

基于目标优化-模拟退火算法的定日镜场布局策略研究

摘要

定日镜是塔式电站的主要组件之一，承担着接收太阳能并将其反射到集热器的关键任务，其布局策略对于塔式电站的单位镜面面积年平均输出热功率具有重要影响。本文根据输出热功率的影响因素推导了其求解过程，并通过**模拟退火算法**来求解单目标优化模型，对定日镜场布局的优化进行了研究。

针对问题一，需要通过计算太阳位置以及定日镜的状态，来求解法向直接辐照度、定日镜采光面积以及定日镜场的光学效率，从而得到输出热功率。其中，大部分参数都可以通过理论公式求解，我们重点关注的是光学效率中阴影遮挡效率以及截断效率的模型建立及求解。对于阴影遮挡效率，本文首先利用旋转矩阵得到太阳照射下定日镜与吸收塔在 xoy 平面上的投影边界坐标，此时真实定日镜场阴影遮挡部分就是投影平面上阴影重叠的区域，再利用**蒙特卡洛法**对整个镜场的投影区域进行随机投点，计算求得重叠阴影与单层阴影的比例，从而求得阴影遮挡效率。对于截断效率，本文采用**平板投影法**和**光线追踪法**计算，首先用**角均分法**将反射光锥分成若干根光线，再通过几何关系分析建立公式求解。

针对问题二，本文提出两个模型。其一是在达到额定功率的情况下，改变定日镜位置、尺寸参数以及吸收塔位置，以获得最大化单位镜面面积年平均输出热功率，属于**多变量单目标**类型问题，因此我们以单位镜面面积年平均输出热功率尽量大为目标函数建立单目标优化模型，此处我们还通过公式的联立，将求解最大单位镜面面积年平均输出热功率问题**转化为求解最大余弦效率**的问题，为模型二的建立提供了思路。其二是利用模拟退火算法求解最优的定日镜位置坐标，在利用模拟退火算法随机生成定日镜坐标前，我们首先对镜场的不同阵列形状进行对比分析，本文使用**DELSOL 布局**。接着对定日镜的布局做初步分析，基于将求最大单位镜面面积年平均输出热功率问题转化为求最大余弦效率的问题，我们得到基本的定日镜场布局思路，即定日镜分布呈现“**北多南少，东西对称**”的排布规律。同时通过此模型获得较优方案后，我们还通过模拟退火算法微调 **10000** 次已生成的定日镜位置，来优化镜场布局。

针对问题三，在问题二的基础上引入可调整不同定日镜尺寸及安装高度的要求，本题的基本算法思路与前两问相同。为了简化计算我们直接采用第二问优化后的镜场布局方案，再通过分析不同区域的定日镜安装高度对输出热功率的影响，得到一个设计定日镜安装高度在镜场中分布的方案，接着在分析影响因素后建立了一个关于定日镜坐标的安装高度计算模型，从而得到最终的最优布局。

关键词：定日镜场 单目标最优化模型 蒙特卡洛法 模拟退火算法 DELSOL 布局

一、问题重述

1.1 问题背景

发展新能源技术是实现“碳中和”“碳达峰”的必经之路，在这样一个绿色经济的大背景之下，一种新型清洁能源技术——塔式太阳能光热发电应运而生。

塔式电站主要由大量排列的定日镜和位于中心的集热器组成。其中，定日镜用于收集太阳能并反射到集热器中心，集热器用于储存热能并通过热交换将其转换成电能。定日镜（图 1）的底座有水平与纵向两种转轴，使定日镜的俯仰角与法向得以调控，两转轴交点即定日镜中心。集热器位于吸收塔顶端，集热器中心距地面的高度即吸收塔高度。塔式电站中的定日镜层层排布成一个定日镜场（图 2），其排布设计对年平均光学效率及输出功率有一定程度的影响。



图 1 定日镜及其底座示意图



图 2 圆形定日镜场示意图

1.2 问题提出

现规划在东经 98.5° ，北纬 39.4° ，海拔 3000m 的空地上建立一个以此为圆心，半径为 350m 的圆形定日镜场。以该区域中心为原点，建立镜场坐标系，其中 x 轴正向为正东，y 轴正向为正北，z 轴正向垂直地面向上。

定日镜场中心设有一座吸收塔，该塔周围 100 米之内用于建造安装发电、储能和控制等设备的厂房。吸收塔上方是直径 7 米、高度 8 米的圆柱形外表受光式集热器。定日镜为平面矩形，左右两条边称为镜面高度，上下两条边始终平行于地面，称为镜面宽度。一般情况下后者大于等于前者。边长在 2 至 8 米之间，安装高度在 2 至 6 米之间，同时要确保镜面底端不会接触地面。为了维护与通行需要，相邻定日镜中心的间距要比镜面宽度多 5 米以上。

为便于计算，本文中所有的“年均”指标计算都是基于当地时间每月 21 日的 9:00、10:30、12:00、13:30 和 15:00 进行。

基于上述背景与附件资料，我们通过建立数学模型解决以下问题：

- (1) 若定日镜尺寸设为 $6\text{ m} \times 6\text{ m}$ ，安装高度设为 4 m，且所有定日镜中心的位置已知，计算该定日镜场的年平均光学效率及输出热功率、单位镜面面积年平均输出热功率。
- (2) 若定日镜的尺寸与安装高度均相同，请设计定日镜场的吸收塔位置坐标、定日镜数目与位置、定日镜尺寸与安装高度等参数，使定日镜场在达到额定功率 60 MW 的条件下，单位镜面面积年平均输出热功率尽量大。
- (3) 若定日镜的尺寸与安装高度可以不相同，其他要求同问题 2，请设计定日镜场的参数，使单位镜面面积年平均输出热功率尽量大。

二、问题分析

2.1 问题一的分析

问题一是在已知定日镜与集热器的各项参数时，计算年平均光学效率及输出热功率、单位镜面面积年平均输出热功率的问题。根据给定条件，我们需要求得天文参数、法向直接辐射辐照度 DNI、光学效率三个基础部分，最终得到定日镜的输出热功率。要确定某一天某一时刻的太阳位置，需要得到此时的太阳高度角以及太阳方位角。而得到太阳高度角 α_s 后，通过计算即可得出 DNI 的数据。对于每面定日镜的光学效率 η ，考虑阴影遮挡损失、余弦损失、集热器截断损失三种随时间变化的动态损失，同时还涉及大气透射率、镜面反射率对光学效率的影响。其中，阴影遮挡损失包括三部分，分别是光线的入射被邻近定日镜面遮挡、光线经定日镜后的反射被遮挡，以及吸收塔的影子遮挡定日镜。而大气透射率在实际中受天气、温度、大气密度等因素的影响，是一个动态变化的数据，无法精确到每天，因此本文采用经验拟合模型^[1]，根据观测距离估计大气透射率。镜面反射率亦如此，其数值受到镜面材料、使用时长、清洗频率、光的漫反射与散射等因素的影响会上下波动，为简化计算，此处取一个合理常数即可。

2.2 问题二的分析

针对问题二，由于需要确定一个在达到额定功率时，让单位镜面面积年平均输出热功率尽可能大时的定日镜排布的最佳方案，因此本题考虑优化模型。相对于第一问而言，第二问的难点在于定日镜的数量、位置、镜面尺寸、安装高度均为变量，但最终年平均输出热功率的求解可以利用第一问的算法。在选择定日镜的布局时，要综合考虑几个关键因素。首先是镜面在镜场中的排布密度，若过于密集，相邻镜面的阴影遮挡就会很高，能量收集会大大下降，造成资源浪费。其次是镜面的阵列形状，通常有圆形、方形、麦田形等，合理地规划镜场布局能够最大化利用占地面积，问题要求我们一个同心圆的区域内设计镜场布局策略，出于题目和利用效率的双重考虑，本题选择类圆形排布。针对本问来说，可以首先建立单目标优化模型，确定待优化的目标函数，也就是单位镜面面积年平均输出热功率，再确定函数的约束条件以及决策变量的取值范围。同时，出于优化算法和减少不必要的排列次数考虑，可以先分析能让光学效率尽可能高的排布方式，再进行简化建模。对于具体的定日镜布局，采用启发式优化算法，求出全局最优解，从而生成最佳的镜面布局方案，以实现高效的能量收集。

2.3 问题三的分析

在第二问的基础上增添变量，即各个定日镜的尺寸与安装高度可以不相同，要求设计一个最优的定日镜场，使单位镜面面积年平均输出热功率尽量大。本题的难度在于增加了变量个数，排列的可能性也就更多。但其基础算法和思路与问题二一致。

三、模型假设

假设一：在计算“年均”指标的时点时天气晴朗，不出现下雨、阴天、雾霾等影响定日镜接收光照的特殊天气情况，也不考虑云层遮挡。

假设二：为简化计算，不考虑经过反射后的光线在接收器上的干涉、衍射情况，以及光线在传播中的漫反射。

四、符号说明

符号	说明	单位
E_{field}	定日镜场的输出热功率	kW/m^2
DNI	法向直接辐射辐照度	kW/m^2
α_s	太阳高度角	°
γ_s	太阳方位角	°
η_{sb}	定日镜的阴影遮挡效率	/
η	定日镜的光学效率	/
η_{cos}	定日镜的余弦效率	/
η_{trunc}	定日镜的截断效率	/
h	定日镜的安装高度	m
d	定日镜边长	m
H	吸收塔高度，即集热器中心离地高度	m
d_{HR}	单个定日镜镜面中心到集热器中心距离	m
A	镜面总面积	m^2

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 模型的建立

本题所求数据为定日镜场的输出热功率，其公式如下：

$$E_{field} = DNI \cdot \sum_i^N A_i \eta_i \quad (1)$$

式中， DNI 为法向直接辐射辐照度， N 为定日镜总数， A_i 为第 i 面镜的采光面积， η_i 为第 i 面镜的光学效率。根据上文的问题分析，要先分别求以下几个部分的参数，最终求解。

1) 天文参数（太阳位置）

太阳在以吸收塔中心为原点的镜场坐标系中的位置示意图见图 3。其中， α_s 表示原点到太阳的直线与 xoy 面的夹角，称为太阳高度角； γ_s 表示以正北方向为起始方向，按顺时针方向所测量的角度，称为太阳方位角。只要确定 α_s 和 γ_s 这两个数值，便能确定太阳在镜场坐标系中的位置。

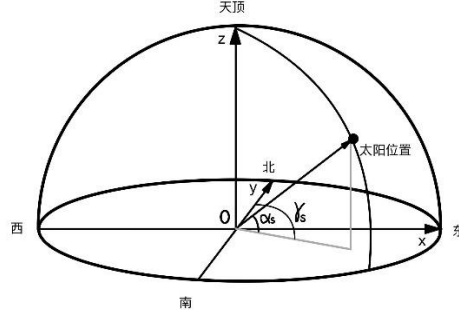


图 3 太阳位置示意图

太阳高度角 α_s 的计算公式如下：

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

式中， δ 为太阳赤纬角， φ 为当地纬度， ω 为太阳时角。由于当地纬度是已知数据，因此还需要解决另外两个量。

首先求每个时间节点的太阳赤纬角 δ ，即太阳在地球坐标系中相对于地球赤道的角度。由于一天之内太阳赤纬角的变化很小，可以忽略不计，这里近似认为不变。本文应用的是近似计算公式（Solar Declination Approximation Formula）：

$$\sin \delta = \sin \frac{2\pi D}{365} \sin \left(\frac{2\pi}{365} 23.45 \right) \quad (3)$$

式中， D 为以春分（3月21日）作为第0天起算的天数。

接着计算每个时间节点的太阳时角 ω ：

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

式中， ST 为当地时间。至此，可以得出太阳高度角 α_s 。

太阳方位角 γ_s 的计算公式如下：

$$\cos \gamma_s = \frac{\sin \varphi - \sin \alpha_s \sin \varphi}{\cos \alpha_s \cos \varphi} \quad (5)$$

2) 法向直接辐射辐照度 DNI

DNI表示在单位时间内，单位面积上垂直于太阳光线方向的辐照度（单位： kW/m^2 ），其数值通常受到太阳高度角、天气、大气散射和阴影遮挡等影响，是一个动态数据，故本文采用的是一种近似计算：

$$DNI = G_0 \left[a + b \exp \left(-\frac{c}{\sin \alpha_s} \right) \right] \quad (6)$$

$$\begin{cases} a = 0.4237 - 0.00821(6 - H)^2 \\ b = 0.5055 + 0.00595(6.5 - H)^2 \\ c = 0.2711 + 0.01858(2.5 - H)^2 \end{cases} \quad (7)$$

式中， H 为当地海拔高度， G_0 为太阳常数，取 $1.366kW/m^2$ 。

3) 光学效率

定日镜的光学效率的公式如下：

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

右边分别代表阴影遮挡效率、余弦效率、大气透射率、集热器截断效率以及镜面反射率（见图4）。其中，镜面反射率可取为常数，大气透射率只要计算镜面中心与吸收塔中心的间距即可得到。因此本题主要针对另外三种动态损失建立模型。而在求解这三种效率前，要先求每个时间节点中各个定日镜的朝向并表示出此时的法向量，使每个定日镜能将太阳光线反射到集热器中心，图示见图5。

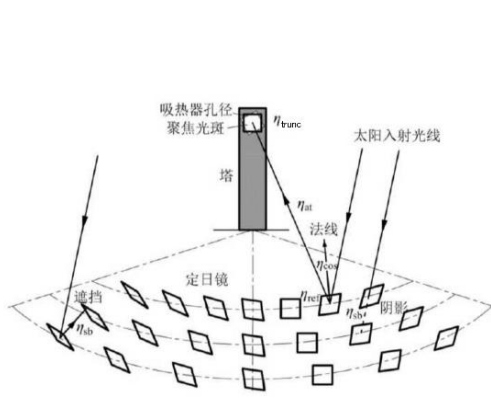


图4 光学效率示意图

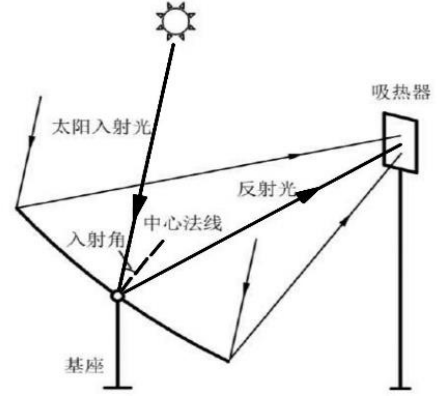


图5 定日镜工作原理

分析上图，对于某个指定的定日镜，已知镜场坐标系中太阳、定日镜中心、集热器中心的位置，则可利用向量表示入射与反射光线，再根据几何关系计算定日镜镜面所在平面的法向量。

a. 余弦效率 η_{cos}

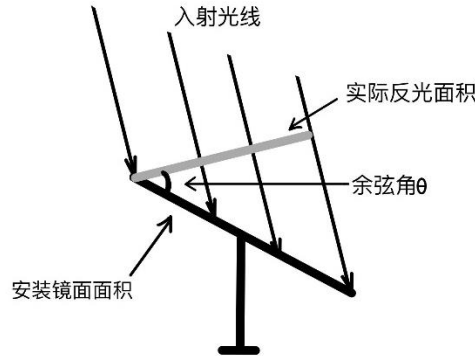


图6 定日镜实际反光面积示意图

在实际应用中，定日镜的俯仰角与方位角随着时间的改变而改变，太阳的入射光线与镜面并不垂直，而是呈一定的倾斜角度的，因此其接受的能量会产生损失，这也是能量损耗中占比最大的。这部分损失即体现在定日镜的余弦效率上。余弦效率是指定日镜的实际反光面积与安装面积的比值，其值等于入射光线与定日镜镜面法线夹角的余弦值（见图6）。余弦值越大，说明实际反光面积越大，损耗的能量越少。基于以上的分析，我们可以得到余弦效率的公式：

$$\eta_{cos} = \cos \theta_1 = \frac{\vec{e}_s \cdot \vec{n}}{|\vec{e}_s| \cdot |\vec{n}|} \quad (9)$$

式中， θ_1 为入射光线与定日镜镜面法线的夹角， \vec{e}_s 为指向太阳的入射光向量， \vec{n} 为定日镜镜面的法向量。

b. 阴影遮挡效率 η_{sb}

阴影遮挡效率在物理意义上指的是有效反射光线的镜面面积与安装镜面面积的比值，即 1-阴影遮挡损失。阴影遮挡的损耗共分为三部分，其一是在太阳的照射下，定日镜的阴影落在了邻近定日镜的镜面上，使得那一部分镜面无法接收太阳光；其二是太阳光经定日镜镜面反射到集热器的过程中，反射光被邻近定日镜的背面遮挡；其三是吸收塔的影子落在了镜场的某一片区域，使区域内的定日镜无法接收太阳光。在所有效率中，阴影遮挡效率是最复杂、最难于计算的，而在 9:00~15:00 这段时间内，阴影遮挡造成的光学损失率占总体损失的比例非常小^[2]，因此，为了保证数据准确性的基础上提高计算速度，本文对入射光线的遮挡不作处理。以下对阴影遮挡效率的计算进行具体阐述。

太阳光不是平行的光线，而是一束锥形光线。但考虑到在太阳光入射时，太阳与定日镜的距离极大，锥形的影响极小，可以将入射光线等效处理为平行光。因此计算阴影遮挡部分的面积时，实际上可以转化为在平行光照射下，将所有定日镜的镜面投影到镜场坐标系的 xoy 平面上（即影子在地面上的范围）。同理，将吸收塔也如此投影。那么在 xoy 平面上就会形成一层或多层重叠的阴影区域，多次重叠部分的面积即阴影遮挡损失的面积，即定日镜遮挡部分在 xoy 平面上的投影，图示见图 7。

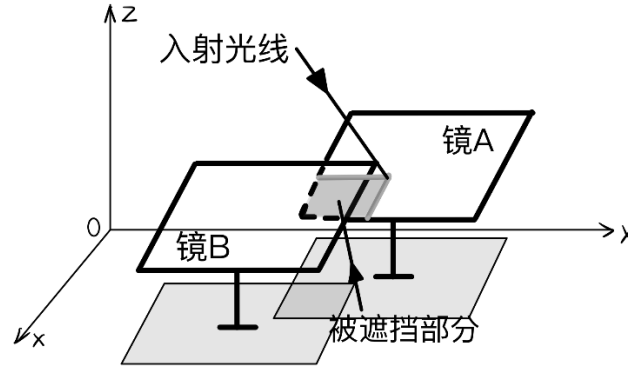


图 7 被遮挡部分投影示意图

计算每个定日镜在 xoy 平面的投影。根据每个定日镜的法向量、中心点位置和入射边长的几何关系，求出四个顶点在 xoy 平面的具体坐标。但在实际应用中，四个顶点位置与定日镜中心、俯仰角、方位角相关，几何关系极复杂，且计算量庞大，因此本文采用的是一种等效圆投影。为了简化计算，将正方形的定日镜镜面取一个圆心为定日镜中心的近似圆，计算出近似圆半径以及投影到 xoy 平面上的椭圆的长短半轴。

对于一个定日镜镜面中心（即圆心）为 $A_0(x_0, y_0, z_0)$ ，半径为 r ，倾斜角度为 θ （相对于垂直于 xoy 平面的投影面）的等效圆，其投影到 xoy 平面的圆心坐标 $A'(x', y')$ ，我们使用旋转矩阵计算：

假设将三维坐标系中的点旋转 xoy 平面内的点的旋转矩阵为 R ：

$$R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

然后，对于某个定日镜镜面中心 (x_0, y_0, z_0) ，我们可以通过如下公式计算其在 xoy 平面上的投影位置 (x', y') ：

$$[x', y'] = [x_0, y_0, z_0] * R \quad (11)$$

xoy 坐标系中新投影圆(x, y)的长轴通过计算 x 坐标的最大差值得到，同理，短轴通过计算 y 坐标的最大差值得到。

在作出 xoy 平面上所有的投影图形之后，我们要对单层阴影区域面积所占全部阴影面积的比例进行计算。考虑到蒙特卡洛算法是一种灵活性强、精度可控的随机抽样模拟算法，只要将抽样次数设置的较高，精度就能高，可用于概率与统计问题。因此本文采用蒙特卡洛法对阴影遮挡效率进行计算。对所有阴影区域（包括一层或多层）进行随机投豆概率计算，记投在阴影区域为一层上的次数为 T_1 ，投在阴影区域为多层上的次数为 T_2 ，则阴影遮挡效率计算公式如下：

$$\eta_{sb} = \frac{T_1}{T_2} \quad (12)$$

c. 截断效率 η_{trunc} 。

对于截断效率而言，光源与目标物的距离、光束形状、光强、光的衍射和散射都会对其产生影响。其中，光的强度会随着距离的增大而减弱，而在镜场中定日镜到集热器的距离对光来说很小，光强的减弱可以忽略不计。同时，在实际应用中，光束相对集中，并且定日镜到集热器的距离不大，光的衍射和散射在这种情况下的影响很小，可以不做计算。因此，本文主要考虑的是光源与目标物的距离（即定日镜镜面中心到集热器中心的距离 d_{HR} ），以及光束形状对镜场截断效率的影响。

前文中已有阐述，太阳光并不是平行光，而是一束锥形光线。为了准确评估以及提高精度，不能忽略太阳光是锥形光这一特性。一束锥形光的半角展宽是 4.65 mrad，光源与目标物的距离越大，溢出的光线就越多，截断效率越低。

基于此，本文利用平板投影法与光线追踪法^[3]来计算截断效率，具体原理见图 8。

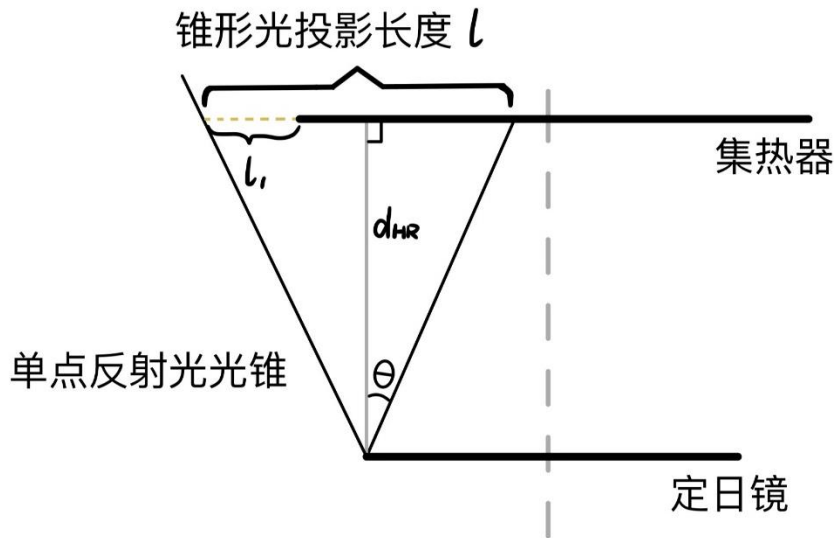


图 8 定日镜边界点反射光锥示意图

首先，经计算得，定日镜的中心即使是锥形光，想要从直径为 7m 的集热器周围溢出，至少需要 700 多米，远超我们所规定限制的定日镜场的边界到集热器中心的距离，所以，定日镜中心所反射到集热器的光不会有溢出。又考虑到大部分的锥形光不会从集热器上下两端溢出，大部分溢出光出现在集热器水平方向，所以下文在计算光线能量时将投影面积比均转换成投影长度比，简便计算。

接下来，我们就可以从定日镜边界开始考虑光线溢出问题。运用光线追迹法追踪

边界光线，由图可知边界光锥有一部分投影在集热器上，一部分溢出而造成损失。从边界开始，向定日镜中心移动极小距离，追踪下一条光线，会发现该光锥比边界光锥少极小溢出部分。按上述方式从定日镜边界向定日镜中心逐步移动观测的光锥会发现光锥溢出部分会逐渐减小到 0，不难发现从边界到溢出为 0 的临界点，光锥的溢出部分是线性减小的。

所以我们可以假设每个光锥内存在能量为单位 1 的无数光线且他们的投影均匀分布。那么投影部分的能量便与投影长度正比例相关。于是，我们便可以根据投影部分的长度计算投影部分的能量，同时可以计算溢出部分的能量，从而求得截断效率。因为该几何模型是关于集热器中心和定日镜中心连线对称的，所以我们只需要计算左半部分便能求出截断效率。

已知定日镜中心到集热器中心的距离 d_{HR} ，同时由几何关系可知，定日镜边界到集热器的垂直距离也为 d_{HR} ，又知锥形光半角展宽，那么可以求出锥形光投影长度 L，即：

$$L = d_{HR} \cdot \tan \theta \quad (13)$$

又根据溢出长度= $L/2 - 0.5$ ，那么能量损失比 $n = (L/2 - 0.5) / (3.5 + L/2)$ 。

根据题目所给的截断公式可知，集热器截断效率为集热器接受能量与定日镜成功反射出的能量之比，即：

$$\eta_{trunc} = (1 - n) \quad (14)$$

5.1.2 模型求解与结果分析

针对定日镜的镜面反射率，目前的镜面反射率通常在 0.93~0.94 范围内，但它的数值与镜面材料、使用时长、清洁程度等密切相关。若经过 30 天的自然积灰，镜面反射率的损失甚至会达到 25%以上。为了使数据更加贴近实际投入使用的定日镜镜面反射率情况，我们查阅了相关文献中实践应用的镜面反射率的受控因素以及变化情况，本文采用该文献中的平均镜面反射率 90.82%^[4]。而对于大气透射率，由以下公式可得：

$$\eta_{at} = 0.99321 - 0 - 000117d_{HR} - 1.97 \times 10^{-8} \times d_{HR}^2 \quad (15)$$

对于最终定日镜场的输出热功率公式，我们

最后，根据问题一所给的数据，我们建立模型计算后得到结果如表 1 所示。其中所有数值计算完毕后取小数点后三位：

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 (kW/m ²)
1 月 21 日	0.607	0.720	0.964	0.974	0.508
2 月 21 日	0.638	0.740	0.986	0.974	0.592
3 月 21 日	0.659	0.761	0.990	0.974	0.648
4 月 21 日	0.674	0.779	0.990	0.974	0.687
5 月 21 日	0.683	0.789	0.990	0.974	0.706
6 月 21 日	0.686	0.792	0.990	0.974	0.712
7 月 21 日	0.683	0.789	0.990	0.974	0.706
8 月 21 日	0.674	0.779	0.990	0.974	0.685
9 月 21 日	0.658	0.760	0.990	0.974	0.646
10 月 21 日	0.634	0.738	0.984	0.974	0.583

11 月 21 日	0.605	0.718	0.964	0.974	0.502
12 月 21 日	0.585	0.711	0.941	0.974	0.455

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)
0.64867	0.75647	0.98075	0.97360	38.89998	0.61923

根据表 2，我们可以得知定日镜的余弦效率是光学效率的主要影响成分，而对于指定的定日镜场，余弦效率是由太阳的高度角以及方位角共同决定的，以下我们就针对四个典型的时间分析相应的太阳高度角与方位角：

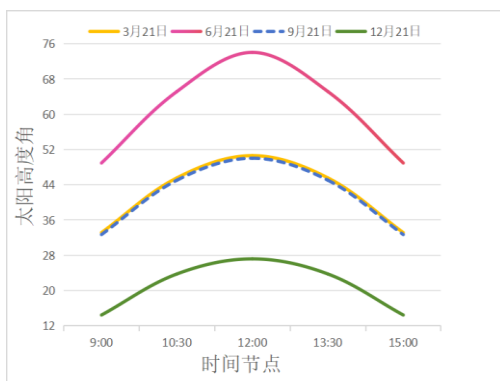


图 9 典型日太阳高度角变化图

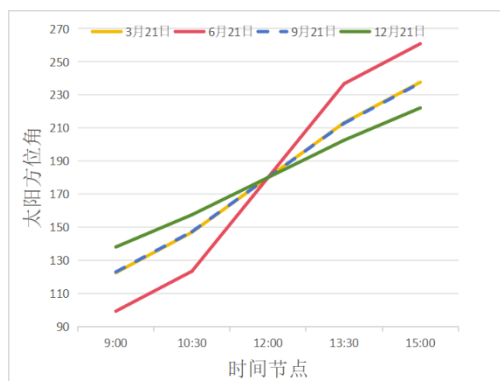


图 10 典型日太阳方位角变化图

根据以上二图，我们可以看出夏至日（6 月 21 日）的太阳高度角要显著大于其他月份，并且这一天的太阳方位角变化幅度也是最大的。经查阅文献，发现从仿真模拟镜场的余弦效率分布情况来看，有总体来说太阳高度角越大，镜场的余弦效率越高的规律^[5]。即从理论上来说，6 月余弦效率应该是最高的，12 月余弦效率最低；同时，在一天当中，正午 12 点的太阳高度角最大，是一天当中余弦效率最高的时候。

5.2 问题二模型的建立与求解

5.2.1 模型的建立

本题的目标是求一种最优方案，通过改变定日镜边长、安装高度、位置坐标以及数量和吸收塔的高度，从而使单位镜面面积年平均输出热功率在约束条件下尽可能大。问题中要求定日镜场的额定功率要达到 60MW，定日镜要安装在距吸收塔 100~350m 的范围内，且镜面边与安装高度分别要在 2~8m、2~6m 的范围内，并且定日镜随水平轴旋转时不会碰到地面。据此我们建立一个单目标优化模型。

决策变量：

定日镜边长、安装高度、位置坐标、定日镜总数量、吸收塔高度，即

$$\{d, h, x, y, N, H\} \quad (16)$$

目标函数：

单位镜面面积年平均输出热功率最大，即

$$\max f(d, h, x, y, N, H) = \frac{E_{field}}{A} \quad (17)$$

目标为最大单位镜面面积年平均输出热功率，其中， A 是定日镜镜面总面积。
上文已经提到：

$$E_{field} = DNI \cdot \sum_i^N A_i \eta_i \quad (18)$$

将其拆分分析，得到：

$$\begin{cases} \sum_{i=1}^N A_i = A \cdot \eta_{sb} \\ \eta = \eta_{sb} \eta_{cos} \eta_{at} \eta_{trunc} \eta_{ref} \end{cases} \quad (19)$$

又因为在光学效率 η 中，镜面反射率 η_{ref} 是定值，大气透射率 η_{at} 、阴影遮挡效率 η_{sb} 、集热器截断效率 η_{trunc} 由于损失影响很小，也可以近似为定值计算。因此定义一个参数 K ，假设此处的参数 K 为一个可求的定值，使：

$$K = \eta_{at} \eta_{trunc} \eta_{ref} \quad (20)$$

将上述式子整合并化简，得到目标函数的最简表达式：

$$\max f(d, h, x, y, N) = DNI \cdot K \cdot \eta_{sb}^2 \cdot \eta_{cos} \quad (21)$$

约束条件：

定日镜的边长 2~8m，安装高度 2~6m，并且要求镜面旋转时不会触及地面，即

$$\begin{cases} 2 \leq d \leq 8 \\ 2 \leq h \leq 6 \\ h - \frac{d}{2} > 0 \end{cases} \quad (22)$$

定日镜场要达到额定功率，即

$$E_{field} = DNI \cdot A' \cdot \eta = 60MW \quad (23)$$

其中， A' 表示定日镜的有效镜面面积，即

$$A' = N \cdot d^2 \cdot \eta_{sb} \quad (24)$$

将以上二式联立，得

$$DNI \cdot K \cdot N d^2 \eta_{sb}^2 \eta_{cos} = 60MW \quad (25)$$

从上式我们可以看出定日镜的实际总镜面面积 Nd^2 与其平均余弦效率 η_{cos} 是互相牵制的。而根据式（25），要让单位镜面面积年平均输出热功率尽可能大，就要让定日镜的平均余弦效率尽可能大。要让平均余弦效率 η_{cos} 尽可能大，就要使实际总镜面面积 Nd^2 尽可能小，这个会在后续建模中详细介绍。综上，在寻找最优布局的过程中，我们要优先考虑将定日镜安装在余弦效率高的区域。

针对镜场的阵列形状，经查阅文献得，一般镜场排布后反射光在吸收塔表面有三种聚焦策略，分别是简单聚焦策略、交替偏移聚焦策略、正态分布聚焦策略。在一天的不同时刻，正态分布聚焦策略的能流密度较低，所以我们考虑，反射光斑在吸收塔表面呈现正态分布的规律。为了使光斑在吸收塔表面更容易产生正态分布的光斑，我们在排布定日镜场时，使用 DELSOL 布局。

首先尽可能使吸收塔一圈都存在定日镜，其次，大致排布总体格局是以吸收塔为圆心，向外进行同心圆排布，并且前后两层同心圆中的定日镜成交错排布，使所有定日镜在吸收塔形成的反射光斑更接近正态分布。具体排列图示见图 10。

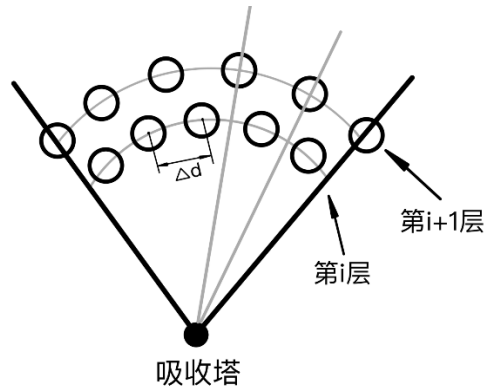


图 10 定日镜排列示意图

在确定了定日镜场的基本阵列形状后，为了初步判断定日镜场的分布规律，我们根据问题一中的数据可以先作出四个典型日的正午时刻余弦效率热图，仿真图见图 11。

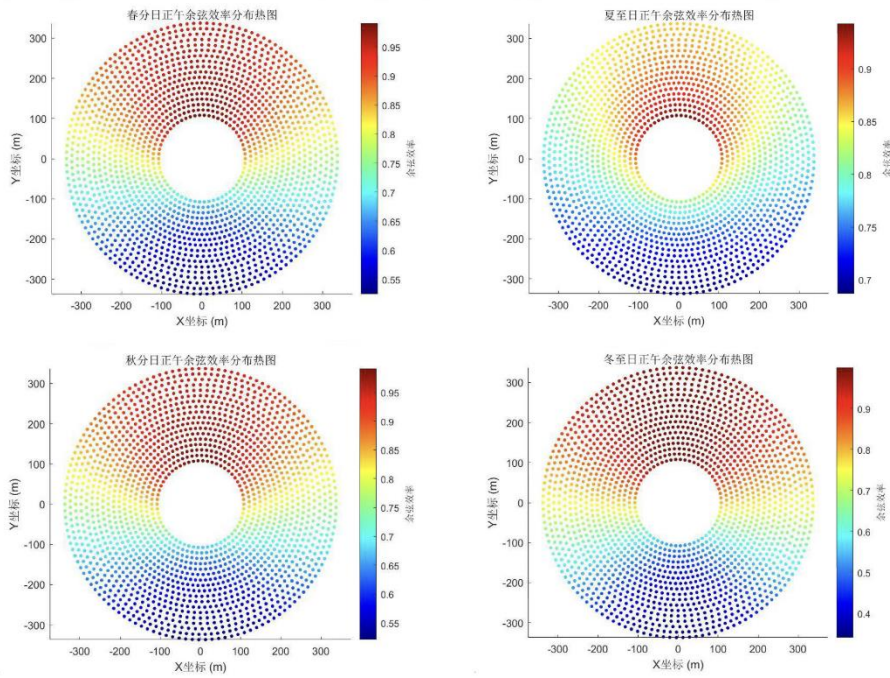


图 11 四个典型日正午余弦效率分布热图

根据图 11，易知无论在哪个季节，在 y 坐标为正数时余弦效率普遍更高，即靠北的定日镜余弦效率高，光学损耗小。其中，由于夏至日的太阳高度角更大，南北边定日镜的余弦效率相差相对较小；而冬至日南北相差极大，靠北的定日镜的余弦效率要远高于南边，反光能力相差较大。以上是针对正午时刻的分析，而对于同一天内非正午时刻的时间节点，由于太阳方位角根据 12:00 对称分布(见图 10)，所以东西两边的平均余弦效率对于某一天来说也是对称相等的。

综合上述两点，定日镜场的基本分布规律是北多南少，东西对称。得出这个结论后，我们就能在 MATLAB 的代码实现时从能让光学效率尽可能高的较优布局入手，减少不必要的排列次数，让计算效率大大提高。

基于模拟退火算法的定日镜坐标求解：

接下来，对定日镜场中各个定日镜的位置坐标计算。要在求解问题复杂程度高时找一个最优解，我们对比了粒子群算法以及模拟退火算法，二者都适用于求复杂问题的全局最优解，但考虑到要避免产生局部最优解，我们最终采用模拟退火算法这种全局搜索能力更强的算法。

对于这部分的算法，我们需要建立一组针对定日镜坐标的新的目标函数、决策变量以及约束条件，这与上文的单目标优化模型要区分开来。事先说明，在此模型中，变量仅为定日镜的位置坐标，而其他需要求解的变量如 d ， h ， N ，在此处都视作参数。

决策变量：

第 i 个定日镜的在镜场坐标系 xoy 平面的坐标。

$$\{x_i, y_i\} \quad (26)$$

其中 i 的范围是 $1 \sim N$ 。

目标函数：

定日镜的余弦效率总和最大。在求最优布局时我们的原则是让平均余弦效率 η_{cos} 尽可能大，从而达到单位镜面面积年平均输出热功率最大的要求。

$$\max \frac{\sum_{i=1}^N g(x_i, y_i)}{N} \quad (27)$$

其中，

$$g(x_i, y_i) = \cos \theta_i = \frac{\vec{e}_s \cdot \vec{n}_i}{|\vec{e}_s| \cdot |\vec{n}_i|} \quad (28)$$

这是问题一中计算余弦效率的公式。

约束条件：

对于定日镜的位置，要求吸收塔即原点的 100m 内不能安装，且要建在距原点的 350m 范围内，同时对邻近定日镜的间距也有要求，即

$$s.t. \begin{cases} x_i^2 + y_i^2 > 100^2 \\ x_i^2 + y_i^2 < 350^2 \\ (x_i - x_j)^2 + (y_i - y_j)^2 > (d + 5)^2 \end{cases} \quad (29)$$

5.2.2 模型的求解与结果分析

对于最终优化目标只要确定定日镜的坐标后就容易推得，因此本题的求解重点在于定日镜坐标的计算，下面给出利用模拟退火算法确定定日点坐标以及其它参数的具体思路以及求解步骤：

Step1: 确定待优化的目标函数以及决策变量，确定取值范围。此步骤在模型建立中已有阐述。

Step2: 最优解初始化。按照约束条件随机生成关于 y 轴对称的 1000 个坐标，根据模拟退火算法计算其最大平均余弦效率，多次进行该操作，确定初始平均余弦效率。

其中，对于随机坐标在 MATLAB 中代码实现的具体思路如下：在前文中我们已经分析，想让单位镜面面积年平均输出热功率尽可能大，定日镜的基本布局是“北多南少，东西对称”。因此我们设立 n 个随机点，首先让它有 0.9 的概率落在北侧，即第一、二象限。落点前要判断是否满足约束条件，即北侧是否还能放下定日镜、是否距原点

100~350m 以及与邻近定日镜的间距是否大于 $d+5$ 。将北侧的空间最大化利用后，则开始落在第三、四象限。这个思想的具体实现代码关键部分如下：

```
% 使用概率来生成 y, 更有可能生成在正数范围内,即北边
if rand < 0.9
    y = rand * max_value;
else
    y = (rand - 1) * max_value;
end
% 检查是否满足约束条件
distance = sqrt(x^2 + y^2);

if distance > 100 && distance < 350 && all(pdist2([x, y], coordinates(:, 1:2)) > (5+r))
```

其中 $\text{max_value}=350$ ，用于缩小随机数生成范围。

Step3: 确定参数 N 与 S 。(S 为定日镜面积，其值等于 d^2 ，为便于定义，以下都使用 S 代替 d^2)根据初始平均效率和所需要达到 60MW 的条件我们可以计算出 $N \cdot S$ 的值。根据定值 $N \cdot S$ 求得一系列 S 与 N ，将这一系列 S 与 N 分别代入我们的模拟退火算法中，计算随机坐标在该 S 与 N 条件下的最大平均余弦效率。比较该系列平均余弦效率，确定最大平均效率下的一组 S 与 N 作为后续求解的参数，并保证 $h > d/2$ 。

Step4: 通过不断提高每个点的判断条件最低余弦效率，低于该余弦效率的点不作记录，来不断压缩可以取的点的空间，并且同时在不改变点数 N 的情况下提高所求的平均余弦效率。

Step5: 重复执行循环后，选取平均余弦效率最大的一组坐标作为我们后续计算的坐标。再通过固定坐标，改变吸收塔高度 H 和安装高度 h 来计算最大的平均余弦效率，以及吸收塔高度 H 和安装高度 h 。最后再通过多次微调（10000 次）来对坐标分别进行优化，即略微改变一个坐标的位置，观察其余弦效率是否改变。若增大则保留，反之不变。经过大量的微调及对比判断，我们最终得到的定日镜的分布图如图 1 所示。我们发现结果非常符合之前对定日镜场布局的初步分析，北侧接受光能的能力更强，数目也更多。

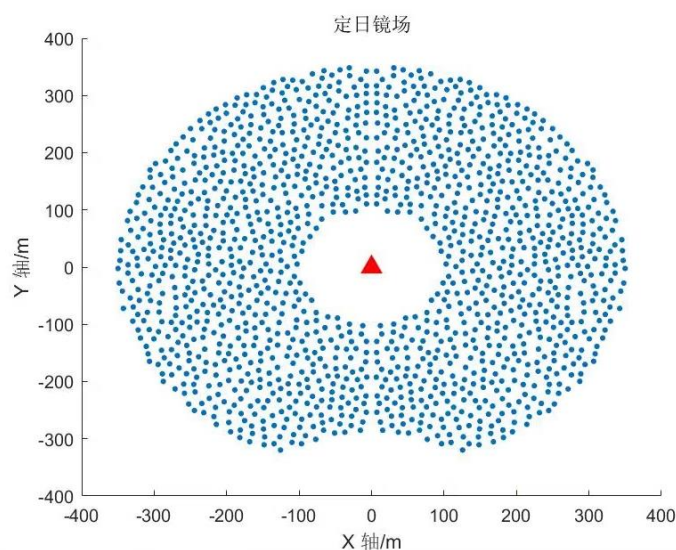


图 12 定日镜分布图

Step6: 输出最优情况下的目标函数值以及定日镜坐标情况。定日镜场的设计参

数见表 3。（定日镜的具体坐标见附录）

表 3 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)
(0,0)	8*8	5	1420	90880

得到以上参数后，我们再根据问题一中的计算过程得到在最优情况下每月的平均光学效率及输出功率（见表 4），最后求得其年平均值（见表 5）。

表 4 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	0.700	0.818	0.980	0.974	0.596
2 月 21 日	0.725	0.839	0.990	0.974	0.676
3 月 21 日	0.743	0.859	0.990	0.974	0.731
4 月 21 日	0.758	0.877	0.990	0.974	0.772
5 月 21 日	0.765	0.886	0.990	0.974	0.791
6 月 21 日	0.768	0.888	0.990	0.974	0.797
7 月 21 日	0.765	0.886	0.990	0.974	0.791
8 月 21 日	0.757	0.876	0.990	0.974	0.770
9 月 21 日	0.742	0.858	0.990	0.974	0.728
10 月 21 日	0.723	0.836	0.990	0.974	0.668
11 月 21 日	0.698	0.816	0.980	0.974	0.589
12 月 21 日	0.678	0.809	0.960	0.974	0.539

表 5 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)
0.73510	0.85400	0.98583	0.97360	63.9844	0.70405

5.3 问题三模型的建立与求解

5.3.1 模型的建立

本题中定日镜的尺寸以及安装高度都可以各不相同，通过合理的排布各个尺寸的定日镜，可以使定日镜场的单位镜面面积年平均输出热功率最大化。这是求解相对问题二来说的更优布局方案，对实际应用中使塔式电站的效率最大化有重要意义。然而在这样的背景下变量更多，求解复杂程度及难度大大增加，计算量巨大。为了简化计算，我们采用问题二求得的全局最优解的数据。要在第二问的基础上使效率最大化，可改变定日镜尺寸和安装高度。而考虑到额定输出功率的限制，在第二问中已有阐述， $N \cdot S$ 是一个定值，由于这里的 N 已经确定，因此 S 也不更改，即定日镜的尺寸与第二问最终取得的数据相同，仅调整各个定日镜的安装高度 h 。

针对不同定日镜安装高度的选择，我们为了尽可能避免邻近定日镜的阴影遮挡损失，使单位镜面面积年平均输出热功率尽可能大，我们采用一种中间低外围高、北侧高南侧低的喇叭状分布。其中，北侧高于南侧是因为在当地纬度的条件下，太阳始终位于镜场的南面，为了让靠北侧的定日镜更充足地接收到光线，我们将其安装高度调高。

综合以上分析，我们建立了一个安装高度计算公式：

$$h = 4 + \frac{\rho - 100}{250} + \left| 1 - \frac{\vartheta}{\pi} \right| \quad (30)$$

为了便于计算，在本题的求解中我们使用的是定日镜的极坐标。其中，极径 ρ 是定日镜底座到吸收塔底座的距离，极角 ϑ 是定日点底座这个点与正向极轴（即正北）之间的角度。

$$\begin{cases} \rho = \sqrt{x^2 + y^2}, \rho \in (100, 350) \\ \vartheta \in (0, 2\pi) \end{cases} \quad (31)$$

据此：

$$\begin{cases} \frac{\rho - 100}{250} \in (0, 1) \\ \left| 1 - \frac{\vartheta}{\pi} \right| \in (0, 1) \\ h \in (4, 6) \end{cases} \quad (32)$$

对于安装高度的计算公式，其中，4 是指定日镜边长为 8m，因此要使转动过程中定日镜底端不会碰到地面，安装尺寸要大于 4m； $\frac{\rho - 100}{250}$ 的作用是使定日镜的安装尺寸从外圈到内圈依次降低； $|1 - \vartheta/\pi|$ 的作用是让安装高度从北侧向南侧依次降低。

5.3.2 模型的求解与分析

根据上述模型，我们用 MATLAB 运行得到了每个定日镜安装高度的数据，并基于此绘制了更为直观的定日镜场模拟图样，见图 13。

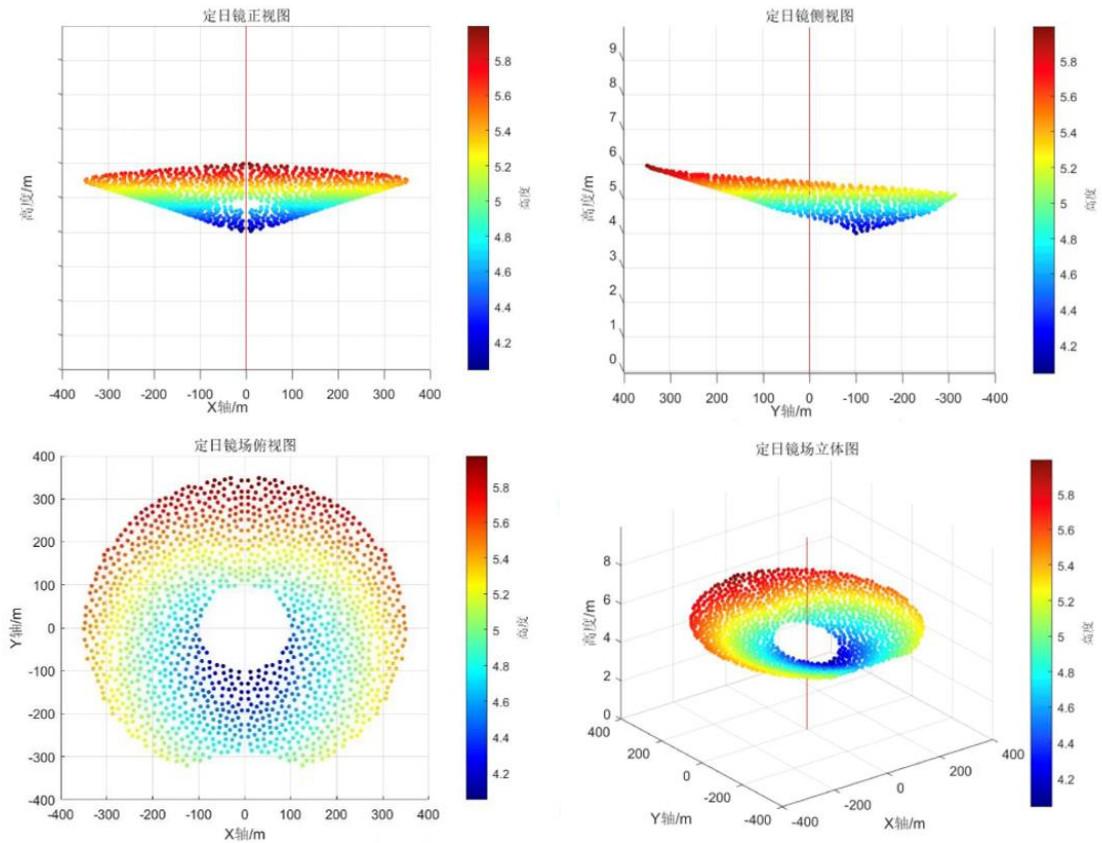


图 13 定日镜场安装高度模拟图

根据求解出的数据，我们又可得到以下三张表：

表 6 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)	单位面积镜面年平均 输出热功率 (kW/m ²)
0.74525	0.86254	0.98975	0.97360	65.04109	0.71568

表 7 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	0.713	0.826	0.989	0.974	0.613
2 月 21 日	0.732	0.847	0.990	0.974	0.683
3 月 21 日	0.750	0.868	0.990	0.974	0.738
4 月 21 日	0.765	0.885	0.990	0.974	0.779
5 月 21 日	0.773	0.895	0.990	0.974	0.799
6 月 21 日	0.775	0.897	0.990	0.974	0.805
7 月 21 日	0.773	0.894	0.990	0.974	0.799

8 月 21 日	0.765	0.885	0.990	0.974	0.778
9 月 21 日	0.749	0.867	0.990	0.974	0.736
10 月 21 日	0.730	0.845	0.990	0.974	0.675
11 月 21 日	0.705	0.824	0.980	0.974	0.595
12 月 21 日	0.712	0.817	0.998	0.974	0.588

表 8 问题三设计参数表

吸收塔位置坐标	定日镜总面数	定日镜总面积 (m^2)
(0, 0)	1420	90880

六、模型的评价

6.1 模型的优点

在利用模拟退火算法求定日镜的坐标时，我们采用的是先粗略分析缩小范围，再通过代码实现精确分析的方法。我们首先对定日镜的大致排列形状以及最合理的分布规律进行分析，得出“北多南少，东西对称”的结论，基于此我们对算法进行优化。从而在编程时可以随机经过初步优化后的方案，由于经过初步分析后排除了许多效果较差的方案，计算量减少，计算速度比起未经过初步优化时大大提升。

在计算阴影遮挡效率时，我们使用蒙特卡洛法近似计算，将复杂问题简单化处理，有助于节省运算时间。

6.2 模型的缺点

在问题二求解时，为了简化计算，我们将大气透射率 η_{at} 、阴影遮挡效率 η_{sb} 、集热器截断效率 η_{trunc} 这三个量近似为一个定值。虽然这三个部分的损失在光学总损失中占比很小，但也存在一定的误差，计算精度不够高。

七、参考文献

- [1]O. Farges, J.J. Bezian, M. El Hafi, Global optimization of solar power tower systems using a Monte Carlo algorithm: Application to a redesign of the PS10 solar thermal power plant [J], Renewable Energy, 2018, 119:345-353.
- [2]胡叶广. 塔式太阳能热发电系统的多级反射式聚光镜场的研究[D].哈尔滨工业大学,2019.
- [3]张国勋,饶孝枢.塔式太阳能聚光系统太阳影象方程[J].太阳能学报,1982(02):172-178.DOI:10.19912/j.0254-0096.1982.02.008.
- [4]师志鹏,龚俊,王伟志等.积灰对塔式太阳能定日镜反射率的影响研究[J].热力发电,2019,48(06):134-137.DOI:10.19666/j.rlfid.201811200.
- [5]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.DOI:10.27205/d.cnki.glttec.2022.001089.

[6]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.

附录

Matlab 代码:

```
1. （部分代码因数据表改动问题无法直接运行，代入相关数据仍可运行）
2. %% 导入数据，作分布图
3. options = detectImportOptions('附件.xlsx');
4. options.VariableNamingRule = 'preserve';
5. data = readtable('附件.xlsx', options);
6. x = data("x 坐标 (m)");
7. y = data("y 坐标 (m)");
8. z = 4;
9. % 创建散点图
10.scatter(x, y, 10, "blue", 'filled');
11.hold on;
12.title('定日镜场');
13.xlabel('X 轴');
14.ylabel('Y 轴');
15.% 手动添加坐标原点的红色三角形
16.plot(0, 0, 'r^', 'MarkerSize', 10, 'MarkerFaceColor', 'red');
17.
18.
19.%%计算太阳相关参数
20.%太阳赤纬角
21.data1 = readtable('日期D.xlsx');
22.D = data1.D;
23.% 计算  $\sin(\delta)$ 
24.delta = sin(2*pi*D./365) * sin(2*pi/360*23.45);
25.cos_delta = sqrt(1 - delta.^2);
26.%太阳高度角三角函数
27.sin_delta =data1.sin;
28.cos_phi =data1.cos;
29.cos_omega = cos(39.4/180*pi);
30.sin_omega = sin(39.4/180*pi);
31.sin_alpha_s = cos_phi * cos(0*pi) * cos_omega + sin_delta * sin_omega;
32.cos_alpha_s = sqrt(1 - sin_alpha_s.^2);
33.%太阳方位角三角函数
34.cos_gamma_s = (delta - sin_alpha_s * sin_omega) ./ (cos_alpha_s * cos_omega);
35.
36.
```

```

37.%%
38.%求角度
39.%太阳高度角
40.data2 = readtable("工作簿 2.xlsx");
41.angle = asin(table2array(data2));
42.angle_degrees = rad2deg(angle);
43.disp('反正弦值（弧度）：');
44.disp(angle);
45.disp('反正弦值（度数）：');
46.disp(angle_degrees);
47.%太阳方位角
48.data2 = readtable("方位角度数.xlsx");
49.angle = acos(table2array(data2));
50.angle_degrees = rad2deg(angle);
51.disp('反余弦值（弧度）：');
52.disp(angle);
53.disp('反余弦值（度数）：');
54.disp(angle_degrees);
55.
56.
57.%%
58.%大气透射率
59.% 导入数据
60.options1 = detectImportOptions('附件.xlsx');
61.options1.VariableNamingRule = 'preserve';
62.data1 = readtable('附件.xlsx', options1);
63.h=80;%吸收塔中心高度
64.
65.results = [];
66.for j = 1:size(data1, 1)
67.    % 获取每一行的 x、y、 z 值
68.    x = data1("x 坐标 (m)")(j);
69.    y = data1("y 坐标 (m)")(j);
70.    z = data1("z 坐标 (m)")(j);
71.
72.    % 计算 dHR
73.    dHR = sqrt(x^2 + y^2 + (h-z)^2);
74.
75.    % 计算  $\eta_{at}$  的值
76.    eta_at = 0.99321 - 0.0001176 * dHR + 1.97e-8 * dHR^2;
77.
78.
79.    results = [results; eta_at];
80.end

```

```

81.
82. disp('每一行的结果: ');
83. disp(results);
84.
85.
86. %%
87. %太阳光锥临界
88. angle_in_mrad = 4.65e-3; % 将 4.65 mrad 转换为弧度
89. h=250;
90. % 计算正切值
91. tan_value = tan(angle_in_mrad);
92. d =h*tan_value;
93.
94.
95. %%
96. %投影长度
97. % 导入数据
98. options1 = detectImportOptions('附件.xlsx');
99. options1.VariableNamingRule = 'preserve';
100. data1 = readtable('附件.xlsx', options1);
101.
102. results = [];
103. for j = 1:size(data1, 1)
104.
105.     x = data1("x 坐标 (m)")(j);
106.     y = data1("y 坐标 (m)")(j);
107.     z = 4;
108.     % 计算 dHR
109.     dHR = sqrt(x^2 + y^2 + 76^2);
110.     angle_in_mrad = 4.65e-3; % 将 4.65 mrad 转换为弧度
111. h=dHR;
112. % 计算正切值
113. tan_value = tan(angle_in_mrad);
114. d =h*tan_value;
115.     s = (d-1)./(d+7);
116.     results = [results; s];
117. end
118.
119. disp('每一行的结果: ');
120. disp(results);
121.
122.
123. %%
124. %蒙特卡洛法计算阴影效率

```

```

125. % 从两个 Excel 文件中读取顶点坐标
126. data1 = readtable('全部投影顶点坐标.xlsx');
127. data2 = readtable('重叠区域顶点坐标.xlsx');
128. x1 = data1.x;
129. y1 = data1.y;
130. x2 = data2.x;
131. y2 = data2.y;
132. polygons1 = {[x1, y1]};
133. polygons2 = {[x2, y2]};
134. p = 10000;
135. n = 0;
136. m = 0;
137. hold on;
138. for k = 1:p
139.     xP = rand * 2; % 随机生成点的 x 坐标
140.     yP = rand * 2; % 随机生成点的 y 坐标
141.     isInsideAnyPolygon1 = false;
142.     isInsideAnyPolygon2 = false;
143. % 遍历每个多边形并检测点 P 是否在内部
144.     for i = 1:size(polygons1, 2)
145.         x3 = polygons1{1, i};
146.         y3 = polygons1{2, i};
147.         isInside1 = inpolygon(xP, yP, x3, y3);
148.
149.         if isInside1
150.             isInsideAnyPolygon1 = true;
151.             break;
152.         end
153.     end
154.     for i = 1:size(polygons2, 2)
155.         x4 = polygons2{1, i};
156.         y4 = polygons2{2, i};
157.         isInside2 = inpolygon(xP, yP, x4, y4);
158.
159.         if isInside2
160.             isInsideAnyPolygon2 = true;
161.             break;
162.         end
163.     end
164.     if isInsideAnyPolygon1 && isInsideAnyPolygon2
165.         n = n + 1;
166.     end
167.     if isInsideAnyPolygon1
168.         m = m + 1;

```



```

169.     end
170. end
171. axis equal;
172. P = n / m;
173. disp(['落在两个多边形内部的点的比例: ', num2str(P)]);
174.
175.
176. %%
177. %模拟退火法计算最大平均余弦效率
178. %通过随机产生坐标计算约束条件下的最大值
179. r = 8; %正方形边长
180. % 设置坐标点的数量（这里是总点数的一半，因为会生成对称的点）
181. half_num_points = 700; % 假设总点数是 1400，即需要生成 700 对对称
    点
182.
183. % 设置坐标值的范围
184. min_value = -350; % 坐标值的最小范围
185. max_value = 350; % 坐标值的最大范围
186. fixed_z = 5; % 固定的 z 坐标
187.
188. % 初始化坐标和温度数组
189. coordinates = zeros(2 * half_num_points, 4);
190.
191. % 模拟退火参数
192. initial_temperature = 1.0; % 初始温度
193. final_temperature = 0.01; % 最终温度
194. cooling_factor = 0.9; % 冷却因子
195.
196. % 生成随机初始坐标
197. for i = 1:half_num_points
198.     while true
199.         x = (max_value - min_value) * rand + min_value;
200.
201.         % 使用概率来生成 y，更有可能生成在正数范围内,即北边
202.         if rand < 0.9
203.             y = rand * max_value;
204.         else
205.             y = (rand - 1) * max_value;
206.         end
207.         % 检查是否满足约束条件
208.         distance = sqrt(x^2 + y^2);
209.
210.         if distance > 100 && distance < 350 && all(pdist2([x,
            y], coordinates(:, 1:2)) > (5+r))

```

```

211.
212.         symmetric_point_index = 2 * half_num_points - i + 1
213.         ;
214.         symmetric_x = -x;
215.         symmetric_y = y;
216.         symmetric_distance = sqrt((x - symmetric_x)^2 + (y
217.         - symmetric_y)^2);
218.         if symmetric_distance > (5+r)
219.             temperature = funs(x, y, fixed_z);
220.
221.             if temperature >= 0.56
222.
223.                 coordinates(i, :) = [x, y, fixed_z, tempera
224.                 ture];
225.                 coordinates(symmetric_point_index, :) = [sy
226.                 mmetric_x, symmetric_y, fixed_z, temperature];
227.                 break;
228.             end
229.         end
230.     end
231. end
232.
233. % 初始最优解
234. best_coordinates = coordinates;
235. best_temperature_sum = sum(coordinates(:, 4));
236. best_average_temperature = best_temperature_sum / (2 * half_num
237. _points);
238.
239. while initial_temperature > final_temperature
240.     for i = 1:2 * half_num_points
241.         % 随机选择一个坐标点
242.         idx = randi(2 * half_num_points);
243.         x = coordinates(idx, 1);
244.         y = coordinates(idx, 2);
245.
246.         % 生成随机扰动
247.         dx = (2 * rand - 1) * 10; % 在[-10, 10]范围内生成随机扰
248.         动
249.         dy = (2 * rand - 1) * 10;
250.         u = x + dx;
251.         v = y + dy;
252.
253.         new_distance = sqrt(u^2 + v^2);

```

```

249.
250.         % 检查新坐标是否满足约束条件
251.         if new_distance <= 100 || new_distance >= 350 || any(pd
           ist2([u, v], coordinates(:, 1:2)) <= 5+r)
252.
253.             continue;
254.         end
255.
256.         % 计算扰动后的温度
257.         new_temperature = funs(u, v, fixed_z);
258.
259.         % 计算温度变化
260.         delta_temperature = new_temperature - coordinates(idx,
           4);
261.
262.         % 如果温度变化为正, 或者按概率接受负变化
263.         if delta_temperature >= 0 || rand() < exp(delta_tempera
           ture / initial_temperature)
264.             coordinates(idx, 1) = u;
265.             coordinates(idx, 2) = v;
266.             coordinates(idx, 4) = new_temperature;
267.         end
268.     end
269.
270.     % 更新最优解
271.     current_temperature_sum = sum(coordinates(:, 4));
272.     current_average_temperature = current_temperature_sum / (2
       * half_num_points);
273.     if current_average_temperature > best_average_temperature
274.         best_coordinates = coordinates;
275.         best_temperature_sum = current_temperature_sum;
276.         best_average_temperature = current_average_temperature;
277.     end
278.
279.     % 冷却温度
280.     initial_temperature = initial_temperature * cooling_factor;
281. end
282.
283. % 最终的坐标和温度
284. x_coords = best_coordinates(:, 1);
285. y_coords = best_coordinates(:, 2);
286. z_coords = best_coordinates(:, 3);
287. temperatures = best_coordinates(:, 4);
288.

```

```

289. % 绘制热图
290. figure;
291. scatter(x_coords, y_coords, 10, temperatures, 'filled');
292. xlabel('X 轴');
293. ylabel('Y 轴');
294. title('最优的热图');
295. colorbar;
296. colormap(jet);
297. axis equal;
298. grid on;
299.
300. % 输出最大平均温度和最优坐标数组
301. fprintf('最大平均温度: %f\n', best_average_temperature);
302. fprintf('最优坐标数组: \n');
303. disp(best_coordinates);
304.
305. % % % 继续优化最大平均温度坐标集合
306. additional_iterations = 10000; % 设定微调迭代次数
307.
308. for iter = 1:additional_iterations
309.     % 随机选择一个坐标点进行微调
310.     idx = randi(2 * half_num_points);
311.     x = best_coordinates(idx, 1);
312.     y = best_coordinates(idx, 2);
313.
314.     % 生成随机扰动
315.     dx = (2 * rand - 1) * 10; % 在[-10, 10]范围内生成随机扰动
316.     dy = (2 * rand - 1) * 10;
317.     u = x + dx;
318.     v = y + dy;
319.
320.     new_distance = sqrt(u^2 + v^2);
321.
322.     % 检查新坐标是否满足约束条件
323.     if new_distance <= 100 || new_distance >= 350 || any(pdist2
        ([u, v], best_coordinates(:, 1:2)) <= 2*r)
324.
325.         continue;
326.     end
327.
328.     % 计算扰动后的温度
329.     new_temperature = funs(u, v, fixed_z);
330.
331.     % 计算温度变化

```

```

332.     delta_temperature = new_temperature - best_coordinates(idx,
333.         4);
334.     % 如果温度变化为正, 或者按概率接受负变化
335.     if delta_temperature >= 0 || rand() < exp(delta_temperature
336.         / initial_temperature)
337.         best_coordinates(idx, 1) = u;
338.         best_coordinates(idx, 2) = v;
339.         best_coordinates(idx, 4) = new_temperature;
340.     % 更新最优解的平均温度
341.     best_temperature_sum = sum(best_coordinates(:, 4));
342.     best_average_temperature = best_temperature_sum / (2 *
343.         half_num_points);
344. end
345.
346. % 输出微调后的最大平均温度和坐标数组
347. fprintf('微调后的最大平均温度: %f\n', best_average_temperature);
348. fprintf('微调后的坐标数组: \n');
349. disp(best_coordinates);
350.
351. % 最终的坐标和温度
352. x_coords_final = best_coordinates(:, 1);
353. y_coords_final = best_coordinates(:, 2);
354. z_coords_final = best_coordinates(:, 3);
355. temperatures_final = best_coordinates(:, 4);
356.
357. % 绘制微调后的热图
358. figure;
359. scatter(x_coords_final, y_coords_final, 10, temperatures_final,
360.     'filled');
361. xlabel('X 轴');
362. ylabel('Y 轴');
363. title('微调后的热图');
364. colorbar;
365. colormap(jet);
366. axis equal;
367. grid on;
368. scatter(x_coords_final, y_coords_final, 10, 'filled');
369. hold on;
370.
371. title('定日镜场');

```

```

372. xlabel('X 轴');
373. ylabel('Y 轴');
374.
375. % 手动添加坐标原点的红色三角形
376. plot(0, 0, 'r^', 'MarkerSize', 10, 'MarkerFaceColor', 'red');
377.
378. %%函数 temperapute 计算余弦效率
379. function temperapute = funs(x, y, z)
380.     % 输入反射终点的坐标
381.     x1 = x;          % X 坐标
382.     y1 = y;          % Y 坐标
383.     z1 = z;          % Z 坐标
384.
385.     x2 = 0;          % 反射终点的 X 坐标
386.     y2 = 0;          % 反射终点的 Y 坐标
387.     z2 = 80;         % 反射终点的 Z 坐标
388.
389.     % 输入入射光线向量
390.     Vi = [0, -0.634730513, 0.772733573];
391.
392.     % 计算反射光线的单位向量
393.     Vr1 = [x2 - x1, y2 - y1, z2 - z1];
394.     Vr = Vr1 / norm(Vr1);
395.
396.     % 计算点积
397.     cosine_theta = dot(Vi, Vr);
398.
399.     % 计算余弦效率
400.     temperapute = sqrt(0.5 * (1 + cosine_theta));
401. end
402.
403.
404.
405.
406. %%
407. %将 xy 坐标转换成极坐标
408. % 导入数据
409. options = detectImportOptions('问题二坐标.xlsx');
410. options.VariableNamingRule = 'preserve';
411. data = readtable('问题二坐标.xlsx', options);
412. x = data.x;
413. y = data.y;
414.
415. r = zeros(size(x));

```

```

416. theta = zeros(size(x));
417. % 计算每个数据点的极坐标 (r,  $\theta$ ) 使用 y 轴作为极轴, 逆时针方向
418. for i = 1:length(x)
419.     r(i) = sqrt(x(i)^2 + y(i)^2);
420.     theta(i) = mod(2*pi - atan2(x(i), y(i)), 2*pi);
421. end
422. % 将角度转换为以  $\pi$  为单位
423. theta_pi = theta / pi;
424.
425. for i = 1:length(x)
426.     fprintf('数据点 %d - 直角坐
        标 (x, y) = (%.2f, %.2f)\n', i, x(i), y(i));
427.     fprintf('数据点 %d - 极坐
        标 (r,  $\theta$ ) = (%.2f, %.2f $\pi$ )\n', i, r(i), theta_pi(i));
428. end
429.
430.
431.
432. %求每个定日镜的高度 h
433. % 导入数据
434. options = detectImportOptions('问题二坐标.xlsx');
435. options.VariableNamingRule = 'preserve';
436. data = readtable('问题二坐标.xlsx', options);
437. r = data.r;
438. a = data.pi;
439.
440. h = zeros(size(r));
441. % 计算每个数据点的 h 值
442. for i = 1:size(r)
443.     h(i) = 4 + (r(i) - 100) / 250 + abs(1 - a(i));
444. end
445.
446. for i = 1:size(r)
447.     fprintf('数据点 %d - h = %.2f\n', i, h(i));
448. end
449.
450.
451. %作效果图
452. % 导入数据
453. options = detectImportOptions('问题二坐标.xlsx');
454. options.VariableNamingRule = 'preserve';
455. data = readtable('问题二坐标.xlsx', options);
456. x = data.x;
457. y = data.y;

```



```

458. z = data.h;
459. % 根据 h 的不同值设置颜色
460. colors = z; % 使用 h 的值作为颜色
461. colormap jet; % 选择颜色映射
462. caxis([min(colors), max(colors)]);
463. % 创建一个立体图
464. figure;
465. scatter3(x, y, z, 10, colors, 'filled');
466. xlabel('X 轴');
467. ylabel('Y 轴');
468. zlabel('高度');
469. title('定日镜场的立体图');
470. % 添加颜色条
471. cbar = colorbar; % 获取颜色条的句柄
472. % 设置颜色条的标题
473. cbar.Label.String = '高度'; % 设置颜色条的注释为"高度"
474. % 手动设置 z 轴的上下限
475. zUpperLimit = max(z) + 4; % 调整上限
476. zLowerLimit = 0; % 调整下限
477. zlim([zLowerLimit, zUpperLimit]);
478. % 添加平行 z 轴的细线，作为吸收塔
479. hold on; % 保持图形处于活动状态
480. line([0, 0], [0, 0], [zLowerLimit, zUpperLimit], 'Color', 'r',
        'LineWidth', 0.5); % 添加细线
481. hold off; % 结束图形的活动状态

```