)=(360-d*i)*cos(d*j/(360-d*i));

```

```

        tot_center(num,2)=(360-d*i)*sin(d*j/(360-d*i));
        tot_center(num,3)=h;
    end
end
daan=tot_center;
end

filename = '附件.xlsx';
[tot_center,txt,row] = xlsread(filename,'A2:C1746');%tot_center代表的是每个镜子的坐标
[hang,lie]=size(tot_center);
filename1 = 'wh.xlsx';
[w,txt,row] = xlsread(filename1,'A2:A1746');
[h,txt,row] = xlsread(filename1,'B2:B1746');
%location_tower为塔的坐标
location_tower=[0,0,84];
T2_func(tot_center,h,w,location_tower,hang)

```

附录 D 问题三的模型算法

```

function ANS=T3_da(now)
    filename = 'id.xlsx';
    [id,txt,row] = xlsread(filename,'A1:A2049');%tot_center代表的是每个镜子的坐标
    h1=now(1);
    w1=now(2);
    height1=now(3);
    h2=now(4);
    w2=now(5);
    height2=now(6);
    h3=now(7);
    w3=now(8);
    height3=now(9);
    h4=now(10);
    w4=now(11);
    height4=now(12);
    h5=now(13);
    w5=now(14);
    height5=now(15);
    m=length(id);
    h=zeros(m,1);
    w=zeros(m,1);
    H=zeros(m,1);
    tot_center=T3_get_location();
    if h1>height1*2
        h1=height1*2;
    end
end

```

```

if h2>height2*2
    h2=height2*2;
end
if h3>height3*2
    h3=height3*2;
end
if h4>height4*2
    h4=height4*2;
end
if h5>height5*2
    h5=height5*2;
end
for i=1:m
    if id(i)==1
        h(i)=h1;
        w(i)=w1;
        H(i)=height1;
        tot_center(i,3)=height1;
    elseif id(i)==2
        h(i)=h2;
        w(i)=w2;
        H(i)=height2;
        tot_center(i,3)=height2;
    elseif id(i)==3
        h(i)=h3;
        w(i)=w3;
        H(i)=height3;
        tot_center(i,3)=height3;
    elseif id(i)==4
        h(i)=h4;
        w(i)=w4;
        H(i)=height4;
        tot_center(i,3)=height4;
    else
        h(i)=h5;
        w(i)=w5;
        H(i)=height5;
        tot_center(i,3)=height5;
    end
end
tot_center;
%ANS=T3_func(tot_center,h,w,[0 -15 84],m,now);
end

function ANS=T3_divide()
    tot_center=T3_get_location();
    area_rou=[100,150,200,250,300,350];

```

```

area_ceta=[0,2*pi/5,4*pi/5,6*pi/5,8*pi/5,10*pi/5];
cnt1=zeros(5,1);
cnt2=zeros(5,1);
[m,n]=size(tot_center);
id_rou=ones(m,1);
id_ceta=ones(m,1);
id=zeros(m,1);
%%进行笛卡尔系到极坐标系的转换
for i=1:m
    for j=1:5
        if
            area_rou(j)*area_rou(j)<=tot_center(i,1)*tot_center(i,1)+tot_center(i,2)*tot_center(i,2)
            &&
            area_rou(j+1)*area_rou(j+1)>=tot_center(i,1)*tot_center(i,1)+tot_center(i,2)*tot_center(i,2)
            id_rou(i)=j;
            break
        end
    end
    end
    %求ceta角
    an=atan(tot_center(i,2)/tot_center(i,1));
    if an<0 %第四象限
        an=an+2*pi;
    end
    if tot_center(i,2)>0
        if tot_center(i,1)<0 %第二象限
            an=an+pi;
        end
    end
    if tot_center(i,2)<0
        if tot_center(i,1)<0 %第三象限
            an=an+pi;
        end
    end
    end
    for j=1:5
        if area_ceta(j)<=an && area_ceta(j+1)>=an
            id_ceta(i)=j;
            break
        end
    end
end

num=0;
while num<5
    hang=rand;
    lie=rand;
    hang=ceil(hang*5);

```



```

lie=ceil(lie*5);
if cnt1(hang)+cnt2(lie)>=1
    continue
end
cnt1(hang)=1;
cnt2(lie)=1;
num=1+num;
for i=1:m
    if id(i)>0
        continue
    end
    if id_rou(i)==hang
        id(i)=num;
    end
    if id_ceta(i)==lie
        id(i)=num;
    end
end
end
ANS=id;
end

function ANS=T3_divide_2()
    tot_center=T3_get_location();
    area_rou=[100,150,200,250,300,350];
    area_ceta=[0,2*pi/5,4*pi/5,6*pi/5,8*pi/5,10*pi/5];
    cnt1=zeros(5,1);
    cnt2=zeros(5,1);
    [m,n]=size(tot_center);
    id_rou=ones(m,1);
    id_ceta=ones(m,1);
    id=zeros(m,1);
    %%进行笛卡尔系到极坐标系的转换
    for i=1:m
        for j=1:5
            if
                area_rou(j)*area_rou(j)<=tot_center(i,1)*tot_center(i,1)+tot_center(i,2)*tot_center(i,2)
                &&
                area_rou(j+1)*area_rou(j+1)>=tot_center(i,1)*tot_center(i,1)+tot_center(i,2)*tot_center(i,2)
            id_rou(i)=j;
            break
        end
    end
    %求ceta角
    an=atan(tot_center(i,2)/tot_center(i,1));
    if an<0 %第四象限
        an=an+2*pi;
    end
end

```

```

end
if tot_center(i,2)>0
    if tot_center(i,1)<0 %第二象限
        an=an+pi;
    end
end
if tot_center(i,2)<0
    if tot_center(i,1)<0 %第三象限
        an=an+pi;
    end
end
for j=1:5
    if area_ceta(j)<=an && area_ceta(j+1)>=an
        id_ceta(i)=j;
        break
    end
end
end
for i=1:m

    if abs(id_ceta(i)-id_rou(i))==0
        id(i)=1;
    elseif abs(id_ceta(i)-id_rou(i))==1
        id(i)=2;
    elseif abs(id_ceta(i)-id_rou(i))==2
        id(i)=3;
    elseif abs(id_ceta(i)-id_rou(i))==3
        id(i)=4;
    elseif abs(id_ceta(i)-id_rou(i))==4
        id(i)=5;
    end
end
ANS=id;
end

function DAAN=T3_func(tot_center,h,w,location_tower,hang,now)
%%
%%prepare
ANS6=0;
ANS7=0;
for month=1:12
    month;
    Time=[9,10.5,12.0001,13.5,15];
    %ANS代表表一中的五个元素
    ANS1=0;
    ANS2=0;
    ANS3=0;

```

```

ANS4=0;
ANS5=0;
E_field_average=0;
for CLOCK=1:5
    %%
    %%model_1
    day=21;%日期
    D=Date(month,day);%从春分起的天数
    sin_delta=sin(2*pi*D/365)*sin(2*pi/360*23.45);
    delta=asin(sin_delta);%太阳赤纬角
    phi=39.4/180*pi;
    ST=Time(CLOCK);%当地时间
    omega=pi/12*(ST-12);%太阳时角
    sin_phi=sin(phi);
    sin_alpha_s=cos(delta)*cos(phi)*cos(omega)+sin(delta)*sin(phi);%太阳高度角
    alpha_s=asin(sin_alpha_s);
    %太阳方位角
    cos_gama_s=(sin_delta-sin_alpha_s*sin(phi))/cos(alpha_s)/cos(phi);
    gama_s=acos(cos_gama_s);
    %求v0
    v0=[-cos(alpha_s)*sin(gama_s),-cos(alpha_s)*cos(gama_s),sin(alpha_s)];
    v0=v0/norm(v0);

    %%
    %%model_3
    H=3;%海拔高度(单位:km)
    G0=1.366;%太阳常数
    a=0.4237-0.00821*(6-H)*(6-H);
    b=0.5055+0.00595*(6.5-H)*(6.5-H);
    c=0.2711+0.01858*(2.5-H)*(2.5-H);
    DNI=G0*[a+b*exp(-c/sin_alpha_s)];

    %%
    %%coordinate system
    S=location_tower;%塔的坐标
    tot_Dir_X=zeros(hang,3);
    tot_Dir_Y=zeros(hang,3);
    tot_Dir_Z=zeros(hang,3);
    for i=1:hang
        tmp=adjust_coordinate(v0,i,S,tot_center);
        tot_Dir_X(i,:)=tmp(1,:);
        tot_Dir_Y(i,:)=tmp(2,:);
        tot_Dir_Z(i,:)=tmp(3,:);
    end

    %%
    A_tot=zeros(hang,1);

```

```

eta_sb=zeros(hang,1);
eta_cos=zeros(hang,1);
eta_at=zeros(hang,1);
eta_trunc=zeros(hang,1);
eta_ref=zeros(hang,1);
R=7;
H=8;
%total为总的表面积
total=0;
%ans都为答案，以表一的顺序推进
ans1=0;
ans2=0;
ans3=0;
ans4=0;
ans5=0;
for id=1:hang
    A_tot(id)=w(id)*h(id);
    total=total+A_tot(id);
end
for id=1:hang
    %eta_sb
    eta_sb(id)=1-A_s(id,v0,S,tot_center,w,h,tot_Dir_X,tot_Dir_Y,tot_Dir_Z)/A_tot(id);
    if eta_sb(id)<0
        eta_sb(id)=0;
    end
    %eta_at
    d_HR=norm(S-tot_center(id));
    eta_at(id)=0.99321-0.0001176*d_HR+1.97*0.00000001*d_HR*d_HR;
    %eta_ref
    eta_ref(id)=0.92;
    %eta_cos
    eta_cos(id)=norm(cross(tot_Dir_Z(id,:),v0))/norm(tot_Dir_Z(id,:))/norm(v0);
    %eta_trunc
    epsilon=0.88;
    eta_trunc(id)=epsilon*min(1,R*H/w(id)/h(id));
    %eta
    eta(id)=eta_sb(id)*eta_cos(id)*eta_at(id)*eta_trunc(id)*eta_ref(id);
    ans1=ans1+eta(id)*w(id)*h(id)/total;
    ans2=ans2+eta_cos(id)*w(id)*h(id)/total;
    ans3=ans3+eta_sb(id)*w(id)*h(id)/total;
    ans4=ans4+eta_trunc(id)*w(id)*h(id)/total;
end

%求E_field
E_field=0;
for i=1:hang
    E_field=E_field+DNI*h(i)*w(i)*eta(i);
end

```

```

        end
        E_field;
        ANS1=ANS1+ans1;
        ANS2=ANS2+ans2;
        ANS3=ANS3+ans3;
        ANS4=ANS4+ans4;
        ANS5=ANS5+E_field/total;
        ANS6=ANS6+E_field/60;
        ANS7=ANS7+E_field/total/60;
        E_field_average=E_field_average+E_field;
    end
    ANS1=ANS1/5;
    ANS2=ANS2/5;
    ANS3=ANS3/5;
    ANS4=ANS4/5;
    ANS5=ANS5/5;
    E_field_average=E_field_average/5;
end

DAAN=-ANS7;
end
end

function tot_center=T3_get_location()
    gao=3.4859;
    tot_center=T2_location(18,gao);
end

now=[6.9718 8.0000 3.4859 8.0000 8.0000 3.4097 6.8000 8.0000 5.0000 6.8000 8.0000 3.0000
    6.8166 8.0000 4.3708];
filename = 'id.xlsx';
[id,txt,row] = xlsread(filename,'A1:A2049');%tot_center代表的是每个镜子的坐标
h1=now(1);
w1=now(2);
height1=now(3);
h2=now(4);
w2=now(5);
height2=now(6);
h3=now(7);
w3=now(8);
height3=now(9);
h4=now(10);
w4=now(11);
height4=now(12);
h5=now(13);
w5=now(14);
height5=now(15);

```

```

m=length(id);
h=zeros(m,1);
w=zeros(m,1);
H=zeros(m,1);
tot_center=T3_get_location();
if h1>height1*2
    h1=height1*2;
end
if h2>height2*2
    h2=height2*2;
end
if h3>height3*2
    h3=height3*2;
end
if h4>height4*2
    h4=height4*2;
end
if h5>height5*2
    h5=height5*2;
end
for i=1:m
    if id(i)==1
        h(i)=h1;
        w(i)=w1;
        H(i)=height1;
        tot_center(i,3)=height1;
    elseif id(i)==2
        h(i)=h2;
        w(i)=w2;
        H(i)=height2;
        tot_center(i,3)=height2;
    elseif id(i)==3
        h(i)=h3;
        w(i)=w3;
        H(i)=height3;
        tot_center(i,3)=height3;
    elseif id(i)==4
        h(i)=h4;
        w(i)=w4;
        H(i)=height4;
        tot_center(i,3)=height4;
    else
        h(i)=h5;
        w(i)=w5;
        H(i)=height5;
        tot_center(i,3)=height5;
    end
end

```

```

end
%ANS=T3_func(tot_center,h,w,[0 -15 84],m,now);

filename = 'id.xlsx';
[id,txt,row] = xlsread(filename,'A1:A2049');%tot_center代表的是每个镜子的坐标
[m,n]=size(id);
num=zeros(5,1);
for i=1:m
    num(id(i))=num(id(i))+1;
end
ans=num(1)*6.9718*8+num(2)*8*8+num(3)*6.8*8+num(4)*6.8*8+num(5)*6.8166*8

narvs=15;
x_lb=[6 8 2 6 8 2 6 8 2 6 8 2 6 8 2];
x_ub=[8 8 6 8 8 6 8 8 6 8 8 6 8 8 6];
%T3_da(x_lb);
[x,fval]=particleswarm(@T3_da,narvs,x_lb,x_ub);
x
fval

```

附录 E result2.xlsx